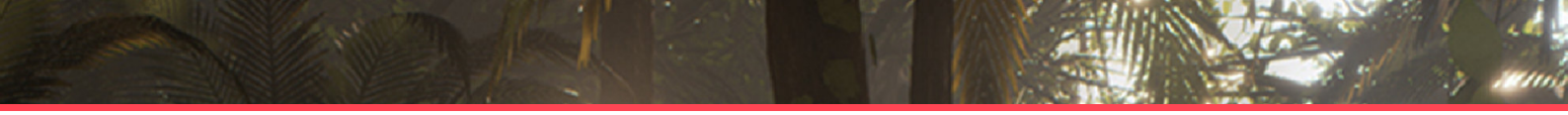
THE ENVIRONMENT ARTIST HANDBOOK

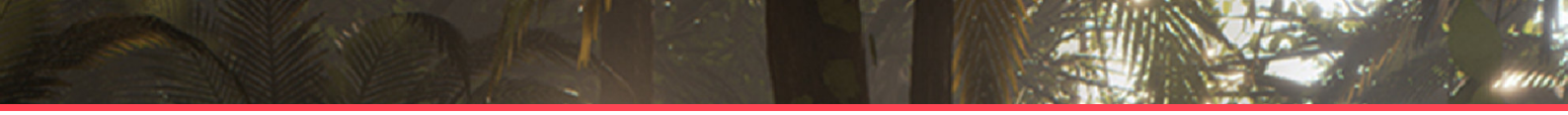
2023

COMPILED AND WRITTEN BY | TIMOTHY DRIES

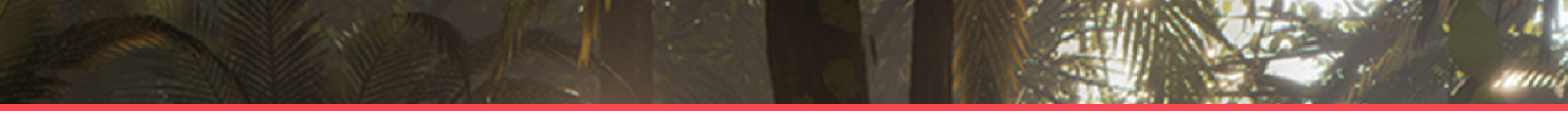


Page numbers are clickable and will take you to the corresponding page

Page numbers are clickable and will take you to the corresponding page



Page numbers are clickable and will take you to the corresponding page



00 NEW THIS MONTH

Broken Landscape Sectors Quick Fix

Introduction

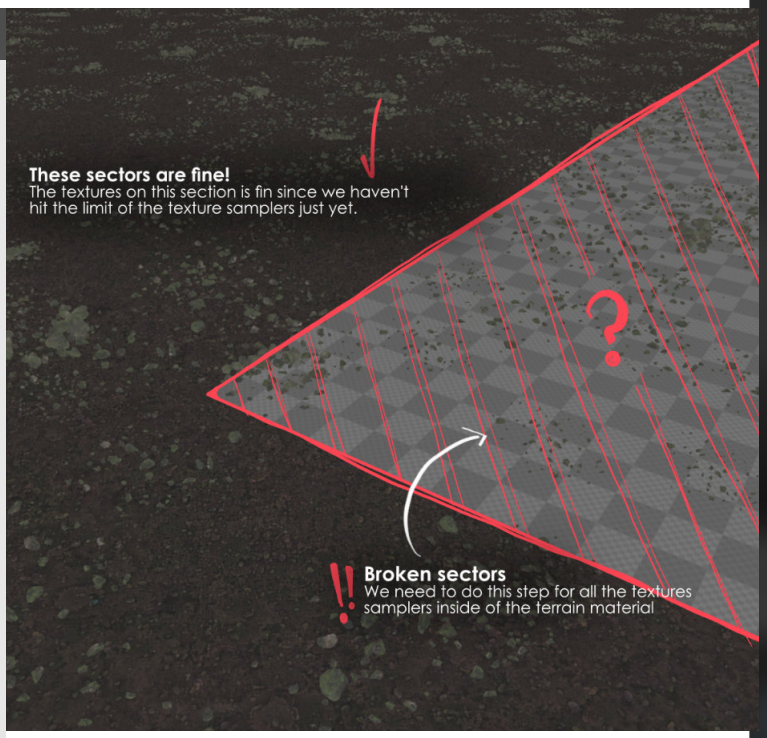
When creating large landscapes in Unreal Engine and you are adding multiple materials layers onto it you might run into the issue where it turns your landscape into the standard grey material if you try to paint multiple layers.

1.Where does this come from?

There is a built in limitation on the amount of texture samplers that you can use in a material inside of Unreal Engine, with the hard limitation being a total of 16 different texture samplers (or even lower at 6 on mobile).

So when we are working with landscapes, we need a lot of textures to make a good looking landscape material especially with blending included. Assuming that we use Basecolor, ORM, Normal and Height for each material we want to paint on the landscape we we quickly end up on that texture sampler limit.

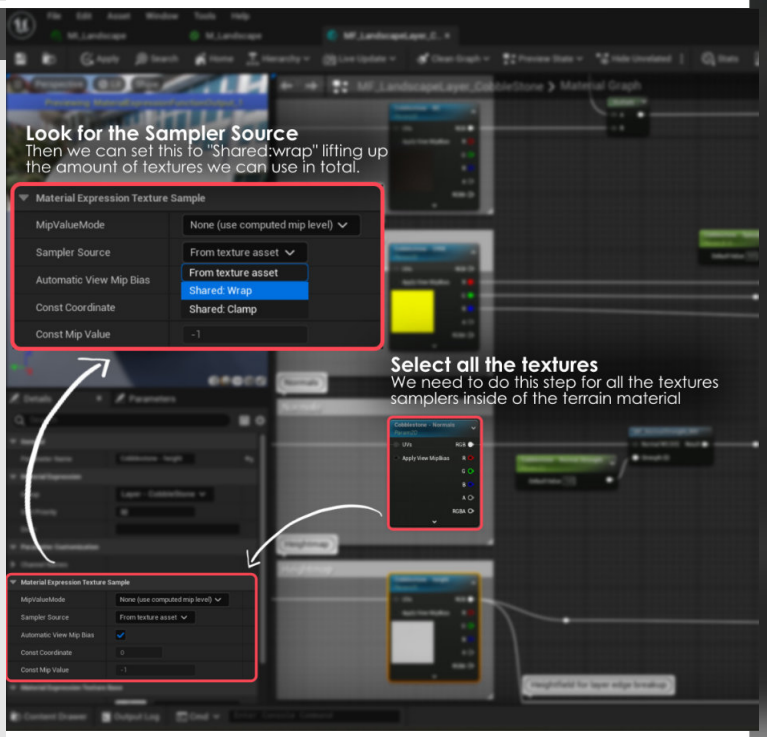
This then results in that Grey default texture popping up or you not being able to paint any new layers. When this happens, it's usually because of that limitation being hit and thus it can't render it anymore.



2. How do we solve this?

The solution to this is actually pretty simple. We need to go through each of the texture samplers and inside the details panel look for "Sampler Source" and set this from the default setting to **"Shared:Wrap"**.

What this does, is it will set the maximum texture samplers from the previous mentioned 16 to 128 instead, thus unlocking more space to add in more layers for your different layers.



Custom thumbnails in Unreal Engine

Introduction

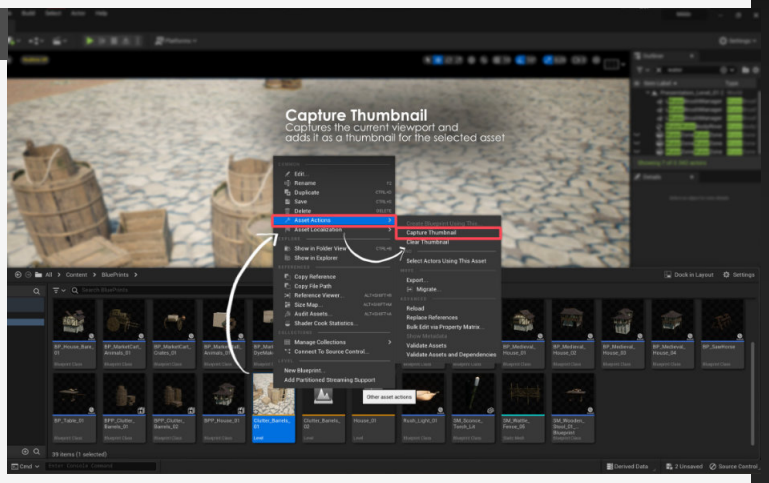
Tired of those boring looking thumbnails for any of your assets, levels and projects?

Let's dive into a couple of tips to help you create new thumbnails to make things easier to find.

1. Level Thumbnails

For normal levels this is not that big of a deal, but when working with the new level instances this becomes more crucial. You want to be able to see at least at a glance what these instances look like before you start adding them to the level.

You can right click on the level itself (of that level instance in your content browser) go to asset actions and then **"Capture Thumbnail"**. This will then capture the viewport and apply it as a thumbnail.

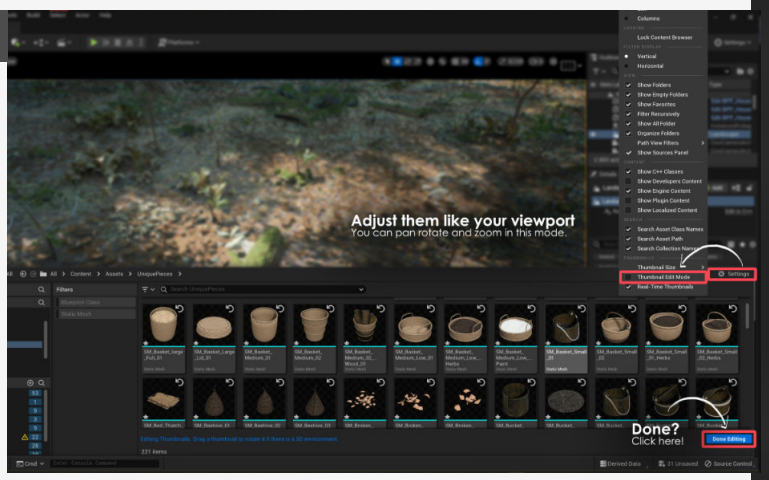


2. Asset thumbnails

We can also change thumbnails up on individual assets by going into a special viewport edit mode, which is hidden inside of the little **"Settings"** menu in the content browser.

We can then adjust these thumbnails as we would normally work in our viewport.

And then once we're done just press **"Done editing"**

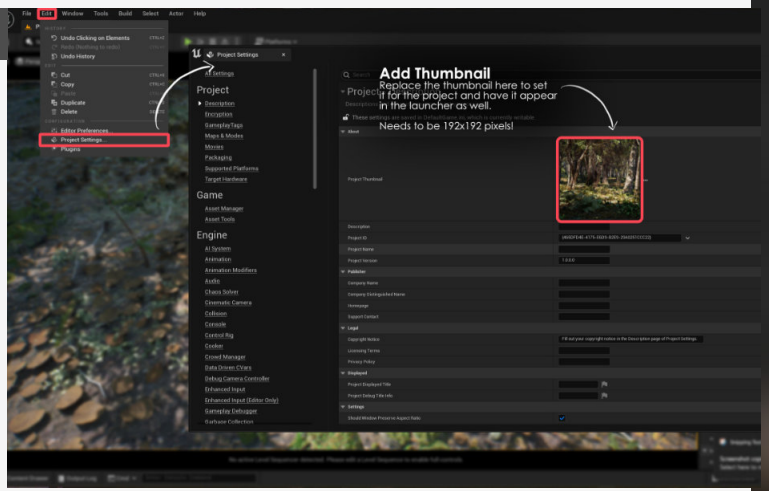


3. Project thumbnails

To change this we go to **"Project settings"** under the **"Edit"** menu on the top toolbar, once this opens we can change the project thumbnail under the project description.

You can't really add any dimension in here though you have to make them very specific to 192x192 pixels.

So you will need to quickly edit any screenshots you've taken of the projects before you can use them here.



Color Correction Regions

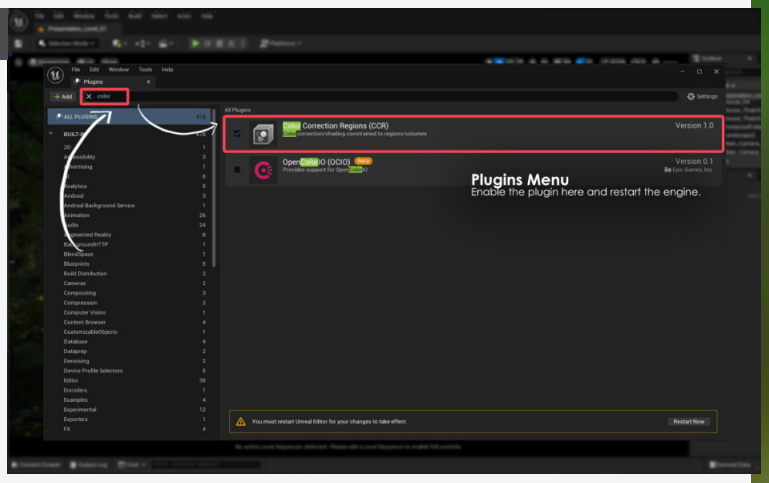
Introduction

Recently discovered this neat little trick from some friends and wanted to share it here as well. These volumes allow you to change specific color correction settings inside of volumes in Unreal Engine, giving you control over the look of certain area's.

1. Plugin and where to find them

For this to get working we first need to install the plugin first so we get access to them first.

You can find these in the "Plugins" under "Color correction Regions (CCR)" and will need to restart the Engine to take this into affect.



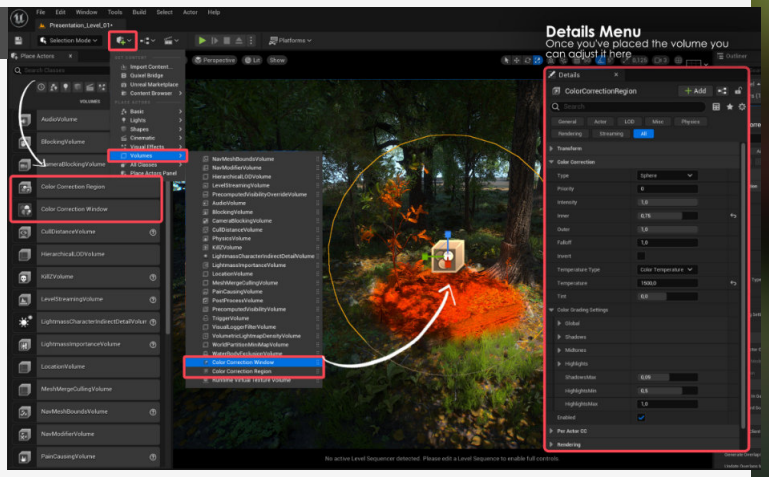
2. How to use these

Once we've got it installed, we can look for these volumes in the actor panel or in the "place actor panel" under Volumes.

You see we have two new additions here one for volumes and one for a window, volumes add a controllable volume based on a simple shape like a cube, sphere, cone or cylinder and window will add a plane where everything behind it being altered.

Once placed we then have control over the majority of our post processing controls especially the colour correction ones, allowing us to change the the visuals either in the volumes or behind the window.

We also have control over the falloff area and the strength of this volume giving additional control.



3. Application methods

HIGHLIGHTING A FOCAL POINT: Adding a little volume around an area and upping the saturation and contrast slightly can be a great little tool to highlight a certain area in our scene.

ART DIRECTING A SCENE: What if we take a scene and have a volume around our little fireplace in the middle and then we invert the volume to affect everything else apart from the fireplace. Then turning the saturation down on this volume to almost make the rest of the scene black and white, giving nice art direction control.

EMPHASIZING FOR-MID AND BACKGROUND: With windows instead of volumes we can do separate colour correction based on a distance in our scene. We can then highlight the area that contains our focal point and de-emphasise the other area's slightly.

And so much more, you can see why I love these little area's!

01 ART FUNDAMENTALS

Environmental Storytelling

Introduction

Environmental storytelling is the soul that breaths life into your environments and should always be at the core of the scene or project that you are making, so why is that and how can we tackle this for our own needs?

That's what we will quickly look into today.

1. The importance

There are obvious advantages for this ofcourse, but do keep in mind that it will break all the texture links that are embedded in the project.

So it is really important if you are making environment art, even if you just add a little bit of it, either in the textures of the assets or by adding additional props.

It will add to the scene and make it feel more relatable and alive.

2. Planning and immersion

So how do you add a story to your environment, well normally you don't add one, it's going to be fused with all the assets that you are going to create for that environment so you need to establish at least the basics of the story before you start on it.

A good way to do this is to start thinking about the basic story, then think about what would have happened to the environment or the characters within them and continue to build upon this base until you have some- thing that you like, is fitting to the scene or is unique in it's own way.

3. Leaving room for imagination

The sense of wonder is a powerful thing, so ideally taking this away is and feeding the entire thing to the viewer/ player is something that I avoid when creating environments.

I love to look at Environments that take me somewhere, immerse me and still leave enough room for questions.

For example in the scene "Last Bastion" I played around with the balance between hope and despair so you can see that the light is still on, there are still banners present, the sun shines down with rays of hope, but why is there no-one in the lookout tower?

4. The choice is yours

This is such a wide topic that I can't address this in a short format like this, so there might be more tips/tricks coming that support this topic.

But what I wanted to end with, is that you have the choice how you construct your story, you can keep it simple or have it really on the nose.

to convey to the people viewing this and get as close as possible.

And most importantly, keep learning and experimenting, I'm still doing that myself.

Leading the eye - 01

Introduction

Let's have a look at something that is often discussed in concept art and traditional painting, but can be used inside of environment art and games as well, this being how to lead the eye. With this we will have a look at the different tools that creators can use to really get the most out of the visual aspect of your work and create a piece that is pleasing to the eye.

1. The tools

When it comes to making a pretty picture it's important to understand the rules that make up the fundamentals of a picture that works.

In this entry we will have a look at some of the different tools we can use to guide the person looking at a picture towards the thing you want to focus to be on.

2. Contrast

Contrast is the different in colors or values, so this will actively determine how bright some things will be compared to other things in the scene.

In general, less contrast will flatten out the image. But we can use this our advantage artistically.
In general, less contrast will flatten out the image. But we can use this our advantage artistically, because higher contrast is also going to draw the attention of the eye towards the contrasted area more than towards a less contrasted area.

3. Light

On the right are some examples of this both in concept art and 3D art. You are pulled in by the use of light on the house section creating a contrast with the darker foreground.
When it comes to 3D we can make use of all different kinds of light and further accentuating it with fake god-rays for instance.

Another golden tip is to look at the masters of painting, because they really used this fully to there advantage and pushed the meaning of using light into the symbolic realm at times.

4. Details

The eye is automatically attracted to something that looks more detailed and complicated than areas of rest, which are as important as details by the way!

If everything is busy and detailed, nothing is.
For some examples the use of this is probably most visible in concept art work as well, but can be super useful for Environments as it will allow you to optimize your time spend on assets that need to be in focus or be used as interactable props or gameplay.

This is a really big topic, so we will continue this in the next one!

Leading the eye - 02

Introduction

In the second part of leading the eye we will continue the exploration of different tools that we can use to guide the eye through the image.

1. Aspect ratio

Let's look at some other tips and tricks to help you guide people through the environment/ art that you are making starting with the Aspect Ratio.

Something that we don't really tend to think of because we are limited by the end goal of our medium being the computer screen.

But for personal work we can have a little bit of a play with these dimensions and maybe switch for a vertical aspect ratio for vertical compositions.

Another thing that I love to do, Is adding smaller black bars at the top and bottom of the image and add a cinematic feel to the image.

2. Rule of thirds

This is the old time classic rule that I love to use myself, the core of this rule is that you divide your image in three equal chunks both horizontally and vertically.

Then once we have this grid, try and put the focus point of the image on one of these intersections or along one of these lines.

The intersection of these lines is called a Power-point (or a crash-point apparently? but power sounds better right!?)

You can see an example of this technique in use on the left.

3. Depth of field

Another trick to really pull in the focus on your environments is to use depth of field.

I use these in my environments but in a really minor way, to slightly make the foreground blurry and pull the focus in on the main focus of the image.

For most example environments, using them in a subtle way is the way to go but there are other great examples who use depth of field in a bigger way.

The first of one of these environments that comes to mind are the environments of Unravel, heavily influenced by shift-tilt photography.

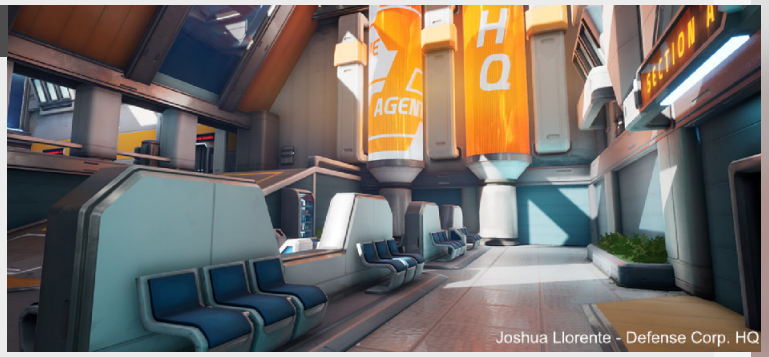
4. Patterns

This is a deeper topic and really justifies a breakdown of it's own for all the things you can do, these patterns can range from using patterns in colors, shapes, lines

This entire section is based on rein- forcing the feeling that you want to convey with your image.

Some more simple examples are round shapes feel safe and inviting (see the image on the right) where as sharp shapes feel dangerous and should be avoided by the player.

Another example is the use of repetition to create a sense of com- fort, structure and convey the sense of hierarchy.



PBR Theory for artists - 01

Introduction

This time we are going to dive into the magical world of texturing, this little introduction to some of the theory behind texturing, which will be crucial to understand what makes good textures tick and how you can leverage all those different things to your advantage.

1. What is PBR?

Physically Based Rendering, this way of texturing and rendering is meant to be a closer representation to the real world. Why is why we call it Physically Based Rendering as it's not truly physically accurate but a really close approximation.

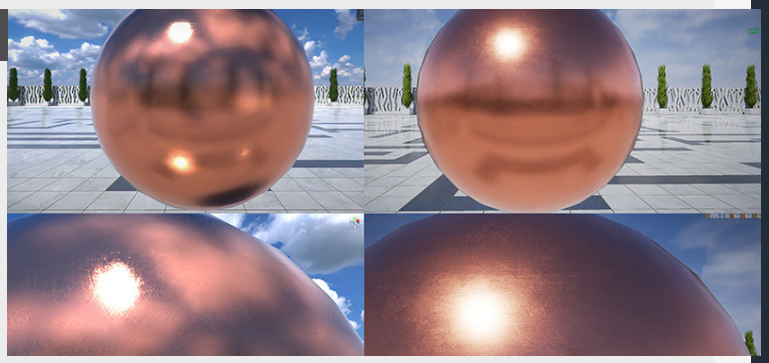
This workflow has been invented to come closer to real life examples where everything is considered shiny and will also allow artist to author materials independent of lighting as these textures are a closer representation to the real world.
This comes with a bunch of adjustments in how we author different textures and how the shaders using these textures will be rendering them on screen.

2. Different shading models

When approaching this shading model we need to have a look at the different shading models, generally there are two ways of working.

Specular/gloss and metalness, where the first one uses the specular and gloss to control light bouncing back where the second workflow focuses more on if the material is dialectic or not, which will influence the color of the reflection and the roughness will control the amount of reflection.

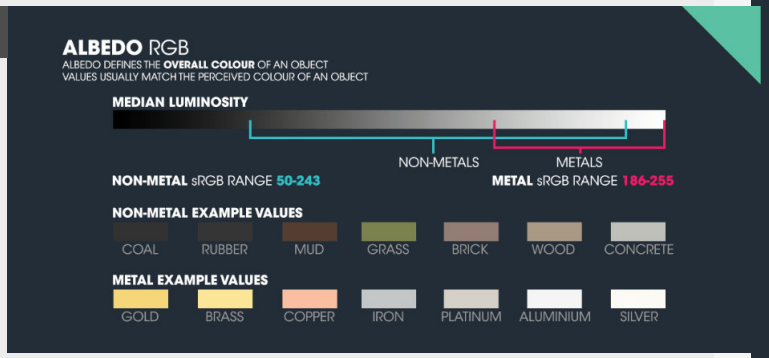
These differences do not imply that one is better than the other, it just implies a different way of exporting/authoring your materials.



3. Albedo and it's values

With trying to approach it more realistically that also means that the values of some materials can be measure too in the form of texture scanning (photogrammetry). But more specifically about albedo for the moment, these are the color values behind the texture, most of these can be found on charts if you really want to be as accurate as possible.

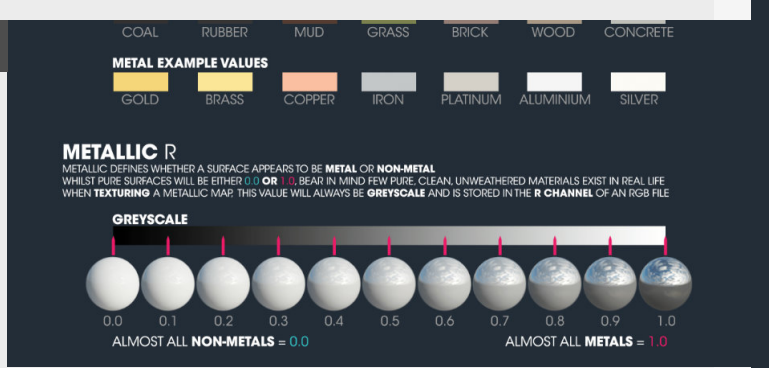
Another good source of this information could be looking at some Megascan textures and sample some of the colors that can be seen on these examples. Keep in mind that this texture does not contain any light information. But of course everything said here are the base rules and they can be broken for artistic purpose.



4. Metalness

With the metalness workflow comes a new way of handling different maps for your material, for this texture specifically we use a black and white texture where white is metal (or Conductive) and black is just a normal surface (Dielectric). Dielectric being the inverse of being conductive. Why is this important you ask? This changes the way the material is that it a conductive material will use the basecolor in its reflections (thus have a colored reflection) where normal material don't have this behavior.

But, we are artists after all, we should always respect the rules that some of these charts dictate, but allow some personal freedom!



PBR Theory for artists - 02

Introduction

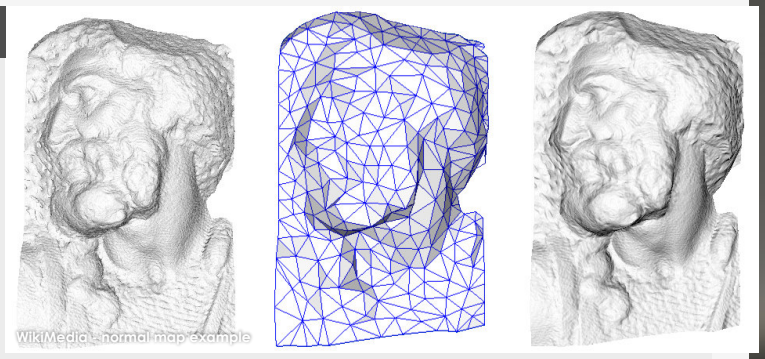
In part two of this introduction to PBR we will have a look at the remaining textures that we need to build our PBR shaders, there are some additional textures that we won't be discussing here, but they are also less vital to the creation of shaders.

1. Normal map

This isn't something new to the PBR pipeline specifically but I want to give an outline of what it is used for anyways, so people new to these textures can still get to grips with them.

A normal map is this weird blue/yellow/green texture, this is because this texture will fake the lighting coming of the texture and will add extra detail in most

cases baked down from the highpoly meshes down onto the lowpoly mesh.



2. Roughness

Arguably the most important map in this entire lineup, this map will control the porosity of the material, the more porous the material is the more the surface is going to be scattered with minuscule little details which are going to bounce the light in different directions and will thus alter the spread of the light being bounced back and will thus make the specular highlight bigger or smaller.

The closer this value gets to zero (or black) the more it will reflect the light back without being altered in any way and the closer it gets to one (or white) the more light absorbent it will get.

3. Specular/Glossiness

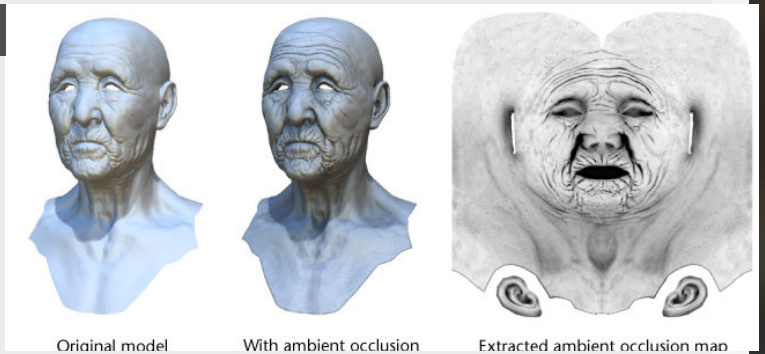
This is mainly used in the Specular/Gloss workflow and replaces parts of the Metalness and Roughness texture. This map controls the surface shine as well as the color of the shine it will reflect back from the surface. This means that we need to balance the shine of all the materials in this one map including both metal and non-metal parts too.

Lastly, this also means that if we have shiny metals we need to push the contrast in this map between metal and wood for example.

4. Ambient Occlusion

Ambient occlusion is something that has been here for a lot longer than the PBR workflow, this texture will fake shadow and is mostly know for grounding smaller elements together by adding some shadow on where different elements connect.

This is super useful to simulate shadows in crevices or other things that got added that require some shadow or depth.



Composition - Rule of thirds

Introduction

So it's been a while since we touched some fundamentals, so hopefully this one will help me kick-start a whole bunch of them! First up we have Composition, starting with a basic rule I lend from Photography and use in all of my work since, the rule of thirds!

1. Rule of thirds

What does it add to your work? The rule of thirds is a trick that's mostly used in photography to add a more dynamic framing and balance to your pictures, if you are looking for a nice framework to build your compositions from this is one of the best ways to get started.

It's fairly simple, we just divide the frame up in three equal division on the horizontal and vertical axis. Then we use the intersections of these lines as a rough guide for our focal points in the scene.

Unreal engine even has a built in tool for this, which makes it really useful to check your compositions on the fly under "Perspective

2. Balancing the horizon

Let's start with the beginnings though, the first thing that I do is balance my horizontal elements to offset everything from the middle of the frame, which is where the horizontal lines come into play.

The general rule is either using 2/3 for the ground, or use 2/3 for the sky, depending on what is in the scene this will also reinforce the feeling of things towering over you as things take up more space in your frame especially using the 2/3 for the ground rule.

3. Using the cross sections

Another thing that really helps me in my compositions is focusing on the intersections between these lines and putting the focal points of the scene on these (or close to these!) points. Using this connection help reinforce a composition that has space to breath and gives a more meaningful connection to these elements.

Also if you are working with this rule in mind and you are looking for a more dynamic composition it's better to use the diagonals, so using one of the points of the top and then the opposing corner from the bottom,

4. Creative uses of this rule

It's good to start with this rule, but keep in mind that all rules can be used in creative ways once you are familiar with them so for example you might want to have a central focused composition. putting the focal points not purely on the intersections, however even in this case the same rules still apply, because if you center these elements on both axis, the frame is going to feel static and create a different mood in the frame (see picture), which might be what you are looking for...

So play around with this once you are more comfortable with this rule, when in doubt, stick to the basics and keep looking for compositions in other people's work.

Composition - Different focal planes

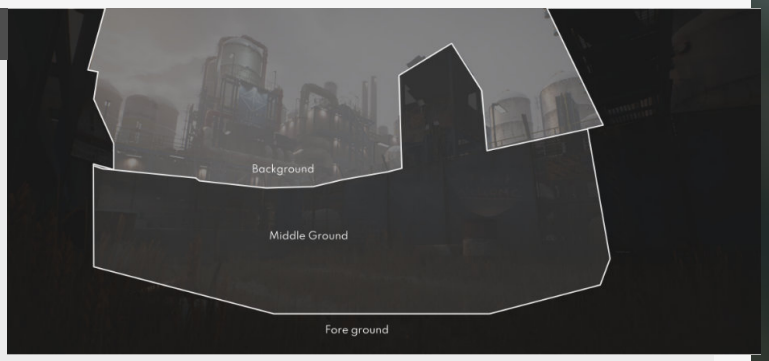
Introduction

Different focal planes can be used to divide your environments in clearly defined sections that can help create a more balanced and organized composition, in this tip we will have a look at different ways that these can be used, but you can play around with the rules a little bit for this one, so make sure to experiment.

1. Fore, middle and background

Another great trick that comes from photography and is also widely used inside concept art, however when creating your environments you can also use this separate between foreground, mid-ground and background.

This is helpful for focusing the attention of the viewer (or the player) towards the intended focal point of your image and also helps to create a more balanced composition.



2. Using the foreground as a blocker

The way that I love using shapes in the foreground is to emphasize the focal point that in the back more, we can use certain shapes here to guide the composition a bit better. This becomes really useful for your own personal scenes that focus on one or more camera angles, but don't depend on 360 rotation of the camera.

On the right you can see an example of this in use in one of my scenes.

3. Choose where to put the focal point

If we are going to focus on Environment Art, most of the focal points will either be in the middle or back. However, they don't have to be, you can play around with this rule, but make sure to understand if before breaking it.

On the right we can see a comparison between a scene of mine on the left and a scene made by Otto Ostera, which serves as a really good example of a different focal point on a different focal plane.

4. Creating some rest

Another major part of following this rule is that it will help you maintain a sense of order and rest in your scene, the person viewing your art is going to subconsciously identify these area's and it will help them understand the scene itself.

This will also help organize the details in the scene into clear separations, which is definitely a plus in bigger environments with a lot going on

Composition - Perspective usage

Introduction

In this one we will be looking at the main component of the scene when it comes to manipulating the view that the person has of your work, so for this one we will be diving into some tricks that we can use to affect the mood of our artwork though adjusting the camera.

1. Impact on the viewer?

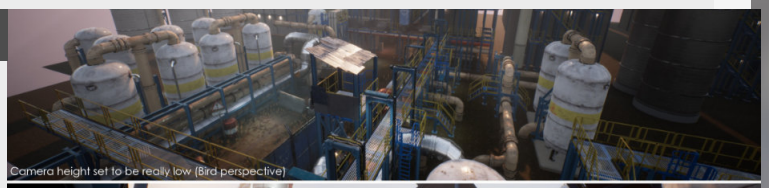
Changing the perspective of the camera can have a big impact on the viewer's mood and how they perceive the artwork itself, the impact of this can really move the viewer into a different emotion and be led through the picture in a completely different way.

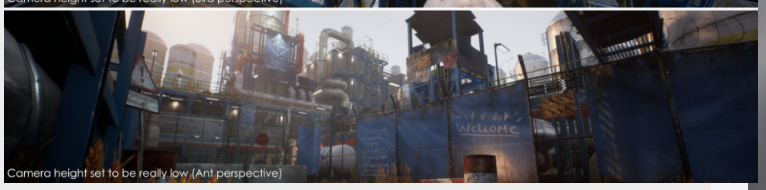
As environment artists we should not forgot that a lot of the power of our work comes from manipulating the the viewer's eye, meaning the camera. so let's have a look at some different perspectives that we can use to do so.

2. Birds vs ants perspective

If you have ever done a concept art class on perspective then these might sound familiar, the bird eye perspective is the perspective you take when flying really high above all the buildings, this is perfect if you want to create he feeling that your world is much larger then it is.

On the other side the ants perspective does the opposite, it can instill this kind of underwhelming or dominated kind of feeling, because all the structures that we add in the scene are going to feel like they are looming over the player.


Camera height set to be really low (Bird perspective)



Camera height set to be really low (Ant perspective)

3. Camera rotation

Angling the camera can have a dynamic or static effect on the viewer depending on the angle, most of the works we as environment artists will be making will feature a normal flat angle camera as the subject matter we present it static as well.

Adding a rotation to your camera can be used to create the feeling of unease or more typically it's used in action packed shots in movies. Then this might not sound like something you can use, but it might still be worth giving a shot as this is also not a rule that is set in stone you should always try and experiment with this and see if it makes you feel different about your work.


Forza Horizon Screenshot - Reddit

4. Field of view

Let's talk about the field of view of your camera, this can also have a big impact on how you perceive the level, it can make it feel super open, or it can induce claustrophobic to a point where the depth is sucked out of the image and it becomes isometric.

A normal FOV depends on the program you are using, but if we are using Unreal Engine this is going to be set at 90, going lower will push it more towards isometric and going higher will push it to be more 360. This in combination with a lower camera standpoint can really help you with making elements in your scene feel towering, as seen a little bit in my last scene "Last Bastion".


FOV set to 90


FOV set to 45

Composition - using contrast

Introduction

Cras in neque. Sed lacinia, felis ut sodales pretium, justo sapien hendrerit est, et convallis nisi quam sit amet erat. Suspendisse consequat nibh a mauris. Curabitur libero ligula, faucibus at, mollis ornare, mattis et, libero.

Aliquam pulvinar congue pede. Fusce condimentum turpis vel dolor. Ut blandit. Sed elementum justo quis sem. Sed eu orci eu ante iaculis accumsan. Sed suscipit dolor quis mi. Curabitur ultrices nonummy lacus. Morbi ipsum ipsum, adipiscing eget, tincidunt vitae,

1. Shape contrast

The first thing that is worth discussing is the contrast of different shapes, simply explained this is the language of the shapes that you are using, you can see clear examples of this in use in stylized games where dangerous areas are often indicated with spikes and safe around often have softer shapes to them.

For example round shapes often make you feel a safe where as in the opposite side of the spectrum spiky shapes feel dangerous, so when the player sees this they will want to avoid this most likely. Then in between we have square shapes, and most shooters especially the cover based ones have turned this shape into cover, because it feels safe and structural.

2. Value contrast

The next thing is looking at the different values in your scene and how they can influence the players moving through it. People tend to be attracted to area's that provide higher contrast and stand out more compared to other area's. For example people will be more likely to investigate area's that are darker if they have been running around in a light environment.

But this can also translate into light for example, where if we have a dark warehouse and we put one light, the player will go towards it because of the contrast different (and also to feel safe, I wouldn't want to be alone in a dark warehouse!)

3. Color contrast

We touched upon it briefly before but this is one trick that is often being used in games, and that is through the consistent use of color throughout the game players will realize that if they see that particular color they can go there to for example climb, or find lost treasure, etc...

You can see this being used in climbable ledges or ladders in Farcry 5 for example where they use a color that is bright to indicate such ledges. So if the player sees something blue on a cliff they know that there is a spot there that they can then use to climb it.

4. Texture contrast

This is a more nuanced trick to pull the attention and is more seen in Concept art then in Environment art, but that doesn't mean that you can't apply it here either. This trick makes use of being attracted to loads of details.

This for example means that if you have a calm landscape with nice round shapes and you put a highly detailed asset in the middle of it people will be attracted to this. This can really help guide the player through your environment, especially used in conjunction with all the other tips and tricks.

Thanks for going through this one!

70 - 30 Rule

Introduction

This rule is the foundation of so many things, giving you a good idea on how to balance elements in your scenes or props, but this can also apply to other things in your life too, say for personal schedule, marketing yourself, etc...

1. Overview of the rule itself

The 80/20 rule (or the 70/30 rule for others) is a rule that applies to a lot of things in life, but for the sake of this tip we'll keep it to art. So with this rule we determine the balance of details versus areas of rest in the image, on a prop or when building a location.

With 70 being the spaces of rest and 30 being the busy areas, this will lead to a better balanced environment that makes it more pleasing to look at or walk around in.

2. In Environment Art

So when it comes specifically to Environment art, this is where you can construct a scene that doesn't feel to overwhelming by applying this rule to the environment. So in this case we can apply the rule of 30 percent busy area that pulls the player or the viewer to these spots.

But for these things to stands out we need area's of rest to help with contrasting them against it, which is where the other 70 percent comes in to help us with this contrast.

3. Prop Art & Design

We can apply this rule to the creation of props too, where you add all these nice details on 30% of the mesh, you can really see this playing out really nicely with good hard surface design or nicely constructed props.

The idea here as with all the other parts is that you focus the player or viewer on the details you added instead of being lost by all the noise you add onto these props.

4. Applying this to other subjects

This can apply to a lot of other stuff outside of just the pure art creation, for example you can spend it on marketing your artwork or sharing it on social media too. Some have experimented with spending 20% of your time marketing yourself, your products, your art or whatever, when keeping 80% of your time to work on the stuff you love is really important for your mental health and the balance in your life.

Colors and emotional impact

Introduction

Colors are important for creating a mood and playing on emotions of the viewer and player. Colors can heavily impact the way we feel about situations, and in games where we have control over these colors we can really play with those to play on the heartstrings of people.

1. Colors overview

Colors have an incredible impact on people's emotions on a subconscious level, we can see that in a lot of real world examples going from fast food chains to the high fashion brands.

In fast food chains they are typically used to get you to start thinking about the food and capture your attention with it's bright colors so that you start thinking about the food even before you enter the restaurant

Different brands
Will use different colors to help set a specific mood

2. Color breakdown

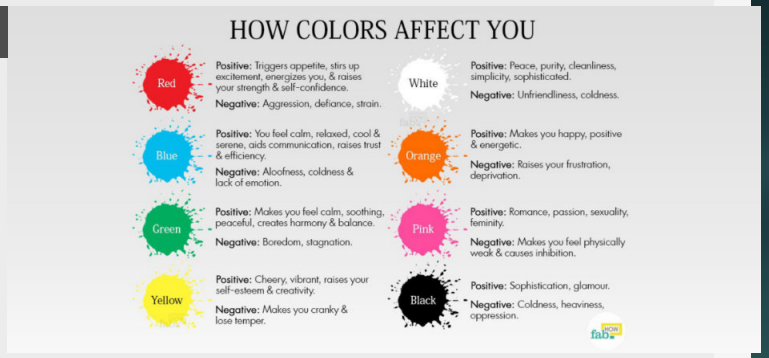
This image on the right does a great job at breaking down how the different colors affect your emotions. Notice that they can both have positives and negatives, which we will dive more into with the next point.

BLACK	GRAY	WHITE	BROWN	RED
Sophistication	Neutrality	Purity	Natural	Danger
Power	Stability	Cleanliness	Dirty / Earthy	Alertness
Mystery	Maturity	Modernity	Authentic	Anger
Formality	Security	Goodness	Richness	Romance
Evil	Authority	Hope	Simplicity	Excitement
Death	Strength	Freshness	Tradition	Energy
YELLOW	**GREEN**	**BLUE**	**PURPLE**	**PINK**
Caution	Life	Peace	Royalty	Romance
Energy	Growth	Stability	Luxury	Compassion
Cheerful	Nature	Calm	Passion	Beauty
Friendliness	Money	Tranquility	Magic	Love
Intellect	Envy	Integrity	Wisdom	Sensitivity
Cowardice	Sickness	Sadness	Extravagance	Femininity

3. Negative and Positive sides

Colors can affect you both on a positive and negative role. For example if we look at the McDonald's example again we know that they use red to evoke appetite and excite you, but it can also mean danger or anger in other cases.

So it's important to keep in mind that you reinforce it with some other elements, like using a nice orange sunset to bring new energy for the survivors, like seen in my last scene "Last Bastion"



4. Examples in scenes

This recent example by Peter Tran is a perfect example of how colors can completely change how you feel about the environment.

Change in colors impact mood
Don't forget about these, make sure that you have some detailed assets in there to add some nice visual interest

Shape language in Games

Introduction

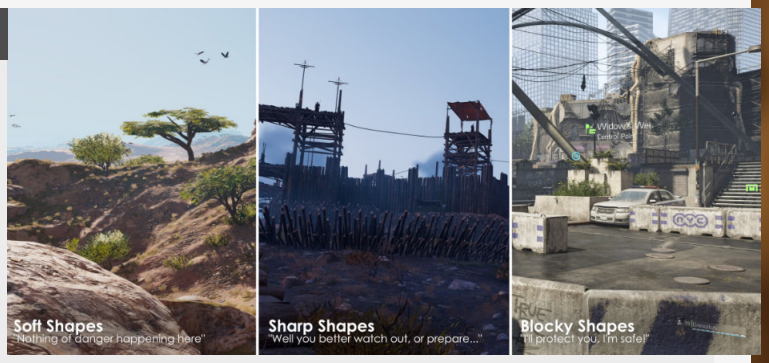
Inspiration for this has been taken from a really nice article provided by 80.lv all about this topic, so consider this a deep dive into that topic, but I'll do my best to give you a short and concise overview of the good stuff.

1. Shape language overview

What's this all about? Shapes have a big influence on our psychology and level designers and environment artists make use of this to influence

. But different shapes have different impacts on the players, so how can we make the most use of the different shapes we have in our tool belt?

Some assets will have predefined shapes when we stick to realistic games for sure but if you are working on a fantasy game, we can really broaden the shapes we can use and use them to put extra emphasis on the emotional impact.



2. Sharp shapes

Let's start with the easiest of shapes to identify are the sharp ones, these are most commonly used for indication of dangerous area's or foreboding danger ahead.

This is also why if you have camps build by baddies they are often surrounded with these kind of shapes to make them stand out from all the other more safe looking shapes that surround it. In this last example they also go hand in hand with more darker colors plus a dead looking landscape that has been stripped from natural resources. Really driving the point home.

3. Smooth shapes

Then on the other side of the spectrum we have round shapes that are directly linked with comfort and a safe feeling.

You will often see these in friendly hiding spots, camps or safe zones.

We can see an example for this on the main base in Wolfenstein.

4. Blocky shapes

Protective, sturdy and more seen in cover based shooter to really drive the point home. In these they often provide a gameplay element too, where you can visually see that if something has the rough height of cover (1m for half cover and 2m for full cover for example) you can take shelter there.

This is also reflected in architecture and other forms of design as well, where banks will often be housed in structural blocky buildings further emphasizing safety, but in this case for your assets instead of protection from bullets.

Where do I put my focus?

Introduction

With this one we're highlighting the importance of being able to focus on things that are important to focus on in your scenes or props to make sure that you finish your projects within a good timeline and without burning yourself out. Because not everything deserves the same amount of attention to look good.

1. Why does this even matter?

This might be a weird question to start out with, but people step into this trap more often then they realize themselves and it just disappoints me when watching people not finish or throwaway valuable work because they seriously underestimated how much time it takes to create a bunch of unique props with unique texture maps, especially when making a sizable scene. The beautiful/awful thing that happens on bigger scenes is that you also learn so much that new knowledge at the end might make you question assets you did in the beginning.

So here's a little guide on what to focus on when working on either props or scenes.

2. The focus of the scene

Pick the focal point for your scene, for a prop this is simple, it's the prop itself. However it's with a scene where it gets complex.

For me it all starts with a story, and the core of that story is where I want you to focus your attention too, making this stand out should be the main focus of your attention.

It doesn't have to be a story though, like in the images on the right it can also be the most important structure like in the work from Yinuo Chen.

But since these sections of your environment or props needs to stand out we should opt for some nice unique textures, dedicating a lot of time and love onto making this thing really stand out.

3. Supporting assets

Next up are the assets surrounding the hero asset, which still deserve a fair amount of time spent, but we can already start looking at more optimized or reused ways of creating these. This also depends on how important they are.

For example if we have a weapon scene, we should not consider this a supporting asset since it's so heavily tied into the main prop for the scene itself.

However if we're making a table diorama , then maybe you can make the table itself out of a trimsheet and details in a way where you can reuse them again. Like the second UV workflow for example.

4. The rest of the scene

Now we're getting to the items that are less visible, less important to the scene OR if you are working on your skills and upping your portfolio don't match with the goals of the project. With goals not matching I mean that if you are doing a piece on learning trimsheets it wouldn't make sense to then also make all the foliage in the background yourself, do yourself a favor and get those from a marketplace or reuse them from another project if you can.

This way of working guarantees that you're only focusing on the skills you actually want to improve, improve them and then move on instead of delaying the end of the project unnecessarily.

Texel Density

Introduction

or pixel density or texture density as commonly referred to. It's the measure to which all artists stick when they want to know how big they need to go for their texture sizes for all their assets.

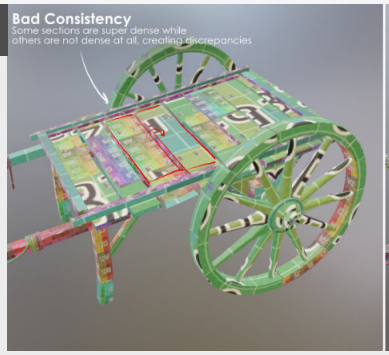
1. What is it?

Texel density (px/cm) is a number that correlates the size of our textures to the size of the assets throughout the entire game or your scenes.

"Texel" (texture element) is similar to textures in the same way that a Pixel is to your screen resolution size. It's the base unit measurement when it comes to textures.

This number then gets used to create consistency within your scenes by standardizing the amount of texture space gets used on all the assets in your scene.

It's expressed in the form of **px/cm** so we can convert the size of assets to texture sizes.



2. Why is it important?

We use this to create consistency between texture resolution depending on scene assets, but it will also stop us crazy artists from going completely nuts with adding way to high resolution textures to any asset we create that might be to small to get big textures.

A more player facing point of importance is that it retains immersion when in the game. When players see a really low resolution texture it can have the effect of pulling them out of the experience. Like seen in the example on the right.

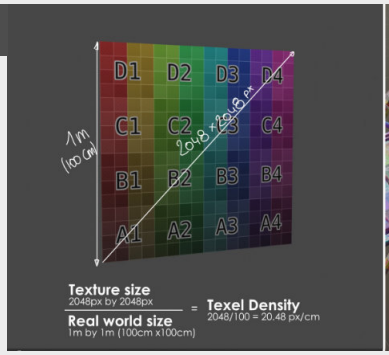
3. Deciding your texel density.

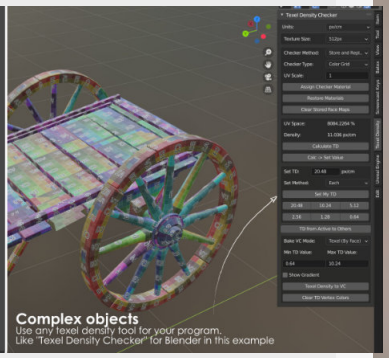
As for calculating it, let's start with the basics. A simple plane of 1x1m (or 100cm by 100cm), that has it's entire surface unwrapped to it's UV's.

Let's say we then add a 2048px x 2048px texture map to this for an example and we love how this looks in terms of visuals. Now to get our final texel density, we can divide the size of the texture with the size of the wall itself. Which looks like 2048/100, that gives us a texel density of **20.48 px/cm**.

Texture size / Asset size = Texel Density

We can then input this figure in most texel density calculator tools.





4. When to break texel density

Not everything needs to match perfect to the texel density set, there are assets in games that can break texel density set for the entire game as they are more or less important visually.

Some examples for these are weapons, menu items and cinematic props for higher texel density and background props for lower texel density examples.

Another situation where you change the texel density is when you are working with partially hidden sections that are not that visible but still need to be modeled, these sections that are smaller can also help you increase the size of other sections that are important to the asset.

Player Leading with Visuals

Introduction

Player leading is a term that's often used in the games industry as a term to guide players towards a certain goal or stage in the level. Player leading can serve different purposes and be done by multiple disciplines, think voice lines, level design, narrative, etc... but for today, we're sticking to the visual examples.

1. Scale and height

This is the most common example that you've already noticed, making something big is the best way of making things stand out in the distance and will also serve as a great reference point should you ever get lost.

A great and more recent example that takes this to a whole other level is Elden Ring, with a massive tree that almost takes up the entire screen serving as your main reference point for a lot of the game.

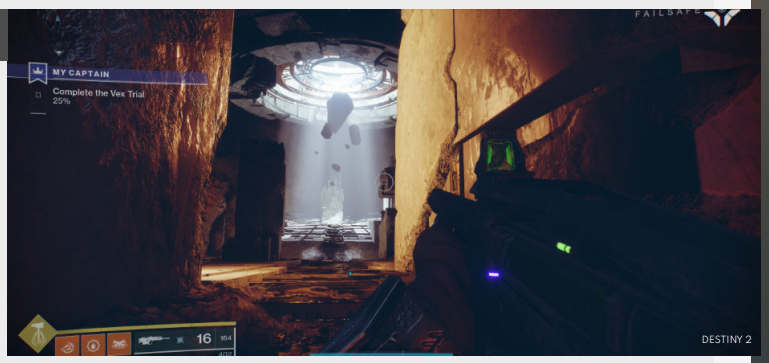
Additional tips can be found under **"Landmarks and Point of Interest"**

2. Composition

Obviously I can't skip a talk about player leading by talking about composition.

You can always direct and compose shapes and assets in a way where they highlight the object you are trying to lead the player towards, easier to achieve in linear games compared to open world games.

Composition is a deep topic onto itself, so you can always check out the different sections on this broad topic.



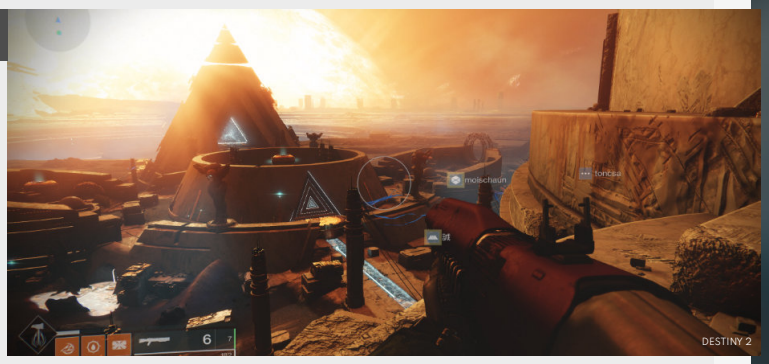
3. Colors

and especially the contrast between them.

This is something we used extensively for each location we worked on while working on the Special Operations in Farcry 6.

We knew that most of the world was dominated by foliage which was green. So we decided to pick a color that was as opposite as possible (blue for us, since we had other restrictions) and then used that in specific locations to pull the players towards that.

Again, coming back to Destiny this game also has countless examples of great use of color for directing the players.



4. Shapes

Can be used effectively to lead the eye, especially when used in contrast to each other.

Control is an amazing example for this, set with brutalistic shapes that make up a large proportion of the game. This then makes more organic shapes stand out by virtue of contrast. To add on top of that the developers Remedy Entertainment did a great job by infusing the more subdued brutalistic concrete with splashes of color, so really combine them both.

Again, we did a bit of a deeper dive on it in our tip on **"Shape Language"**

How is something made?

Introduction

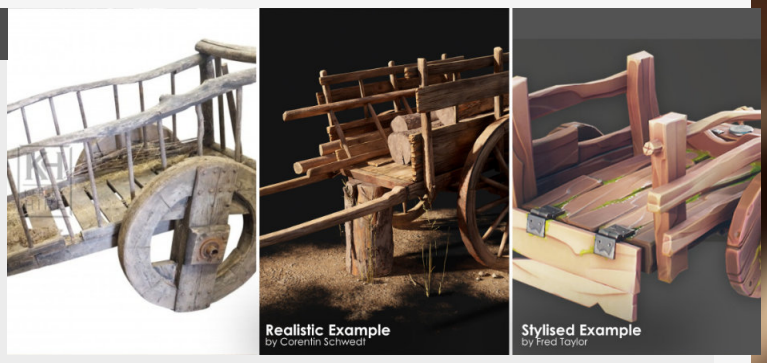
Figuring out how something has been constructing and the logic used to do so is an amazing way to get closer to creating an environment or prop that has both a lot of believability and character. This can be applied to both the props and environments.

1. Believability and relatability

We are constantly pushing ourselves to create more and more detailed objects, whether they are stylized or realistic. We strive for our artwork to be believable, and one way to achieve this is by grounding it in reality.

By doing so, we can lend a sense of relatability to the assets we create, allowing the viewer to connect with them on a deeper level.

Construction of an asset it one of the factors that helps us with achieving this and lend a sense of relatability to the assets we create.



2. Good references is crucial

Good references are crucial for accurate and high-quality work. We can turn to sources such as blogs, auctions, and YouTube videos. Gathering information from these sources allows us to make informed decisions when working on projects.

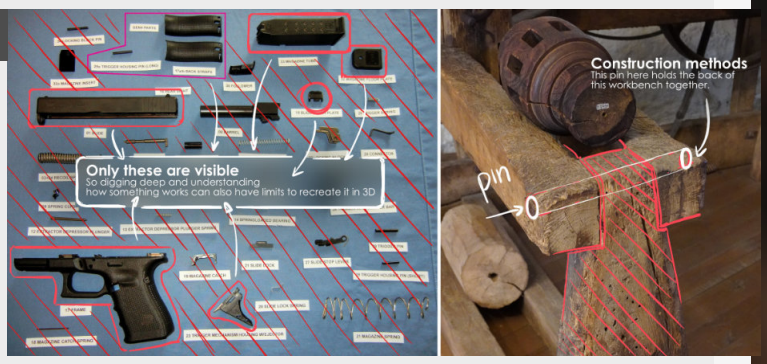
For example to create medieval assets, try finding explore niche resources. One great resource is www.thomasguild.blogspot.com, a blog about recreating medieval woodworking.

It's important when working with a subject, to find the exact thing if it's a recreation or use it as a solid base foundation.

3. Translating this into 3D

The challenge is deciding which aspects of the model are essential and which are not. For instance, when creating a rifle, the internal components may not be necessary, but understanding how they function and impact the textures can still be beneficial.

The level of detail required for the model typically depends on the desired style. If aiming for a simplified appearance, focus on the most critical features. For a realistic approach, additional details must be added and more time dedicated to considering how each component functions.

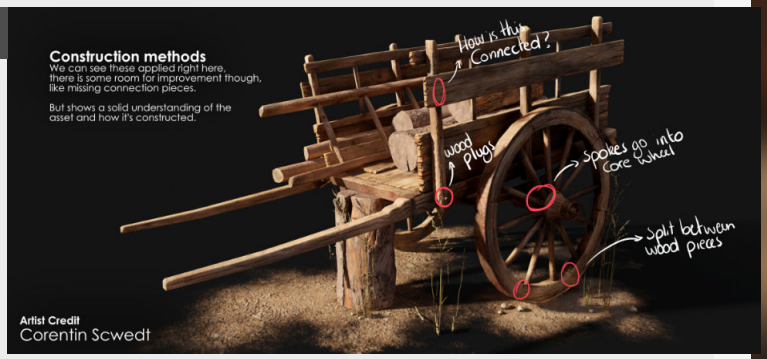


4. Some examples

Our wood cart is a great example of something that can have a lot of detail and character but usually we tend to keep it pretty simple.

There are some amazing people out there that still create them using authentic techniques.

On the right you can see how this research influences our model.



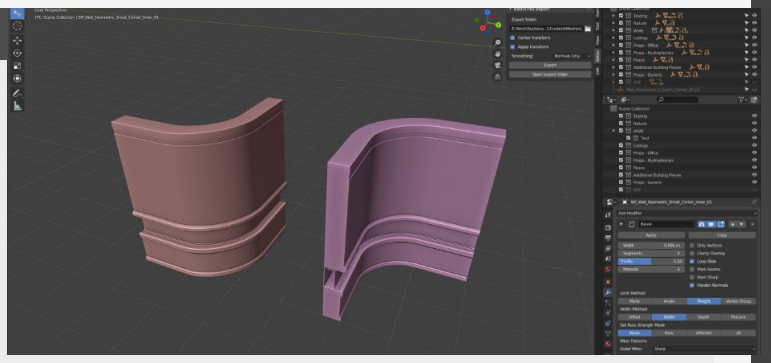
02 ENVIRONMENT WORKFLOWS

Destructive versus non destructive work

Introduction

You might have heard or even have been using this for a while now, but this is the difference between being able to go back to a previous state of the asset or not. More down below.

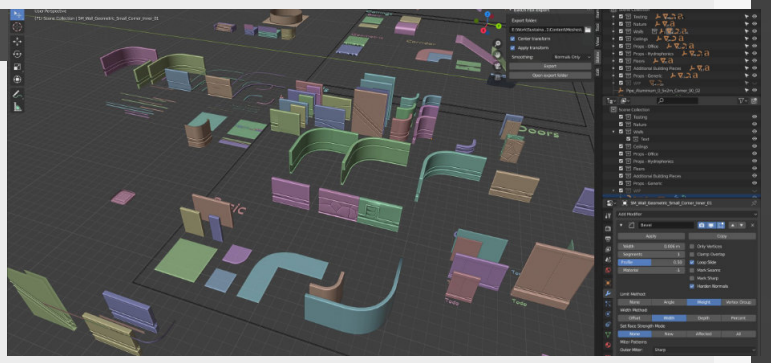
1. The difference



Destructive workflow is your normal way of working where you are just modeling things like you normally would and don't have the option of going back to edits that you have done further then your reverted states allow you to, so if you want to go back you need to start over or salvage things.

So logically non destructive is trying to keep the option of going back to previous States as much as possible, for example using an edit poly modifier in max will allow you to go back to states before adding this modifier. Another example is the curve modifier in Blender to deform a mesh with a curve, where you can always attach a different curve or delete the modifier.

2. Minimizing risk

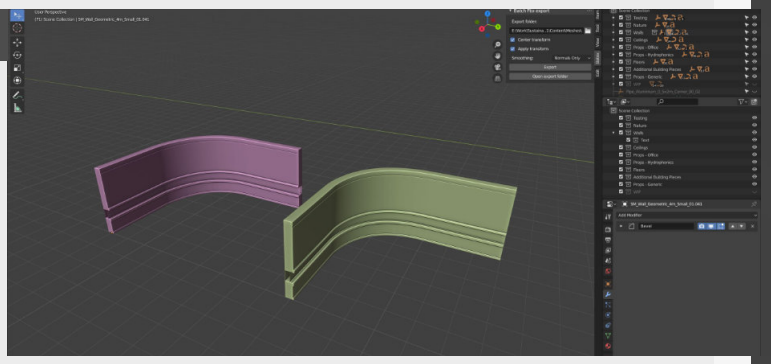


The reason why we do this is to minimize the potential risk if we want to do adjustments later on, it's way easier to do so when you only have to go back a couple of steps instead of starting all over.

For example when building a modular kit in Blender you can just predefined your curves for all the rounded corner pieces and adjust them on the fly, this will ensure all the pieces will have the same roundness.

Most modeling packages have some sort of options available but it might be a little bit less intuitive in Maya compared to Blender or Max for instance.

3. Manual Backups

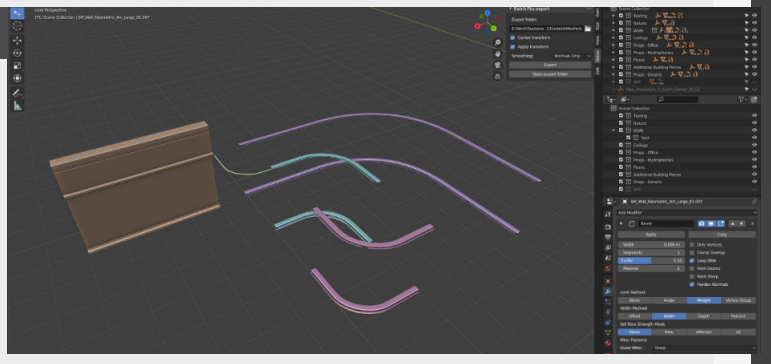


So in times when it isn't possible to continue with a non-destructive workflow the way forward is getting rid of the non-destructive layer and committing to the actual mesh. When doing this make sure that you create a manual backup by duplicating the mesh with all the modifiers attached to it.

This is going to save you a lot of time when working on a really expansive modular set for instance. You can also do this with incremental saves for the file itself.

I love to do is once I have a backup mesh is saving it on another layer where I store all the backup meshes for instance.

4. Different options



Some programs are better at it then others too. Houdini for example is amazing for keeping everything non-destructible because of it's node-based setup and you can go all the way back to the start input that you gave it and do adjustments or swap it out completely.

The runner up (from all the packages that I have used myself) are Blender and Max, they both have comparative ways of working with modifiers that you can adjust on the fly

So in short, wherever possible try and use these modifiers, layers or procedural where possible, keep things as fluid and streamlined as possible.

High to Lowpoly Workflow

Introduction

The bread and butter for most artists when they are getting started, especially when it comes to creating props that need a lot of details, like gameplay related items, weapons, or assets that are important to your world.

1. Overview

The core of the workflow is making a highly detailed mesh that has all the nice details, then make a lower poly more optimized versions on the same mesh, unwrap it (this is the 2D representation of your 3D object) and then **"Bake"** (transfer) these details from the High quality mesh (Highpoly) to the lower resolution (Lowpoly) mesh.

This workflow is mainly used for props and assets that need a lot of detail and are going to be really big on screen, this is why you see this workflow see used a lot in Weapons, Gameplay and object that are really important to the narrative side.

2. Pro's

The pro to this workflow is that this is the most typical pipeline that allows you to be really specific with the details and the amount of it too, which for assets that are important is great, because you can add a lot of unique details to them to make them stand out.

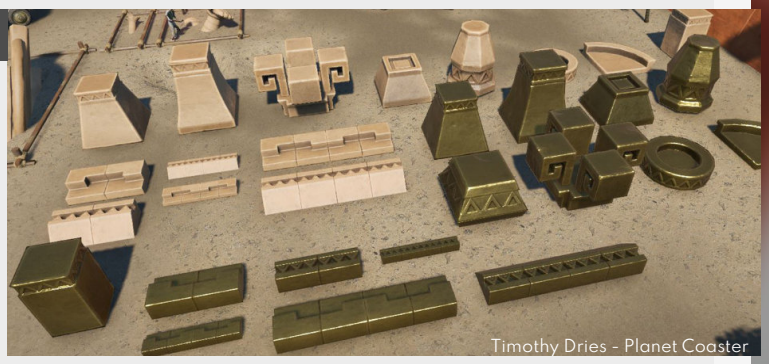
It also makes for an easier workflow overall, because you can work on these assets in a vacuum and in a lot of cases don't have to worry about sharing resources from other assets.

3. Cons

This workflow isolation also has a big downfall, especially in terms of performance, these textures are big and can really take up a lot of space and can possibly become a strain on other resources needed for other assets in the game.

Plus, these textures can only be used on this specific asset (unless you are adding some shader tricks like colorization or something) and it's UV's, so this makes it so that you can't really use this texture on other assets either.

Baking is the last one, because you need a highpoly object with your details to bake down onto your lowpoly mesh, which just adds a ton of iteration time by itself.



4. More on this topic?

Even tho this is the basic for of any pipeline there are a lot of things you can add on top of this base to make it more interesting.

Adding masks:
You can add different masks on top of your texture to add layers to it that you can adjust on the fly (Example on the right is a world based dirt layer on top of these unique pipes more on this can be found in "Adjustable pipe material" in our PDF).

Adding reusable sections:
You can also add sections of reusable meshes (like cables, bolts, etc... are great contenders for this)

Hand painted Workflow

Introduction

Popularized by World of Warcraft, this workflow takes back artistic control to it's roots in 2D.
The sheer time and dedicated you need to put into this is definitely the thing that makes this workflow so appealing to artists because it's so close to the core artistic approach.

1. Overview

The core of the workflow is basically you painting everything by hand or at least in part. A popular technique for when there was no normal map available yet and there we're strict limitations on the amount of polys you could use.

Painting everything into the texture like highlights and shadows manually requires a really good understanding of how light works in real life since you only have one diffuse map to work with if really limiting yourself.

Newer additions from this style break those limitations a bit, stylized PBR that adds normals and roughness maps, cell shaded that adds nice edges, etc...

2. Pro's

The main thing here is the artistic style choice really, it's a really handcrafted style that contains a lot of work and dedication to get it right. Which is also why it stand apart from all the rest and feels really unique.

Honestly, this workflow can be really creatively fulfilling as it's so light on the technical aspects that it's just fun to work within.

3. Con's

This style is energy and time intensive, which is probably the main thing that hold a lot of people back if you wanted to build an entire scene or world with.

World of Warcraft is definitely the staple example for this style. So if you are sticking to the older style of texturing this also means that you can only rely on a diffuse and emissive texture (if needed) so no normal or roughness/specular map was used in the early days.

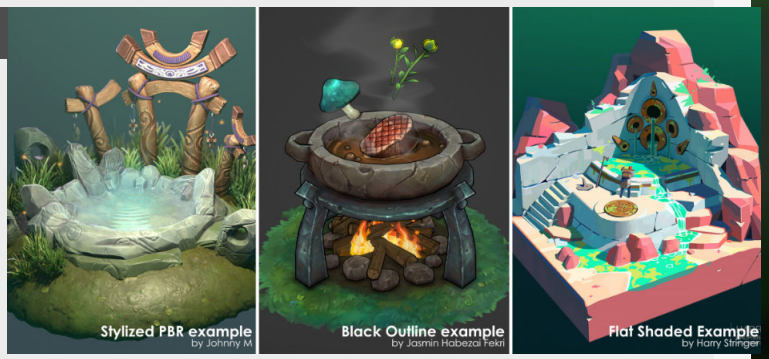
Since then we have more texturing techniques and new maps we can use in this technique so the style has evolved since then. But more on that in the next step!

4. Hand painted in the modern age

Stylized games and workflows are currently living through a full Renaissance.

There are so many different styles breaking away from the pure hand-painted look, like Stylized PBR (adding normals, roughness and ao into the mix), adding filters through Substance Painter, Baking information and using that as a base for your information, etc...

So you can really push your own inspiration into these styles with all these new additions, so try some stuff out!



Masks and tileables workflow

Introduction

A highly reusable workflow often used in games that have to render a lot of assets on screen, see open-world games. Making use of assets that already exist or are made to be highly reusable.

People can also call this the 2nd UV workflow (since we will be using a second UV to store unique data)

1. Overview

Large open spaces that are filled with assets are often limited in resources. A way around this is to share resources between assets that are less important then others on your screen by making use of tileable materials or shared trimsheets, and then adding additional detail layers on top of them by using other tileable materials. For example a metal bucket that is rusty or dirty.

The way that we blend these together is to use a new UV that can store a unique unwrap so we can store a unique mask in this UV, this mask can also be relatively small compared to your normal textures as you usually blend this with a different mask (higher resolution) for the transition between the material and the layer on top of it.

2. Pro

The high re-usability of the workflow is definitely the powerful side of the workflow, where you can make assets that become more and more versatile the more textures and resources become available.

A simple kit might start out with one type of metal, but expand to multiple types of metal later, combining this with welds, rust and dirt can make for a powerful kit that can be used on hundreds if not thousands of assets.

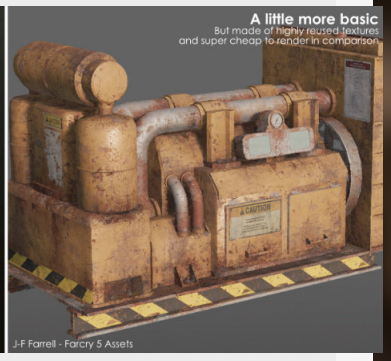
The amount of detail can also be decided on an asset by asset basis, making simpler assets easier to render and more complex ones more detailed as needed.

3. Cons

From an artistic perspective you can't really go all out with adding a lot of details in your props that use this workflow, unless you start adding additional materials, decals or layers to it (but then you are losing the performance gain of doing this in the first place).

But if we take a step back, you can't even see this "lack" of artistic detail on these assets, because of the density of the assets in modern day games and how fast paced a lot of it usually is anyways.

The other small side effect is that if you want to do some adjustments to the blending you will have to go back to your texture and adjust it there.

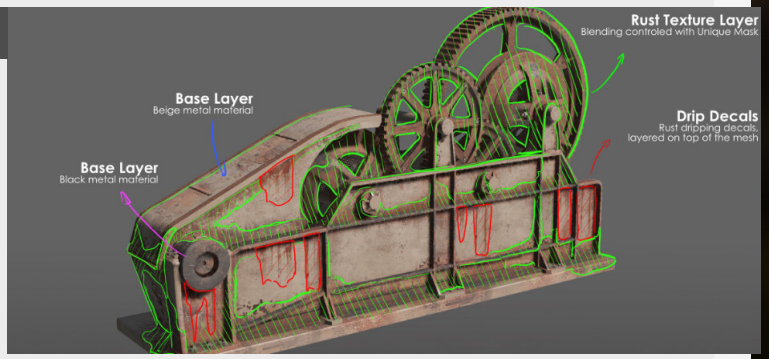


4. More information

You can find a complete step by step breakdown later in the document (if you have access to the PDF, you can find this on Beyondextent.com), which goes over the different steps you have to take to get through the whole process, from setting it all up, to the mask generation and the UV setup.

This is a really well used technique which can be seen in a lot of games, including Farcry 5 for example.

This workflow can also be combined with trimsheets, decals and more for additional details.



Hybrid Trimsheets Workflow

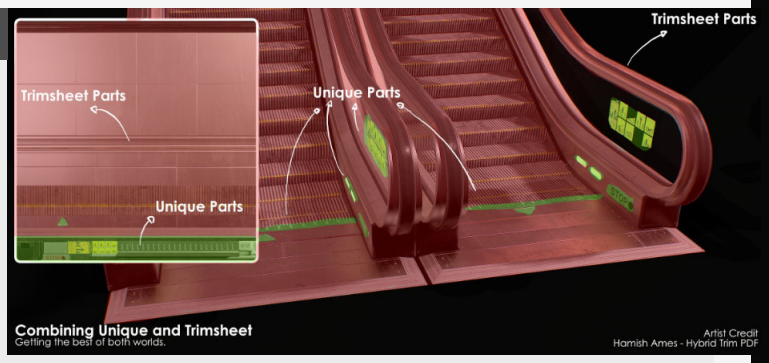
Introduction

This one is named as such because it's a combination of unique and trimsheet textures, adding reusable sections in the Unique Textures to make them a bit more versatile. Let's talk about it a bit more.

1. Overview

Using a combination of unique and trimsheet textures this is a great technique to make your trimsheets more unique with some additional unique elements. Usually we look at this the opposite way where we add reusable pieces into the texture to add more versatile sections onto the texture itself.

A good example for this are cables, wiring, pipes, etc... these need only a little bit of room on your texture to support the geometry it can be built with. So instead of filling out he space of the texture completely with a trimsheet we just need to make sure to add a little space for unique elements to allow us to fill them with unique textures.



2. Pro's

The big advantage to this technique is that you are adding bit's of unique texture to your trimsheet to make them more unique and get rid of the typical "Trimsheet" look. Which is great to give back artistic control over specific bits of storytelling or narrative that you want to add onto your models.

And even with smart use of your texture you can get more specific details like dirt or damage on your textures. This makes it perfect for unique textures that are to big to texture them with a normal unique texture.

3. Con's

The limitations of this technique are that we make the trimsheet way less shareable between different assets in one scene.

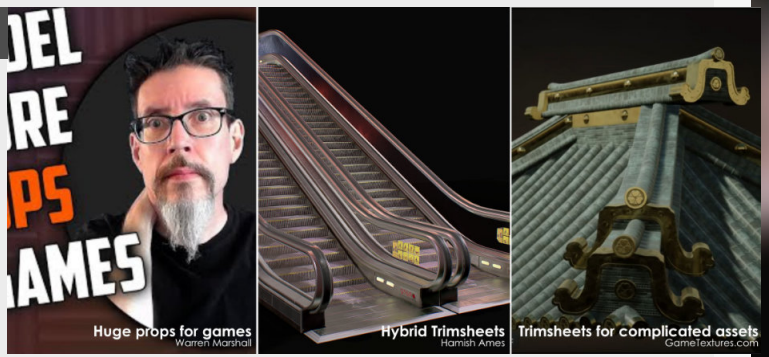
The more specific stuff we add to the texture the less usable it is on other assets or it becomes heavier on performance when doing so, as you are potentially loading in a texture that you aren't fully loaded in by other assets that don't need those unique details.

4. Additional Information

Usually this is something that people figure out through the limitations of using Trimsheets for a while and want to add a little bit of uniqueness to them.

For some more resources we actually had a workshop on this technique which you can find on YouTube.

Hamish Ames also offers a great free PDF on the topic as well, which you can find on his Artstation, so make sure to check that one out!



Detail Textures workflow

Introduction

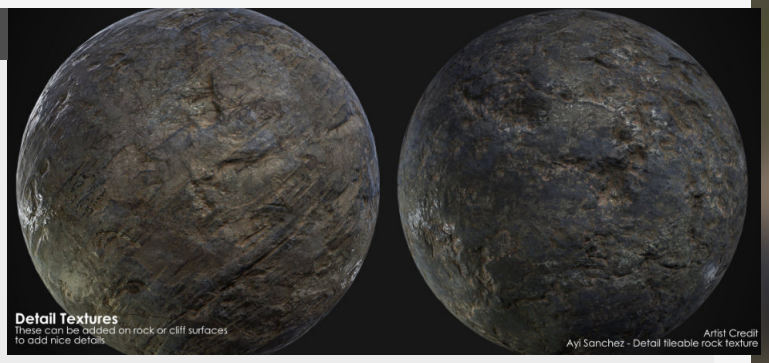
Sometimes also called secondary textures are mostly used on characters for skin, rocks or other natural assets that are too big for a base normal texture by themself.

By adding an additional layer on top of it we can add extra details on top of the already existing texture which makes it look way better up-close.

1. Overview

A workflow that is mostly used on rocks, cliffs or other natural hard surface elements. it makes use of your base texture, which in most cases is unique. And then adds extra details textures on top of that through the use of a detail normal map.

This is a great little tricks that is also scalable, because these natural assets are pretty huge most of the time and can thus not really support a texture that fit's their texel density most of the time.



2. Pro's

It's a great and relatively cheap way to add a layer of crisp details on top of your textures, these textures can also be reused on multiple different assets throughout your environment too.

Another great thing here is that you can control this layer independently from the base layer, and you can scale it on the fly if you want. Allowing you all the control and optimization you would want.

3. Con's

Usually this workflow is only used for these natural assets as they are very noisy by themself and thus the texture detail we are adding on top of it will work to it's strength here. That being said, this layer of noise can end up feeling, well, noisy!

So be careful on how intense you go and make sure that the textures you're adding on top of them are well balanced and don't add too much.

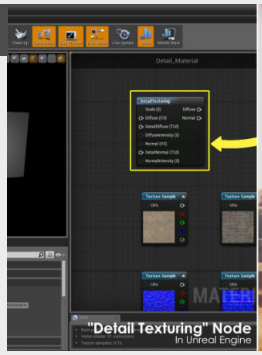
4. More resources

Mre on this workflow are usually found under **"Detail textures"** if you search for it, but it can also be found under **"detail normals"** as this is the cheaper and more widely used approach just with a normal map.

In Unreal Engine you already have a node available for blending detail and base textures together called **"DetailTexturing"** which also has some documentation under "Adding Detail Textures".

Unity also supports "Detail" textures as standard materials.

Casper wermuth also shows a more in depth version of this in his tutorial



Mid-poly (Star Citizen) workflow

Introduction

This workflow (often also called the Weighted normals and Decals workflow) is picking up steam just because of it's versatility and because hardware is getting more powerful removing restrictions on the amount of poly's that you can throw at it. The most typical example is Star Citizen ofcourse, but games have been using and continue to use different implementations and variations of these.

But let's give you an overview on how this differs from the traditional pipeline.

1. Overview

Heavily used in star citizen and probably the ones that are carrying the banner when it comes to this particular workflow.

The workflow itself uses a higher amount of polygons in general as instead of baking the bevels down from a highpoly mesh you just model simple bevels in and then solve the shading with **"Face Weighted normals"**.

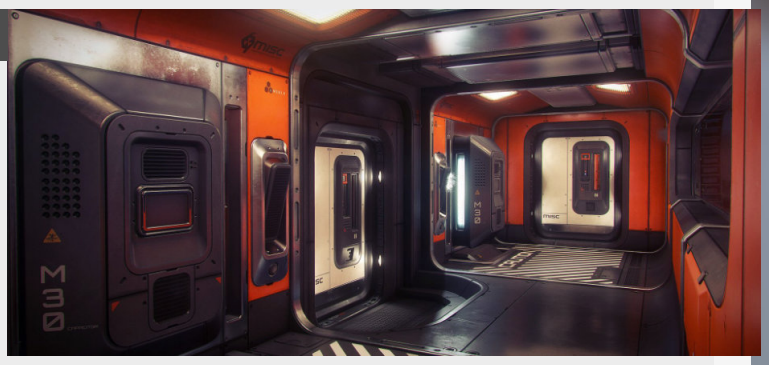
They then combine this with the use of tile able materials as a base and then add additional details on top with **"Parallax occlusion decals"** (more info on what Parallax is in our Compilation or blog #52)

2. Pro's

It makes it easier to iterate as you don't need to go back to baking with this workflow and it also allows you to combine lower resolution textures because you aren't relying on unique textures for details but add them with decal textures.

It also allows you to easily make adjustments on the shapes and work around feedback you might get during the process.

It's also way more scale-able since usually the base of it relies on a tileable material and a decal sheet for additional details. So if you need to change the base material you can just plug in another tileable material and you are good to go.



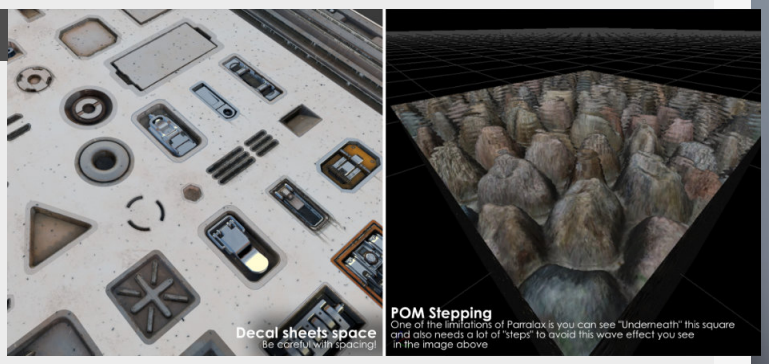
Matthew Trevelyan Jones

3. Limitations

The main limitation is in most engines that POM decals have a limitation in viewing angle, so this means that if you look at Parallax at really harsh angles you can tell that's it's faking the depth of them and you can even see the pattern in some cases.

From what I've seen Star Citizen has put some extra resources behind this to make this less of an issue, but you can still tell in some area's.

Also, a lot of planning needs to go into the decal sheet itself and can become tricky to add specific things onto the more it fills up, so a lot of experimentation or planning needs to go into this.



4. Looking for more on this?

Star citizen is the example for this and there are some great video's by digital foundry (https://www.youtube.com/watch?v=TUFcerTa6Ho&ab_channel=DigitalFoundry)

There is also a great breakdown on how to achieve this in Unreal by Thomas Woodward's blog (https://www.artstation.com/minimacman/blog/gwY2/high-detail-pom-decals-inside-ue4-brief-reference-guid).

But it's not unique to Star Citizen tho it can also be found in other games to varying degrees like The Ascent more recently.

Star Citizen

Reuse in environment art

Introduction

Let's dive into the topic of Reuse and Modularity for Environment Art, I wanted to call this Modularity in Environment Art at first, but people might get a bit confused because this comes with a specific type of reuse, so here we go.
Reuse and modularity is something that is well know in the games industry as the limitations are everywhere especially when building Environments. These will help performance greatly and are a great way to optimize your scenes and games.

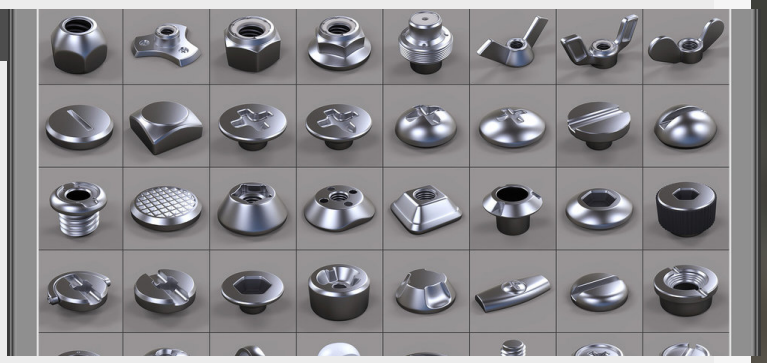
1. Texture reuse

First up in the list is texture modularity.

There are many different techniques that you can use when it comes this specific types. First one is the most common one, tileable textures, which are modular in their own right, because you can blend between multiple textures, scale them up or down and move them around. This applies to all the different texture maps, for example you can use multiple normal maps on the same asset.

Some other examples are trimsheets or channel packed textures.

2. Mesh modularity

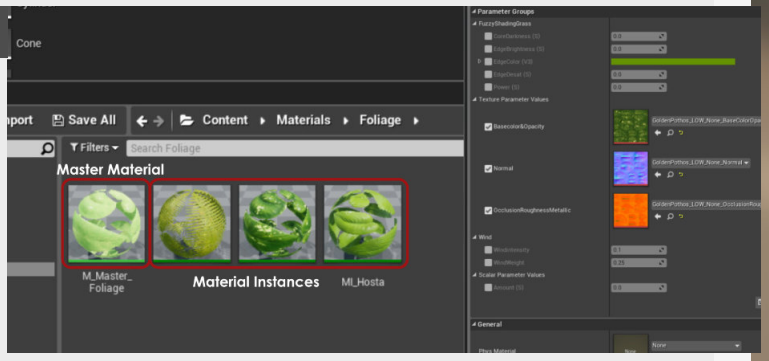


Mesh modularity is specific to modeling and Unwrapping, this stands for is the reuse of certain parts of the mesh. For a simple example, using symmetry on the mesh making it so that you only use half of the UV space compared to unwrapping it in full.

Another example could be cables, bolts, nails, etc... The generic pieces that are there to solidify the structural feel of meshes in the world. Making these over and over is just a waste of time, bake them all down onto a large decal sheet and keep reusing them on your models.

Additionally to this save the source file in your library!

3. Material use



The following section might be a bit specific to Unreal Engine as this is my preferred engine to work in, but can be applied to other programs as well in different ways. So if you are really looking to save your budget, you can texture a full environment using one master material and accompanying Material Instances, this will be way cheaper than using a bunch of master materials , plus will save you time creating them! We can then create some additional variation using vertex colors, masking, etc... This in combination with channel packing we can make a really performant environment.

4. Asset modularity

What we mean with modular environments is that we are trying to split up a full environment into multiple pieces that can then be reused in ways to make different sections of that environment.

So for example a building kit that is split up into multiple different walls, pillars, windows, etc... With all these pieces we can then build our full environment.

You can see these as big Lego pieces, we want to find a balance between the granularity of the assets and the quickest building experience.

2nd UV Workflow - Base setup

Introduction

We're diving into the workflow that I recently started using in my personal projects. So if you are using tileables or trimsheets and are wondering how to make them look really good with additional layers, then this one is for you!

1. Overview

If you're using trimsheets or tileables you will quickly start to notice that they feel flat and don't allow for additional details to make them look as refined as unique textures.

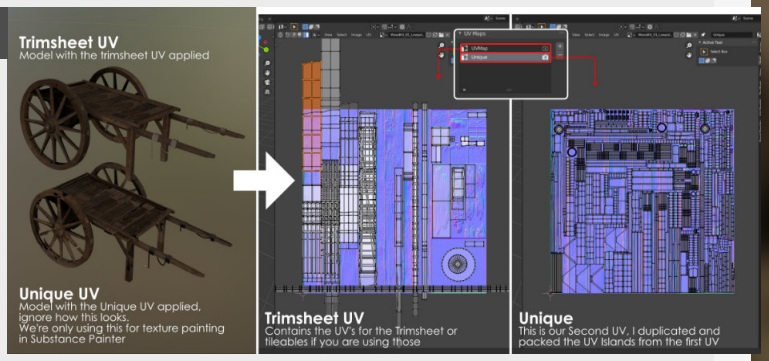
This is where we can start using a **second UV** that will give us a unique unwrap for the object, and we can use that to layer multiple masks on top of these trimsheets or tileables.

There are some new costs that come into play though, one is the extra UV set and another are the masks that we use. These masks are usually packed in different RGBA channels and are often lower res then the actual texture resolution. (More info on what **"Channel packing"** is can be found in the our Environment Art collection.)

2. Setting up the second UV

Go to your asset that you want to use and make sure that you've unwrapped it properly for your trimsheet or tileable already, this makes it easier to make a second UV channel.

In my case using a trimsheet I can quickly and easily setup a second UV channel because I unwrapped my first UV already an squared most of the UV's, so I can just pack all the islands in my second UV channel. We're looking for a nicely packed and unique unwrap that doesn't waste too much space.

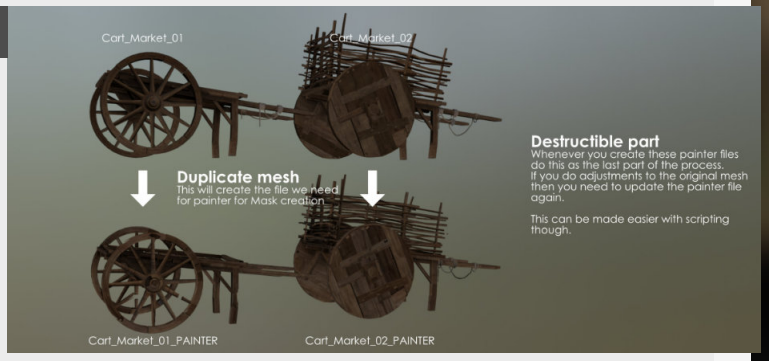


3. Destructible part of the workflow

Now that we have this setup and because Painter only allows painting on the first UV map (unless using UDIM's) we need to duplicate the mesh and then get rid of the Trimsheet UV so we can start painting on the Unique UV and building up our masks.

The reason why I call this the destructible part is because this duplication will mean that if you adjust the UV mapping of the base you will need to redo the duplication part. Minor adjustments are easy to do but adding new things will force you to go back.

Keep this in mind and do this once you are happy with the Unwrapping of the trimsheet/ tileable layer of the asset.



4. Setting up painter file

Set's setup the different channels that we will be using, for this I use the Custom Channel **("User")** 1-4 for the different Channels that will eventually be packed into a single texture.

I have multiple groups setup, one for each channel and then build up the masks with Generators to make it easy to drop in other assets and have the same sort of outcome.

This part can improved a lot tho, you can build your own smart material to handle this and also you can automate a lot of these steps so you don't have to do it all by hand.

Which I want to experiment with more later on.

2nd UV Workflow - Mask Generation

Introduction

This blog entry follows last weeks introduction to this topic **"2nd UV Workflow - Base Setup"**

Now that we have the texture done and we have the models setup we're ready to dive into Unreal Engine, where we're going to finish this little breakdown with our setup in Unreal Engine.

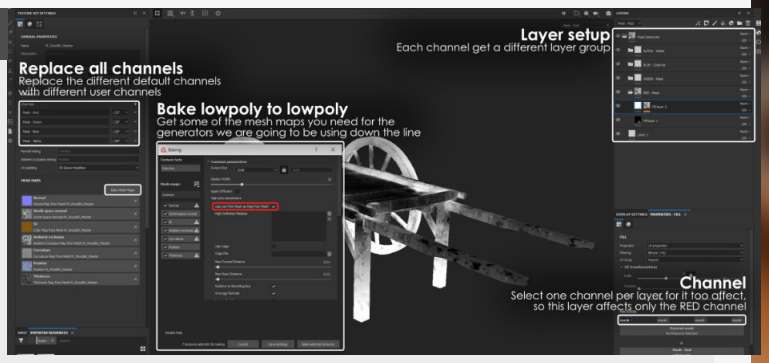
1. Starting out

Baking lowpoly to lowpoly

We do this to generate some information that we will be using in the generators for our non-destructible mask setup later, the masks generated there such as AO, Curvature, Position etc... will then be used in these.

Setting up groups for different channels

Now for every channel that we eventually want to pack in our texture we set up a different group so that it's easy to recognize which group is for which channel too.

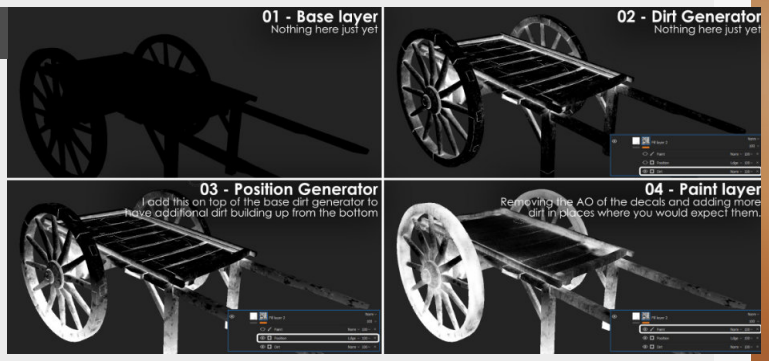


2. Keeping it non-destructible

Using Generators
Really useful to start. In this case I'm looking for a dirt mask, so we start out with a **"Dirt"** generator and adjust some of the values to where I'm happy with it, so lot's of dirt for the dirt mask and less dirt for the moss for example.

Tri-Planar Projections
these doesn't make advantage of all the maps we baked in the beginning and is just a projection from all sides on top of the assets.

Filters
These can add on top of the generators and projects to make them more unique and break away from the generator look.



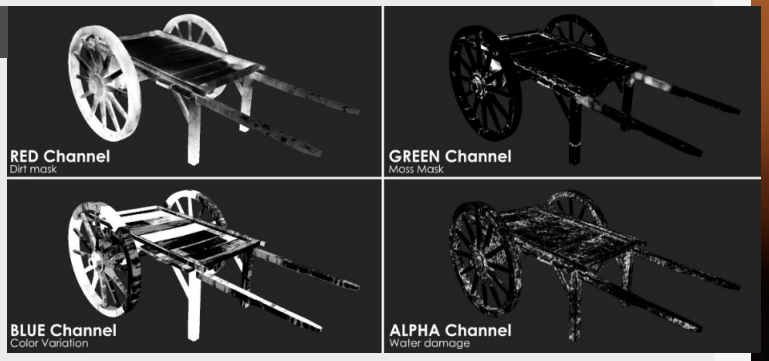
3. Different masks

So every mask that I use is built differently in this same way, for this prop the **RED** = dirt, **GREEN** = moss, **BLUE** = color variation and **ALPHA** = water.

RED and **GREEN** are roughly the same with a dirt generator as a base and then projections, filters and a paint layer for little personal details.

BLUE starts out with a **"UV Color Random"** at first, this gives each separate UV islands a different shade of gray (that we use in Unreal as color overlay strength), then we can add additional tweaks on top of it.

ALPHA is for water damage, just using different alpha's as projections, to make it look more like water damage.

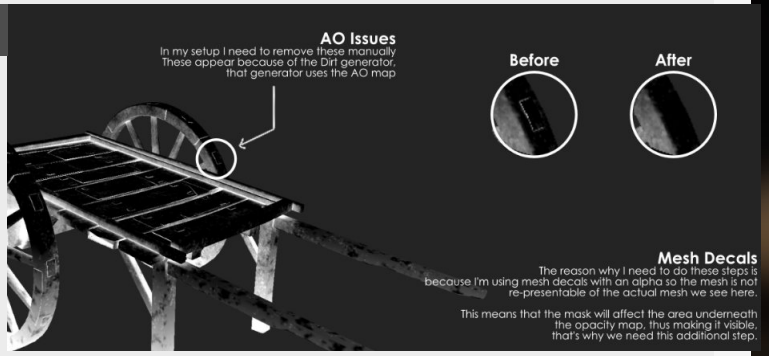


4. Details and other meshes

If we want to add additional details to this setup we can add a paint layer on top of the stack and paint in some additional details to make them more unique. For example I need to go in manually and remove some of the AO that is baked, because it will show the mask through the opacity.

Now we can just drag in the asset, bake lowpoly to lowpoly again and then we have the same sort of texturing style as the other props too and allow you to quickly go through a ton of assets.

Just make sure to save your asset under a different name if you want to come back to it.



Second UV Workflow - Unreal Setup

Introduction

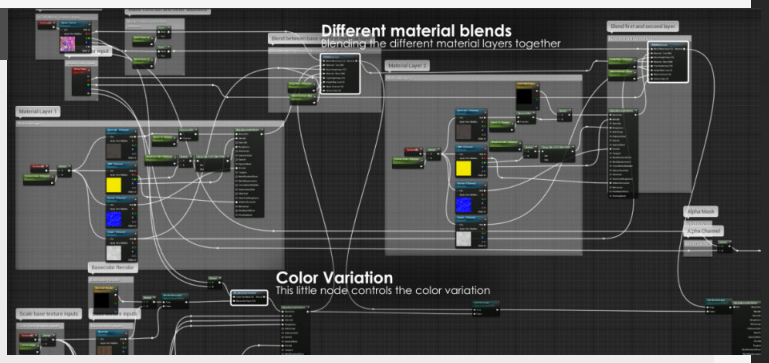
This blog finalised the series on how to use the 2nd UV workflow, this time being specifically for the shader setup in Unreal Engine. We'll dive into the shader setup, some material functions, additions to the shader to make it more customizable and more...

1. Unreal Engine Shader

This might seem like a big bowl of spaghetti right now, but in fact it's not really that complicated. In this one we are going to break down.

We have four different section here, two are for **blending material layers** that use textures, one for **color variation** and one for **water damage**.

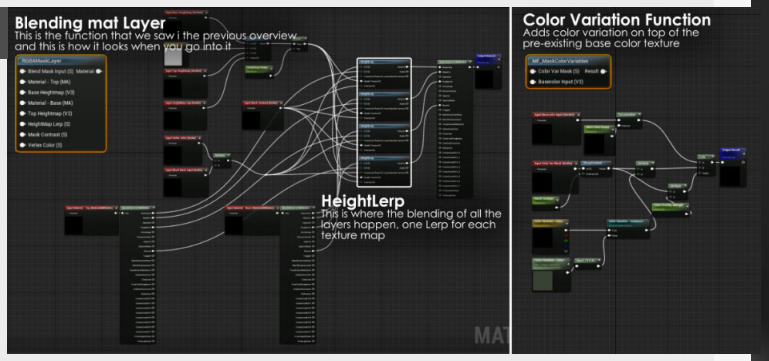
The biggest section are definitely the materials layers that are blended, a lot of it is hidden inside of the material functions which are the bigger nodes that have a lot of connections.



2. Material functions

Material functions allow you to combine a bunch of nodes that you reuse over and over in a separate setup. The function takes all inputs and blends them together. (basecolor A with basecolor B, etc...) using a Height Lerp, then I add a multiply of the vertex color and the mask we get from Substance painter and add that to the transition phase of the Height Lerp. (You can also use a normal lerp node).Then the output of these nodes we then combine into a material attributes node again so we keep it nice and clean.

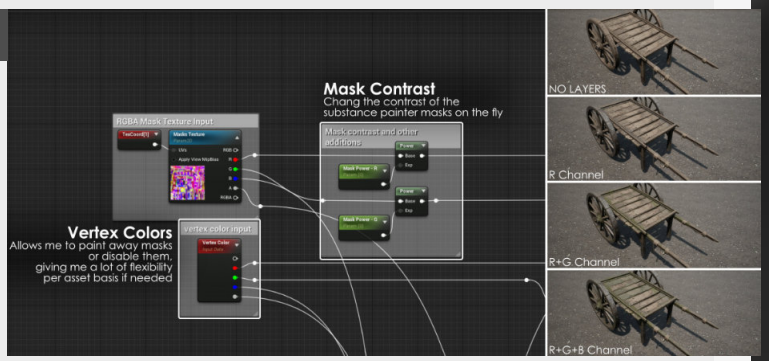
I also use a function for the color variation, taking the basecolor texture and adding a color on top of it based on the mask that we get for Substance Painter.



3. Customisability

Mask Contrast
Adding some simple "Contrast" after our masks we get from Substance painter will allow us to change how tight we wants these masks. Keep in mind that we need to have some Grey values to give us something to work with when it comes to increases in contrast.

Vertex painting
I like to build a lot of modifiers/ customisability into my materials, so we already have these masks that we get from substance that are really amazing, but I'm taking this a little further by adding vertex paint to it too, this allows me to disable different layers based on vertex paint on asset instance base.

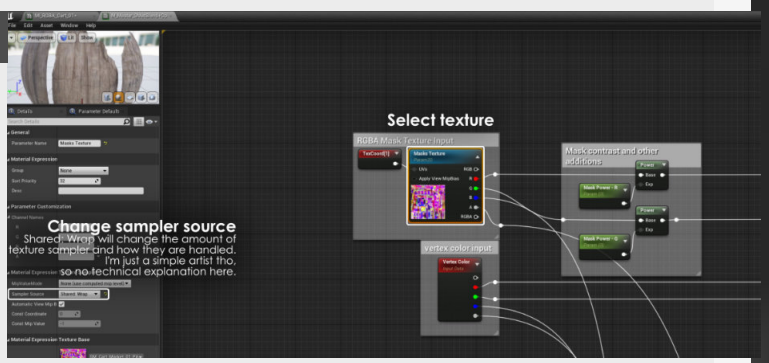


4. Limitations and workaround

The reason why I don't have 4 different texture layers setup is because there is a 16 slot limitation to the amount of texture samplers that you can use inside of the material.

However, you can avoid this by ignoring this limitation if you're only building for PC anyways.

You can do this by setting the texture sampler's to **Sampler Source > Shared: Wrap**.



Planning your trimsheets

Introduction

Trim sheets are the bread and butter for environment artists, so today let's have a look at how we go abut planning them. We'll be diving into what makes for good trimsheets, building for different scales, how to start blocking the shapes and more.

1. Good Trimsheet conditions?

The main thing I look for when thinking about trimsheets are:

Reuseability:
If you have a scene with loads of recurring elements then you know you can turn those into trimsheet elements.

Adjustability: Use your trimsheet in more then one way by just adding a paint layer on top of it for example through a shader.

Feel free to mix it up, my example will be dealing with wood in isolation, but you can combine multiple materials into one comprehensive trimsheet to even texture a larger section of your scene.

2. Break down different scales

First, let's find the biggest features that can be textured with our trimsheet and then find the smallest thing and then make pieces that can accommodate both of them.

So in my example the biggest thing are the thick wooden beams and the smallest section will probably be the sides of the planks.

The next step is then bridging the gap in between them, we want to do this to allow for stretching of the different UV islands if we need too.

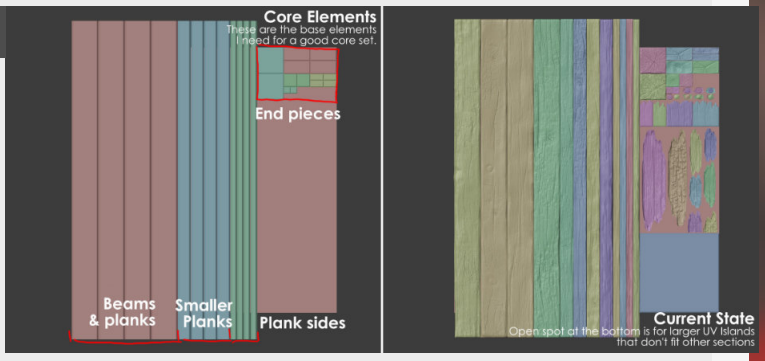
I did this by simply scaling the larger section in half, and then we have one shape to bridge the gap between large and small.

3. Blocking in shapes

So now that we have something to work with it's time to start blocking in the pieces (you can also do this in 2D! if you prefer and are faster in it) So first of all let's just make a plane that we have can use as a guide for our different sections.

Then we can start blocking in simple sections with primitives to give us an idea how many pieces you need to make the set work for you.

Make sure that these sections also stick out from the template if you want them to tile in either direction. For example my planks are longer then the template to make sure the edges continue if I we're to add longer sections.



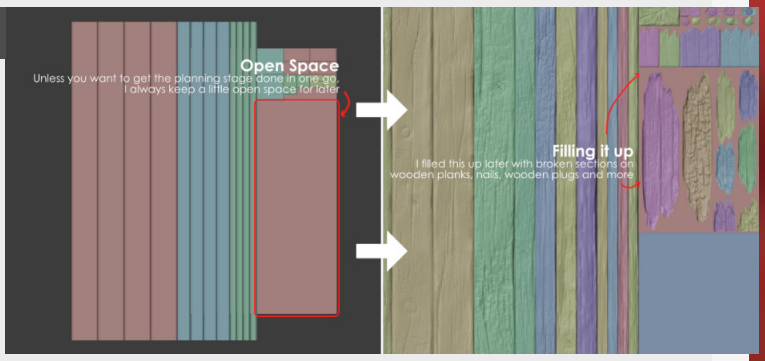
4. Keep some open space

Last trick, but such a good one to keep in mind when building trimsheets.

Sometimes in the later stages you notice that you need some extra space on your UV set to add some little sections, like for example adding little nails, bolts and broken sections on my trimsheet.

If you don't have any open space anymore on your trimsheet you are going to have to either move the existing stuff around or waste budget by adding a new texture for these items.

Have fun creating!



Trimsheet - Adding details

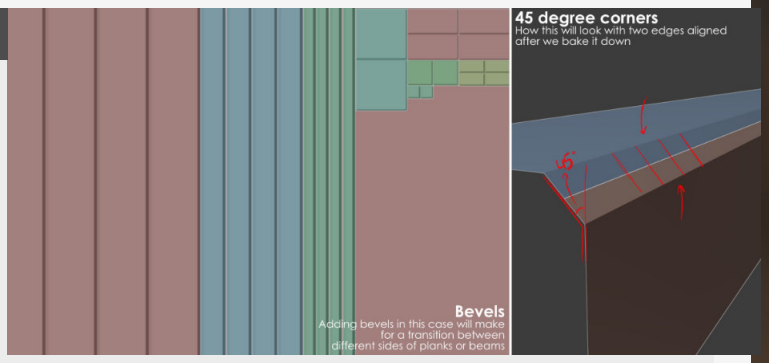
Introduction

Building a trimsheet isn't as daunting as it sounds, so let's dive into some techniques on how to add more details to these pieces and how to prepare them for our baking step.

1. Adding bevels

This is following straight from the last weekly tip we did abut trimsheet planning, at this stage we already have a blockout of the shapes that we want to have in our trimsheet. It's time to start refining them a bit more. In this case, because we're working with wood, we want to have a nice transition between the edges of the wood going to the top sections of the wood.

We can do this with adding a single bevel to all of our edges. This will make it so that if we bake that information down we face a bevel into the texture, if we do this on both sides then they should align up nicely.



2. Adding Details

This also differs from trimsheet to trimsheet, some hard surface ones don't need a details pass, you will most likely doing that from the start anyways, but for this specific trimsheet there are some additional things we need to keep in mind when we're doing a details pass on it.

For example, those nice bevels we added before, we should try to keep them intact, especially at the edges. If we don't do this then we might end up running into issues where the two sides of a wooden beam meet.

3. Mesh decals

Trimsheets can only be the trims we've been working on until now, but you can also be a little more creative and add some details to your trimsheet too, I chose to do this in my example so we can get some extra damaged sections we can add as see-through geometry on top of our normal geometry.

We need to keep spacing in mind for these meshes tho, if we put them too close to each other we run into issues where we might need to add more geometry in the lowpoly to be able to cut them out without intersecting with any of the other mesh decals.

4. Baking it all down

Baking distance: First thing we need to keep in mind, we need to make sure that our bake distance is large enough, so in this case that is a plane above our lowpoly that will capture all the information in between.

If Baking opacity: You will need to do some thing to make it easy on yourself, you can find more tips on this in "Substance Painter Opacity Map Bake" or (Blog #97)

AO causing issues: Especially when using floaters, make sure that you give them enough space between elements to AO baking won't have an effect.

Trimsheet UV Tricks

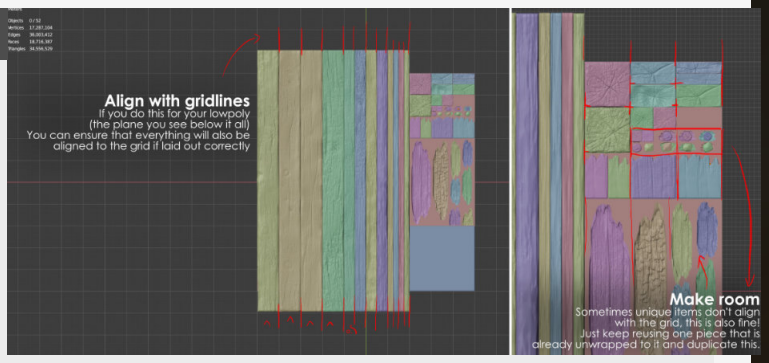
Introduction

This time we're going to be focusing on UV tricks that you can do when you already have your trimsheet setup and ready to go. If you are wondering how to create them, check out the previous entries about planning, and creating a trimsheet.

1. Use Snapping

This time we're going to be focusing on UV tricks that you can do when you already have your trimsheet setup and ready to go. If you are wondering how to create them, check out the previous entries about planning, and creating a trimsheet.

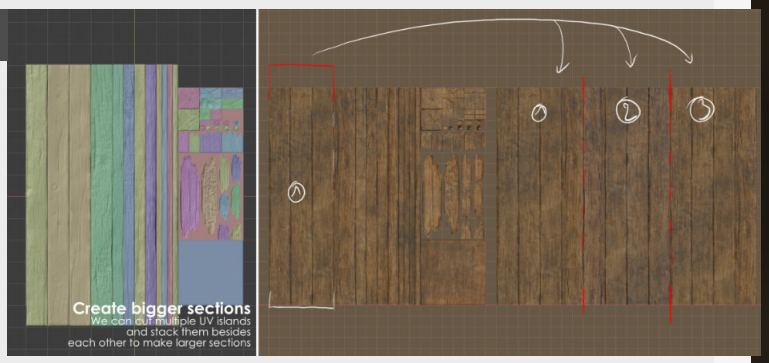
Snapping will greatly increase the speed of which you can iterate on getting UV Islands to align with your textures, this is especially handy in our case if we want them to snap with the sides of the wooden sections.



2. Making tileable sections

Working within tight restrictions and limitations is often times the thing we do most so saving a texture can make a big difference. So instead of making a fully tileable texture of the item we need, maybe there is already something on our trimsheet that we can use to make a tileable section out of.

For example in my case we can use a section of these 4 beams to make a tileable section out of, if we then add a couple of these right beside each other we can get pretty far!



3. Curves and bend

what about curved sections? Well, don't overthink it too much, we can just add geometry to it and start deforming them without touching the UV's.

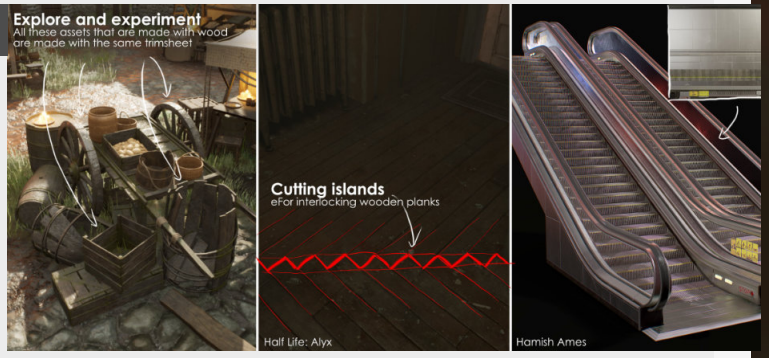
We do need to add a bit more geometry if we use this method because we can't fake a nice curve with a unique texture like we would do in other cases, so adding more geometry is a solution to this.

4. Be creative with it

Don't let these couple of tricks stop you from trying something else tho, you can do a lot of amazing stuff with trimsheets and a little bit of geometry.

And also with combining them with unique sections or mesh decals you can build a texture that is super versatile and allows you to texture a lot of the assets in your scene.

Another interesting example is the interlocking bits of geometry to create an interlocking wooden floor like seen in Half life: Alyx

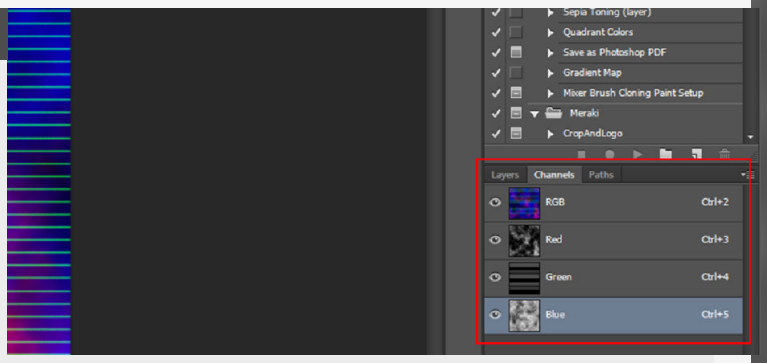


Channel packed textures

Introduction

Channel packing is one of those great things to save texture memory and reduce the number of instructions that materials use inside of Unreal Engine, which will reduce the footprints on the memory for them.

1. Channel separation in Photoshop

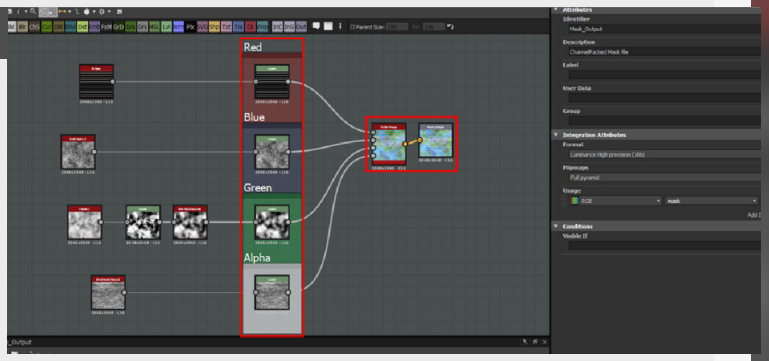


You can find the individual channels inside of the channel menu (next to the layer menu by default) and if your image > mode is set to RGB or anything not grayscale.

To copy these individual masks it is the easiest to deactivate all the other channels that you don't want to affect.

It is the easiest to copy them over from another PhotoShop file into this file.

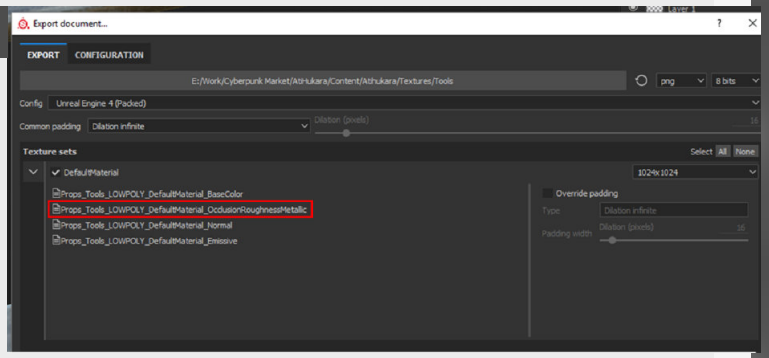
2. Substance Designer



If you are doing texture generation inside of Substance designer then you need to keep an eye out for the RGBA - Merge node, this will combine the selected grayscale maps into a single texture.

This can then be used in the same way as the other methods, but is a far easier method.

3. Substance painter

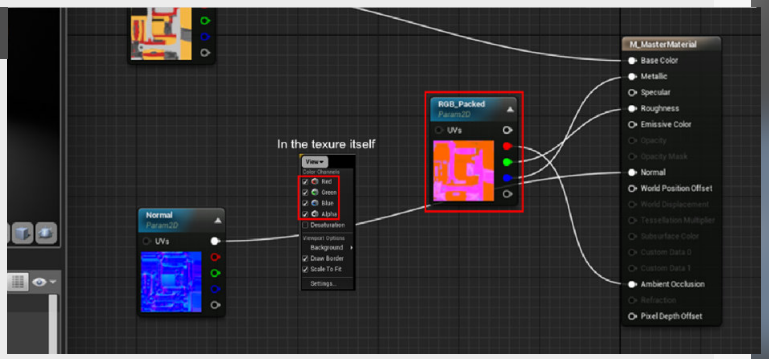


There is already a pre-existing preset for this to export all the different channels to combined textures.

The one that I use (because I use Unreal Engine so much) is ofcourse the Unreal Engine preset.

This preset makes it sure nice and simple to export all the maps you need and then hook them up in the Unreal Engine material.

4. Texture preview



Now that we generated the texture and imported it into Unreal Engine we can have a look at all the different channels using the Texture Preview.

To do this, you need to double-click the Texture and in this view you can click the little drop-down menu on the top left and there you can filter the different channels.

Pipeline Introduction

Introduction

This allows you to copy and past bits or entire materials to a text file. You can also do this with blueprints as well, allowing you to quickly save and share important bits of materials and code.

A really handy trick because it is so simple in it's use.

1. Pipeline overview by focus

MODELING

Hard-surface:
3DS Max and Maya are probably the most know or typical 3D softwares used in most modeling pipelines in studios. Blender is good example for a really solid upcoming 3D package with integrated sculpting options and an increasingly interesting toolset with a strong supporting community offering a bunch of plugins. Other options can include Modo ,Sketchup ,etc...

Organic:
When looking at making organic models the first one that comes to mind is Zbrush known for being an innovative sculpting program and really nice for retopping and optimizing your meshes. Some other programs such as Blender and 3Dcoat, but as of writing this, Zbrush is still the most known program. There are other programs like speedtree, tree-it, etc... that are suited for making something specific.

Procedural:
A little section to the side, but probably the most important one going into the future are procedural tools like Houdini and other integrations into programs like it.

TEXTURING

Baking:
Let's start out with Normal Map baking, Substance Painter is probably the most straightforward option because you can bake and texture in the same program, but there are some other great options like Knald or Marmoset Toolbag with nice realtime cage adjustments and skew painting.

Painting:
Substance painter is the king when it comes to building your own textures, especially because Quixel Suite is no longer being actively worked and and replaced by other standalone programs.
The non-destructive masks and generators workflow that Substance painter uses is really nice and will allow for quick iteration.

Procedural:
Substance Designer has been here for a while now, and it remains a really fun node-based program that allows you to fully procedurally make your own materials will all the adjustability that you can decide on yourself.

OPTIMIZATION

Exporting:
Some really handy trick for a small optimization, but more than necessary when working with modular kits is getting a good batch exporter tool which will set all your meshes to 0,0,0 and export it to a selected folder, huge timer-saver so make sure to look one for your program.

Poly-reducing:
When it comes to optimizing your models there are lot's of different ways of doing this and almost all programs have some ways to do this, but still the most reliable one is going to be Zbrush because it generates the cleanest flow onto the model. That being said, if you want complete control nothing beats retopping by hand if you need stuff to be really specific!

Lodding:
Again, best result when done manually but there are great options to do this, other than some built in tools that reduce your model (only geometry and don't take into account number of materials etc) there is also the option to get Simplygon (used in most studios) which will take control of this step and optimize the mesh and textures for you.

PRESENTATION

Game engines:
For presentation purposes you can go a lot of routes here and the one that I like to do is present them inside of a game engine, which shows that you can take an asset through he full pipeline. Most commonly these will be Unity and Unreal, and I would say out of the box Unreal is more artist friendly.

Standalone:
if you are looking for even more artist friendly and standalone programs, then having a look at Marmoset Toolbag is probably a good way to start, allowing you to quickly import textures and models to then light and do some post-processing as you see fit.

Obviously, these are not the only options to present your projects in and there are some really nice ways build into your 3D packages like Maya, 3DS Max, Blender, etc...

All the different programs might make it more difficult to choose from them, and there are a ton more than I list, or could ever list.

Damage Decal Workflow

Introduction

This time we're looking at the damage decals workflow. Where we use a layers of decals on top of our base mesh to add damages sections to our assets. Which is really useful for games with a lot of broken architectural or rubble pieces.

1. Overview

It's part of different mesh decal workflows, but when people refer to damage decals they are more specifically talking about decals that add damaged sections to your assets. Like let's say you have concrete pillars that are broken and you want to differentiate the broken section from the non-broken version.

For this to work you need your base mesh with a tileable texture and then where you want damage you add another piece of mesh on top of your base that had the damage decal (texture with opacity map applied to it) added to it, so you can get a nice looking damaged section.

2. Pro's

It's really versatile, where once you have a texture filled with different damage Decals you can keep reusing this one over and over on appropriate meshes with the same texture.

Which in a game setting can save you a lot of drawcalls / texture memory and will thus make your technical artists happy.

Another pro to this workflow is that it's super easy to work with, because you can just remove/switch out textures for these damage decals if needed and if you have laid them out in the same manner.

3. Drawbacks?

The more traditional approach to this technique is definitely to have a decal that's standalone (see example on the right) so if you are going that way you can't really use it as a "Normal Decal" anymore. Depends on how unique you go for this look.

This technique is also more expensive in terms of polycount budget, since you will have to add a whole lot of mesh for adding these decals.

Another minor drawback is that you can't easily more them like you can with projected decals, so you will have to go back to your 3D program of choice if you want to update the position on the model.

4. More resources?

An expansion on this:
If the decal itself doesn't look to unique we can also turn them into **"Normal decals"** that are used to specifically only affect the normal map of the underlying object.

If you want a video overview you can head over to this link here (https://youtu.be/FPzPGm_WB7E) where you can see the workflow of creating one yourself.

Also one of my favorite blogs (Simon Schreibt) that always inspired my also did an excellent breakdown of this technique used in fallout 3, as seen here: https://simonschreibt.de/gat/fallout-3-edges/

03 WORLD BUILDING

World & Location logic

Introduction

To construct a believable location or world we need to focus on building a solid supporting logic for it, it acts like the foundation of your whole environment, allowing you to go back to it if you are ever lost and find your footing again or even expand on it.

Brought to you by some holiday pictures :D

1. What does this mean?

We talked about this in a previous blog post but it might be worth looking into this topic deeper than we did back then.

When starting out creating your own location it's important to establish some rules and guidelines to establish the location and make it feel alive, believable and grounded in reality.

For this we need to start out with planning the location and thinking about some questions we need to ask our self.

2. Form follows function

What is the function of the location? Is a great question to ask about your environments.

This question allows us to figure out if we want to build a fishing village that this location is going to require a body of water, but also so much more!

For instance a small docking yard, to repair smaller ships, a local fishing shop near the docks, the docks itself, small village, etc...

All these things are there because they follow the function of the place, because people need to eat, etc…

3. History

Manmade
For instance a museum that is now re purposed as a military base can add a lot of interest to such a location, how would people re purpose that location to make the most out of it for their own purpose?

Natural
A small settlement that grew over time and now has an old part of the village versus the new part build in a different style with different construction materials. Like you see in most major cities that have been around for a really long time.

4. Keeping it logical

When coming up for ideas for a location, make sure that you keep it grounded in reality.

When talking about this you can look at the 80/20 rule, so 80% grounded in reality and 20% imagination.

The thing that is important is that you need to keep checking if the environment makes sense for yourself and maybe show it to someone who hasn't seen it before at a stage where all the elements are in and make sense for you so you can double-check that way.

Hope this helps!

World Building - An Overview

Introduction

World building is super important for games in general, people want to be immersed in worlds and want to feel a part of them. So as game developers we do the best to give you the richest possible world so that you can really feel immersed in them. Let's have a look at an overview of this.

1. Introduction to worldbuilding

World building has many different facets, but in it's truest meaning it literally stands for the process of constructing an imaginary world in which in our case our players can then go on adventures and explore all the facets of that created world. Which if done correctly and with enough room for creative input it can create stunning results that do a good job of being as elaborate as the one we live in right now.

As for environment art specifically we will most likely take inspiration from the narrative setup by narrative designers and writers (or a "bible" in some cases) which we can then turn into art assets to make a space from.

Farcry 5

2. World building in environment art

When we talk about world building we have full control over the implementation on the visual level, which is an awesome power to have. Especially in the combination with level design and if the two come together in the creation of a nice playable space that looks good and tells a story, which is where we come in!

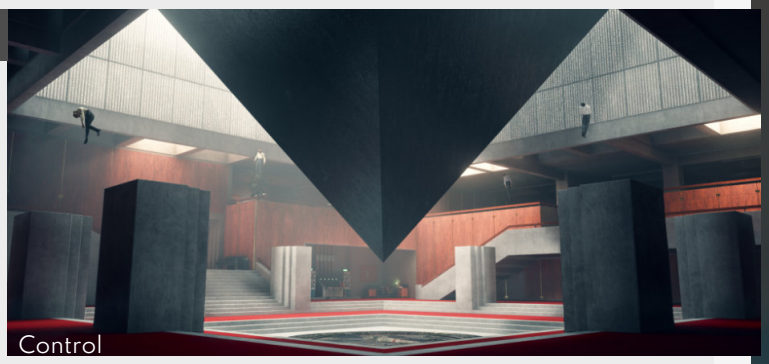
We can implement this on so many different levels, this can be by placing rocks and sculpting terrain in a specific way to indicate a possible landslide, on the asset level to tell micro stories that tie into the larger world narrative and more... The possibilities are endless and awesome!

Red dead redemption 2

3. Balance between different aspects

Now when creating worlds in games there are a lot of aspects and disciplines that come together to create this world. For most games if not all games is that the gameplay comes first, and many worldbuilding aspects come to life specifically to support to gameplay.

When it comes to games in a lot of cases the game can look at well constructed and pretty as it should, but if it runs at 5fps and is boring as it can be then who will play it? So in most cases art can not be pushed as far as we as artists want it too.

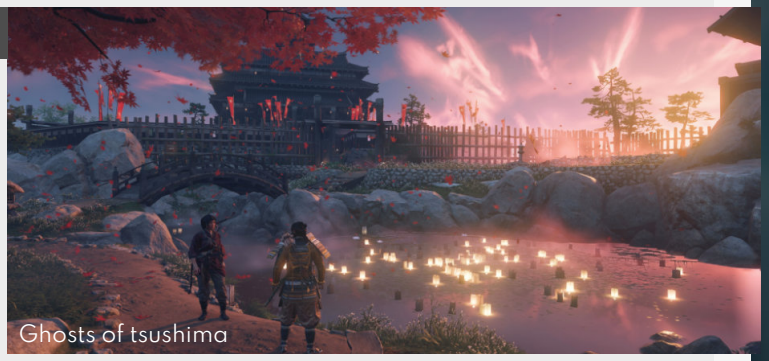


Control

4. World building elevates the experience

This is the best part of the job, in all aspects of Environment Art really, people are looking to play games that engross them into the world and the stories in that world. So we as the visual representation of that should be proud of owning a small part of it and really be proud on bringing this visual experience to the players.

So don't underestimate this section and build your own rich worlds that people want to explore for themselves.



Ghosts of tsushima

World Building - Basic human needs

Introduction

So let's dive into some specifics when it comes to world buildings that's applicable for us Environment Artists. This is super useful to start with in the beginning when planning your next piece (but can still be applied during working on it). Starting with basic needs!

1. Why do I need to think about this?

This might be the first question you might ask yourself. But this is important for the creation of scenes, especially when creating worlds that are interesting for people to explore and to view. It will elevate your environments from being nice technical achievements to a world.

This will add depth and will open the space for mini stories even if it's just a single image. When you start thinking about the following points, it will inspire you to create new unique assets and those will make your work stand out more.

The following examples can be applied to individuals as well as collections of people.

2. Water and food

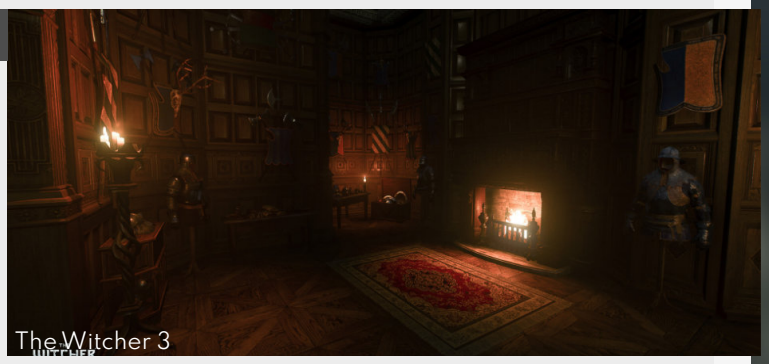
Water: The most important source of life (if sticking to the human realm of course). Finding out where people have access to it, transport it and store it can really open up some interesting possibilities. For example on a planet that is always freezing. Water, collection, storage and transporting it is totally different compared to earth.

Food: The same applies for food as well. Do they hunt? Is everything automated? etc... Even when building abandoned or destroyed environments how we think about this factors will still have an impact on how we construct these environments.

3. Clothing and heating

The same applies to the clothing they wear, even what kind of clothing is is self-made (hinting at a lack of commercial infrastructure or rural life), super fancy (hinting at high society), has reinforced parts, military colors, metal plates, (think post apocalyptic, mad max style) maybe a combination of them all!

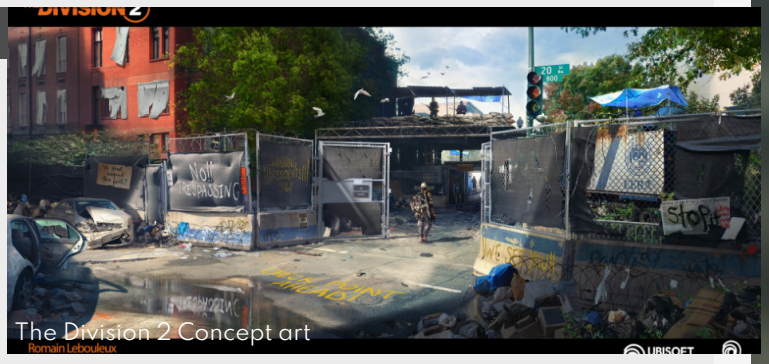
Also are the clothes linked to how they stay warm? if not then how do they stay warm? These can all have a big impact on the types of structures, what kind of equipment they would use to keep them self warm and survive.



4. Safety and protection

Then last one for this tip, safety and protection. Are they a peaceful society? A smaller community? Nomadic tribes? all these things have impact on where they live and how they live. All of these factors have a good impact on the assets you make.

Some examples: The people in your scene could be peaceful traveling traders, that would then mean that they don't need weapons, have equipment to hide in hostile terrain and probably travel a bit lighter then others to be agile. If you apply this type of thinking you can quickly start to build up a good foundation that can really serve you in creating your own unique world or scene. Good luck!



Who's in your environment? - Part one

Introduction

When thinking about world building, the people living in the world or scene you are building are just as important, even if they are not visible they can still have an impact on the scene. So let's discuss some of the influences they can have.

1. Characters in Environment art

This is a subject that a lot of environment artists might not be a really big fan off (Trust me, I'm in the same boat). But over time I realized that there is no way around narratives in your scenes without involving or at least thinking about the people that might be a part of them too. This also applies to scenes that have no physical people in them.

When you take this step and you think about the their background, their skills and current narrative it can really open up for some great opportunities when thinking about assets to create for your scene. So let's explore this a little bit.

Steven Hong - The Goblin Den

2. Their background

The origin of the person or people is an important factor, there are so many interesting ideas when it comes to it. Especially when reflected in the environment itself. For example in the scene on the right by Martha Niemczynska you can see that this is a person who is anti establishment, a hacker, and that he/she is on the run now after their hit.

And this is all without the person in question even making an appearance in the picture themself, that's what makes for some good environmental storytelling.

The scene would look totally different if the person here had a different background.

Marta Niemczynska - Cyberpunk 2077 Fan Art

3. Cultural backgrounds

Another big category that can have a massive impact on the scene, you can really pull from your cultural background. For example what does a skyscraper look like set in **X** culture, whole world like the artwork by Jeryce Dianingana. Really tapping into his own cultural background for inspiration leading to an amazingly refreshing style that's inspiring a lot of people.

It doesn't have to be your own culture either, if you do some good research in cultures, there is some awesome stuff waiting to be found and can push your world building to the next level by picking on element and giving it your own spin. This can both apply to both a macro and micro scale in your environments, from massive buildings or entire cities to a single prop.

Jeryce Dianingana - Wakongo Hotel

4. Class & wealth

Lastly for this week's entry, let's look into wealth and class (with this more the upbringing of someone). We can tap into this by portraying a fancy mansion with lavish gold trims and painting from the person to indicate that he/she is really proud of them self and that is the only thing they care about. Or we can do the opposite and show off a slum area where they have to scramble just to get by on a day to day basis. Reflecting this in the environment by having sewage water running down the street etc...

Think about the witcher without it's unique setting and reliance on Slavic myths and role, that is such a big part of that game. So make sure that you use it to your advantage.

Ivanna Liittschwager - Old London Slums

People's personal motivation

Introduction

When thinking about world building, the people living in the world or scene you are building are just as important, even if they are not visible they can still have an impact on the scene. So let's discuss more of them in the second part.

1. Political alignment

So the first one is a thing that games love to play with more and more especially in these politically divided times and that is the political alignment of the people in your scene. However, it doesn't have to be radical to influence your world that you want to create. It can totally be a utopia as well!

There are so many things you can pull from here that can make your scenes super rich, where it's a utopia on the brink of falling victim to a radical movement trying to break it apart or vice versa, and it doesn't have to be black and white either, there is so much space in between, so use that to your advantage.

Edit Ballai - Utopia

2. Personal beliefs

Allright, for this one let's have a little fun! What if you make an environment that is a flat earthers den/garage. Imagine that person building there own rocket to see for them-self if the earth is flat or not. There is so much good stuff from maps with ice walls surrounding the earth, model and blueprints of rockets, posters calling out NASA for being fake.

Could be such a fun little scene to work on and would definitely pull some attention to itself even if that theory is totally bonkers! You can apply this to kids rooms believing that super heroes are real, a scene about santa's workshop, etc...

Beeple - Flat earth sunset

3. Religious beliefs

One of the main reasons that pulled me to Farcry 5 and playing that game (even if the game itself was a little disappointing at times). So there are entire games that are build on this section alone, so you can pull from this section as well.

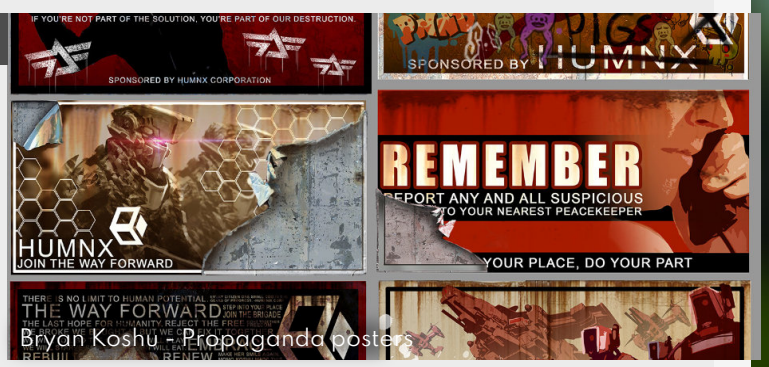
For example you can go this direction of depicting a hidden scene of a cult, playing on religious beliefs that people have and all the outfits and looks that come with that. You could even push it into a more horror directed scene and end up with something more like outlast 2 for example.

Farcry 5 - Concept art

4. Education and other influences

Now there are so many directions to pull inspiration from and we haven't discussed them all here, like Propaganda and how that influences a character and adds to the world either visually or through audio, same applies to educational background of the person involved, their upbringing and so much more.

So I want to leave you with this though, is that you need to take the time to think about the people in your scene, why they are there, how they got there, where do they get their food etc... There is such a wealth of inspiration at your fingertips if you just look for it



Bryan Koshu - Propaganda posters

Nature’s impact on world building

Introduction

In the previous couple of tips sections we've been focusing a lot on the people involved in your environments, so now we're going to go back o our comfort zone and talk about the environment itself a little more, specifically natural elements that can have an impact are under our control when creating an environment.

1. The weather conditions

The weather is a tool that is often used for emphasising a mood in your environment, while not often done by the environment artist them-self when in the industry, in your personal pieces it's good to pay some extra attention to this aspect.

For example if you want to emphasise a really sombre and heavy tone, you can always use rain to further push this mood. On the opposite side you can use the rising sun as a sign for hope and in case of most survival games that you survived the night. Happy and loving moods can be emphasised by using the magic hour type of lighting (around 6ish as the sun goes down). Don't underestimate the power of the weather.

2. Terrain features

Terrain features also play a big an important role when doing natural environments, for example having a big hill right in front of you with a big structure on top of it can really feel overwhelming and intimidating and on the opposite side you can build a lot of intrigue by using smaller spaces and gaps in rock formation to signal that there might be a hidden path for example. Especially if these are used in contrast to really up the dramatic impact. More directly related to environment art, you can use terrain features to pull the player attention to a certain point. These are some excellent examples of player leading and more on that topic can be found in our tip **“Terrain features for player leading”**.

3. Water, rivers and streams

This goes together with other items on the previous list, where people need water to serve their basic needs (In realistic scenario's) but these can also serve to support the world building. If we look at history, settlements have always started around bodies of water or near a river, and then expanded outwards using irrigation methods.

Because of this bodies of water serve as a natural point of reference in that world and can be used by players to find their way though locations using them as guidance.

4. Altitude and verticality

"If you can see it, you can go there" as said by Todd Howard when talking about Skyrim. Players will always use height as places to get an advantage or just to get their bearings and figure out the next places to go.

So when constructing your own scene, keep this in mind. Plus just in general, it's a good rule to try and incorporate some height variation in your own scenes, giving you a more natural looking scene. Even minor variation in roads or just "flat terrain" adds a lot of natural feel to it.

Landmarks and Points of Interest

Introduction

This time we're going to be diving into a concept that is used in almost all games, especially the open world ones and will help the player find his way in massive open world environments. These being "Weenies" coined by Walt Disney actually.

1. What are they?

Big shapes in the environment are really useful for leading the player through the game. This has been used in a lot of games going from single player narrative driven games to open world games. In most cases their size alone makes them stand out.

However, they don't have to be, they can also just stand out by using contrasting colours, shapes or silhouette's. Also, you yourself can use these in your environments to make for an interesting point in your environment to be focused on.

2. Size and silhouette

First and most obvious is having a towering thing in the distance that will immediately stand out and that is the most common of approaches, but they don't always have to be giant and far away. Some smaller items that stand out can be used as an intermediate between you and the target location. Which will be called breadcrumbs.

Another way for these to stand out is to have a silhouette that is substantially different compared to other object in the environment, this is also why man-made weenies stand out in a natural environment.

3. Color contrast

If we can't rely on having bigger shapes we can also use colors to push the contrast and thus making it more visible from a distance, imagine you're walking through a nice lush green forest and all of the sudden there is a painted red building in the middle of it, that will surely pull some attention to it.

Just as seen in the example from the right, you can clearly see the red building in the middle of all the blue fog, which has been put there completely intentionally.

4. Some examples

Let's dive into some examples used in games, you can see that these don't have to be limited to buildings, mountains or that kind of stuff. You can also use fires, bright lights and some others to pull the players attention to it.

This was definitely more of a level design topic technically, but it's so good to have knowledge about the topic so you can use it to your advantage in either your work or when working on games yourself.

Using roleplay for idea generation

Introduction

As an Environment artist it's important to tell stories in your scenes, more so if you are a level artist and are actually building the locations yourself. These stories are what give additional interest to the people walking around in your environments, so in this one we're going to dive into a good method of coming up with some of these stories.

1. What do we gain?

When we start to envision ourselves in our work we think about the smaller details that could get lost when not doing so, you think about your daily routine in the setting you are created.

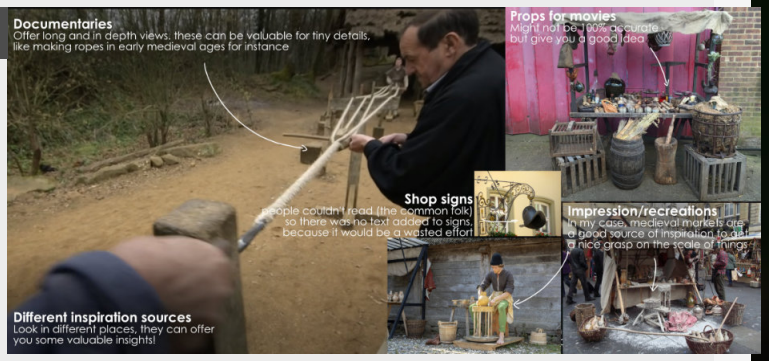
"What would your day to day look like?", "What would the routine of you be?", "What do you eat?", etc...

All of these questions that you encounter are amazing ideas for cool little props that can really add some character to your scene, they will also help you move your scene away from the more standard generic scene and more to something personal and unique. (The world building section in the Complete Compilation holds more of these tips to look into)

2. Doing the research

However, these questions do have to come from somewhere, so we have to do research to start to imagine which questions we can use. Hopefully you have done a large part of this section already, but if we're talking specifically about environmental storytelling, then what I personally love to do is have documentaries on that get me as close as possible to the source material (which is def more tricky when creating fully new spaces, eg. Scifi).

My recommendation on researching is also trying to look in places that are foreign or new to you. Apart from your typical google search, books, documentaries, etc... will give you more interesting result and more unique stories.

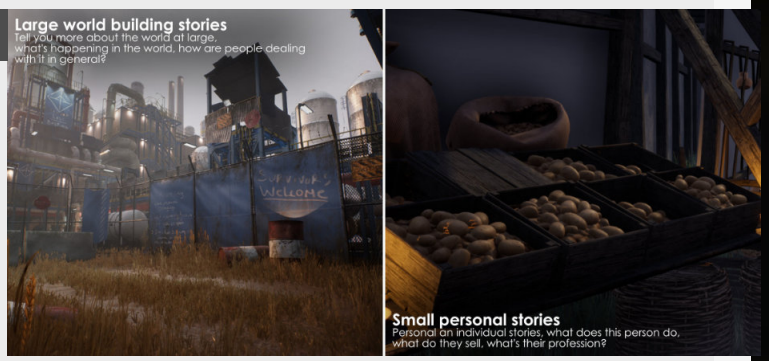


3. What are we looking for?

Larger General stories:
The more overarching story, the world itself, what do people use, what are there habits, what kind of governance structure is there, how to they travel, eat, sleep etc..

Smaller Personal Stories:
The smaller personal stories, what are the personal connections of this person that we are impersonating now, does he have family, what's his occupation, how does he get around...

These questions are really good for coming up with idea's for little unique story props in your scenes.

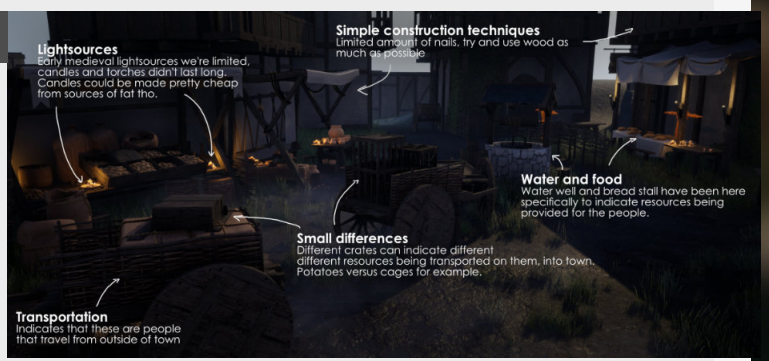


4. How do stories translate to assets?

For example if we focus on my medieval scene and on a personal story.

Imagine your a farmer in the early medieval age, just got done with your harvest and are going into town for the day to sell your goods. For this we'll need a cart, storage for the food you are selling (most likely Jute/Hemp bags) you would also bring something to drink and eat along the way, maybe even store that in a chest that you can lock so the contents remain safe?

You can see that this can quickly build up an interesting list of assets to build for your scene, so try it out for yourself.



Asset clustering

Introduction

It's now only what you place in a level that matters but also how you place things, and often times this might be the thing that is overlooked, so for this one we're diving into placing things in clusters. Why do we do this and how does it help us?

More information can also be found in **"Blending shapes together (The rule of Three)"**

1. The artistic reason

We've talked about contrast a bunch of times already so by now you know that without it you can create interesting compositions or playspaces, they just feel boring and uninspired without it.

But not only just putting thing together makes things looks nice though, you can push it a little further buy adding nice transitions between all the surfaces and assets it's surrounded by if possible. Imagine a garbage bin by itself, it looks okay... But if you add a carboard box next to it, and some smaller pieces of rubbish next to it to give it some flavour it will look so much better.

Obviously different art styles and projects ask for different approaches, so try to apply this whenever you can, but don't break any art direction rules!

2. Navigation and player leading

Eluded to a little bit before already but navigation and player leading is a huge part of games, and if we just have a bunch of noise in our scene then we lose the ability to sign or lead the player.

So in an example from the Deus Ex we can see that the assets have been scattered in the environment but there is intentional space left between object so the player can plan their next move or see where to go.

Especially in cover based shooters which lean more heavily on the tactical nature of things you can definitely tell that this has been a part of the thought process throughout the development and a player always needs to be able to quickly evaluate all their options to move into the next position.

3. Reduces noise

This might be the main reason why I would recommend that you do it, is because if you add details everywhere you add so much noise that you lose the ability to make things stand out (unless you want to do so with nothing).

If you just add a lot of objects everywhere it will start to look like noise to the player, and that will definitely impact how a player/viewer of your art will traverse the scene either by moving or with their eyes.

Make game art assets feel “Real”

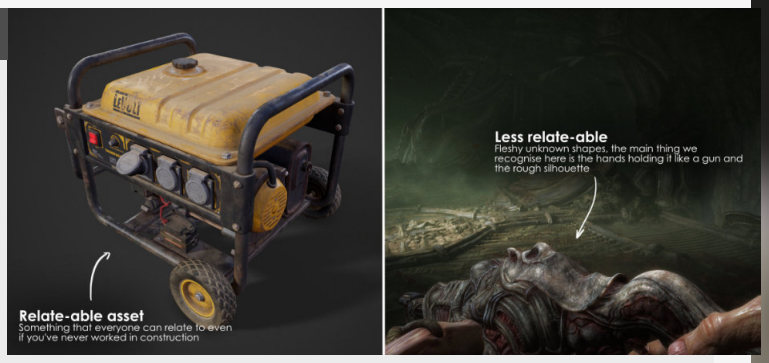
Introduction

Game art assets can range from clutter that is scattered on the floor to highly detailed props that are important to the storytelling or the characters they are surrounded by. With this tip we’re zooming into those highly detailed props and breaking down what makes them feel “Real” or relatable to us as players.

1. Relate-ability

First of all, whether we come up with a new idea or we’re starting with a concept, we need to make sure that it’s relatable. Something that’s relatable can’t be 100% new to the viewer, that would require an entire explanation to even help them understand what this object is.

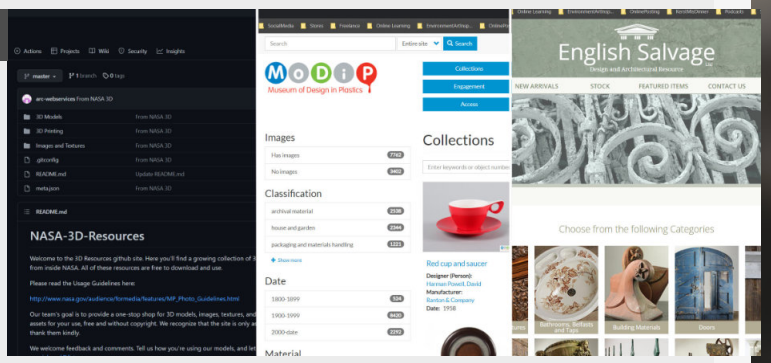
What you want to do especially when building new assets that are different yourself is to use relatable shapes or construction techniques to link it back to something that we have seen already somewhere.



2. Find good references

You need to arm yourself with a bunch of good references, both ones that will help you construct the asset itself but also finding more obscure ones that can help you add unique details to it to help it stand out compared to others.

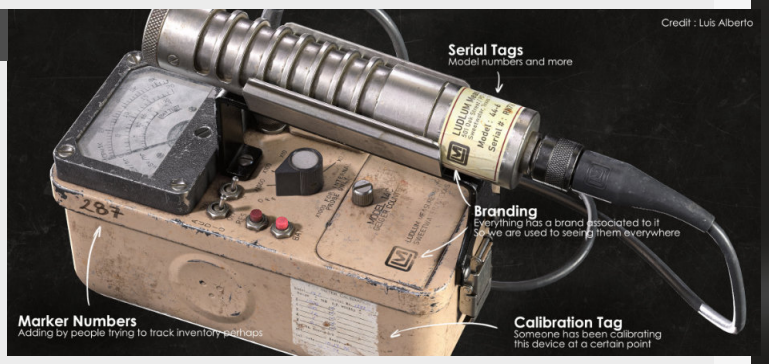
Which is why I love scouring the web for detailed an obscure references to really get into the nitty gritty details on how to recreate them. The deeper you dig, the more unique you can make your asset look. Some interesting recent finds are online museum’s that take have a massive database of images with unique looking items. However, even though this things exist, make sure they follow rule number 1.



3. Adding storytelling bits

As a follow up from number 2, if you find good references this also allows you add storytelling details.

These details can be for example scratches of people carrying this asset around, people spilling coffee on it, the asset being dropped on the floor, so it’s dented a little bit, inspection stickers that with multiple layers to them. Whatever you can think of that is relevant to the asset can be really nice to add as a layer on top of it.



4. Make it personal

And even going deeper with the storytelling is the layer that adds character or references to the person that owns or uses this asset.

If you add this layer successfully people can start to see what kind of person this asset is attached too. Allowing you to get glimpses of the person behind it. giving it the depth it needs and deserves.

04 PRE PRODUCTION

Defining project pillars

Introduction

When starting with your projects it's really important that you know what you are starting to build, and nothing is less important than defining some pillar or guidelines for your project which will help to guide you through the development of these full environments which can take quite a long time.

1. Defining project pillars

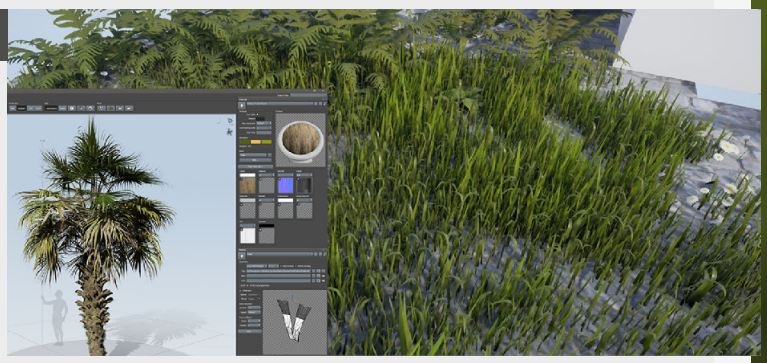
When starting out a new project and being unsure where to start, in this we will have a look at some tips and tricks on finding and defining your pillars for your project, which will act as supports for your project and allow you to look back to them whenever you find yourself lost.
We will have a look at 3 ways that I personally like to start with defining these pillars.

Keep in mind that you can use these pillars in conjunction with each other, so focusing on a story first might also help you with the style you want to pick for that story.

2. What skills do you want to build?

A thing that I did when I was a student or starting in the industry was structuring projects around a learning a new piece of workflow or program that I wanted to learn.
For instance my work "Dense Jungle" which is horribly named btw, was a project that started because I really wanted to learn inside of Speedtree, before doing that project I had no idea how to do any foliage and had only used speedtree standard assets in one of my scenes.

An early WIP screenshot can be seen on the right.



3. Working with a story

Another thing you can do is use a story as the foundation of a good scene, this will allow you to always think back about the story, and if adding an asset doesn't make any sense in the story then you know that you should not spend time making it.

This story is something that you control, some examples can be:

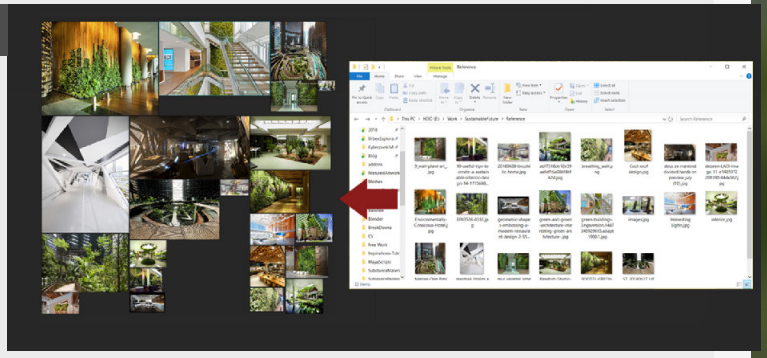
The person living in the environment, what does he do, does he have a darker side with subtle hints pointing to it? etc...

4. Building on style

What is also an option, is starting by picking a certain style, this sounds kind of vague because it's also so encompassing.

Because style could be "Hand-Painted" and then picking a subject that would really stand out if you were to apply Hand painted approach to it.

But style could also mean "Brutalism" for example, focusing on concrete, harsh spaces and a feeling of intimidation.



Metrics and grid settings

Introduction

This edition will dive into some of the most important aspect of creating a modular environment and is especially critical for considering gameplay. We will discuss thing such as creating good blockouts, grid settings and how the whole scene reads as well.

1. Importance

Most commonly used for modular environments and is especially useful giving it to someone else to work with.

So for gameplay and usability the environment should feel nice and to scale when you are walking or playing through them.
Which is why we are delving into the topic of metrics and how to prepare aligning everything to a grid which is going to make the Environment or asset package that you are going to make that much more usable.

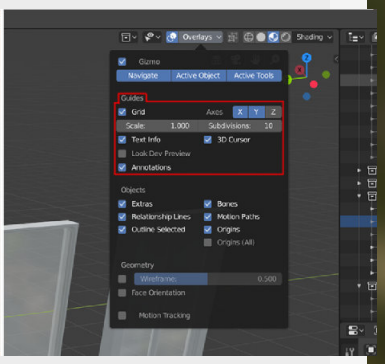
So let's dive into this!

2. How does it read

When we are talking about metrics, the first thing that comes to mind is just the more general dimensions and getting a feel for them.

Changing the height of a wall that should be climbable can make it feel less so if we make it higher.
So adjusting the metrics can have a serious impact on the way the environment feels and how players interact with the space. This is why running through your own environment (or placing dummy characters) is really important to get a feel for the environment.

3. Grid settings

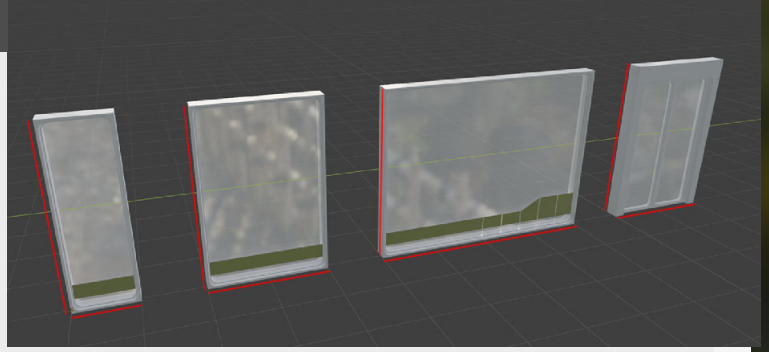


Let's have a look at your 3D package and then jump into the your game-engine.

First, you notice that all of the 3D programs have a grid already there once you open them, if you know what kind of metric that you are working towards.
You are working towards, it's useful to change this if at all needed, especially if you are working with multiple divisions of one meter for instance.

The next step is looking at Unreal Engine where you can set different presets, and make sure to set the grid size beforehand.

4. Blockouts first



When working on a modular environment or an asset package, it's super annoying to having to go back to the drawing board because you didn't take enough time to do some proper blockouts to define different metrics and really work out how the environment is going to be put together.

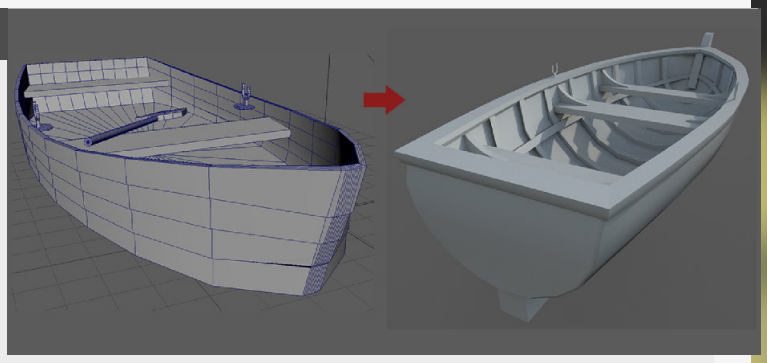
So spend that time working on blockouts, define proper metrics and setup your grid correctly where you have to, it will save you a lot of time down the line!

Using & finding references

Introduction

When talking to a lot of people starting out in their journey to break into the industry as a 3D artist, the major thing that I can see straight away is a lack of attention to scale and details in environments, props and others...
A big reason for this is because is that you are just to eager to get into actually doing the work, and a lot of people forget that actually doing the work includes reference hunting.

1. Importance of references

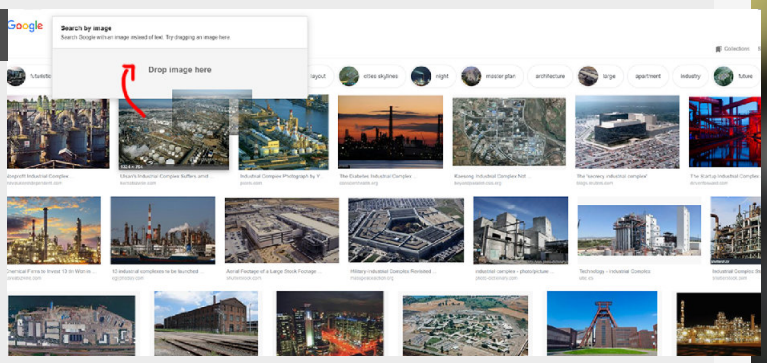


Let's talk about the importance of references.

References are all about understanding the object or scene you are trying to create, made up or realistic, you would still use real life examples to construct both of these things.
References are there to support you when you don't have a clear idea about shape, proportions, details, textures, mood and more...

It may feel a burden at first, but once you start collecting references and use them correctly, you will see vast improvements in your work.

2. Finding references

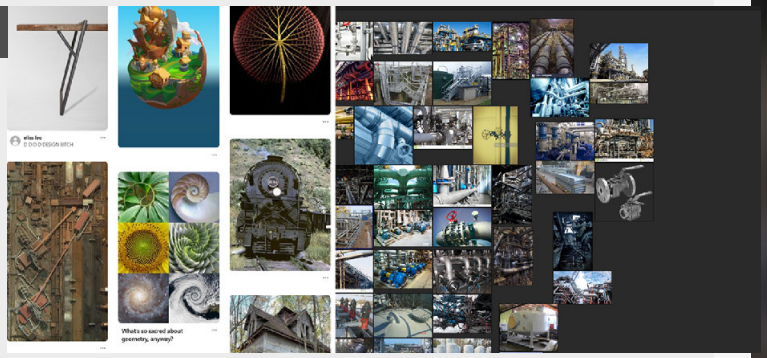


The first thing to thing about is Google, and typing the keywords as direct as possible. Another option for google is using Google image lookup. Just drag and drop images in the top bar.

Pinterest is also a really good tool for this and will also allow you to store it in one place, which is great!

Other stuff include: Ebay listings or spending a long time clicking on links in my search that don't appeal to me, because sometimes these can be reference goldmines.

3. Compiling references

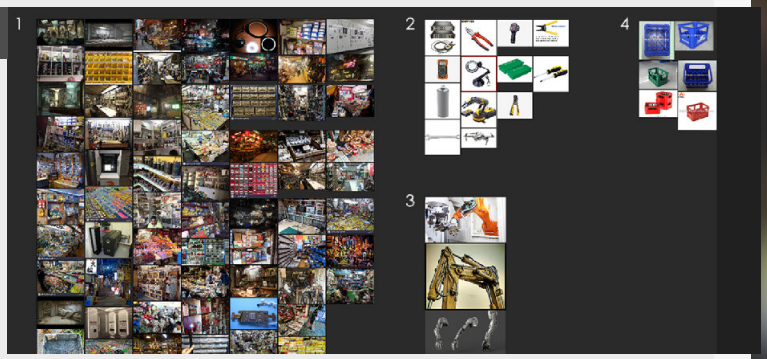


Now we just need to find a proper place to store them.

I always used Pinterest before, but I needed to open a browser and do some extra steps to get me going, which is annoying.

So then I made the switch to PureRef, which is a standalone program that allows you to save boards that you create yourself and the interface is just drag and drop, so just select an entire folder of images and drop them in and watch the magic happen. As a sidenote you can save these boards, to quickly open them up the next time.

4. Using references



There is a balance you need to find in how specific you need to look when it comes to references, and lean on them when struggling.

So find what works best for you and work towards that, for example, I like making multiple boards. These include Moodboard, modelling references, architectural style, texture references and weathering inspiration.

Last tip!
Mainly use real-life reference, try not to work from a game or other persons project, the world has so much to offer.

Overscoping

Introduction

Let's dive into something that plagues a lot of people starting out on their own personal projects and trying to push themself is that they tend to overscope their own projects and then never finish them in the end.

1. It's too much

I see with a lot of people that want to get into environment art, and think big.

However, this can be really dangerous as it might be too much for you to handle.

But especially if you don't have that much experience yet, you might not have an idea how long it takes to finish an asset going through your entire pipeline which you probably are still figuring out as well.

So let's dive in to some tips that help you against overscoping.

2. How to keep it small

So there are a couple things you need to know to keep it small.

Making a smaller environment is not showing less skill, it will only allow you to show more technical skill because you are more focused on the actual quality instead of quantity. Some good examples for these would be looking at diorama's, some of these are really awesome and there are a lot of good examples on Sketchfab.

3. One camera illusion

One thing that I really like to do, is to make the blockout of my scenes and restrict myself in the camera angle that I use and I work towards that one camera angle.

This means that I don't need to worry about making it actually playable and can just focus on that one angle and make it look as good as I can.

After that if you still want to expand this environment, it's easy enough to fly around and dress some other regions for other shots because you already have all the assets these anyways.

4. Focus on the important

Focus on the most important assets, typically your hero assets that are essential for the story,setting,... etc and move on from there.

Once you have blockout out and placed your most important assets, you can move onto the other supporting assets, that can also be cut if you think you have overscoped and need to do so.

Hope it helps!

Reference sheets

Introduction

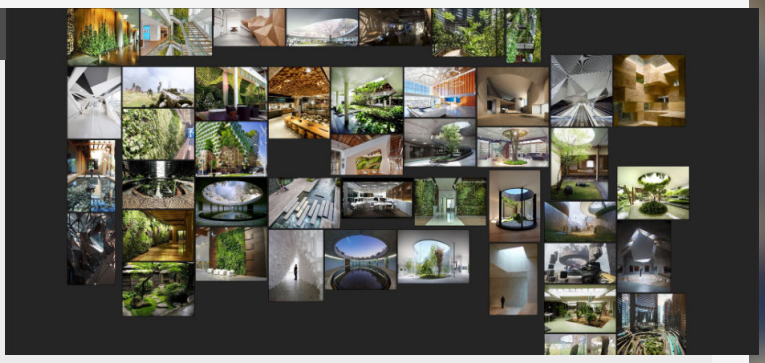
Let's dive into something that plagues a lot of people starting out on their own personal projects and trying to push themself is that they tend to overscope their own projects and then never finish them in the end.

1. All the things at a glance

The reason why I always use this, is that it saves me a lot of time building assets for a scene and it keeps me focused on modeling instead of a balance between

keeps me focused on modeling instead of a balance between modeling and reference hunting.

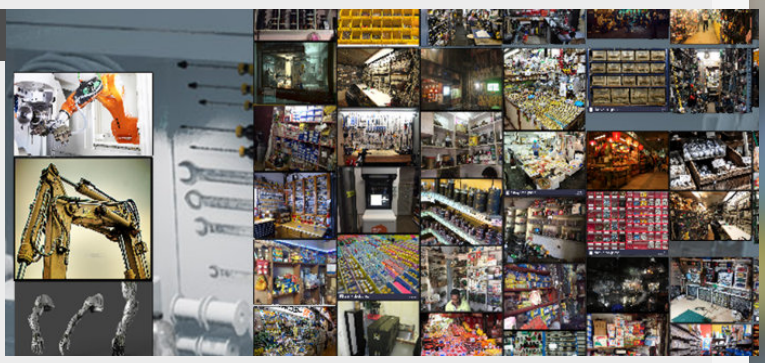
So the reference sheet will just have all the information that you need when modeling all those beautiful assets, and keep you focused while you are working on it.



2. Defining the pillars

So what is in my reference sheets? and at what time do I start building this?

Well, the reference sheet is not the first thing I think of when making the environment ofcourse, first I define the pillars of the environment, the story and what style is.After I have defined this in rough lines I start looking for inspiration for assets, elements and shapes. So that at one glance of the reference sheet I get some good new ideas or know what I want to build/experiment with next.

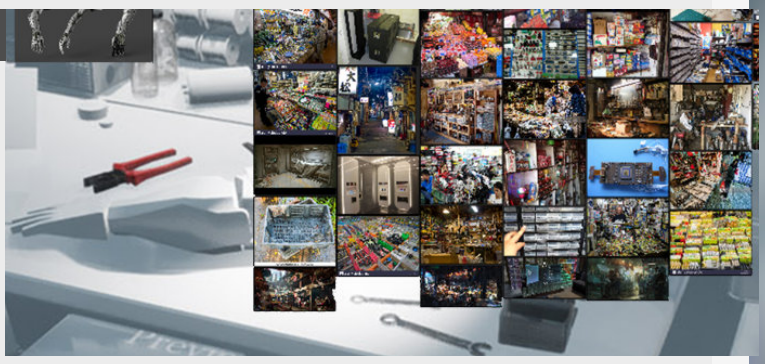


3. Finding and compiling references

For this one I'm using the old school way of dragging it into a reference folder on my PC so I can use another program called PureRef.

PureRef allows you to just drag and drop all the reference pictures into the program, it's going to automatically sort them and present them in a nice manner.

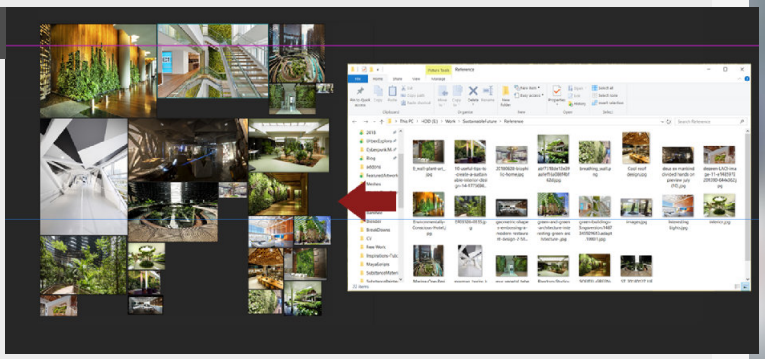
What I like to do is drag them in by section, for instance one section with shape inspiration, modeling details, etc... all as separate sections.



4. Extra notes

So using these sheets, making them easily accessible through the use of PureRef and it's awesome saving feature should minimize the downtime when you are working on your own environments, this will also speed up the process of getting into it and loading all your reference.
Hope this helps you speed your work and modeling process.

Don't forgot to take care of the planning stage, even if you are using a concept for your environment, think about what would make sense for you to be in the scene and make it yours!



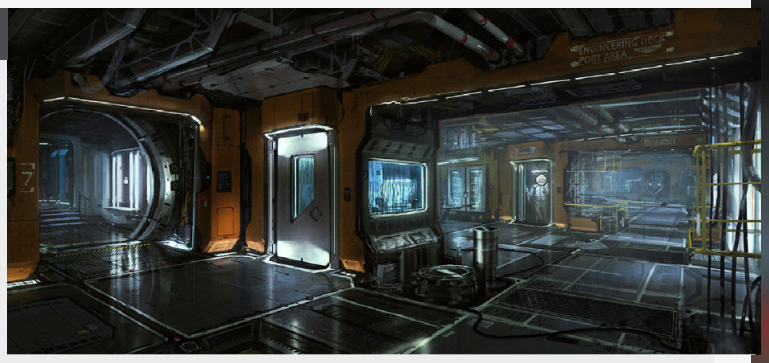
Breaking down modularity

Introduction

One skill that environment artist need to have when it comes to the modular workflow (the ability to reuse the same meshes/textures multiple times to reduce the cost on resources) is the ability to break them down from the reference, now when it comes to examples like these there are a couple of different types of modularity that we can talk about

1. Why would you use this?

Breaking down a concept into a modular piece is really useful when it comes to making the blockout for the modular pieces, the reason that we do this upfront is to save is time when we actually go into the production of the scene itself.

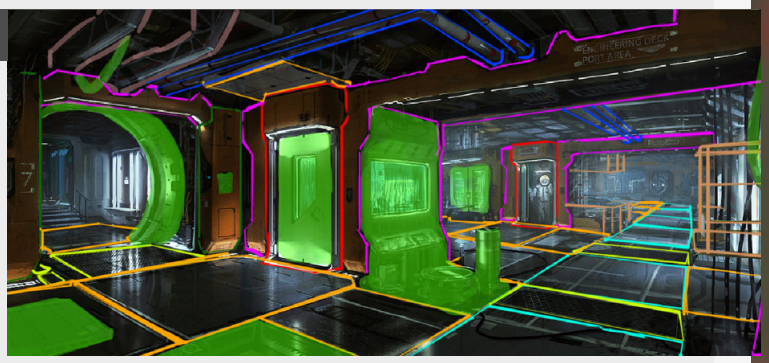


2. Modular meshes

So let's take this screenshot and break it down into it's modular pieces first, a good rule of thumb is to look for repeating patterns first, and then see how big those pieces actually are.

Because for instance you know that most windows are going to be reused, but does that mean that pieces of the wall will also be reused?

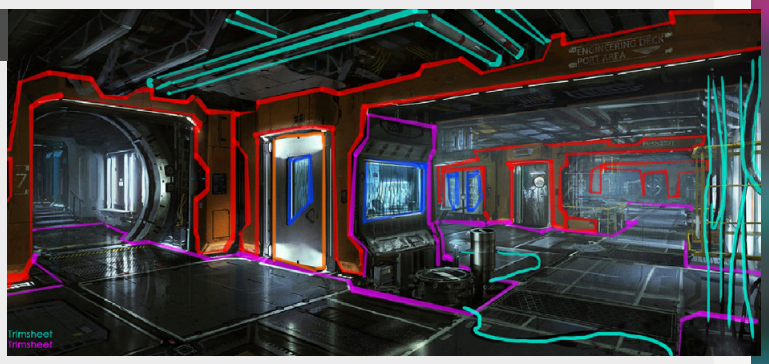
Have a look and really try to spot how big those pieces get.



3. Modular materials

Next up, let's take a look at the amount of materials that we can reuse in our scene. A lot of them might be using the same base material but dressing them up differently with decals is going to make them look more different and set them apart from each other.

Breaking a scene down this way will also help you to reduce the amount of time you spend on making materials that you might end up not using at all.



4. What kind of modularity

There are many types of modularity that we can speak of, but generally on a scene level these are the most important ones that you need to keep an eye out for.

Thinking about larger scenes this way will help you enormously budget/time wise, because the more you will do in a unique way the longer the development time of your scene will get. So work smarter, not harder! (or maybe a healthy balance between the two)

Different reference purposes

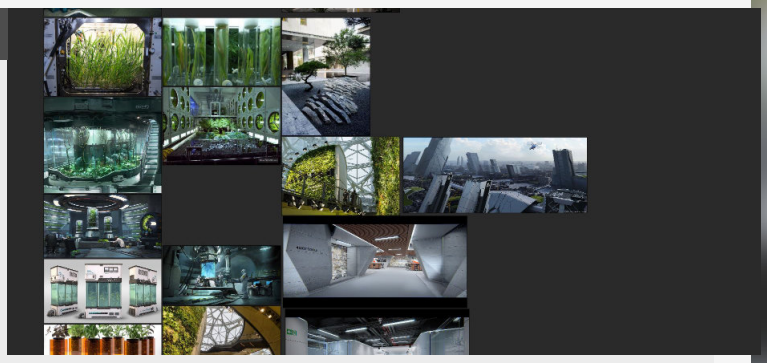
Introduction

So when hunting for references, I love to work my way from the larger scene as a whole to the smaller details. So in these references I would start to focus on the overall scene composition first, move my way down to the specific mood and go smaller and smaller...

1. Overal scene composition

The first thing I start looking for when starting a new scene is the overall scene composition, references and pieces of inspiration that can be used to give me the general idea and get me inspired for the overall composition.

I tend to keep the amount of this limited as t not get confused by all the different references, limitations are good!



2. Moodboard

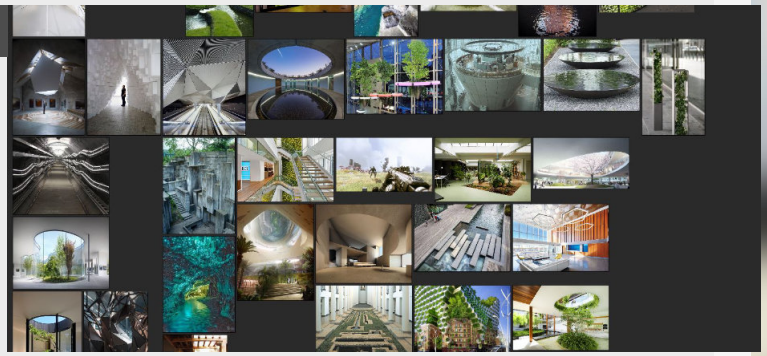
Second thing that I have in the reference sheet is the mood that I'm looking for, this one is specifically based on the mood that I want to go for, in terms of colors and also in terms of weather and lighting in the scene.

Sometimes it's crucial to split this one from the first one s you can spend way too long looking for both in the same picture, unless you are lucky to find that golden reference your looking for!

3. Specific elements to combine

Now we are getting down to the more specific types of references, and these are the specific kind. What I mostly do with these ones is that I find a great reference but there is only one small part of it that is useful to my scene, so I either crop it out or I make a note of it.

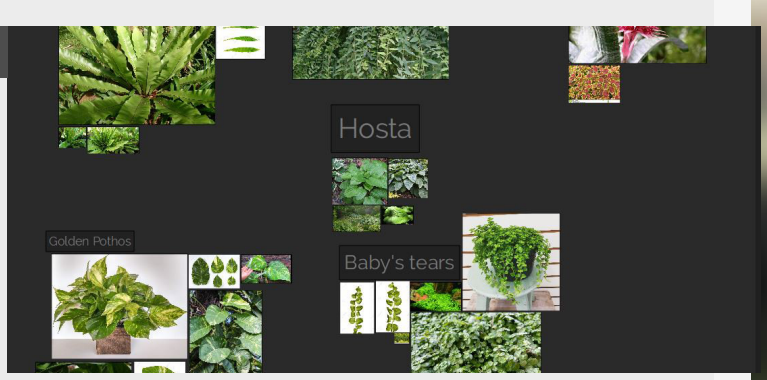
This way I can easily identify which parts of the reference interest me and are applicable to the scene.



4. Some additional notes

Lastly, I just want to add to this that you can spend ages looking for references, but at some point you need to make decision of when to stop.

Also adding too many references can lead to confusion, so it's best to take the references that you think are pretty clear, especially for the high level stuff and keep the number of them to the minimal you need to make your scene.



Environment blockouts

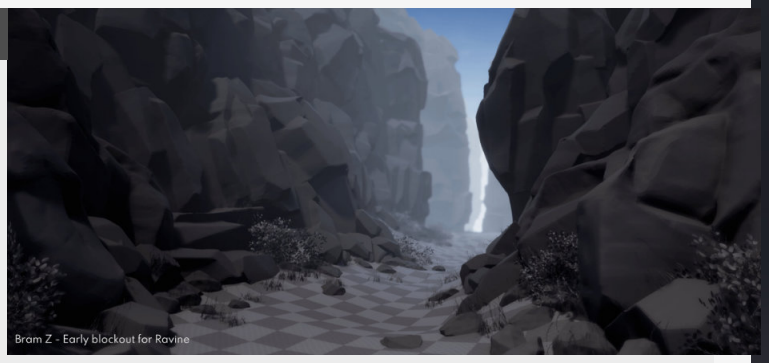
Introduction

Let's dive into something else for this weeks edition, let's talk about blockouts, what to focus on and how to tackle them.

1. What's their purpose?

Maybe you have noticed this in your own work at some point, but you get that nasty feeling that something might be off on your scene and you know that you have to fix it but can't be bothered to do it because you already progressed so far.

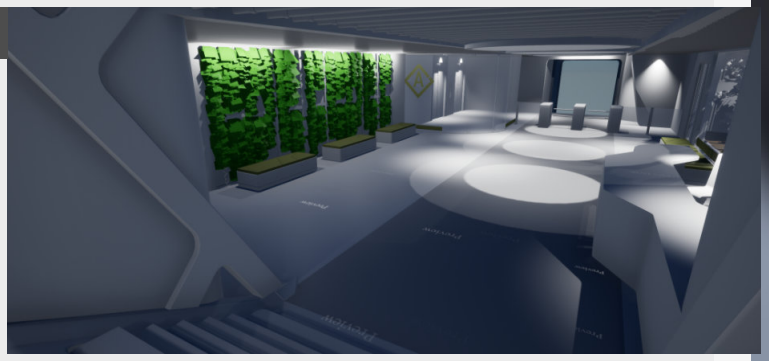
This is why we do blockouts for our scenes, we do this in a way to avoid as much structural mistakes along the way, especially when it comes to the composition of your scene.



2. Focus on the essential

It's all about minimizing the amount of stuff you are working on, blockouts are there to help you figure out the fundamentals of your scene so that you avoid having to do big reworks on your scene later. Once you got the camera angle and composition that you are happy with you can then use these as solid fundamentals and build upon them.

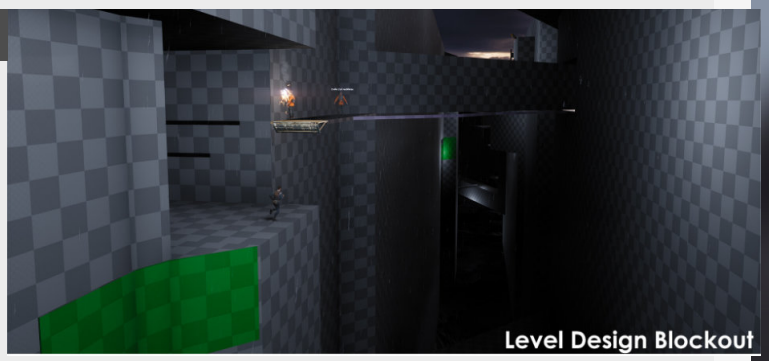
This is also why I'm reiterating on what is important, keep yourself focused on the large shapes, composition, and your camera angle for your scene, the other stuff all comes later.



3. Use placeholder textures

Just use simple color materials to indicate your rough intentions. This will help you define your scene a bit more and give you an idea of the different colors you are going to use without getting carried away with the material creation itself.

In Unreal Engine you can just use a simple Master material for this and setup the whole thing pretty quickly by using multiple instances of that material and just override the color. I want to reiterate, don't spend your time on already adding a roughness, metalness and whatever, focus on the color of your scene. Another option would be to use a color swatch texture to speed things up if sticking to a dedicated color palette.



4. Focus on speed

This might speak for itself after reading all the previous part, but setting yourself some sort of deadline on when you want to finish the blockout stage can really help you focus on the speed of iteration.

This focus on speed will help you make decisions quicker, give you more variation in options and most importantly will cause you to focus on the bigger picture.

Planning your scene's timeline

Introduction

Let's have a look at how we can break down your projects in a way where you can get a better grip on what you are doing in the next couple of months.

1. Starting at the end

When planning your own scenes it's always good to think about the time-frame you want to finish it in and what your end goal is, so let's say you take one month to finish a diorama. Now you know the end result and time frame and you can start making your way backwards and start dividing this environment into smaller tasks like "blockout, modelling, texturing,etc...".

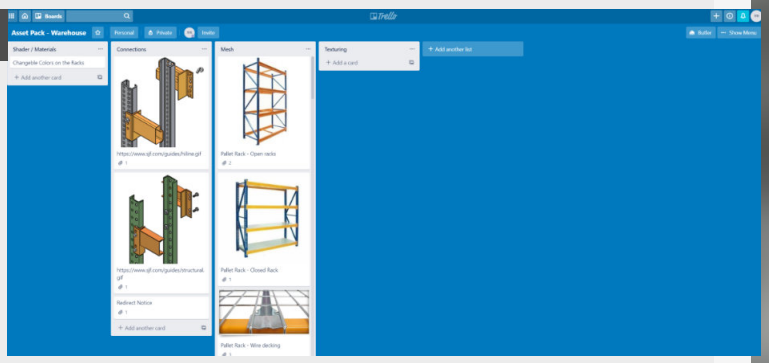
How granular you go is totally up to you at this point, some people do really well with splitting it up in small chunks, while other people try to keep it a bit more free flowing.

Imagine where you want to be in the end and work your way backwards from that point.

2. How to keep track of it all?

Which app or fancy method doesn't really matter that much here, the main focus is making it as easily accessible as possible, so depending on the size of the project this can go from a simple checklist to entire sheets, the only thing you need it that you can check them when you need to and don't have to spend your time looking for them.

However, there are some pretty good digital tools out there like Notion (Which I'm using now) or if you want to go real simple something like Trello is awesome!



3. Build in "save points"

This is a thing that could work for people having issues with over scoping their projects and a thing I have been thinking about a lot recently. But you can help yourself by building "Save Points" into your planning. With this I mean compartmentalized sections of your environment that are still strong enough to stand on their own, especially when building the bigger environments.

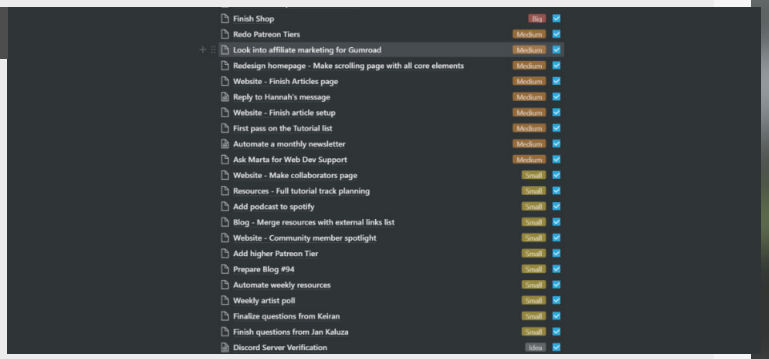
An example for this could be making a cyberpunk apartment, making a living room first and then maybe expanding upon it later with the textures and assets you already have now and adding to it is way better then trying to do the entire apartment in one go and not finishing anything.

4. The day to day

The whole point of planning your environment art workload is to make it easier for you to bit of smaller chunks of the bigger whole and not be overwhelmed by it all. Which also bring me to the to-do list mentioned in last weeks post.

planning your day to day work is an easy way of turning all the ideas in your head into an easy to understand format and will help you get more stuff done.

Holding a to do list and



How to find good references

Introduction

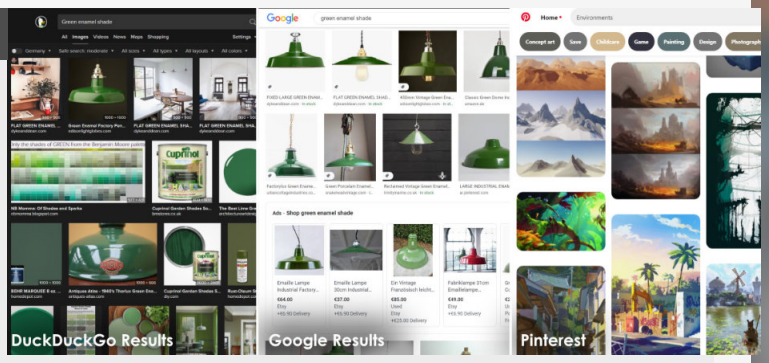
A large part of the environment art workflow is looking and finding references in addition to the references that are provided to you by art direction. But sometimes it might be tricky to find some of them, so in this one we're diving into how to find some good references.

1. Try different search engines

This might sound like a weird one because most of us are all using google for all of our searching, but we tend to forget that it does serve us items based on our searches and it's always learning from it, so for that reason, if you want to find some more unique references you might want to look somewhere else.

So a couple of alternatives are using **DuckDuckGo** to search, **Google Incognito mode** to give you some other results.

You can also use dedicated apps like **Pinterest** to build up some interesting boards to fuel your creativity.

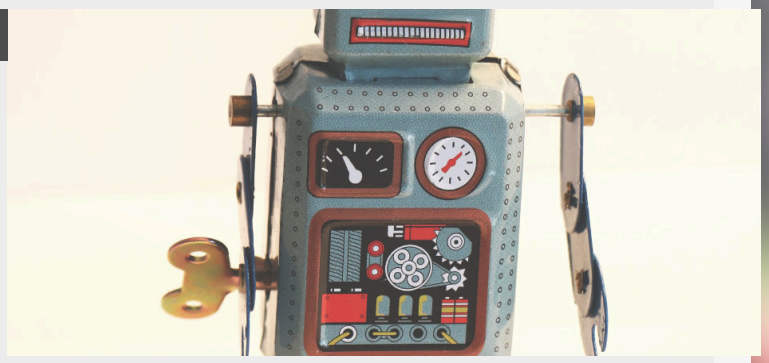


2. Talk like a robot

This also ties into the previous point when looking up stuff in search engines, and that is to really focus on the most important things in a sentence.

If you want to find something really specific, cut out all the fluff and treat it like you are just inputting commands or keywords.

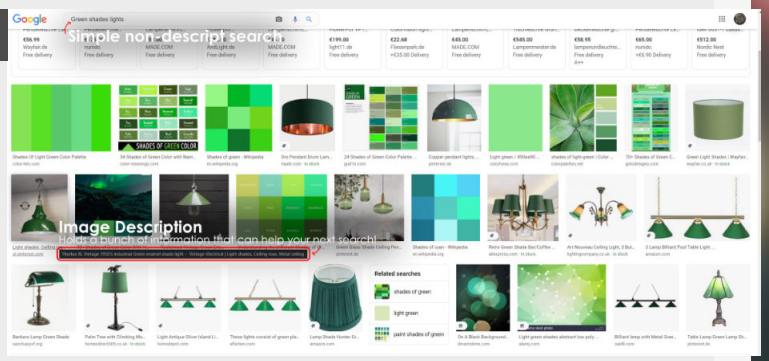
For example, "A nice looking green ceiling light that you would see hanging in bars in the 70's" can be converted into "Green 70's Bar Ceiling light", this will give you much better results and also more results because all the fluff is cut out of the sentence.



3. Image Descriptions as a guide

This is something that I use a lot, especially when I'm not really sure what I'm looking for. I try to get as close to the item I'm looking for using the previous section as a guideline and then look at the descriptions that are within the pictures to find additional details or information about the thing you are looking for and then add them into your new search terms.

In addition to this, using google image search can also really help you with finding similar or relevant images too, so that's another good tool to use in some cases.

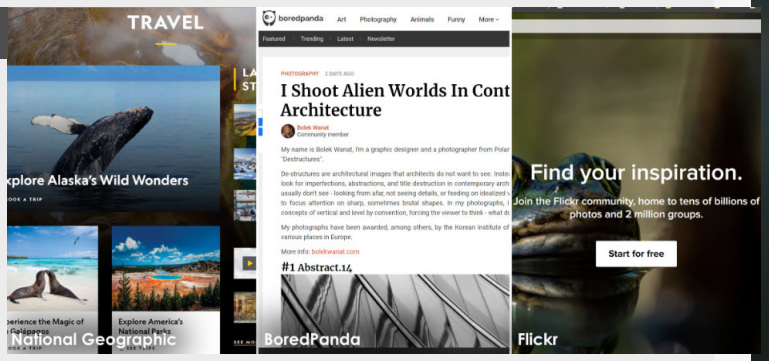


4. Dig deep

The last thing is digging deep, often times the best references/resources are the more obscure ones that take a bit of time to find and you might have to do some digging. Be careful that you don't get sucked into endless blackholes tho!

An example of this can be people's personal travel blogs for environment art references, they can be great for more remote parts of the worlds where google maps don't work and without tourist hotspots.

Some good places to start are: **BoredPanda**, **National Geopgraphic**, **Flickr** and **Pinterest**...



05 MODELING & UNWRAPPING

Intro to Blender 2.79

Introduction

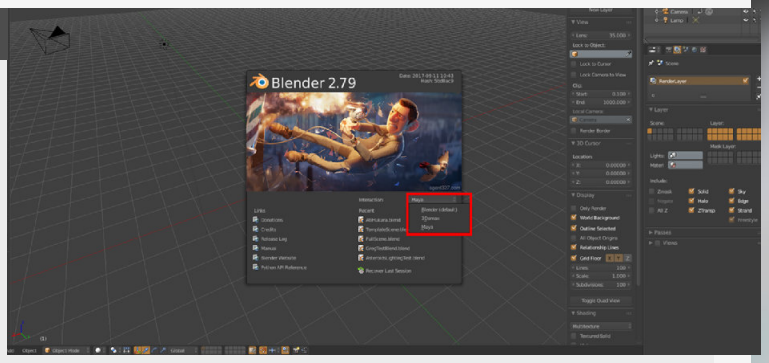
As of writing this 2.8 was not yet in sight, but I had heard some great upcoming things about the program and it was also super interesting to me that this program seemed to be more stable compared to Autodesk competitors, while also being free.

Let's dive into some things that make the switch a little easier!

1. Change the default behaviour

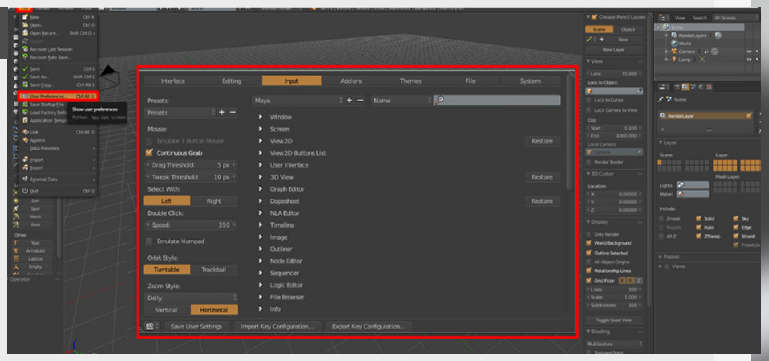
When installing Blender you are going to notice a little pop-up in the middle of the screen, if you are like me I wanted to get straight into the action and clicked this away as soon as possible, but there is actually something really useful here.

Take a look at the top right corner and notice that you can see that you can switch between different presets.



2. Changing hotkeys

A straight forward step maybe, but taking the time to tweak hotkeys further to what you are most comfortable with is one of the most important things to do at the beginning because it can save you tons of time.

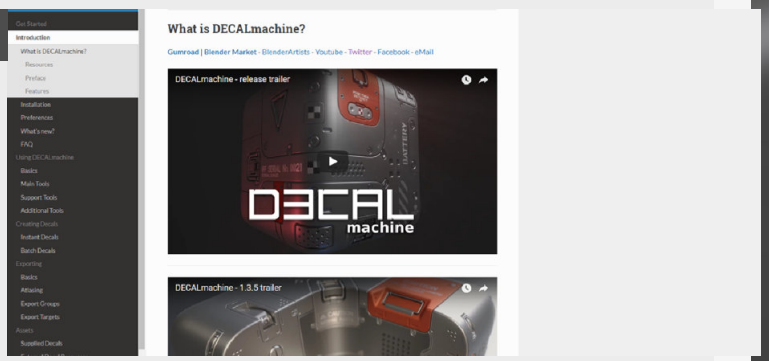


3. Driven by addons

Capsule: one click batch exporter, really useful to export lots of assets at once when working in a game engine

Decal machine: Which I am currently still testing and going through a full asset for, but this seems to be crazy powerful, it allows you to put decals on any mesh and then let's you bake that information down.

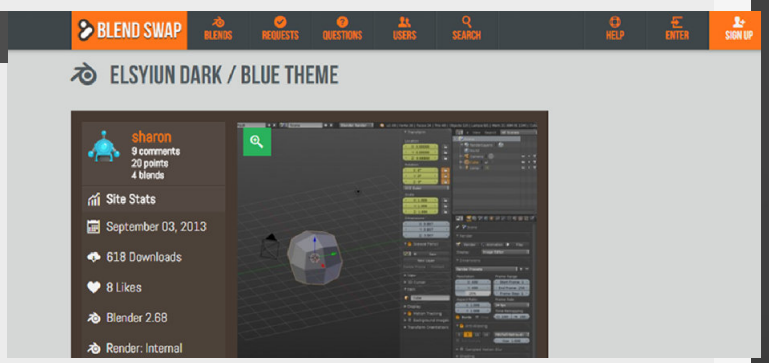
Special thanks to Andreas Strømberg (@Stromberg90 on twitter) for the tips on where to find these addons.



4. Setting the tone

This might seem petty but this was the biggest hurdle when getting into Blender. Take the time to change the theme to something that suits you perfectly, it's super easy but it really helps you stick through the initial "getting used to blender" phase.

I personally use the Elsyiun theme.



How to use instances

Introduction

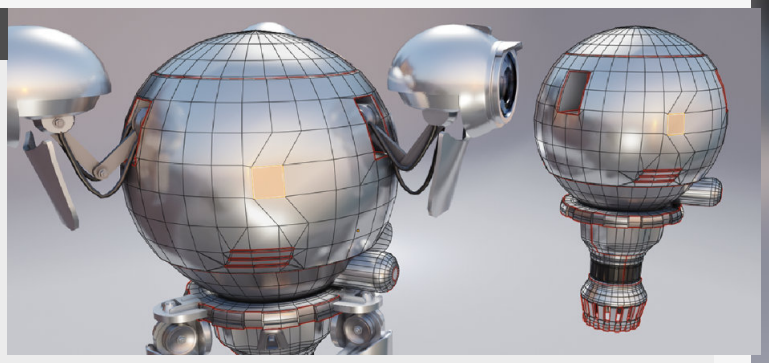
Let's dive into a real timesaver when it comes to modeling, and these are instances, which are super useful for making modular pieces and testing them inside of the modeling package, as instancing will just propagate all the changes that you do to the master or any of the other meshes to all of the others that are instanced.

1. What are instances

This is the use of Instances to quickly make changes along multiple instances of the same asset in a scene.

This is especially useful when working on modular assets that need to be tileable and usable with multiple different other assets, on with this you only need to do he adjustments to one asset instead of multiple assets (which is also going to make it cheaper/more performant once we are using these in a game-engine.)

Let's have a look on how to use these in different packages

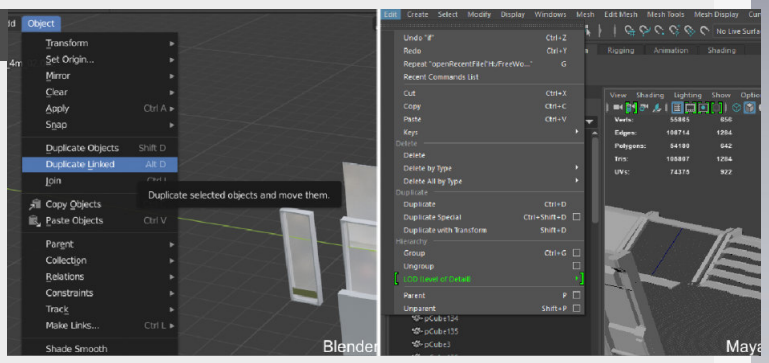


2. Supported programs

Blender: Duplicate Linked
Maya: Duplicate Special (with instance active in the popup menu)
Max: Instances or References

All of these options basically react in the same way, they will provide a two way duplication of the asset, so you can change the master or the instances mesh and they will remain in sync.

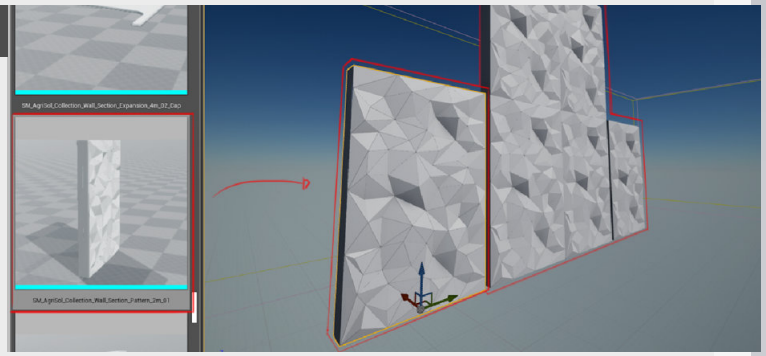
There is one exception to this rule, with the References in Max, the instances only change if you change the master mesh.
, not the other way around.



3. Unreal Engine

Instancing come more naturally in game engines especially if you are using modular pieces and will be duplicating things around and using the same mesh multiple times in the same level/ environment.

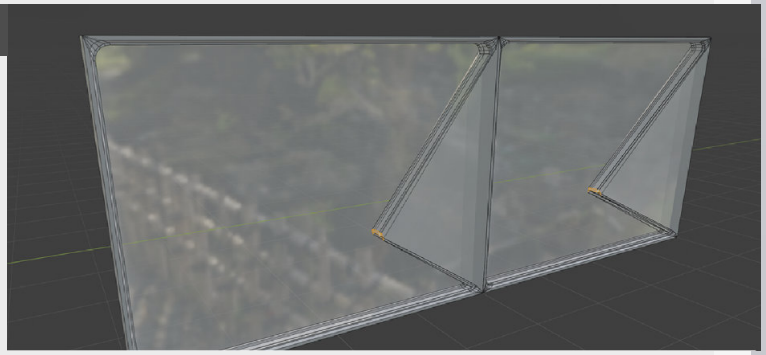
For example when this is done in Unreal Engine, it will automatically "instance" the mesh and make it cheaper on the memory. The reason why I put Instance in those parentheses is because they will still increase drawcalls.



4. Closing thoughts

I was already using this for a long time in Unreal Engine itself, testing modular kits when I was building them inside the engine, constantly switching back and forth doing smaller adjustments in blender, exporting them to Unreal Engine and then doing this over and over if I needed to. (what a waste of time!)

So I decided to switch it up to using Instances inside blender and doing my main bulk of the iteration in there.

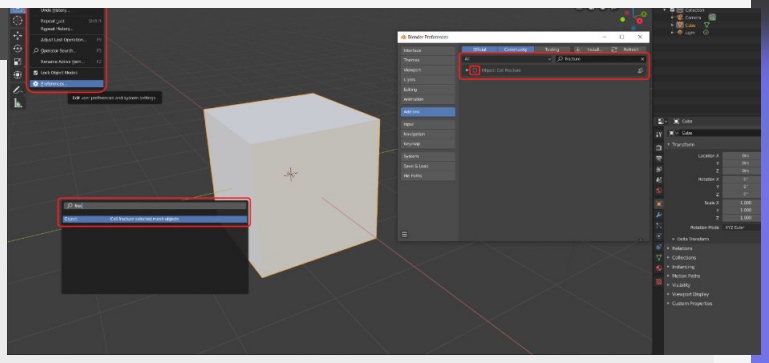


Blender fracture addon

Introduction

The fracture modifier is super powerful for the creation of random looking cracks and geometry. There are things outside off these applications that we can use this modifier for too, so let's dive into it.

1. Activating the addon



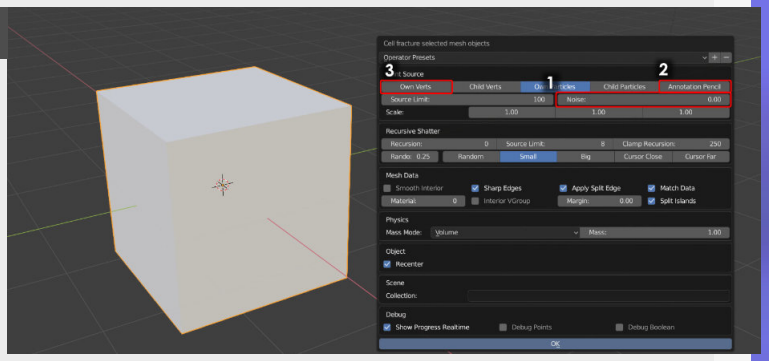
First off all lets see how we can activate this add-on and get us started.

Let's go to **edit > preferences** and in the add-on section we can use the search function to search for **"fracture"** and then just tick the checkbox to activate it.

We can then find the fracture tool in the 3D view by going to: **Object > Quick Effect > Cell Fracture**.

Or another option is that we can look for it using the search menu and look for **"Fracture"**

2. Controlling the fractures

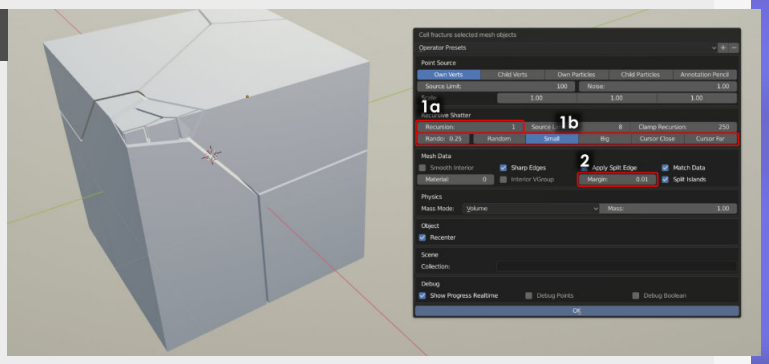


Now we got this fancy add-on installed its time to get to work, first off all the fracture add-on will divide the mesh in straight lines until we start introducing some noise **(1)**. This will then introduce some random rotations and offsets to the lines that are cutting the mesh.

There are a ton of other options that we can play with here, for instance annotations can be used to draw lines to guide the cutting of the mesh **(2)**.

The option that I use is adding more cuts in the geometry and up the density of the mesh in the place where you want more cuts and setting it to fracture according to the vertices **(3)**.

3. More parameters

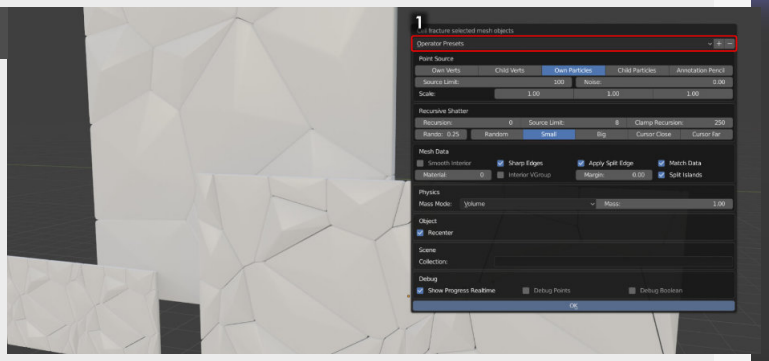


Some other options that are nice to play around with are the recursive fractures **(1a)** where it will do a secondary pass on the fractures, fracturing the already fractured pieces from the first pass.

This comes with some additional parameters like breaking the larger, smaller, close or far to cursor and even random **(1b)**.

There is also a handy offset build in that allows you to control the spacing between meshes **(2)**, this was particularly useful when experimenting with making these shapes and then using them as booleans.

4. Presets and experimenting



Some other options that are nice to play around with are the recursive fractures **(1a)** where it will do a secondary pass on the fractures, fracturing the already fractured pieces from the first pass.

This comes with some additional parameters like breaking the larger, smaller, close or far to cursor and even random **(1b)**.

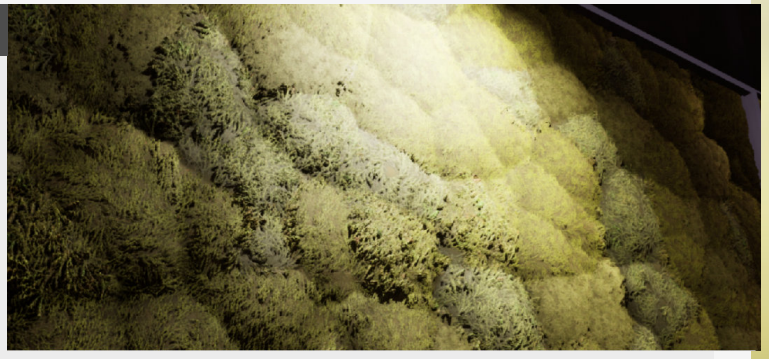
There is also a handy offset build in that allows you to control the spacing between meshes **(2)**, this was particularly useful when experimenting with making these shapes and then using them as booleans.

Unreal Engine Contact Shadows

Introduction

Contact shadows have been here for a while now (since version 4.14), but not that much people know about it or use them. They are a great way to add visual fidelity to a scene by adding a more accurate approximation to shadows with dynamic lights.

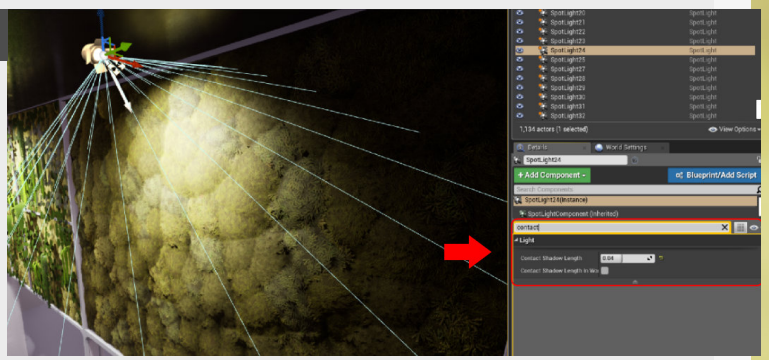
1. How and when to use?



There are some specific usecases that apply for contact shadows, they are not going to do that much if you have a lot of smaller, real fine detail on a texture for instance.

It will need some slightly bigger chunks of geometry to work with. So make sure that you have bigger chunks that the contact shadows really love to use.

2. Usage and location



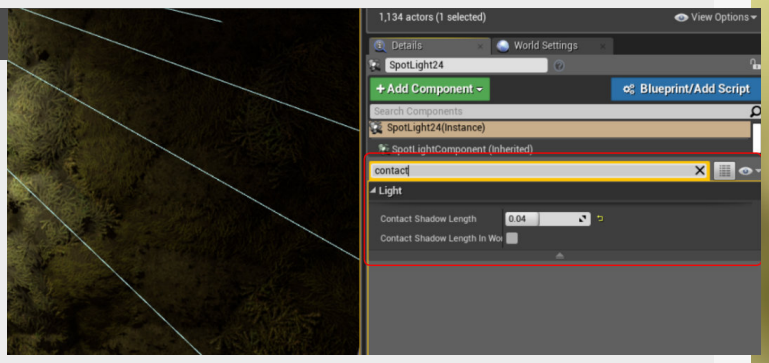
These contact shadows are available on all types of light, except for skylights.

Contact shadows and it's settings can be found inside the light panel's hidden drop down menu.

There is no checkbox, it's just a slider for the length of the shadow.

A quick way to find it is searching for it in the search bar.

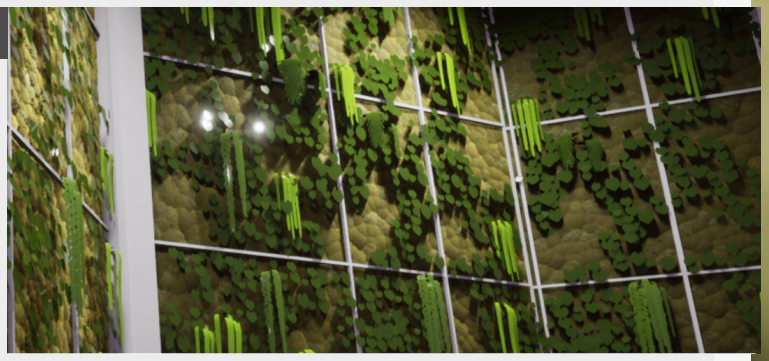
3. Shadow settings



The settings for these kinds of shadows are super simple though, you can just set the distance of the shadows. The only option that you have is changed it from screen spaced to world unit based.

The standard way of adjusting this is based on the screen, so meaning that putting this to 1 is going to cast a ray across the entire screen, where 0.5 only half and so forth.

4. Remarks



Contact shadows are really nice visually, but as most people tend to use the static lightmaps that get baked inside Unreal Engine which includes these more minor shadows.

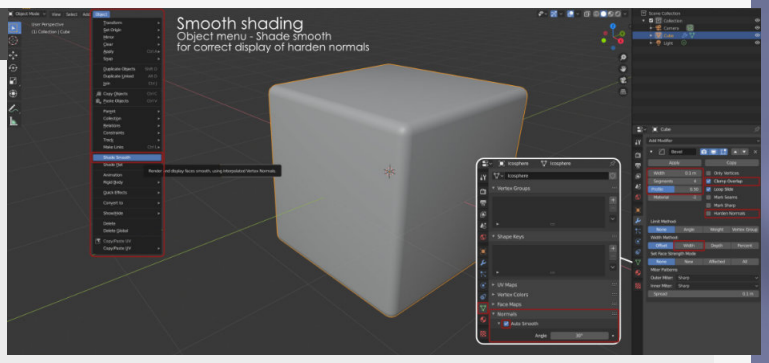
So this is more something that is specific for people that use dynamic lights because baking shadows will be accurate as long as the lightmap resolution allows for it on static objects/ lights.

Bevels in Blender

Introduction

Bevels are becoming more and more important the cheaper geometry is becoming to render for the GPU, with games like star citizen working with a Mid-poly pipeline might be nice to dive into some tips on bevels. So let's dive in.

1. General settings

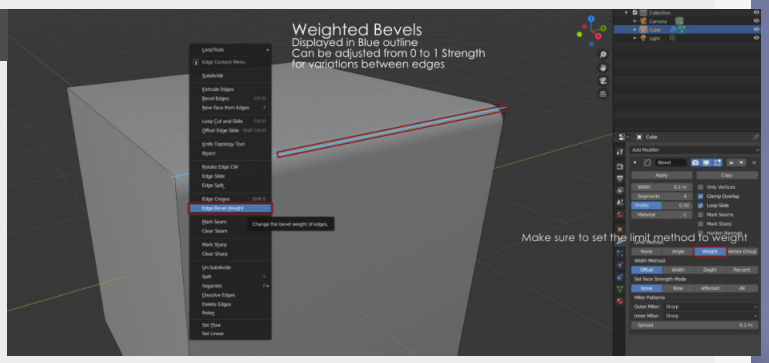


First of all let's dive into some more of the generic settings for the bevel. The most obvious of all settings is just the "Width" of the bevel or the amount of "Segments" as well as "Profile", allowing you control on how the bevel reacts.

Other settings that are useful are "Clamp Overlap" and "Harden Normals". In which clamp overlap regulates if the bevel will extend over the bordering edges. And lastly "Harden Normals" will match the normals to the new adjacent faces.

Make sure that when you do tick the harden normals button that you smoothen the mesh with the autosmooth function in the "Object" menu.

2. Weighting

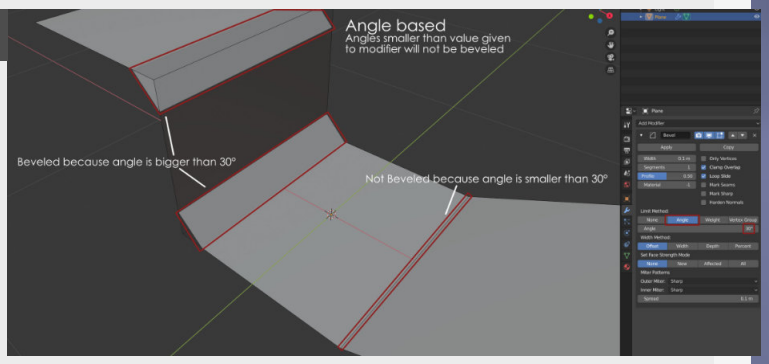


You might have noticed that by using Blender for a bit now that there is the option to assign bevel weights through the edge menu.

This is useful if you want specific edges to bevel, allowing you individual control over separate edges.

Then when adding the bevel modifier we just need to set the "Limit mode" to "Weight" to only have it influence the edges we just tagged. The amazing thing about this is that we can assign different strengths when adding the bevel on an edge, allowing us even more control and opens up options for variation.

3. Angle based

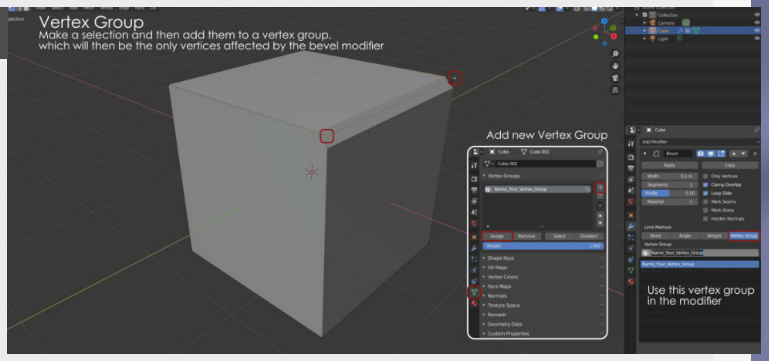


But if you are looking for a more generic option to quickly bevel the entire mesh, then angle based bevels might be the thing to look for. Allowing you the simple control of setting which edge needs to be beveled based on the angle of the polygons neighboring it.

So the value that we give the angle in the bevel modifier needs to be higher than the angle you want it to affect.

Again make sure to have the auto smooth option active under the Normals menu as mentioned in step one or the shading might look a bit weird.

4. Vertex group



Last option here, is adding bevels according to the assigned vertex group, but you might be wondering what the difference is between this and the weighting option is?

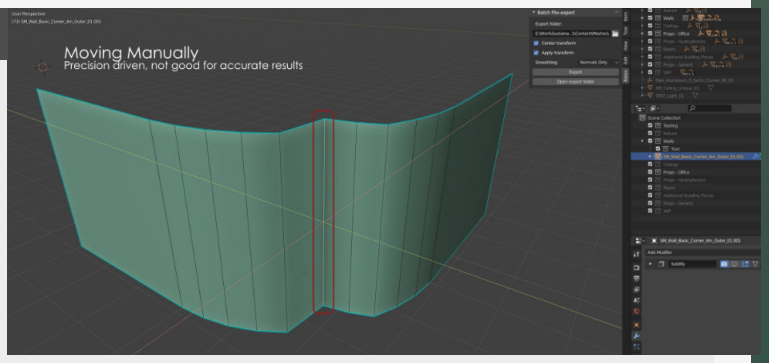
The difference between the two is that even if you were to add your edge selection to a vertex group it will convert this edge selection to verticis. Thus making unforseen connections between them too.

Smoothing curves and bevels

Introduction

Bevels and smooth curves can be tricky to edit, but there are some ways that you can make it super easy on yourself.

1. Things to avoid

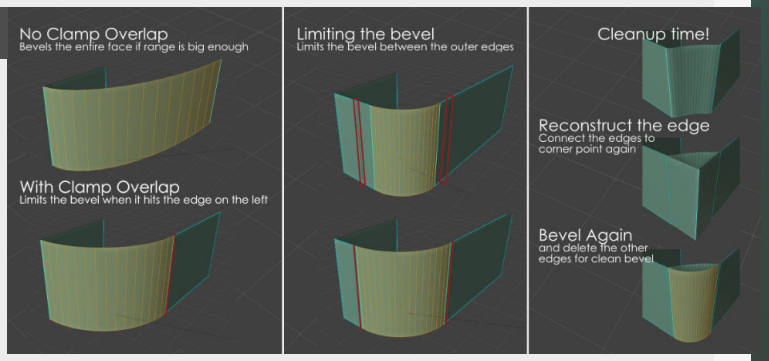


First let's talk about the things you would want to avoid when making these nice curves, the first thing being to try and do this manually, this is never going to be as accurate using some sort of guide or using a bevel to create a nice curve.

This is also going to be way slower than doing it from scratch in most cases, especially for Bevels.

Another thing to avoid as much as possible is doing these curves in a destructive manner, try to be able to go back and adjust this whenever possible, you never know if you m want to do some changes.

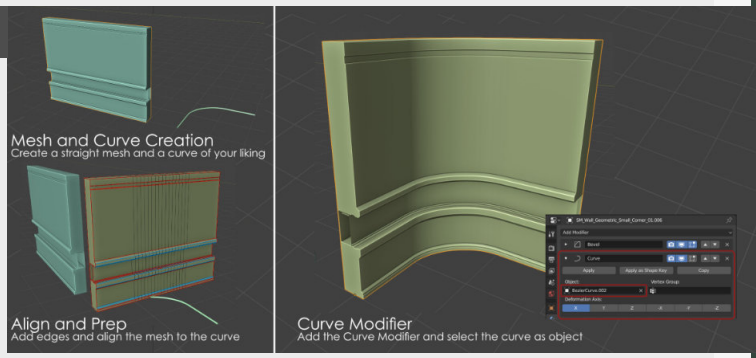
2. Beveling correctly



The first thing to dive into is probably when to do these Bevels as this is going to be you main source of rounded corners, for most of you. There are ways to do this without creating a mess from your mesh though, some examples shown on the right go into using clamp overlap to limit the distance the bevel will go if it reaches another edge or setting this limit yourself with new edges, together with auto merging of vertices this can be a really nice an quick way to remove vertices.

A non destructive way could be using the bevel modifier and having it adjustable all the time, more on that can be found in the weekly tip for **#87**.

3. Using curves

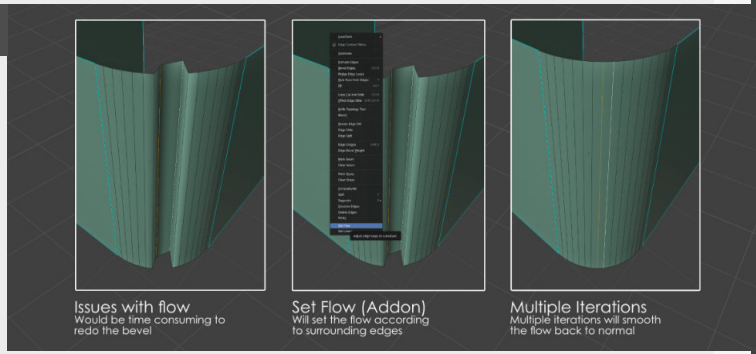


Using curves and the curve modifier in Blender to create easy curves

A thing that I recently got into when moving to Blender is using the curves and curve modifier to make nice curves surfaces. The way this works is that you have your normal straight mesh for my example a wall piece and then make a curve that will be used to bend this straight piece along itself if you use a curve modifier. Make sure to add enough new edges so that it can actually be deformed.

This is not exclusive to Blender though and you should be able to do this in Max or Maya too.

4. Relaxing or edge flow



There is also an easier option when you already have a bunch of edges that should make a nice curve but now don't do that anymore, there are multiple options for this like the **"Edge Flow"** Addon for Blender or the **"Edit Edge Flow"** in Maya, the second actually works more reliably at the time of posting this.

Actually since I'm working in Blender there are other ways that can help you smooth these out in different ways such as the Build in **"Looptools"** Relax function, but for my taste the Edge Flow addon does a better job and is quicker and more reliable.

Mass replace objects in Blender

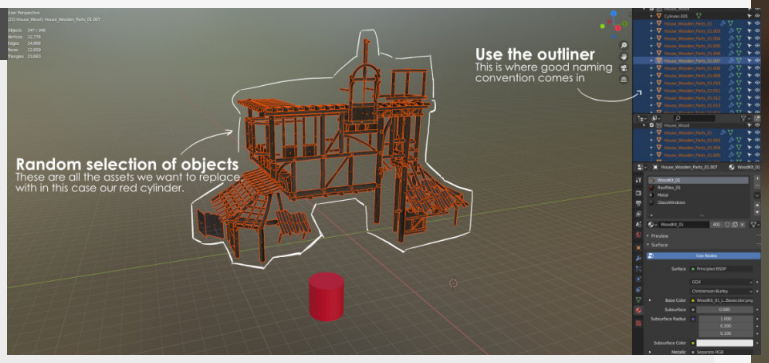
Introduction

Have you ever had this issue where you made this awesome looking wall that uses a combination of a tileable texture and separate bricks. But then you have made an even better looking brick for the wall, so how do you quickly replace all these previously places bricks that aren't instanced? That's what we're going to dive into!

1. Select assets to replace

First of all, let's select all the assets that you want to replace in our scene, good naming convention and an organised scene will definitely help here!

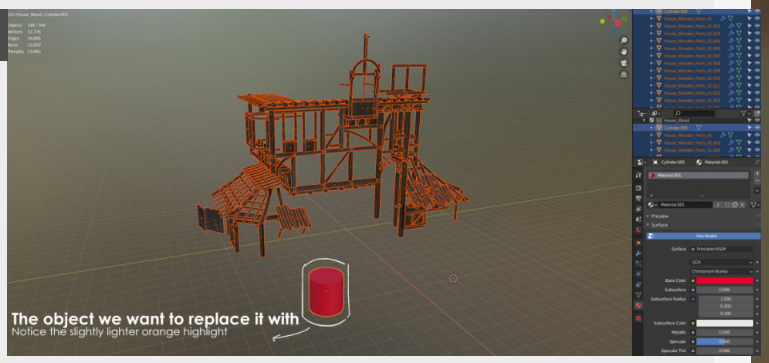
Just filter for the name of the bricks in the outliner and you should be good to go!



2. Select object to replace with

Now for the second step we're going to select the mesh we want to replace it with, once we select it we can see that it also get's an orange outline, but slightly brighter, indicating that this object is the last object selected.

This can also be helpful for other things like liking modifiers, etc...

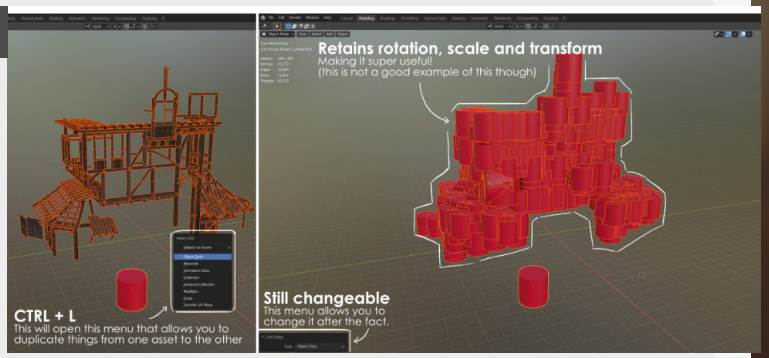


3. Make links menu

"CTRL + L" is going to give us access to the "Make links" menu, this is where you can link object data, modifiers, collections, etc... together to other meshes in your scene.

For this little setup where going to select **"Object Data"** this will then replace all the object date with from the asset we selected last to the other ones.

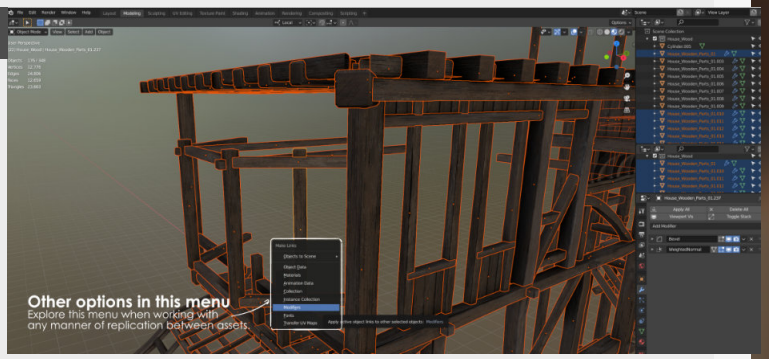
It's good to know that this will retain their transform, so all the scale rotation and transformation data will be retained, which is awesome!



4. Other options and benefits

This is a great way to quickly duplicate date from one asset to the next, and it's good to know that this will retain their transform, so all the scale rotation and transformation data will be retained, which is awesome!

I also use the modifier option from this menu a lot, for example when you have a **"bevel"** modifier and a **"weighted normal"** modifier setup you can quickly add this to a large amount of assets in your scene without having to recreate them over and over.



Making debris with particles

Introduction

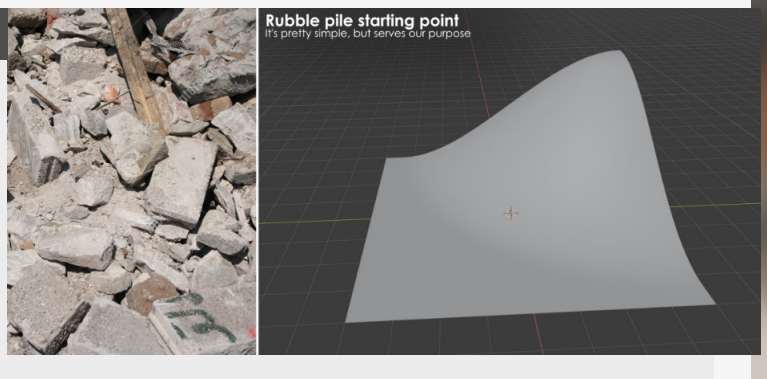
Scattering objects is most of the time done in engine, but it can also be done in different ways inside of your favourite modelling program, this weekly tip might be specific for Blender, but they can also apply to other modelling programs too.

1. Receiving Object

Our starting point, this can be any mesh you can imagine, I've been using this workflow myself for more hardsurface objects but also for foliage. You can let the normals of the underlying mesh decide the directionality of the meshes scattered on them.

(More on how I use this for foliage can be found in "Blog #69 - Blender Foliage with particles")

For this example we will be looking at a simplified rubble pile for example. At this point it looks pretty basic, but it serves as a good point of reference.



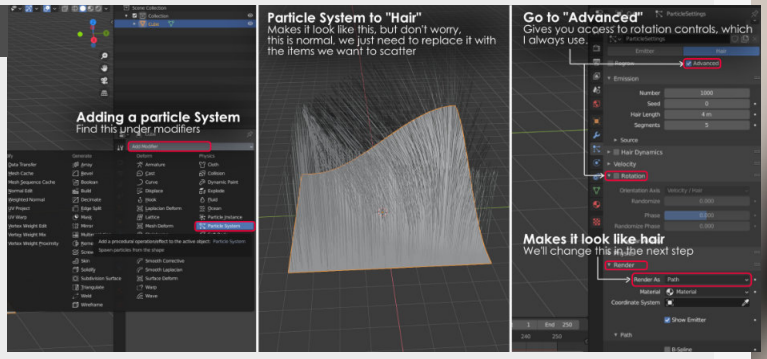
2. Particle setup

First let's add a particle to our mesh:
- Add a particle emitter
- Set it to **"Hair"**
- Tick the **"advanced"** box, this gives you additional control

Most important tabs and setting to play with:

Emission:

Control over the amount of particles and seed it uses.
Velocity:
Can influence the placement of particles, keep this in mind
Rotation:
Speaks for itself
Render:
Controls over what we spawn, see next section.

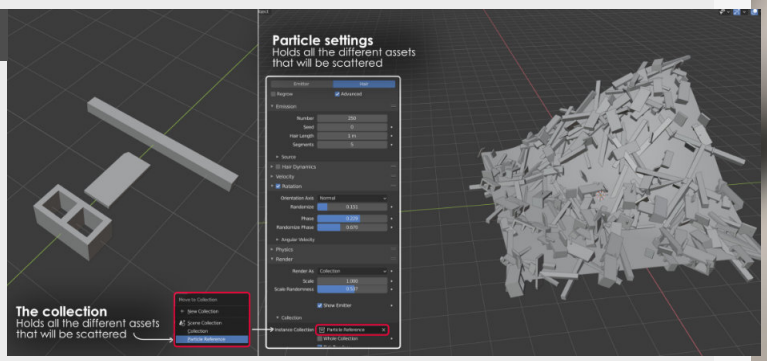


3. "Objects" to scatter

Now let's move onto the objects that you want to scatter, there are a couple of ways that you can do this. But my favorite way and the most versatile way is to do this via the **"Collection"** option, this gives you the option to add objects into this collection and it will update the particles that are getting spawned.

If you only have one object that you want to scatter, then you would select the **"Object"** option instead.

The settings for the particles to spawn can be found in the Particle Settings > Render > "Render As". Here you have different options for what you want to render.



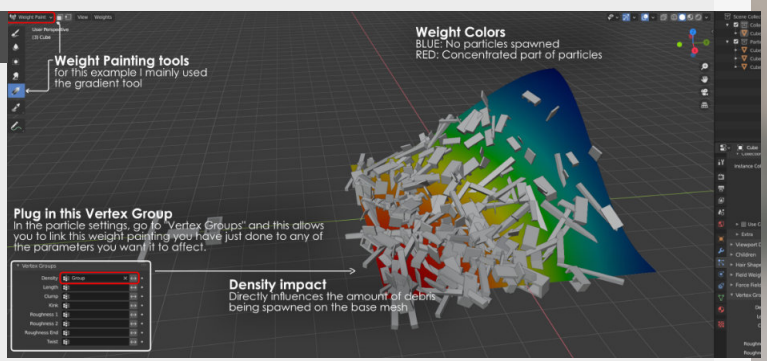
4. Weight painting

Now this is where we get into the interesting stuff, we can build an additional layer on top of this setup by using "Weight Painting" to influence a parameter of these particles.

The colors in this gradient is go from **Blue** (none) to **Red** (All). We can use this weight painting to control a bunch of parameters off these particles, such as **"Density"**, **"Length"**, **"Clumping"**, etc...

These can all be found under the **"Vertex Groups"** dropdown in the particle settings.

Weight painting's density and amount of control is affected by vertices, so keep this in mind when working with it.



Tileables to geometry

Introduction

This time we're looking at a really simple trick if you don't want to use fancy vertex displacement or fancy shaders in your real time environments.

1. Going from base shape to highpoly and lowpoly

We'll be having a look at this little pillar as our test subject for this workflow.

The prerequisites

- Our textures that we applies to the mesh, giving us a representation of the final result and making sure that we have both our textures and heightmap aligned.

- Model with a finalized unwrap, this workflow is destructible in the last stage, so make sure you have

- A Height map (the greyscale texture shown in the image below), we will be using this texture to drive our displacement modifier

Using the Heightmap

Now that we have this heightmap we can then add a **"displacement modifier"** to our pillar which takes our heightmap we put in and offset the vertices based on the values coming from the heightmap. (Closer to white the bigger the offset will be)

But at this point the displacement alone doesn't do that much, because we don't have the vertices or resolution on our mesh to give us a good representation of the heightmap onto the vertex. That's why we're going to be using a **"Subdivision modifier"** (or turbosmooth, etc...).

This subdivides the mesh and gives us more topology to work with, for the top and bottom we use an **"Edge crease"** in blender to sharpen those edges

Displacement modifier

Once we've done so we can see that the heightmap is now influencing the mesh and pushing the vertices outwards. (You might need to turn down the strength of the modifier)

When we add this method we also need to make sure that the modifier is set to do it's displacement on the UV space instead of something else, this will make them match up with the texture that is already on the asset itself.

Making it game-ready

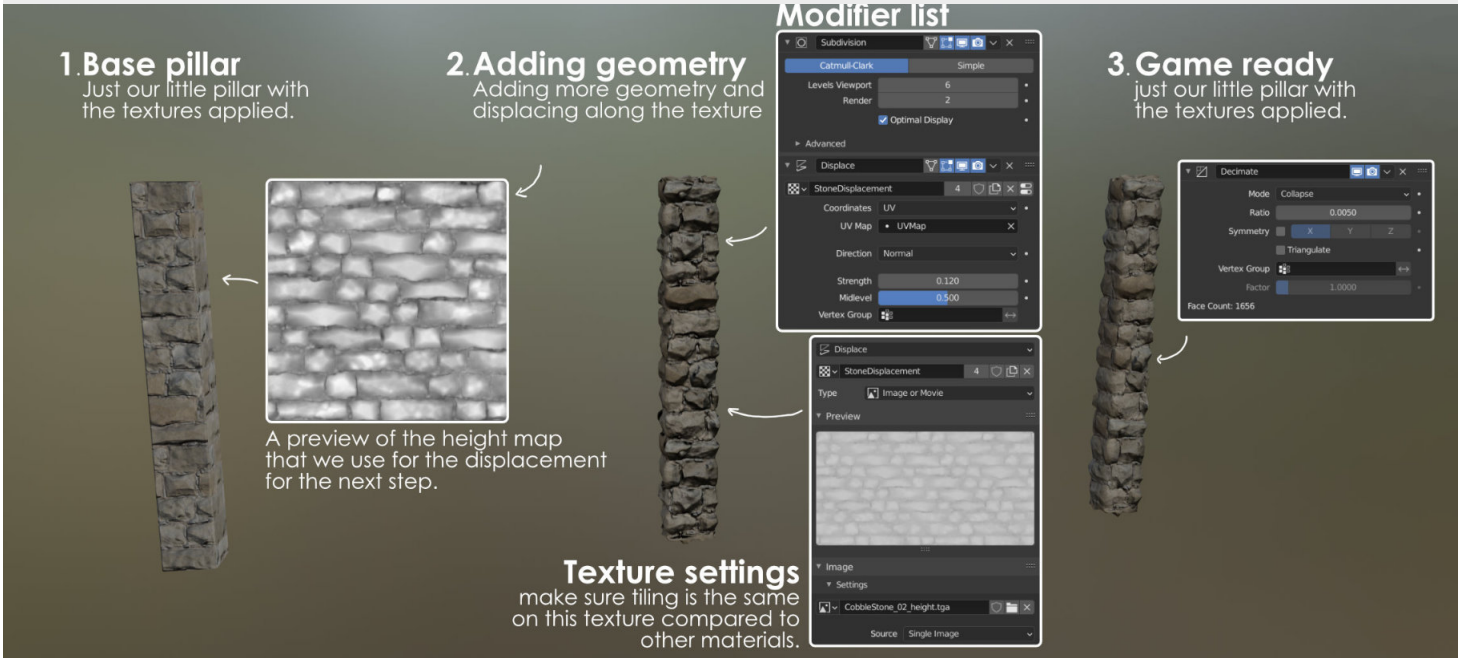
Now we have a nice looking pillar that is nice and displaced and conforms to the texture, but it's not really game ready yet with all the subdivision modifiers we added, so making it so we can use a decimation modifier to push it back down.

For these pillar for example I'm getting close to **1000** to **2000** polygons roughly, but this depends on the details to the texture and how close you are going to add. Plus, the decimation modifier makes it super easy to psh it lower if needed.

Keeping this all non-destructive

For this example I always collapsed after the subdivision + displacement pass, making the mesh a bit easier to work with, but this is purely optional.

You can also just add the **"decimate modifier"** after all the modifiers mentioned before to have it non-destructive and you can adjust it on the fly if needed.



Heightmap to Geometry

Introduction

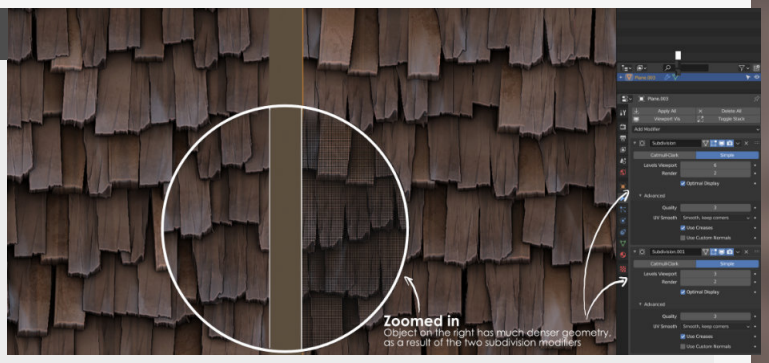
This is a technique I've been using recently in my personal work recently, it's nothing revolutionary but a good tip nonetheless to create really nice detailed geometry based on textures in a really quick manner.

As always I will be using Blender for this, but this can also be achieved in your modeling package in different ways.

1. Adding geometry

First off all we need to add geometry to our subject, we need this to control the resolution or how accurate the heightmap will deform our geometry. The denser our geometry the more accurate the initial projection will be.

For this I just add a subdivision modifier (or Turbosmooth for you Max lovers), depending on the geometry and how dense it was before you might want to add two to create the density you are looking for.



2. Displacing the geometry

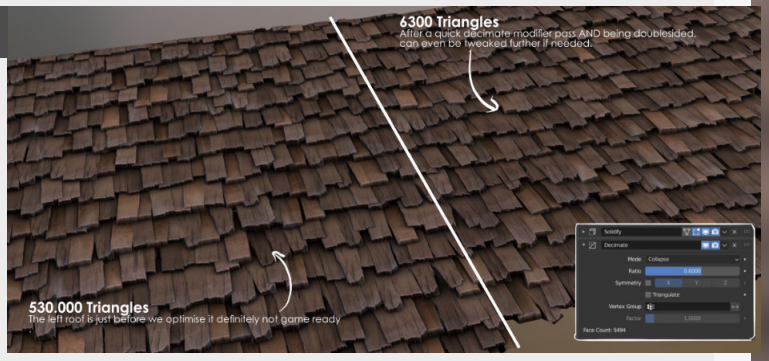
Now that we have nice dense geometry, it's time to move it a bit. Add a "Displacement" modifier next and make sure to add your heightmap texture to it so that it will use the values of them to deform them.

White values will push them out fully where as Black values won't push them out at all. Since my example is pretty simple, I use the "UV" projection method so the heightmap texture is going to be applied as a normal texture on the UV space of the object.

3. Making it game-ready

That's looking nice, but it's also waaaay too much geometry to use in a game engine (Not every engine allows you to chuck everything in it and "Nanite it up", so keep things optimized!) A simple way to do it on this example would be by adding a "Decimation" modifier, this will reduce the geometry based on a factor you can put in.

A good rule to follow here is to reduce it to the point where the visuals of the mesh don't completely break. in my case this



4. Final remarks

This is a pretty quick but demanding way to create geometry based on your textures, depending on the details and complexity of your assets. It's demanding because you always need to feed it enough polygons to actually displace the surface.

One of the main drawback is that it will treat the surface as a single mesh (as seen in my example) so if you want to have nice lose roof patterns with individual tiles it might not be your best way to go.

Face Weighted normals

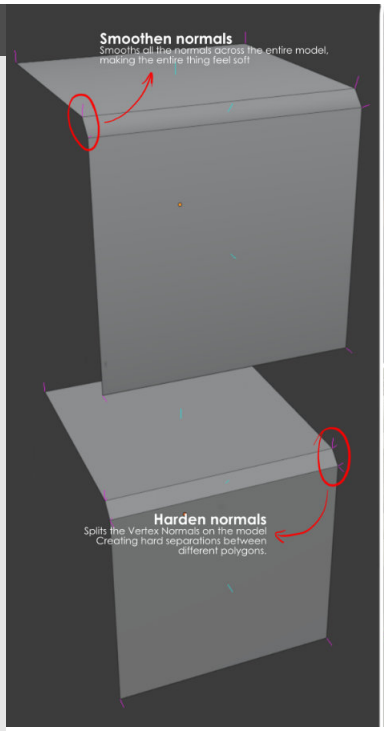
Introduction

With the increasing popularity of the **"mid poly"** workflow (more on that in a previous tip) there is also an increase in the focus on improved shading on their bevels. This time we dig into "Face Weighted normals" and their applications

1. What are face weighted normals?

Face weighted normals are called that way because the normals are weighted or flat towards the faces and have a smooth transition in between. This means that if you have a bevel it will smoothen the transition within the bevel, but not outside of it, like it would normally do on another bevel.

On the image to the right you can see what the difference is when it comes to the normals of the vertices and how this affects the shading of the mesh. This shading will be different because of the vertex normal coming from all the individual vertices, in the next step we'll have a look at the differences.



2. Visual Comparisons

let's look at some examples with a little cube here.

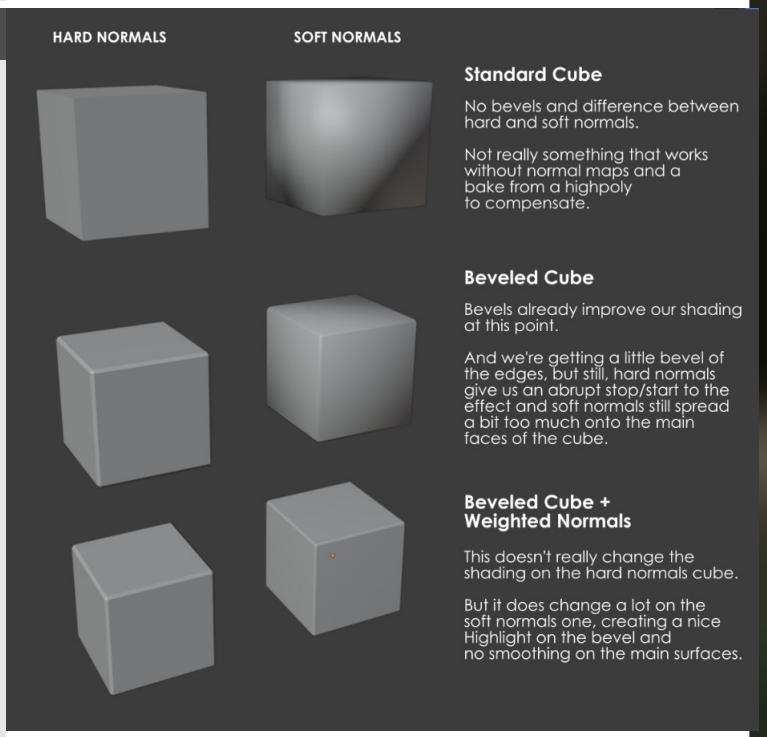
If we we're to smooth all the edges on this cube the shading will look like it's trying to mimic a sphere, and if want to counteract this effect without adding more geometry we need to either make some of the edges hard or use a normal map to counteract this effect.

On the opposite spectrum, if we make all the edges hard then we just have a flat cube that doesn't look really nice either, you can really distinguish all the geometry by the shading.

But in the case where we don't want to have a normal map we need to add a little geometry for some nice shading on the edges. For this we just need to add a little bevel on the edges and then see how we can approach this from here.

If we we're to smooth all these corners, then you can see that it looks a little bit better, but the shading still spills over to the main surfaces. And if we apply hard edges, you guessed it, all these surfaces will just be split.

So, now if we want to get our meshes to look really nice with correct shading, that's where Face Weighted Normals come in. This cleans up the shading and makes it so that the bevels read really well, mimicking what you would achieve with a normal map and less geometry.



UV Packing, optimize resources!

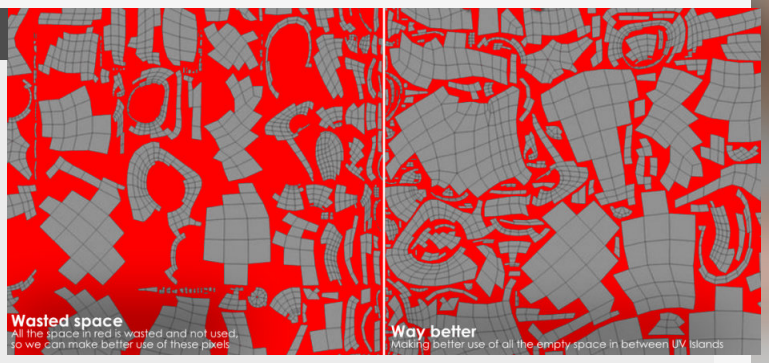
Introduction

UV packing is a really great way to optimize your texture resources and squeeze the most out of them by minimizing the lost space in your UV space, packing the UV islands tighter together and reducing pixels we waste in out textures.

1. Optimizing resources

In games we work within tight memory budgets and have restrictions when it comes to disk space, these have become more lenient over time but it's still a solid practice to keep this in mind.

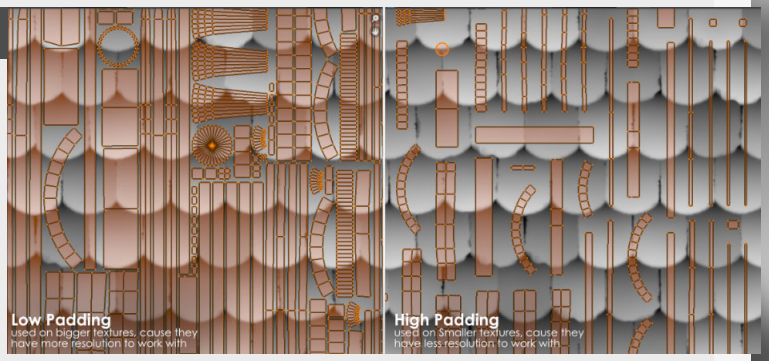
The actual limitations here are usually decided by a bunch of factors such as: Target platform, pipeline optimizations, which workflow is used and how much is reused, etc... and with technology shifting this will never be the same one project to the next.



2. Padding between islands

The only thing that we need to keep in mind when we pack our UV's is that we have padding between the different UV Islands. This padding allows for a little bit of a buffer (or bleed) for the colors from our textures and thus will not cause them to bleed into the UV Island that is next to it.

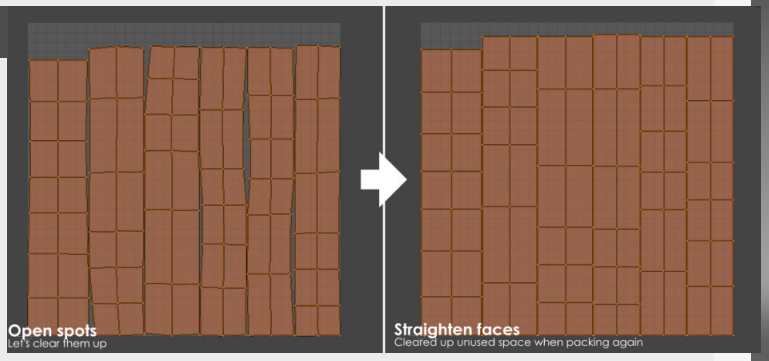
This also means that padding size changes with the texture resolution, as in bigger pixels on the texture will cause bigger color bleeds. Smaller pixel (more pixels on the texture) will minimize the distance it can spread.



3. Straightening UV shells

A trick for good UV's but really applicable for packing is to straighten the UV shells as much as possible.

This for one will make it easier for the packing algorithm to pack things together, reduces bleeding between UV shells (Because we don't have any slightly angled UV edges) and is also better for texture resolution too, as you can get more on one UV space and can thus increase texture size.

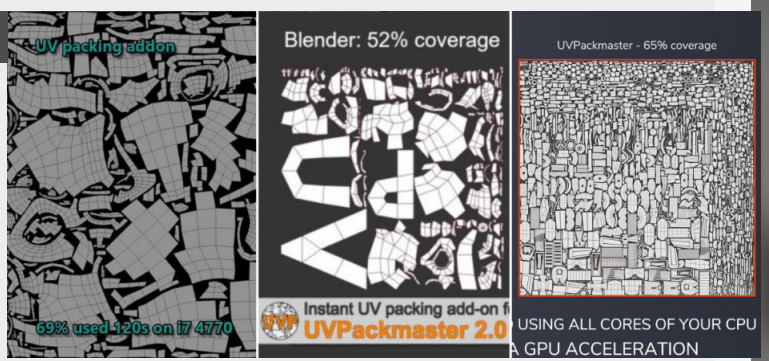


4. How do I get started?

Every program has it's own packing addons or functions. So give these a try if you want to get started or even test standalone tools.

"UV-Packer" or "UVPackmaster" are my recommendations here and can be used both in Blender and in Max.

Standalone tools such as "Roadkill", "UVLayout", "Ultimate Unwrap 3D" or "Rizom UV" can be really great programs, but I've personally always tried to keep the amount of programs I use down to a minimum, but I've heard great stuff about roadkill.



Stylized Asset Design

by **Nikkita Racquel**

Introduction

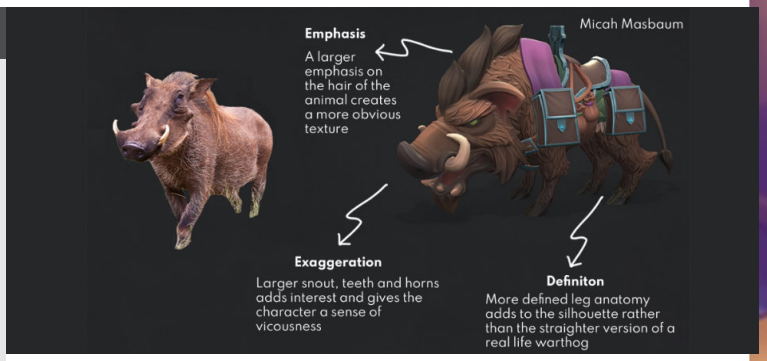
Making stylized art can be very exciting and freeing as an artist however as much freedom as stylized presents as an aesthetic, it can be daunting to dive into for some. This is just a few tips and tricks that might aid those in their own creative process.

1. Knowing the rules

This term of "laws" in this case speaks to the basic form/s of any asset an artist might make. Knowing what something looks like or how you might construct the real thing directly feeds into how you might go about then morphing those shapes. This can be used with anything but tends to be most noticeable in man made objects as straight angles and symmetry is pretty much a standard of manufactured products (furniture, electronics, etc). Reference use is key is this stage of design. Real life references are great to inform the artist of the base form while stylized references can be helpful in determining the stylization the artist might want to create.

2. Shapes and Silhouettes

With creating stylized art, the ability to push and pull things is in abundance compared to that of a realistic art style. Artists can utilize a wider variety of shape language to convey the style, setting and the type of asset that particular piece is. Overall shapes can get pretty weird and wacky at times which is where paying attention to the silhouette of the mesh as it is being made is very important. It informs the artist of how the object might be perceived by someone who doesn't know what it is supposed to be straight away, like someone playing any game for the first time. A simplistic silhouette gives the viewer an idea of what something could be at first glance.



3. Imitation as flattery

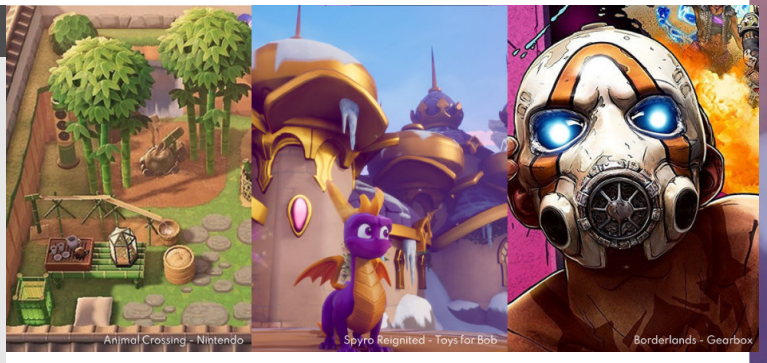
There is absolutely nothing wrong with copying another game's art style to help you learn. It tends to be one of the first suggestions to help artists gain an understanding of the building blocks that make up any given style. Those building blocks can then be used as the foundation for the artist's development of their own style which can be a tricky thing to navigate.

Techniques like this help to alleviate the pressure of constantly actively developing skills as an artist and lets it happen more passively while the artist can enjoy creating, taking the style cues of their chosen aesthetic.

4. Find your own way

As much as you might hear there is a wrong and right way to do things, reality is, there really isn't. The spectrum of stylized is so broad and ever changing that there isn't one way or even ten ways of doing something, there's so many more.

Everything merely serves as advice that you could choose to use as your own guidelines or just go wild! Have fun with it and find your own way to develop your unique stylization :)



Batch exporting

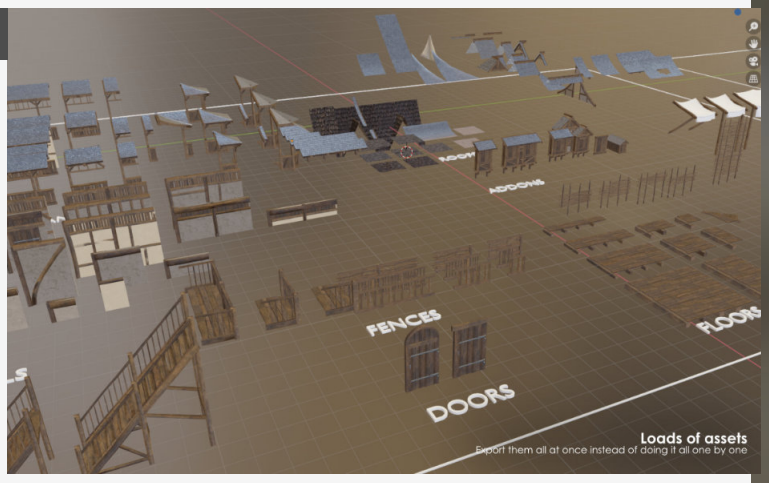
Introduction

Exporting assets from a 3D software is usually done through manual exporting, going through the menus or pressing a hotkey. But what if you are working on an entire modular set that contains 10 to 100's of meshes, that's where batch exporting comes in

1. Overview

Normally a batch exporter takes all the selected assets in your scene, puts them to world origin (this depends on the software you use for Pivot Placements) and then exports each individual asset in your selection as a different FBX file for you to import into your engine of choice.

If you are working on a modular kit this is extremely useful because you don't have to manually update all the assets one by one.



2. Essentials for a batch exporter

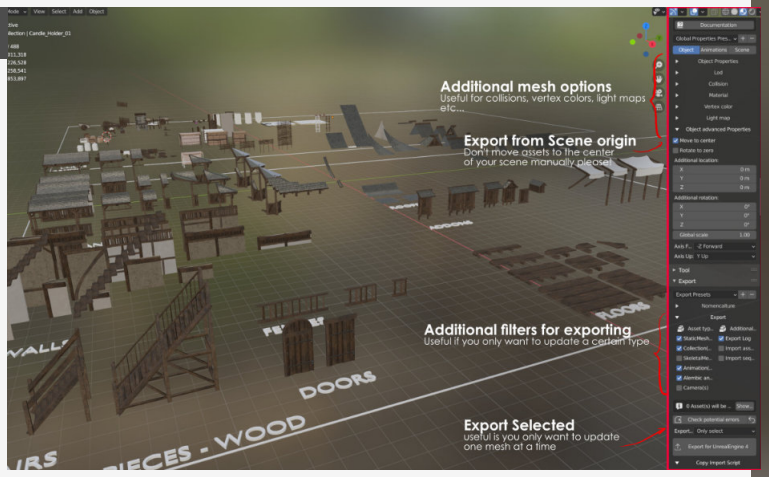
Now if you are thinking about getting a batch exporter tool for exporting, then the following things you just NEED to have in that tool.

Setting to world origin
Especially useful for having the correct pivot placement, this depends on the program you use though, most tools now have this embedded in their default exporter, so might not be required in the addon perse.

Export Selected

The addon also needs to allow you to export the meshes you have selected as well, if you don't have this and are working with a lot of assets then you will need to export all of them whenever you want to export even a single mesh.

Collision exporter
Collisions are also important if you are working with modular kits, the ability to quickly export all the corresponding collisions together with your assets is a must.

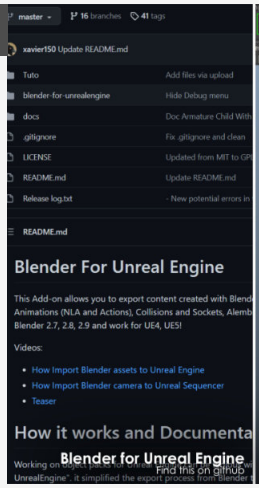


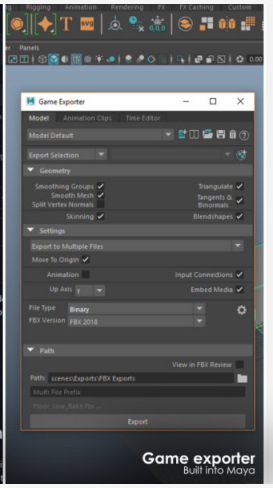
3. Addon recommendations

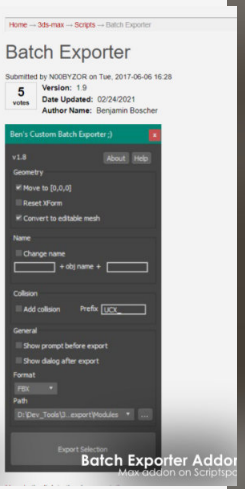
Blender to Unreal Engine (Blender)
this is the one that I've been using personally and has everything I need when it come to my modular kits with collisions

GameExporter (Maya)
there is apparently a built in tool for Maya, I haven't personally used it myself but it seems to have all the options that I would want out of a batch exporter tool like it.

Batch Export/Import (Max)
This older tool still gets the job done it seems for Max, it might even be the same one I was using when I was still using Max?







Putting details in your Geometry

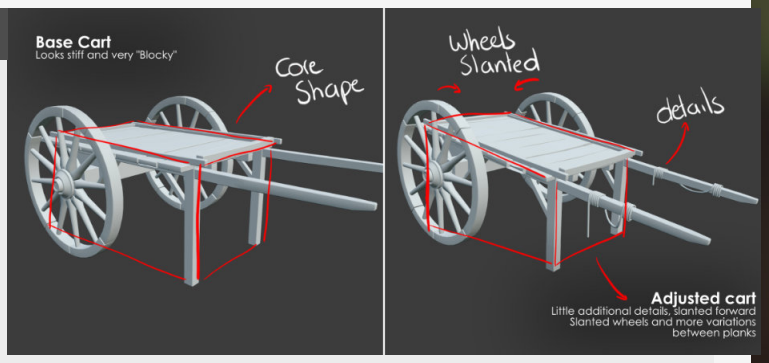
Introduction

As Environment or Prop Artists we're constantly balancing the visuals versus the performance of the assets or environments we create. So today we're diving into how you can improve the visuals while also optimizing the performance a little at the same time, saving you some geometry.

1. Silhouette comes first

The silhouette is the most important part of an asset since that's where people get to focus on first. And thus if we make a silhouette that's lacking in details it will more easily stand out as something that's lacking the details to make something feel "real".

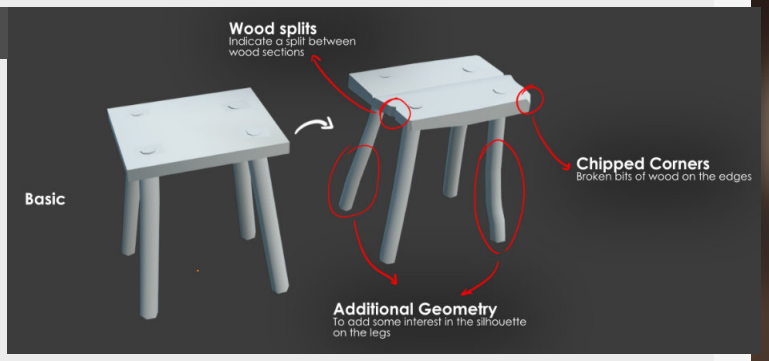
With real here I'm not talking about realism or stylized either, you can make a silhouette stand out in both styles in different ways and it's important in both. Small changes in the silhouette of a shape can have a big visual impact.



2. Adding imperfections

These small changes on the silhouette can also be applied on the internal shapes too, these Imperfections are what makes the world feel lived in and more relateable to players. If you see something that feels 3D it's because most of the time it's missing those imperfections and feels too 3D because of it.

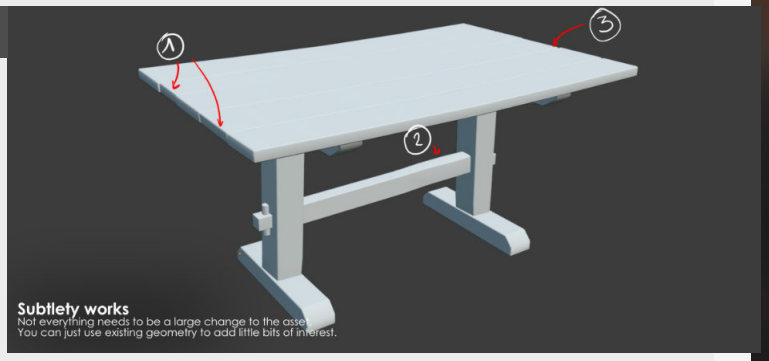
You can easily do this by taking already existing geometry and just moving it around ever so slightly. These imperfections can add a lot of life to anything you do, just look at this example on the right, and how all the clutter and imperfections make it feel lived in.



3. Nothing is straight!

Everything that you see around you is always ever so slightly at an angle, use this knowledge to your advantage and make everything slightly crooked. Obviously how crooked depends on the real world techniques used to create the asset in the first place. A hand made medieval tool will feel more unique and **"Crooked"** than a modern day machined piece of a gun for example.

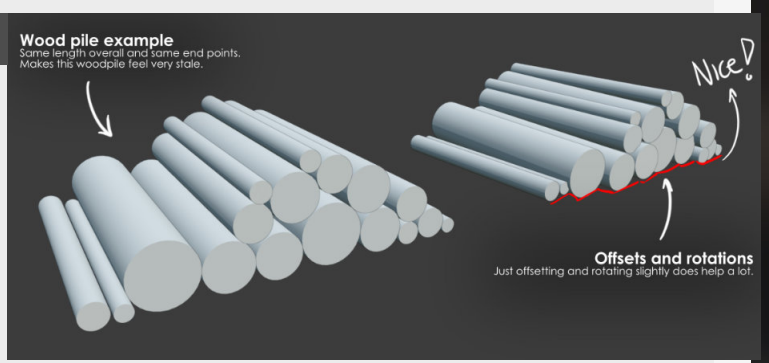
It's best to do this step fairly close to the end, and do it as a polish pass, again, using the existing geometry to really put some nice details on it.



4. Slight scale and rotation differences

Last one, add slight scale and rotation differences to repeating objects, this is a perfect tactic to make something feel handcrafted and give your assets that little love (just make sure to keep in mind the rule of how's it's made in the real world).

This is perfect for things such as bolts, nails, cables, etc… Just give each one the love it deserves!



Creating woodpiles with simulations

Introduction

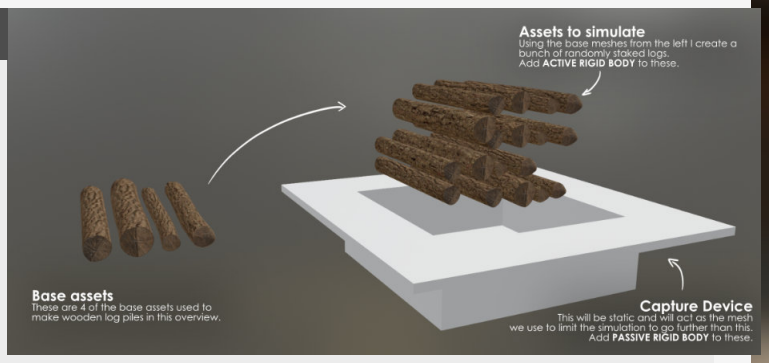
Creating wood piles has never been easier with little technique in blender, while I'm using this for wood piles you can use this on a variation of different assets with a couple of simple adjustments.

1. The Setup

Let's breakdown the little setup here first, you can see we have a a pretty large stack of wooden floating pieces above a plane with a hole inside of this and some thickness. The simulation works on two rigid body components.

Passive Rigid Body
We assigned this to the asset that we want to use to act as our collision and will remain static. So in this case this will be added to the plane beneath the stack of wood.

Active Rigid Body
This we assign to all of the wooden pieces and has a bunch of parameters that we can adjust, but more on that next.

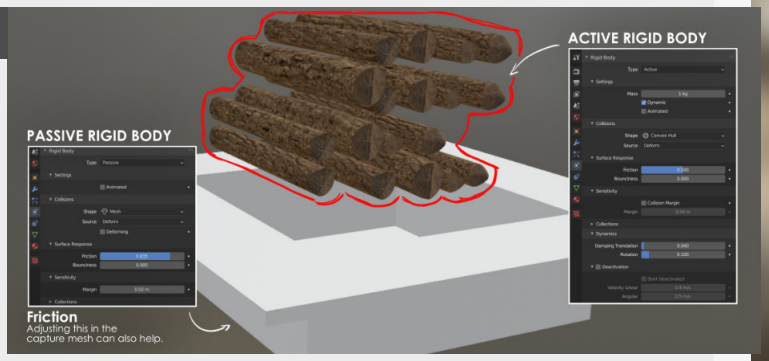


2. Parameter and simulation

Settings to adjust are the **"Mass"** which controls the weight of the object, **"Friction"** which slows the object on contact with others and both **"Damping Translation"** and **"Rotation"** which will slowdown either quicker or slower over time.

Now, let's simulate, for this we need to go to our **"Timeline"** and press play. When done so, we can see that these pieces will start to move. And once it has completed a full run on our timeline we can stop and slide through the timeline and see the entire simulation.

Now we just need to look at which **"Frame"** we want to convert and turn that into a static mesh.

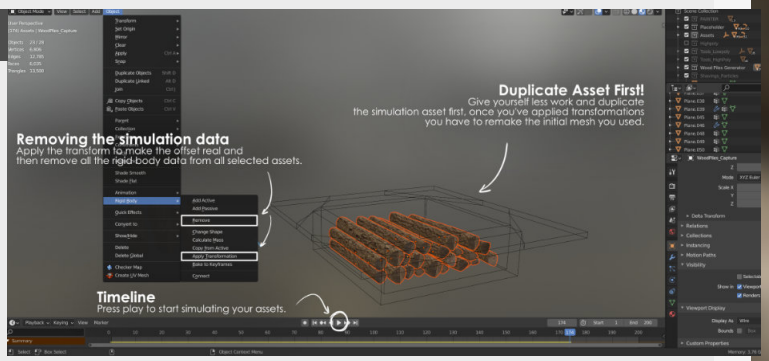


3. Converting to static mesh

This asset still has simulation data attached to it, so to make it a static asset we first need to tell it to remember the transform. To do this you need to go to **"Object"** > **"Rigid Body"** > **"Apply Transformation"**

Even with this offset converted it still has a rigid body and animation data attached as well through the rigid body modifiers attached to it too, so we need to remove them too by going to the same menu and clicking **"Remove"** under the same menu under **"Object"** > **"Rigid Body"**.

And that's it, we now have a static version of these wood piles!



4. Other applications

With a little imagination you can see a bunch of applications to this such as random wood piles all we have to do is make a different capture device, change the meshes we want to scatter and we're good to go again.

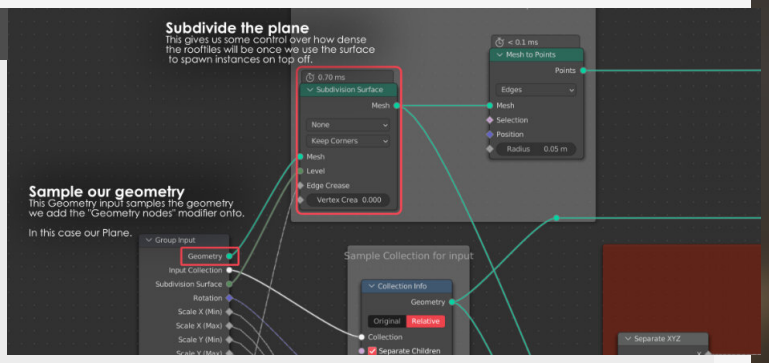
We can also use this technique for the base of tileables of materials, I'm thinking that it could be interesting to have a whole bunch of garbage assets and then bake all the maps we needed from them and you have a tileable garbage material, you just need to make sure the edges are correctly tiling as well.

Geometry Node Roof tiles

Introduction

In this one we'll be looking at the roof tiles I'm currently making for my medieval project. It was also the first time using Geometry Nodes in Blender, so not everything works perfectly with the tool we're going to be looking at, but it's amazing to create some quick rooftops for my scene.

1. Adding control via a mesh

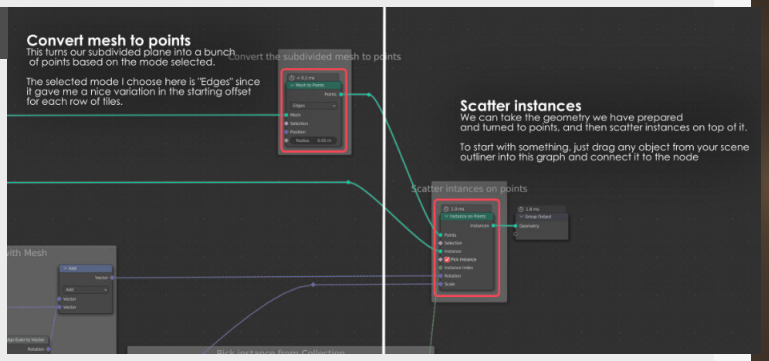


We're going to start with a plane (mesh) just to give us something to work with, we are going to use this mesh to direct the surface area of where we want to generate a roof.

I added subdivision surface (turbosmooth) to give me additional control over how dense the surface will be and the amount of vertices we will be using as points to spawn our roof tiles on top of.

To give us control outside of the tool we can drag any pin to the first group to expose parameters in our modifier.

2. Scattering instances

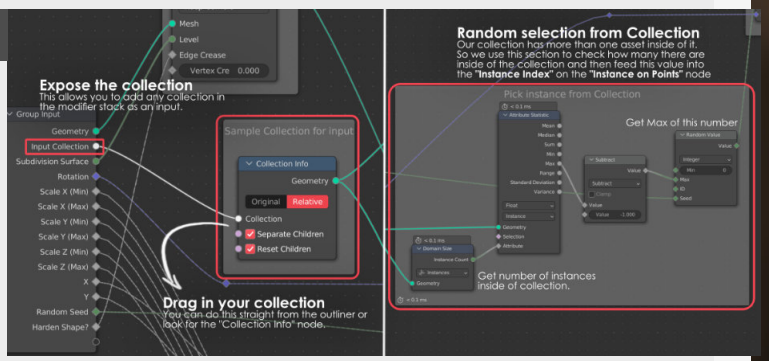


Now that we have a surface that we can work with we can convert this mesh to the points we need to scatter instances on, first we get a **"Mesh to points"** node and then we add an **"Instance on points"** node to do the actual scattering.

For now, the thing we will be scattering on top of this mesh can be anything, and you can just drag it straight from your outliner into the geometry nodes interface and it will add an object info node for you.

We can then hook this up to the **"Instance"** on the **"Instance on points"** node and we have things that are scattered on the object.

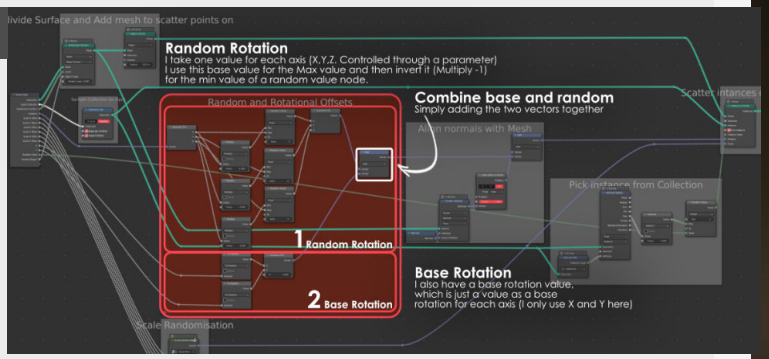
3. Sampling a collection



It's time to get rid of the single object we are sampling and replace it with a collection of objects, this gives us more control over what object we want to spawn and allows us to spawn multiple meshes at once.

To do this we can simply drag in collection like we did before with a single asset and then drag the collection pin to the first group input to expose it to the modifier stack. Now we can switch collections on the fly and try multiple variations pretty quickly.

4. Rotation and scale controls



Now we need to make sure we also align all the instance to the normals of the surface we are trying to control, again, this works for the simple geometry that I have been using so far, but testing on other meshes require some more thought.

My rotational controls work in two layers, first I want to add a base value so i can get the rotation of the roof shingles right first and then add additional randomization on top of it to create some variation.

And the scale works in the same way, where we use a random value between two values defined by us to sample between.

Blender Looptools

Introduction

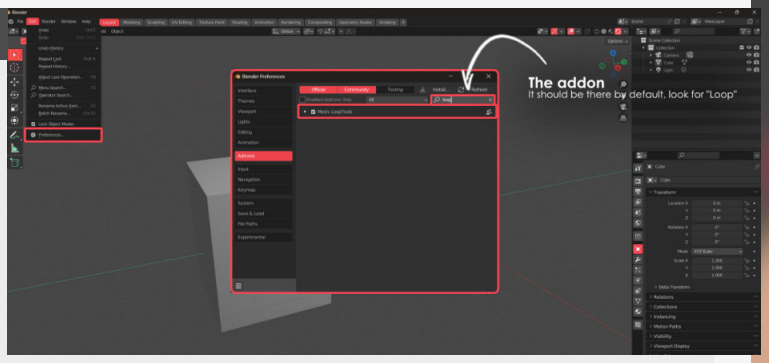
There's a whole bunch of addons that can be used In blender, but today I want to shine a light on one of the tools I regularly use myself, especially for making a selection into a circle again, and it's amazing for doing so.

1. Overview & Installation

Looptools is actually one of the addons that come with Blender itself, so we can just find it inside of the preferences menu and look for "Loop" and it should show up right there.

Once this is installed it can be found in the **"RightClick"** menu when in edit mode on any object.

All the tools also give you control over the percentage of the effect, allowing for precise control.

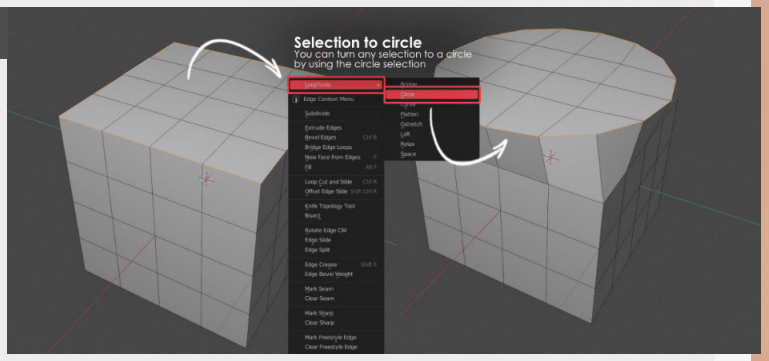


2. Circle

My favorite tool from this selection can turn a selection of vertices into a circle. This is super useful for creating or recreating geometry that you might have accidentally broken.

You can even use it to turn a square selection into a circle to make nice transitions between both shapes.

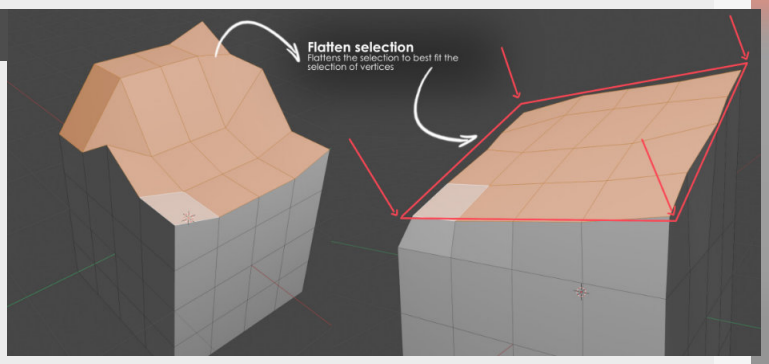
No longer you need to go in and try to fix things manually, because that's always a pain!



3. Relax and Flatten

RELAX: Will relax the selection, handy for smoothening out shapes, however, I still personally prefer the "Edit edge flow" tool that replicates how Maya does it.

FLATTENING: Flattens the selection of faces, with a couple of options of how to flatten all these. Based on normal, best fit or based on view.



4. Other options

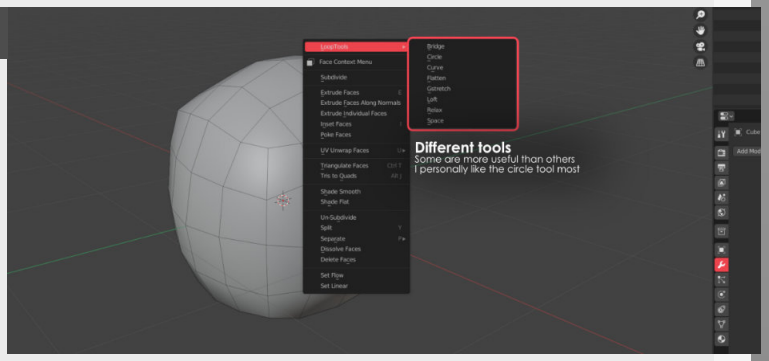
Honestly, I don't tend to use these all too much and they seem pretty specific to certain usecases.

BRIDGE: Connects two faces with a bridge, same as LOFT

CURVE: Seems to soften up the vertices around the curve or edgeloop you've selected.

GSTRETCH: Will move all the edges towards the selected edge, and you have control over the percentage

SPACE: This will space all the vertices along a selection, doesn't even have to be straight for it to work, it will smoothen the selection.



06 BAKING

Baking theory for beginners

Introduction

The first major technical hurdle when it comes to gamedev (well now that I think of it that might be UVing...) But still a technical hurdle by itself. For simplicity's sake, I am going to assume that you are baking inside of Substance Painter, as this will simplify the process a bit, because there are specifics to keep in mind when baking in other programs. But that being said, let's dive straight in!

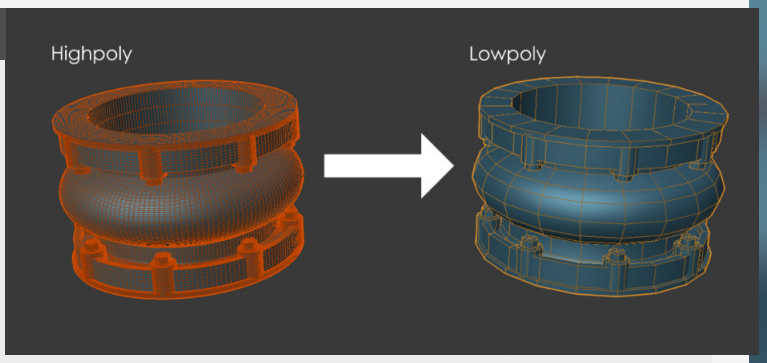
A little thing to mention though, this is a highly condensed artists overview of baking, so add your own research to this!

1. What is baking?

Baking is the act of projecting all the details from a higher resolution mesh that we can't use in game (for performance reasons) to the lower poly game-ready mesh.

For this we need a **Highpoly** mesh (The high resolution mesh) and a Lowpoly mesh (Where we project all the details onto). This **lowpoly** also needs a set of UV's, if this doesn't have UV's then the baking process can't project the details form the highpoly onto the lowpoly mesh and make a texture out of it.

Once we have all these things we can start the baking process, but before we do that, let's dive into some of the theory.



2. Projection or casting rays

Baking works by sampling all the vertices from the lowpoly mesh and cast a number of rays outwards until it hits the limit on the length you set or the cage. It will then bounce back with the information from the highpoly back to the lowpoly.

The result of this bake is dependent on the the flow of the underlying lowpoly, the limit set and the way that the vertex normals are split up (more on that in the next section).

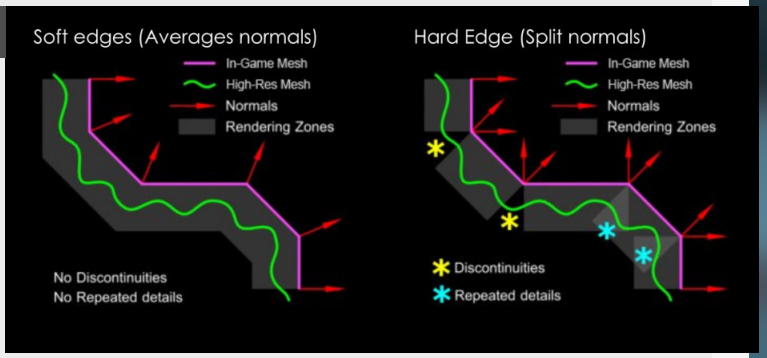
Cages are meshes (Duplicate mesh that encapsulated the highpoly) that dictate when the Rays will be bouncing back inwards when they hit the cage.

3. Hard or soft edges?

Vertex normals are the normals that are attached to individual vertices. The way that the normals coming out from them behave is dependent on the way that you split the edge information.

So in Blender or Maya it would be Sharp or Soft edges, where in Max it would be defining different smoothing groups.

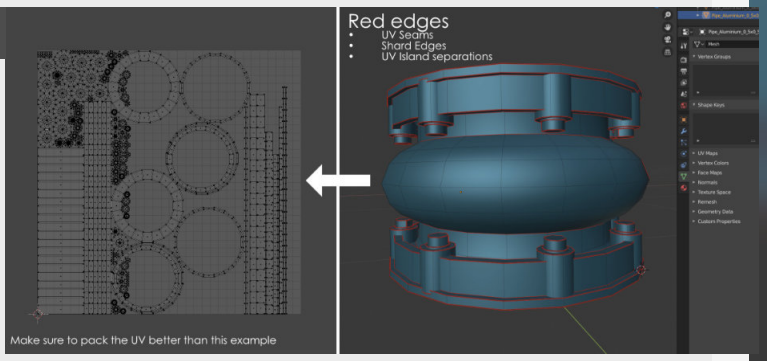
With soft edges it average the normals coming from a vertex between the two polygons it touches. Whereas with hard edge it will split the normals coming from the vertex in two essentially disconnecting the smoothness of the transition.



4. The one rule that matters

All of this sounds super complicated, but the simple reason why this is important is because of this one simple rule. Whenever you have a edge that is at an angle that is close to or higher than 90 you just turn this edge into a Hard edge and split the UV's along that seam and separate them a bit.

This will split up the vertex normals and give the Baking software room to bake this texture between the UV Islands to prevent color bleeding between UV islands.



Baking - Building the Lowpoly

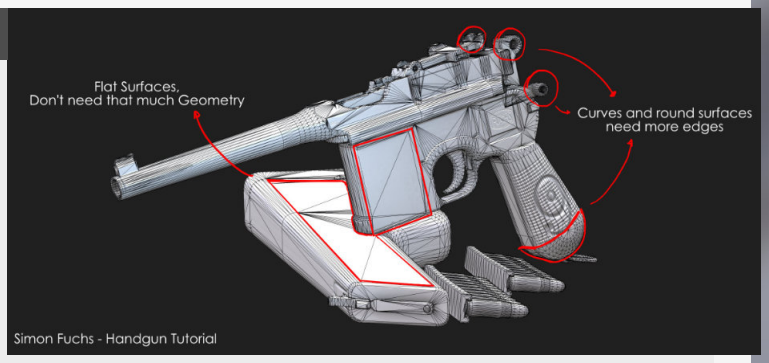
Introduction

This time we will be looking into some tips that will help you build a better lowpoly mesh that can help you get the most out of your lowpoly meshes specifically to get better bakes, let's dive in!

1. Add poly's where they matter

Adding poly's where they matter is important for the whole baking process (and for optimization!) Because baking to a surface that doesn't support the highpoly shape you are going to get artifacts. if you don't have the amount of edges on a complex round object for instance, this might mean that the bake is going to suffer just because of that reason and you will have to apply some tricks to get it right.

Another good thing to keep in mind that removing edge is easier than adding them and where possible try work in multiples of 4 on rounded surfaces so you can safely remove each other edge while keeping an equal number of edges.

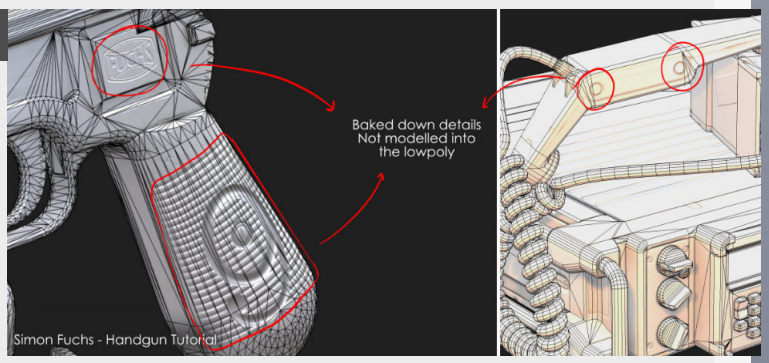


2. Exclude details from the Lowpoly

If you want to build this into your lowpoly mesh make sure that you ask the question if it's really important to have this feature included in the lowpoly.

Maybe there is some other way that you can have this shape, like baking this information down onto the mesh without this shape being present in the lowpoly (flat baking) or simplifying the shape just enough so you don't have to think about building shapes that go into the self for instance.

As an example having some unique text (like a brand or something) on a prop in your scene might be too much to add into the lowpoly.

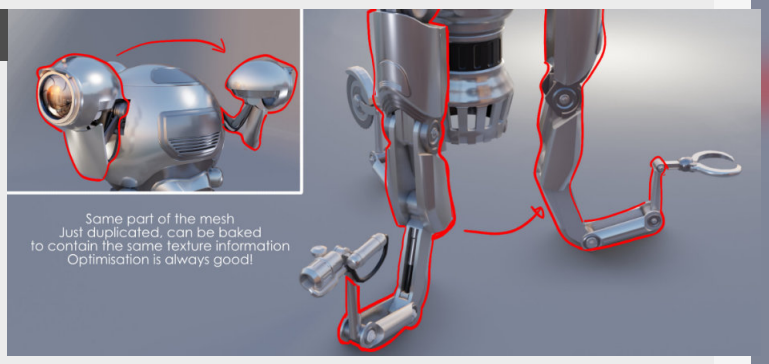


3. Thinking about reuse

Make the most out of your model by reusing as much as possible, you can do this in multiple ways, but the most common one is using symmetry in your mesh where you assume that the other half will be the mirrored version.

Another one could be reusing parts of your mesh that are going to have the same geometry and you don't mind looking the same. This works great for objects that you only want to model once and have all over, like bolts for example.

You are only baking the information down to one of these bolts and if you overlay the UV's if the one you want to bake (Keeping in mind to offset the others on the UV's).



4. Focusing on the silhouette

Becoming more and more important when the polycount goes down or if the style is more exaggerated.

With this I mean it's not only talking about creating a nice silhouette, but also talking about all the things that don't contribute to the silhouette that can be heavily simplified or even cut. Keep in mind that this needs to take the style and polycount budget into account.

Meaning a good balance between the amount of polygons versus too many of them for super smooth surfaces for instance. Try and optimize your polycount and keep that polygons in check!

Baking - UV mapping Tips

Introduction

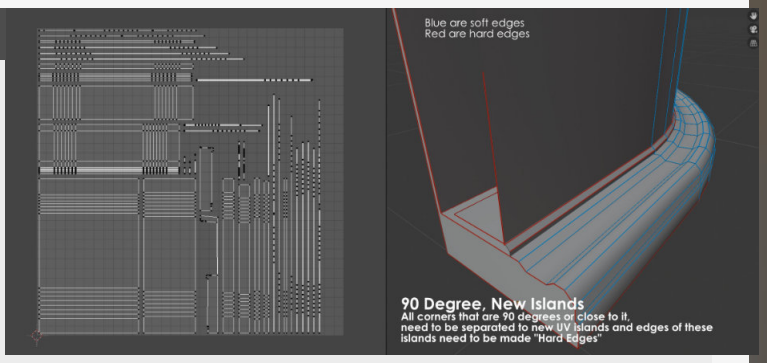
In continuing the baking series let's talk about how you can get the most out of your UV layouts and get the best possible texture bake result. For this I assume you have an understanding of the base concept of unwrapping.

So these tips will be more generalized high level tips for baking specifically.

1. Edges and UV islands

The first solid tip here is about the use of hard/soft edges to define where the cuts in the UV are going to be, the reason for this being is that the the vertex normals change depending on what type of edge (or smoothing group) you assign to it. It will split the Vertex normals if you set this to be a hard edge whereas it will try to smoothen the Vertex normals if you make the edge soft.

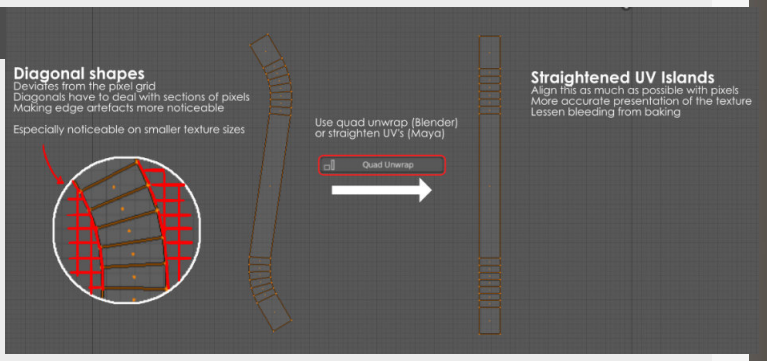
A rule to keep in mind is that every edge on your mesh that is bigger or close to 90 degrees needs to be a hard edge and a cut in the UV's. This will greatly help with baking issues on that part of the mesh. This is probably also the most simple and most useful tips that I can give you.



2. Straightening UV's

You have to make the most of your pixels in your texture space that will eventually be used to texture your model, you will have to try your best to keep away from diagonal UV islands as much as possible especially for smaller textures. You will want to align the UV's as much as you can with the grid of pixels on the texture. If you have diagonals in your UV shells (a lot of the time unavoidable) you might want to think about straightening these out and get a consistent straight edge.

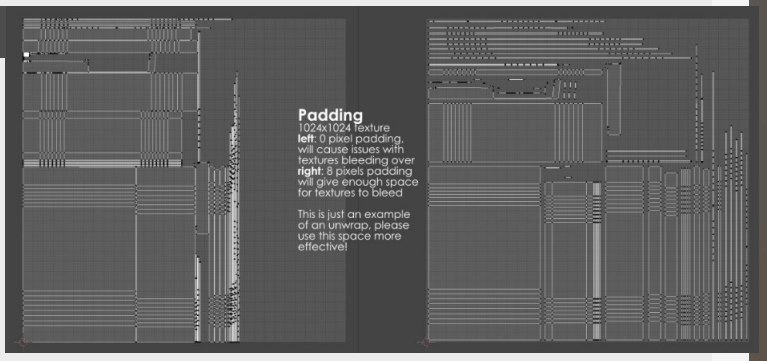
This is also used to get better bake because your edges will not have to deal with diagonal pixels and half pixels. There are some great tools that can help you quad unwrap these sections nice and easy.



3. Edge padding and Dilation

Now there is a caveat to using the most of your UV space and pushing the UV shells as close as possible to each other, this being that you need some edge padding to help bleeding over of the bake. This means that the further it goes away, the edge padding become more visible and bigger, so colors from other parts will bleed over into other parts of they are too close. So to avoid this we introduce a little bit of space between the different UV shells or islands and account for this to happen.

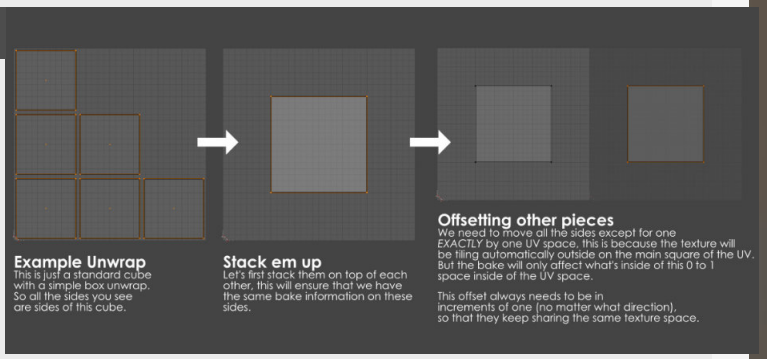
The bigger the texture the bigger the space between the shells. As an example, for a 2k texture this needs to be 8 pixels, for a 1k this needs to be at least 4 pixels.



4. Overlapping UV Separation

For example one way to get more texture space is to use symmetry so we can bake a texture for one half of the mesh and use that same part of the texture on the symmetric part of the mesh, we can also do the same for a reusable piece of the mesh, like for example a bolt. When we do this we need to keep in mind that we need to move one part of the overlapping UV's exactly one space outside of the 0 to 1 space. If you don't the baker will project all the information from all the parts on top of each other, creating a mess.

This is perfect for shareable items like bolts, screws or other items you have that can share texture space.



Baking - Highpoly mesh tips

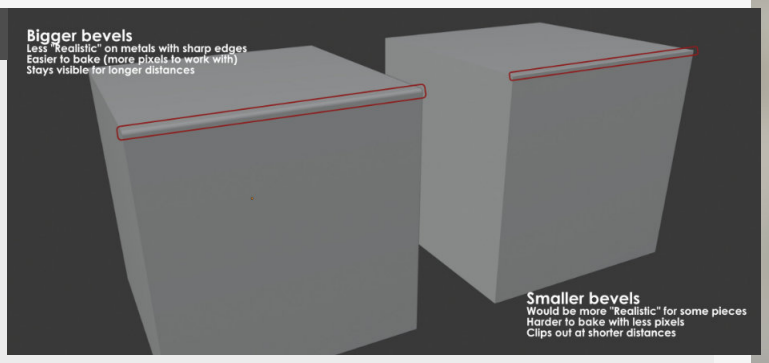
Introduction

Let's dive into some tips for building a highpoly mesh specifically to get better baking results. For this we are looking into how to add thing like bevels, details, using floating geometry and making multiple highpoly meshes instead of one connected mesh.

1. Large Bevels

Bevels are the best way to highlight those nice edges that you get when lights hit the mesh, so making them bigger will make the highlights hit those edges of the mesh for a longer distance through all the MIP maps. So if you are going to bake all this valuable information down to a normal map, you might as well get the most out of it and retain them for as long as possible.

Creating a consistent bevel width for similar parts of the mesh is also something that can help you create consistency between them. So for instance when modeling a gun, all the metal elements have tighter bevels than other parts of the gun like the grip that are made out of softer materials.

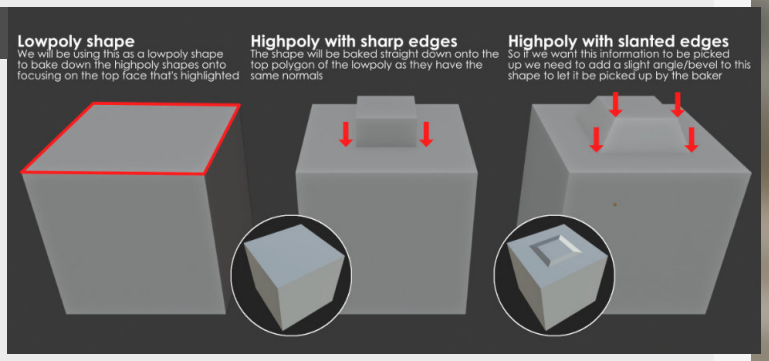


2. Details and sharp edges

When thinking about adding surface details that will be baked straight down onto a surface there are some things to keep in mind. The most important thing building details like this is that you can't use a 90 degree angle for surfaces that come straight (or follow the vertex normal) up from the surface without either giving them a bevel in the highpoly or dedicated lowpoly mesh space.

The reason why to avoid this is because it will never capture the information, like seen in the example on the right.

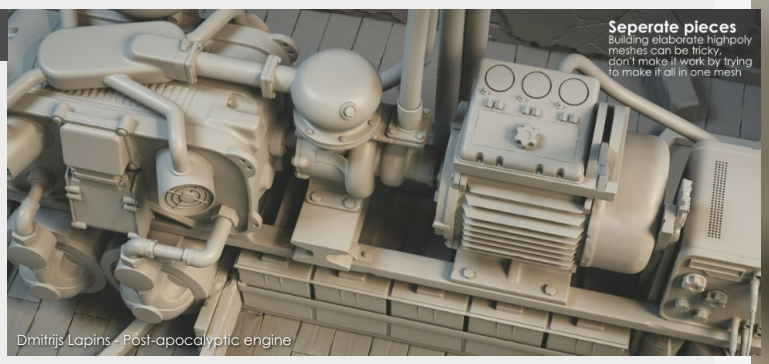
Also think about what kind of details you want to add, it's not always necessary to build meshes, you can use floaters or add them in substance painter.



3. Seperate parts

This one might sound ridiculous for some, but a highpoly doesn't need to be one mesh. Building separate pieces of the highpoly mesh is a huge time saver, especially because some of the connections you are trying to force when keeping it one mesh are unnatural anyways.

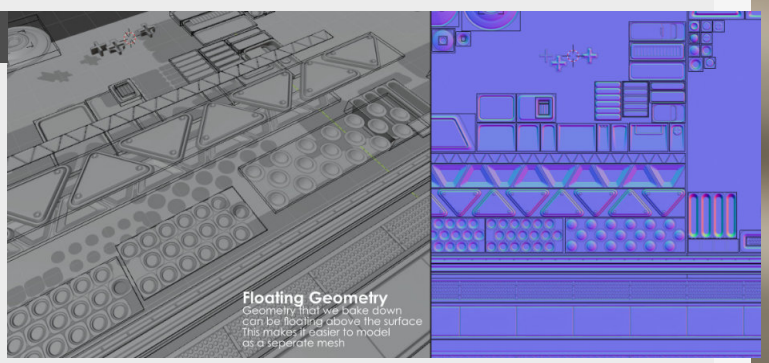
A good way to think about this is thinking about how it was constructed and separating these elements into separate parts of the bigger mesh. Like for example if we think about a table, it has the tabletop, the supporting frame and the legs. Separating the highpoly will make it way easier then doing this all at once.



4. Floating geometry

These are smaller meshes that float above the highpoly surface and then get baked onto the lowpoly together with the base highpoly mesh. Some things need to be kept in mind. First of all is that you need to add these as close to the surface as possible so it can be factored into the baking process, secondly there are some limitations to this because you need to factor in the blending between the edges of the floating piece of geometry and the surface projecting it onto,if these don't match perfectly, you will get normal map artifacts.

This is also why people stay away from trying to use them on curved surfaces unless you have a plugin that matches them to it.



Texture baking - part 1

Introduction

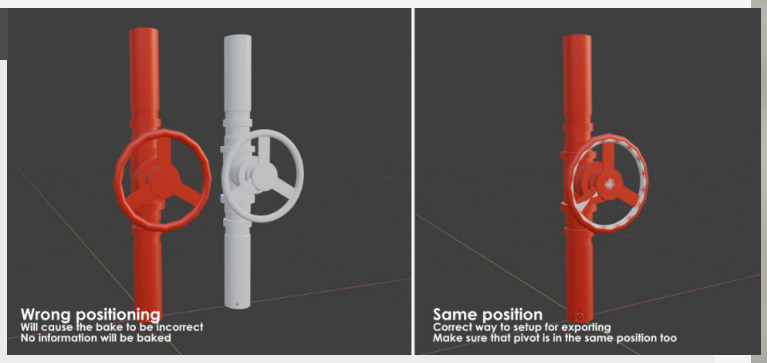
Let's dive into some tips for building a highpoly mesh specifically to get better baking results. For this we are looking into how to add thing like bevels, details, using floating geometry and making multiple highpoly meshes instead of one connected mesh.

1. Aligning all the pieces

Let's start out with some of the basics for starting our first bake and that is aligning all the pieces with each other, they don't need to be at your scene origin point but they need to be aligned.

Also if you are baking without a cage, make sure that all the pieces you are trying to bake don't project onto each other, this means that a piece of your mesh can't be on the same position on ANY axis, or else it will project details from different pieces on top of each other.

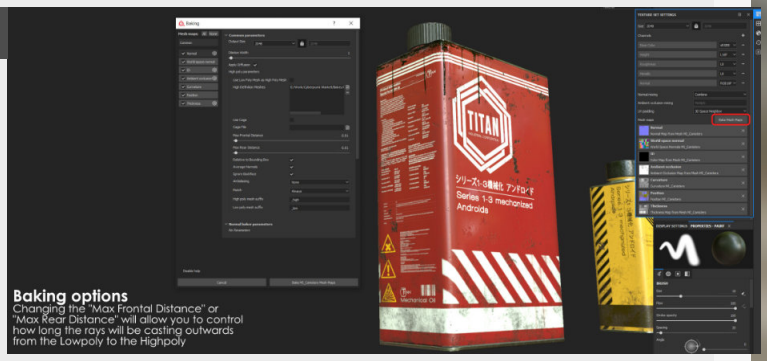
To avoid this really think about either using a cage or baking by mesh name (more on that later)



2. Bake Distance

First one being bake distances, these are dependent on the deviation between the lowpoly and highpoly mesh, so the more your highpoly sticks out the higher the frontal distance will be, same goes for the opposite direction for the rear distance. This replaces what you would normally do with a cage, however a cage gives you a preview where this does not.

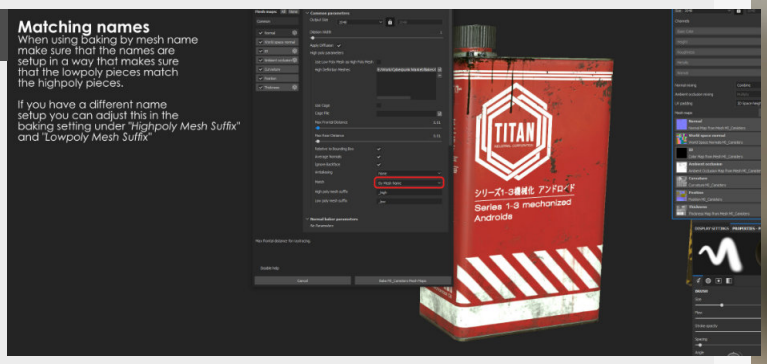
An additional option is to check/uncheck the "relative to bounding box", if we have this checked then the bounding box will be normalized to a 0-1 distance according to the outer dimensions of the loaded asset. Testing this is easy though, you can just pick a value and see if details are missing.



3. Baking by mesh name

Probably the best thing you can do when baking with many pieces is baking by mesh name, this will allow you to not use the explosion or separating them manually method for splitting up your individual pieces so they don't interfere with each other.

The way you need to set these up is give all your highpoly pieces (if you have multiple) a name and give the accompanying lowpoly the same name but replace the hi with low. For example Assetname_Low will match with Assetname_High, keep in mind that there are case sensitive, so make sure you are keeping these things in check.

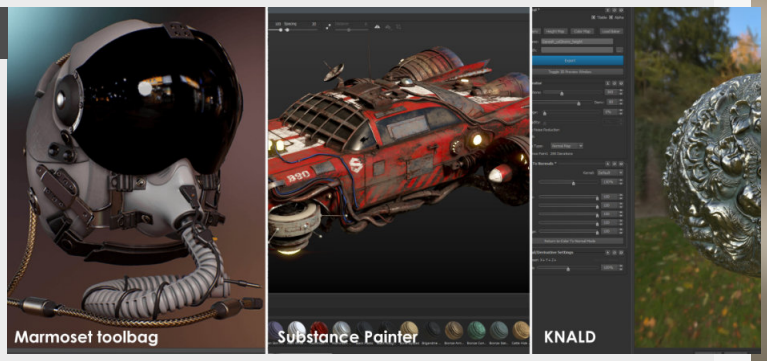


4. Picking your baker

Most of the things explained here are kind of specific for substance painter, however there are other options for baking software too, some of them are really nice actually.

I personally used Knald for a while which has an awesome 3D preview of your meshes so you can easily debug them.

Lastly if you are really keen on having as much options as possible for your modelling package then baking inside of Blender, Maya or Max is also an option too, the reason why substance painter is such a good option is because it integrates with the texture program itself really well, allowing you to start texturing really quick.



Texture baking - part 2

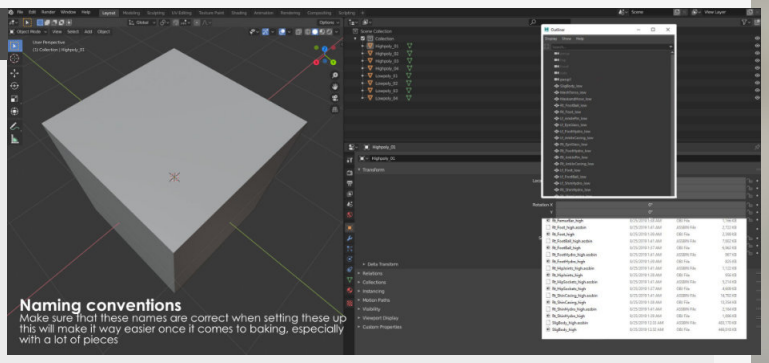
Introduction

In this second part we will be diving into more generic tips about baking. We will be diving into some more common issues that can occur when working on baking your own props.

1. Name your exports properly

This might seem a little on the nose, but naming your exports properly is crucial when moving forward into the baking process. This is good just from an organization perspective, because imagine snapping through over a ton of different assets that have multiple multiple parts and are complicated.

This will also help with using some of the more advanced features that you will be using going forward, especially in point 4 where we talk about baking by mesh name.

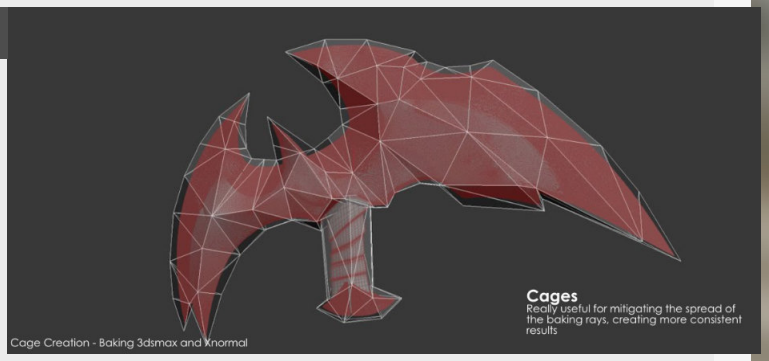


2. Creating a cage

So if you aren't using baking by mesh name or want really specific results we can make a cage for our bake setup which will limit the range of the bake between the lowpoly and the cage.

Which is also why in most cases and with most bakers the cage cannot deviate from the lowpoly when creating the cage as it will compare the vertices from the lowpoly to the cage.

So when you create a cage by duplicating your lowpoly mesh and then exploding it, all you are looking for is to encapsulate your highpoly.

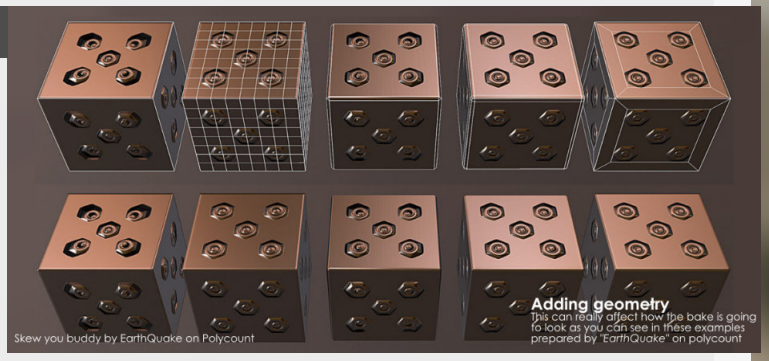


3. Skew maps

These are black and white maps that alter the way that the bake is going to transfer the details from the highpoly to the lowpoly. It will literally skew the details until they are baked flat onto the mesh, making it perfect for floaters and smaller details like nuts and bolts.

The white of the texture will make it project flat onto the mesh whereas the black will leave the area unchanged. This can easily be flipped by toggling the option in the baker (in substance)

However this can also be achieved in add more geometry on a temporary basis to affect the vertex normals too.



4. Debugging errors

There are some specific things to watch out for when baking, because you should never hope for a perfect bake, especially not the first time. Arguable this is the aspect of baking that you will spend most of your time in, because there are a lot of small little errors that can pop up during the baking process.

So getting used to how to solve these will be the most important thing, but this will take time and experience. Aligning your meshes or something being wrong with the transform is just one example so hopefully we will dive into more concrete examples as we go along.

Baking troubleshooting - 1

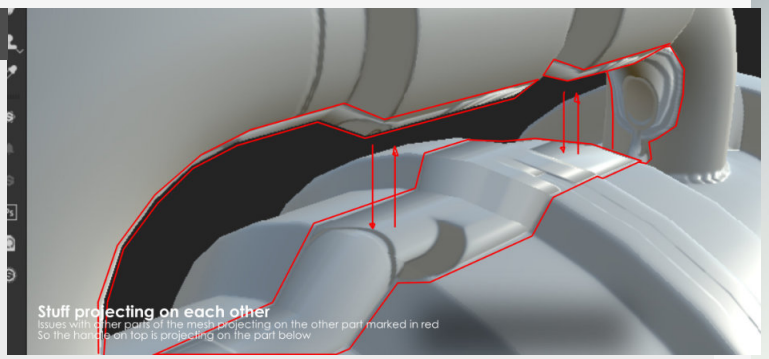
As you might have already noticed, a lot of the time baking is just problem solving with a healthy dose of thinking ahead, so in this entry we will look at some of the common issues with baking and how we can resolve them.

1. Projections from other pieces

When you notice that other parts of your mesh are being projected onto the mesh you are trying to bake currently this might be because you don't have a cage for this mesh or have set a wrong baking distance.

So adding a cage for this mesh will definitely work if it is one big mesh. If the assets consists of multiple smaller parts you can also use baking by mesh name to get rid of this issue.

More on that topic can be found in Blog **#84**.

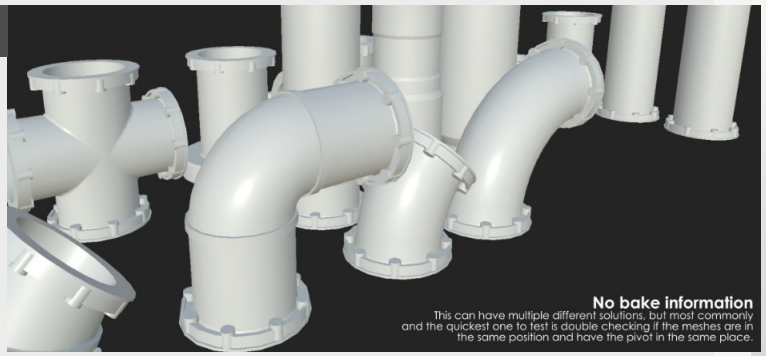


2. No bake information at all

When there is no information for the bake at all the main cause for this is that the Lowpoly and the Highpoly mesh are not in the same place, so double-check your pivots and locations for these meshes.

Clearing the location (or apply transforms in Maya) can help you get rid of weird transform issues.

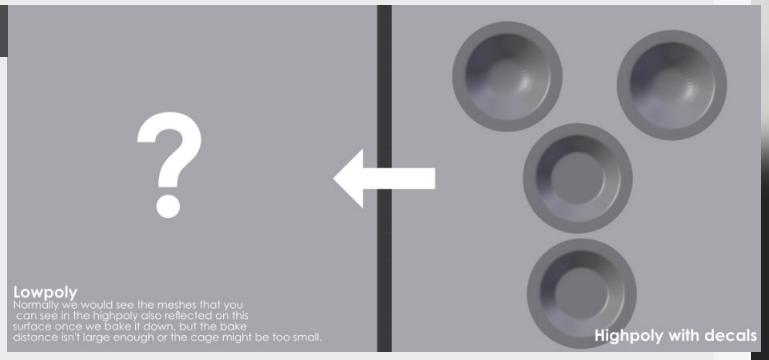
So make sure to check your transforms, because if the pivots are on the same spot these transforms should be the same too.



3. Floating decals not showing

You have floating decals that won't bake? That might be because the distance of the bake hasn't been set correctly or the cage doesn't encompasses these details.

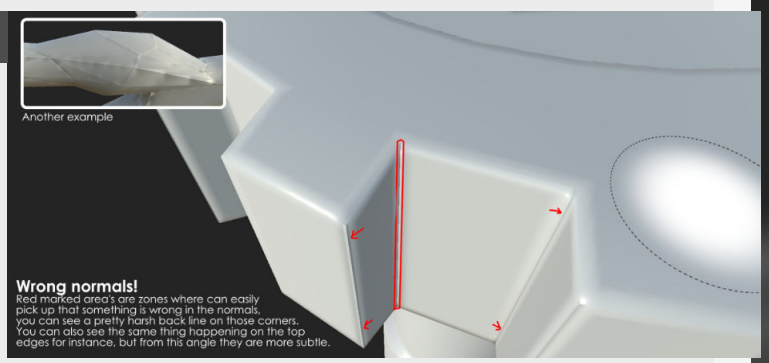
Also be sure to add the floating decals to the right parts of the highpoly when baking with different mesh groups.



4. Split or wrong shading on edges

If you see edges that are not smoothly flowing into each other like you would expect, for example there is this harsh line in the middle of these edges then this might be because you have the wrong edge type for example a hard edge Maya, different smoothing groups in Max or Sharpe she's in Blender

Doing this on edges that aren't split in the UV's will cause you to bake he wrong information down onto these edges, more information on this topic can be found in the earlier blog post about baking Blog **#92**.



Baking - Troubleshooting - 2

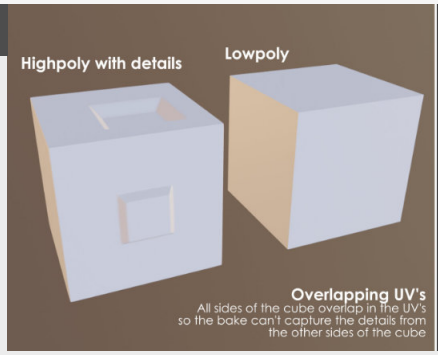
BeyondExtent

As you might have already noticed, a lot of the time baking is just problem solving with a healthy dose of thinking ahead, so in this entry we will look at some of the common issues with baking and how we can resolve them.

1. Multiple surfaces projection

If you have this weird issue where multiple faces are projecting onto the same UV shell and it looks all weird then this is because you probably forgot to move overlapping pieces by one UV space.

Not doing this will basically try to bake the information onto the same space twice from different surfaces on the model, where we only want this once as the UV will take care of the other side.

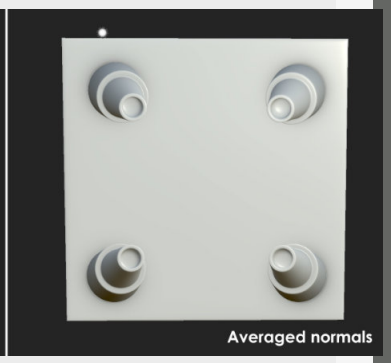


2. Skewed details

When you have tiny details like bolts for example on the highpoly and you want to bake these down you can get this weird stretched look on them, this is because the normals leading up to that point are not pointing straight down.

The traditional way doing one bake with a cage or non-average normals (for all these details to be projected flat onto the surface) and a bake without a cage or average normals (for all the edges etc) then combine these.

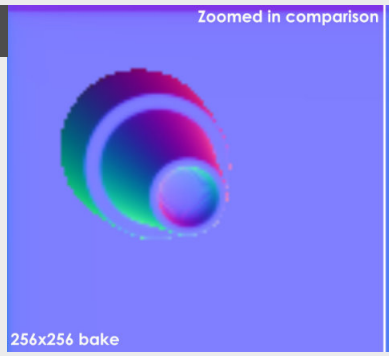
Now you don't have to play around with that anymore and you can just use marmoset with a build in feature that allows you to paint these areas.

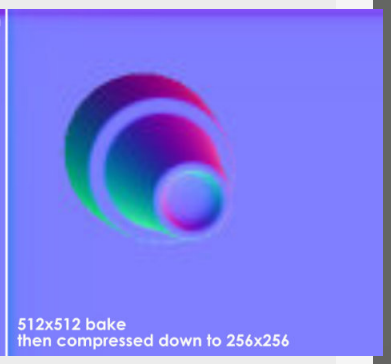


3. Crisper detail workaround

This is a really interesting trick to smoothen out some of the harder transitions that can really help some specific bakes that rely on more detail such as weapons or highly detailed pieces while also working within more strict technical restrictions.

The thing we do in this little step is baking the normal map on a higher resolution and then down-scaling this in an image editor, this will then compress and interpolate these transitions in the normal map differently compared to the raw bake and smoothen them out a bit. Might be a bit specific, but it can be useful in certain aspects.



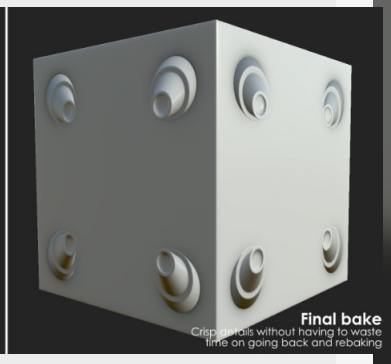


4. Bake taking really long

When your bake is taking really long there are a couple of things you can do to optimize the time that this takes, first one is optimizing your highpoly by reducing the amount of polies.

Another one is checking your baking options, if you are just doing initial test bakes, try and bake on a lower resolution without anti aliasing on, this will significantly improve baking times especially on AO.

Once you are done with cleaning up errors hen you can go and bake full resolution.



Substance Painter Opacity Map Bake

Introduction

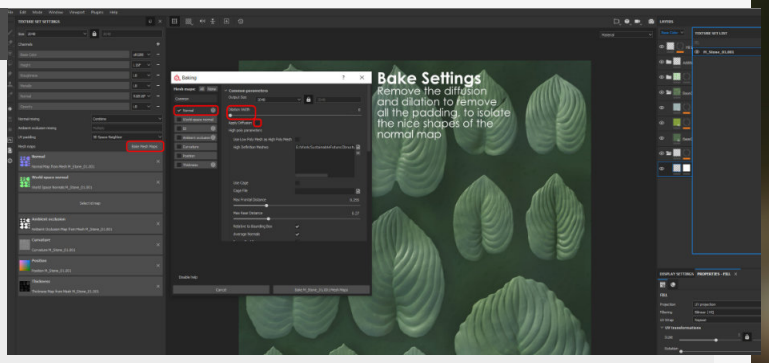
This entry will dive into a workaround way to get opacity maps by baking a normal map that has no dilation and then using a 2D editor to turn this into an opacity map.

1. Baking the texture

Let's dive straight into the baker for this one, as I assume you already have the meshes for it.

For this little workaround we are going to use this baker to bake a normal map and get an opacity map out of it. So disable all the other texture maps in the baker and remove the diffusion and remove all the dilation. This will cause the padding caused from the baking process to be completely removed.

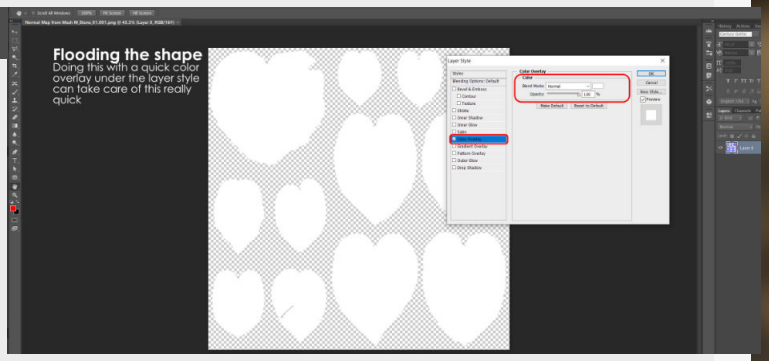
Now that we have this texture (found in the "Shelf" under "Project") we can bring it into photoshop or other 2D software where we go for the next step.



2. Flooding the shape

The next step is super simple, the thing we want to do here is just fill the shape that we get from baking it in Substance Painter with a white color, making it visible for the renderer and removing all the black parts from being visible.

For this we can use a simple color overlay inside of photoshop that is set to white and the style to normal, we don't need to do anything to the background of this image as this will be recognized as black because it doesn't exist anyway.



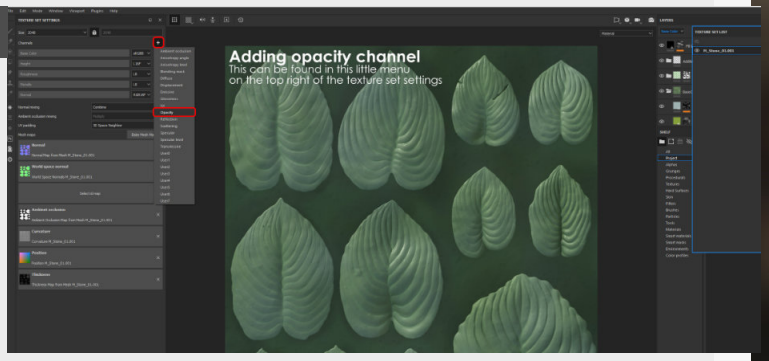
3. Putting it back in the channel

I mean, you must know how to do this if you got this far!

The other thing we need to do is add an opacity channel in the "Texture Set Settings", we can find this under the little plus symbol on the top right of this window.

Now that we have the channel setup properly we can add this to a fill layer inside of the "Opacity Layer" or disable all the other texture channels inside of the properties of the fill itself.

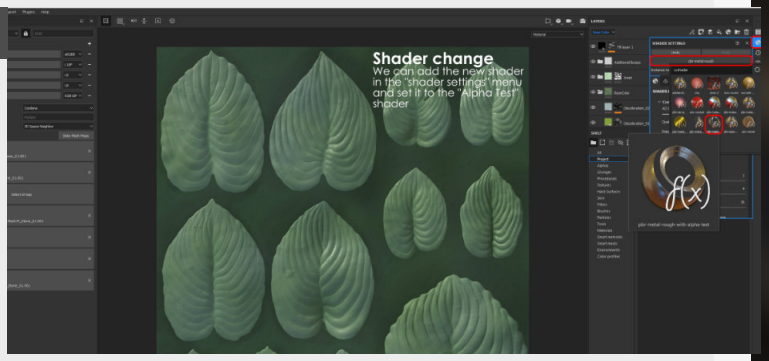
Now that we have this you might see that nothing really has changed at this point, so we need to do one more thing!



4. Changing the shader

Lastly, we need to change the shader from the standard shader used in Substance Painter to the "pbr-metal-roughness-with-alpha-test" for a black and white opacity map.

Now we can see the opacity map in action!
Hopefully this was helpful to all of you.



Baking - Floating geometry

BeyondExtent

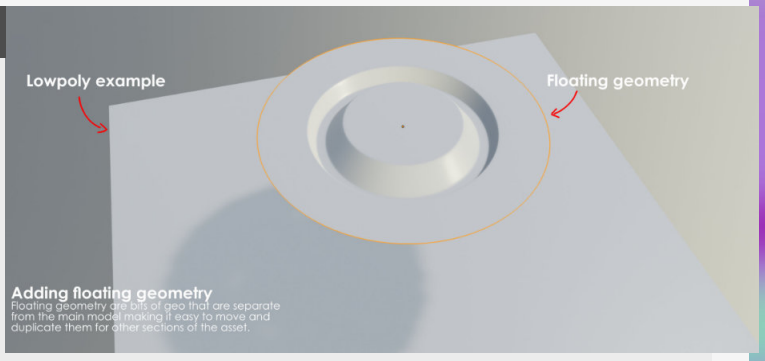
Floaters are pieces of a model that are detached from the main model and float above the surface. This way we can avoid having to model them directly into the main highpoly.

1. Intro to floaters

Floaters or floating geometry is great to add details for the highpoly, these pieces of geometry are detached from the main highpoly mesh and are thus not dependent on the polyflow of the underlying mesh either.

Another benefit of this is that we can easily duplicate this to another part of the asset to realize them, think nuts and bolts as good examples.

As we have them floating above the surface we can easily test if this is the look that we are going for or not and remove them if we decide against them and keeping the highpoly intact.

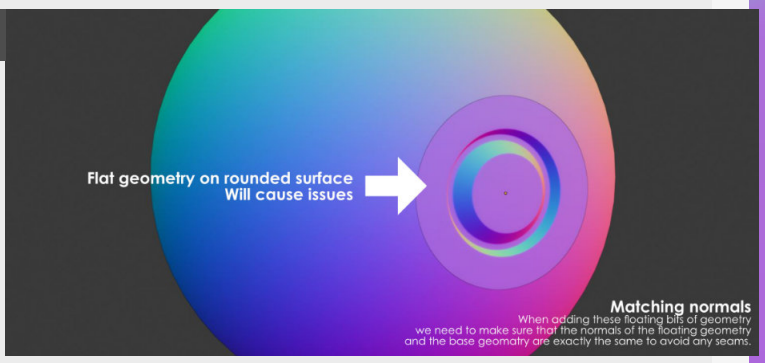


2. Keeping in mind normals

Floating geometry you grab or make will need to have an area where you account for the spilling over between the meshes, this area will serve as the blend between the normals of your floating geometry and the normal of the underlying highpoly.

A big thing that we need to keep in mind is that the normals of the floater, mostly the edges need to correspond with the normals of the underlying mesh normals. If they don't then you are going to notice this in the baked down normals and texture you are going to create using this information.

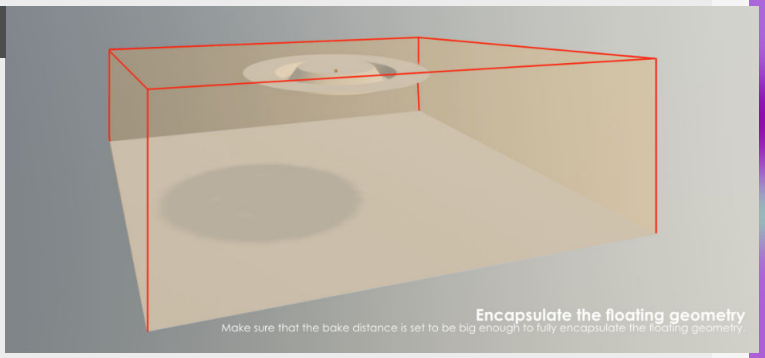
Flat surfaces are perfect for this.



3. Baking these down

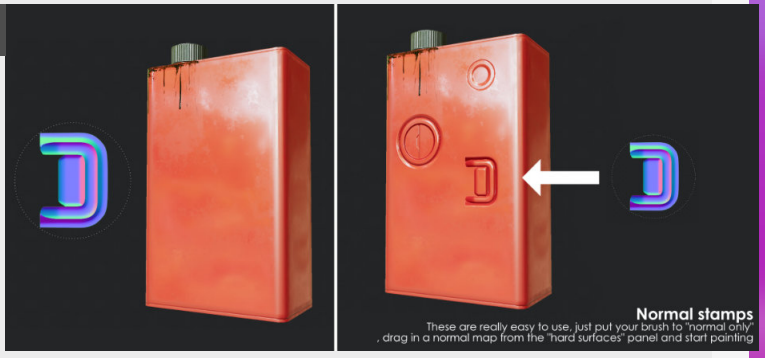
Now that we have these detailed floating geometry, it's time to bake them down. But what we need keep in mind with these details is that we need to make the cage or the baking distance big enough so that it fully encapsulates the details fully.

Now with this floating geometry we need to keep in mind that there are going to be issues with the ambient occlusion bake if we don't separate the floating decals from the normal geometry.



4. Painting floaters

Additionally, and way quicker is that we can also add these later on, after we are done baking. We can do this through substance painter by adding them as normal stamps and if we then combine this with the anchor points that we can use.



Baking by “Mesh Name”

Introduction

So with this one we will dive into some the topics of building and designing your own portfolio and how this will help you with the next job and how to grow towards what you want to achieve with your career.

1. Why do we need to do this?

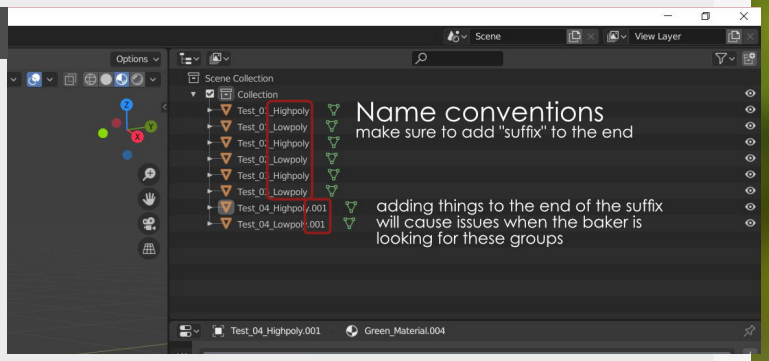
When baking textures it will project the highpoly detail onto the lowpoly mesh, when doing this if you have complex meshes or meshes composed out of multiple different meshes we need to make sure that these different meshes don't project onto each other. Traditionally this was done by "Exploding" these meshes, this will then offset all the different elements of the high and low poly and make sure that other assets don't project onto each other.

This is where Baking by mesh name comes into play, this will bake all these meshes in separate instances and then combine them together, making it super easy for you.

2. Correct mesh names

So for this to work, we need to make sure that we have setup all the meshes with the correct naming conventions. So to do this we need to give all the meshes that belong to the same group the same name, so as and example let's say we name them **"Test"**, then we need to add a suffix for both the Lowpoly and Highpoly. so for our example the lowpoly will be named **"Test_Lowpoly"** and the Highpoly will be named **"Test_Highpoly"**. Once we start baking this will then look at the name and bake them as a group.

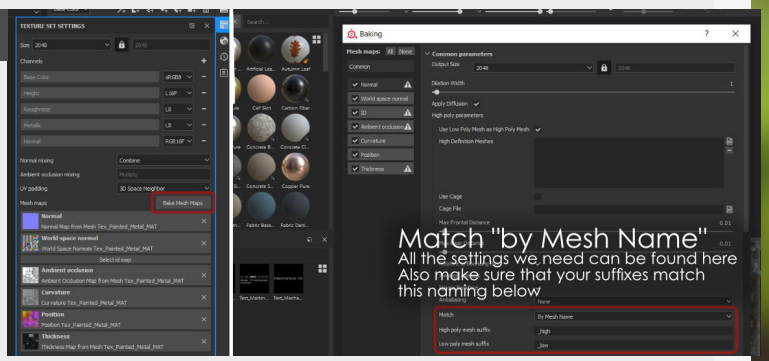
Also important is that the meshes need to keep the same offset in the scene itself so that they remain overlapping if we export them



3. Baking setup

So for this to work, we need to make sure that we have setup all the meshes with the correct naming conventions. So to do this we need to give all the meshes that belong to the same group the same name, so as and example let's say we name them **"Test"**, then we need to add a suffix for both the Lowpoly and Highpoly. so for our example the lowpoly will be named **"Test_Lowpoly"** and the Highpoly will be named **"Test_Highpoly"**. Once we start baking this will then look at the name and bake them as a group.

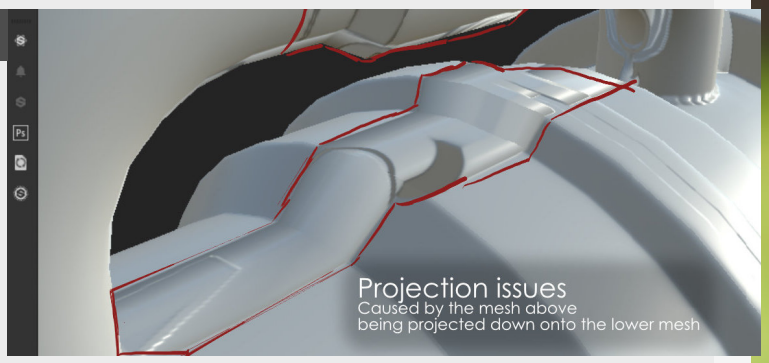
Also important is that the meshes need to keep the same offset in the scene itself so that they remain overlapping if we export them



4. Test baking and troubleshooting

Now once we start baking these maps we don't want to start baking the full textures at maximum resolution yet. So let's start out with doing a test bake on a smaller resolution like 512px. doing this allows us to test our setting and get some initial results.

It will also allow us to see if we have some projection issues, because if we do we need to see if we have setup our naming conventions and our meshes correctly.



07 TEXTURING

Texture maps explained - 1

Introduction

Now that we discussed Baking it's time to dive into the next step and start talking about Texturing, but first we need to take a look at the different maps that are used in the texturing process, so here we go, let's dive in!

1. Normal Map

Normal maps are textures that hold texture information that bounce the light based on the value that is in the texture itself, this is also why this texture has a weird colour.

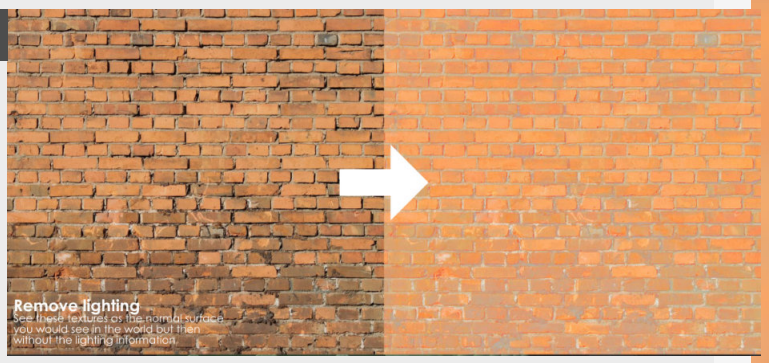
The colours in the map reflect the direction that the bouncing light rays will get adjusted by.

This map is super important in the challenge of adding more details to a model without adding more polygons and most commonly acquired by baking this information down from a much higher detailed mesh down to a lower detailed mesh.

2. Basecolor/Diffuse/Albedo

These are the maps that are used for storing the base color values for the material itself, so why these different names then? There are a couple of different workflows that are used by engines, so to make it simple Basecolor is used in the metalness workflow (eg Unreal Engine) whereas Diffuse is used in the Specular Workflow (eg Cryengine).

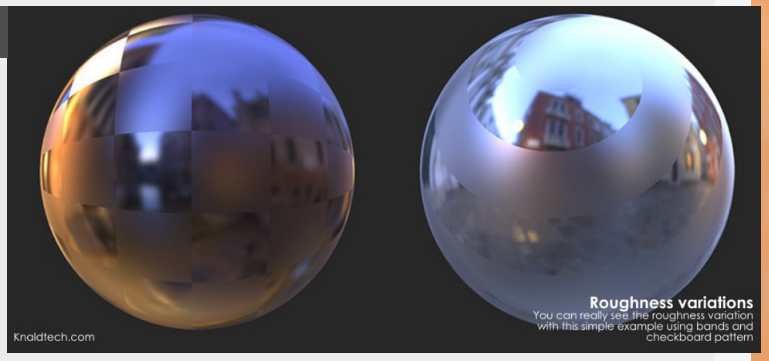
Then for Albedo, this is mostly used in the workflow that is no longer the PBR standard and will be used outside of PBR rendering such as phong or blinn shading for example in modelling packages. These textures are best seen as what you would normally see in daily life with the lighting information removed as seen in the example on the right.



3. Roughness/Gloss

These maps are the interchangeable they are just the inverse of each other. Where Roughness is used in the Metalness workflow and the Gloss Workflow. These have a big impact on how the material will react to the light bounding off of it.

You can think of these maps as the micro surface of the material, meaning that the smoother the surface is the less altered the light bounces are going to be and the more smooth and reflective the light is going to become because of this

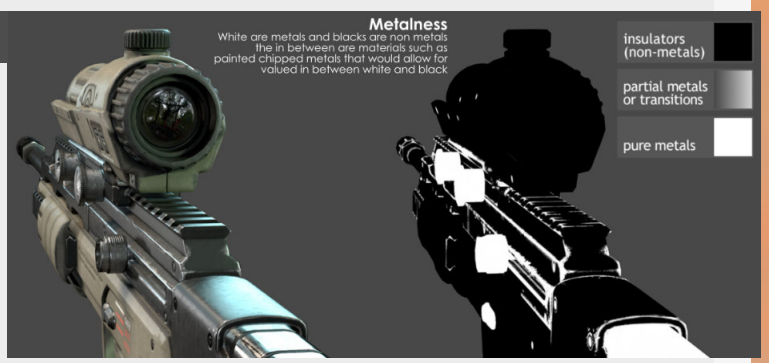


4. Metalness

This map is more specific to the Metalness workflow, I mean it's in the name so it has to be right?

This map controls if the material is a Metal (and has colored reflectivity) or a Non-Metal (and the reflection is going to be white).

So for most (and if you want to be accurate) this map is either black or white, so when the map is white it's metal and when it's black its going to be a non-metal, simple as that!



Texture maps explained - 2

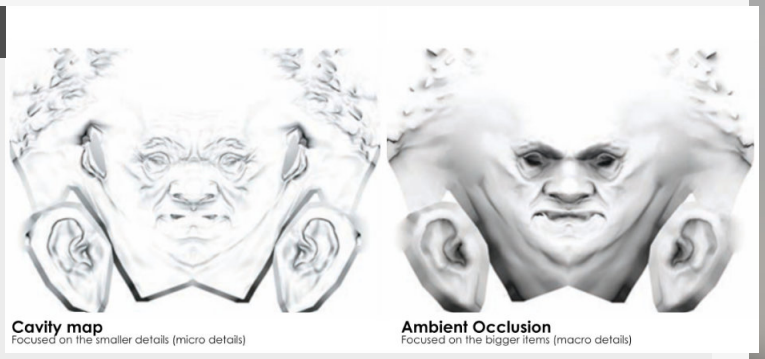
Introduction

Let's take another look at some of the additional maps that are commonly used in texturing, here we go for round number two!

1. Ambient occlusion (AO) and Cavity

First up the ambient occlusion texture, this texture will simulates the shadows cause by surfaces or objects blocking out the ambient light that is bounding around. So to simplify this the deeper the surface is between surrounding surfaces the more shadow will be introduced inside of this cavity.

Now that I mention that word don't confuse this with cavity map, this is basically an ambient occlusion map that is purely focused on the smaller details and will this be calculated into the shader in a different way and will thus affect your meshes differently, so think of it as an AO map that is baked at a shorter distance and has more detail.



2. Opacity

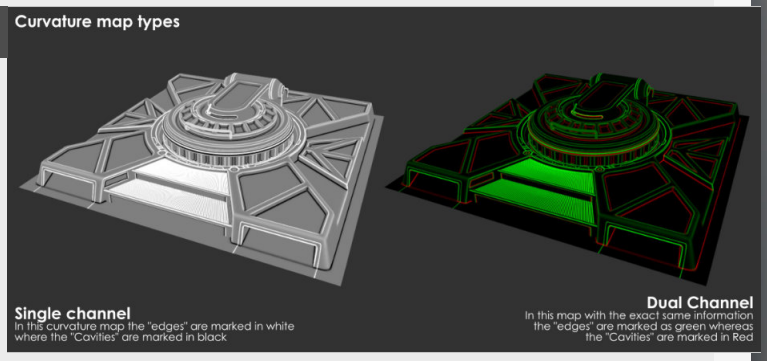
The opacity map is where the magic happens if you're looking to make stuff disappear, it's a super simple one to understand, white on the texture will be visible and black will be invisible.

All the values in between will control the opacity of the material unless you are using a specific shader such as Alpha Clip which can only read black and white values whereas Alpha Blend will allow you to blend to in between values for the opacity.

3. Curvature

A curvature map stores all the information of the mesh where has peaks, valleys and pits. You're probably familiar with this map if you ever baked something in Substance Painter or possibly some other programs that use this to generate masks and edgewear.

A curvature map can be depicted in multiple ways as you can see on the right. The most common one is the single channel one where the edges are represented as white and the cavities are marked in black. You can also see that is uses the full gradient with 50% grey being neutral.



4. Height map

Height map is another map that is fairly common for us Environment Artists especially when it comes to terrain and other forms of depth simulation (for instance in materials). This texture allows you to paint (or author this in any other software and then bake this information down) in black and white where you want your mountains and your valleys to be.

Black will be the lowest point in your terrain and white will be the highest point. But this isn't only useful for Terrain, other programs like Substance Designer, Unreal Engine and others use this texture to simulate depth in materials by using this texture for example for parallax occlusion mapped textures.

Intro to tileable textures

Introduction

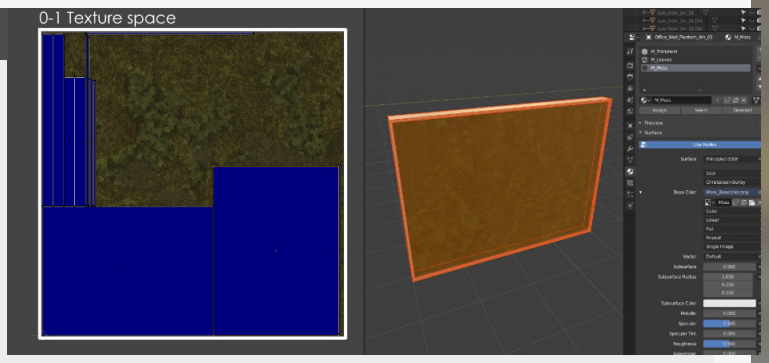
A thing that a lot of artist take for granted at this point but if you are a beginner it might be weird to understand the concept of textures that seem to runs endlessly.

1. UV Spacing

First to understand the concept of tileable textures we need to talk a bit about UV's. Normally when we unwrap stuff we keep it in the 0-1 space of the UV's, there's a good reason for that because all the polygons we move outside the 0-1 space will have the texture repeating onto the mesh, but this can play in our favor!

We want these textures to repeat, this gives us more pixels to work with on the texture.

For example, you have a simple plane unwrapped to the full UV space, say 512x512 but if we double the size of the UV's (Scale it up) This means that the texture is now repeating 4 times on this space while the texture remains 512.



2. 1-way Tileable

These are textures that tile in one way, so this means that if anything pokes outside of the 0-1 space of the UV's to one of the sides (horizontal or vertical) it will tile. When we are talking about these sorts of tileable textures people also call them trim sheets.

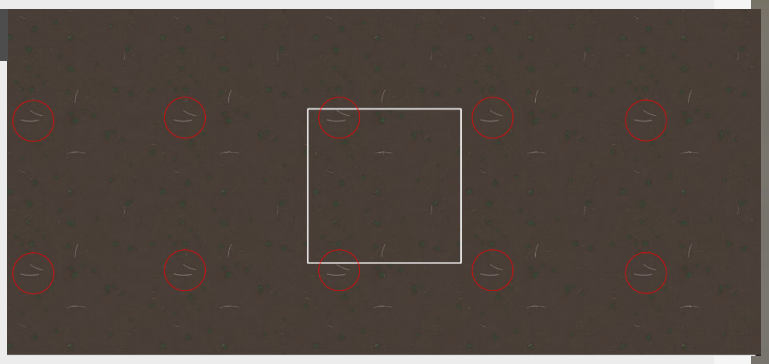
The reason why people call them trim sheets is because you can have many different strips or trims of textures on one texture sheet, as seen in the example on the right.

This topic onto itself is quite worthy of more explanation, so these are the basics for now!

3. 2-way Tileable

The standard form of tileable textures. This means that the texture will tile in all directions and that you can move your uv's around outside the UV space and never destroy the tileable nature of the texture. A downside to this is that the texture itself needs to tile well, this means that if you have any unique spots in the textures this will become super obvious when you tile the texture multiple times.

You can see an example of this to the right, where you can clearly see the tileable nature of the texture because it has distinct patterns.

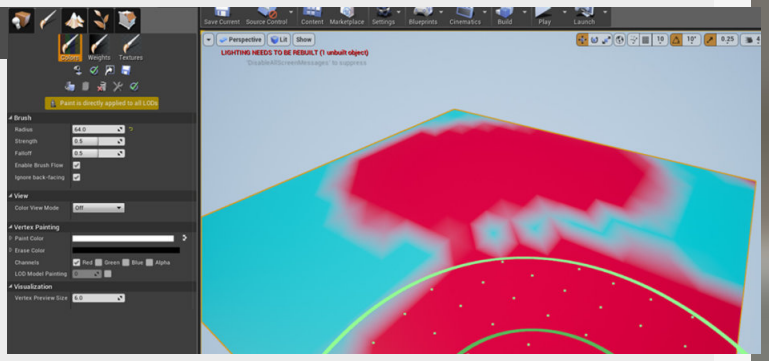


4. Blending it up

The power of tileable textures lies within how you use them though, for example what you could do or your personal projects is buildup a personal library of tileable materials and then blend them two or more of them with masks to get some interesting results.

You can do this in any program you want by using masks, vertexpaint and other things...

Vertexpaint tips can be found in my blog compilation and in one of my previous blog posts!



Making tileable textures - Basics

Introduction

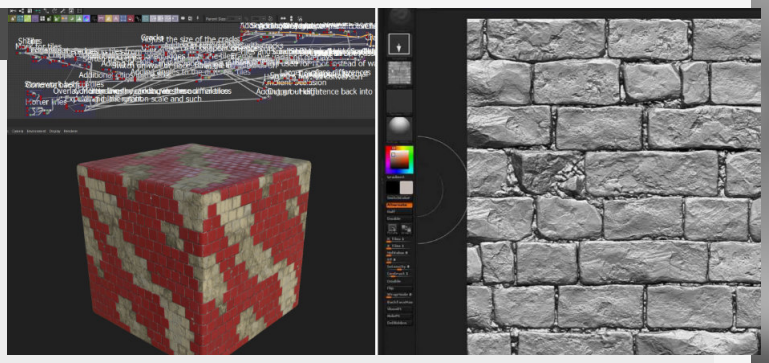
In this one we will have a look at tileable texture creation, this is more of a general overview on how to create them and not really specific about which program to use. In the future we will be looking at program specific creation methods, but this one should give you a good idea about how to start and what to keep in mind when thinking about creating Tileable textures.

1. Different workflows

When starting making tileable textures, there are multiple different ways that we can go about this.

The most common way is using Substance Designer to create these but there are other options too. You can also create these with Zbrush or any other modeling program, there are some neat tricks inside of Zbrush to help you create these, like making the viewport like things for you (flipped normals on YouTube has a good video about this).

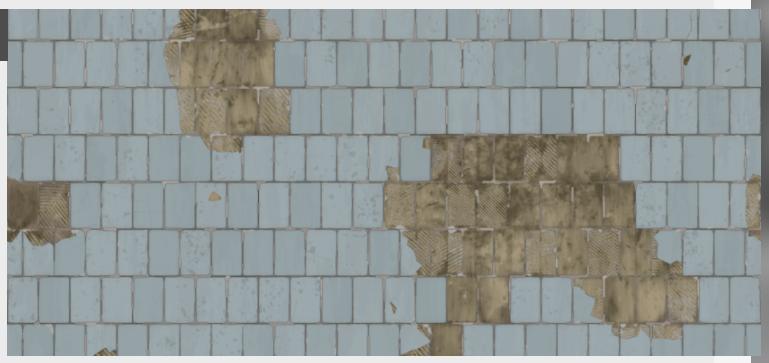
The main difference to keep in mind here is that Substance Designer might be more difficult to get into, but it allows for endless options of variation through it's procedural nature.



2. Building up from big to small

When building the shapes for your tileable texture the main thing you need to keep in mind is that building up the shapes for your textures needs to start with the big shapes building more and more into details.

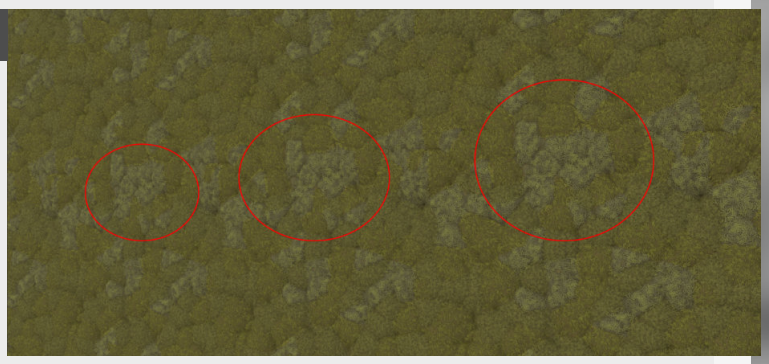
When doing this its important to really analyze the material that you are trying to create and really look at how its build up and try to dissect the different layers.



3. Keeping details tileable

When you get to the part when you start adding smaller details you need to make sure to keep testing your tile-ability of your texture, because most of these details will really make it noticeable when you start repeating it multiple times across a plane.

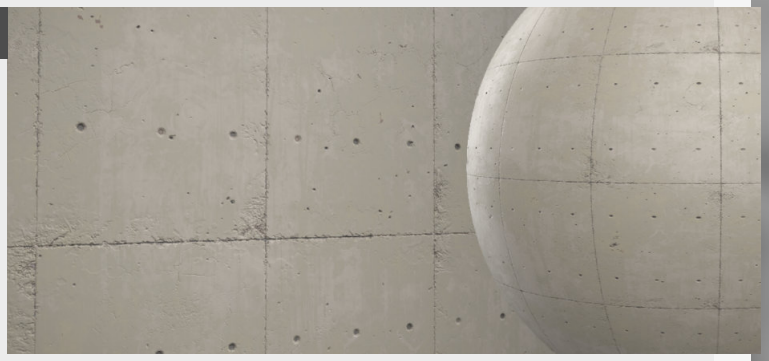
Like in the example on the right where you can really notice these details that really break the illusion of the tileable nature of the texture.



4. Make it the right size

Another important aspect is to make your tileable texture the right size, in most instances a texture above 1k is not even necessary to make it look good, we artist tend to try and get the most fidelity out of our textures and assets we make for our scenes and games with a disregard for optimization.

This changes once you get into a company where the restrictions are way more strict and you need to find tricks to optimize wherever you can, and this starts with how big the textures are, because the step from 1k to 2k in texture size costs is an upscale of 4 times.



Introduction to trimsheets

Introduction

Trimsheets is a texturing technique that is most used on modular pieces to reduce drawcalls and try to maximize reusability of one material. We will be looking into how to set these up, in this short intro to trimsheets.

If you are looking for more information on this topic, Polygon Academy (youtube) has some great tutorials on this topic that are great!

1. What are they?

Trimsheets are called that because of how they are being used and how they are laid out in the UV layout. They are laid out as Trims (long sections) and will in most cases tile on one axis (horizontal mostly) so if you have strips of edges or trims you can use this texture method to dress these up.

Environment artist use them to reduce the amount of materials and textures they need to texture a large part of the Environment by just using one texture for all the modular pieces this entire set will be textured with.

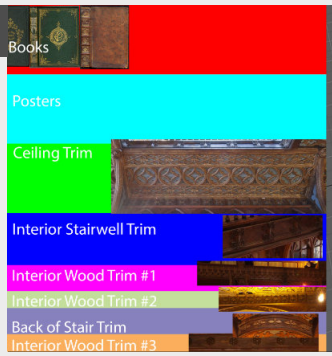

Keegan Keene - Nautical Trimsheet

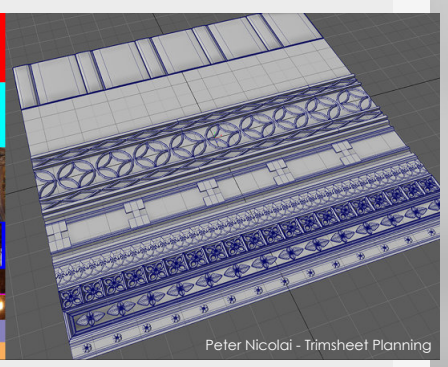
2. Planning ahead

Because you need to make a texture that will got all the meshes you have now and the meshes that you are planning to make in the future.

The nice thing about making a Trimsheet is that they are very versatile in the way you can use them, this means that you are not really restricted in the way you use it because you can cut new edges in the meshes and map a new piece of texture to it.

A good way approach this is planning your chunks in a 2D fashion. Either as a quick sketch or just blocking out some chunks in 3D. Make sure to keep these chunks aligned to the UV grid for easy unwrapping too!




Peter Nicolai - Trimsheet Planning

3. Setting up the texture

What we need to do next is translate this planning into 3D form and bake this down onto a plane.

We can do this with multiple options, by just modeling the highpoly in your favorite modeling package, or doing an additional pass in a sculpting package like Zbrush to add some more edge wear and damage, but if you are just looking for surface details that are pretty flat you can just add these by painted them using Substance Painter.

(A good reference is blog **#50** for **"using anchors in substance painter"**, which will help you embed these details onto the into the material.)


Ben Hale - Trimsheet Texturing

4. Practical uses

Trimsheets are super useful for the texturing sections of modular buildings, adding details to meshes that will give you consistency among all of them or just trying to optimize your scene by using one texture that gets used across all of them.

The best tip I can give you here is to be creative in the way that you use these trimsheets, even when you made the texture yourself you could be surprised by the way that you can use it.

And lastly, keep in mind how it looks, if the difference between a really up-scaled bit in your texture and a normal looking one are close to each other, they might stand out!


Ben Hale - Battlepots

What is texel density?

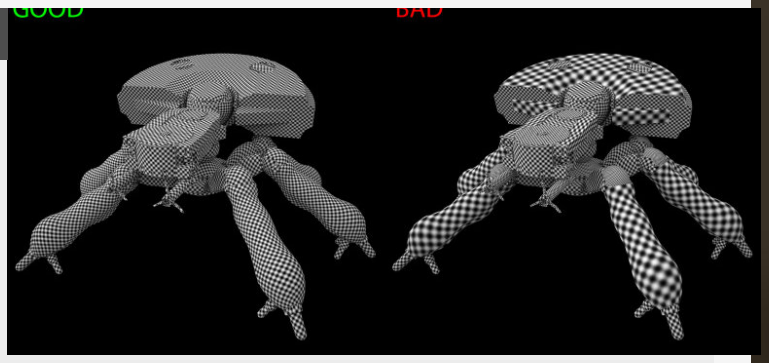
Introduction

Texel density is something that most people recognize in games when they see two textures right beside each other and the one is more blurry than the other. This discrepancy between the textures has to do with the Texel density of those textures on those assets.

1. What is it?

Texel density is the size of the texture and how this will be displayed on the assets or screen. For consistency we try and keep the size of all the different textures or elements you might use to the same standard.

So for a simple example let's say we use a **1024pixel** by **1024pixel** texture on a **4m** by **4m** tall wall. then from this we know that if we were to make a **4m** by **2m** tall wall using the same texture the UV space needs to take up half of the same (using 512pixels by 1024 pixels).
With this we are aiming for consistency between all the elements within a scene or section within assets as you can see in the example.



2. Different use cases

The texel density being used is does depend on the kind of game or scene you are making too. For example in a first person game you can get closer to the textures so that means that they are usually going to be higher resolution because of this reason.

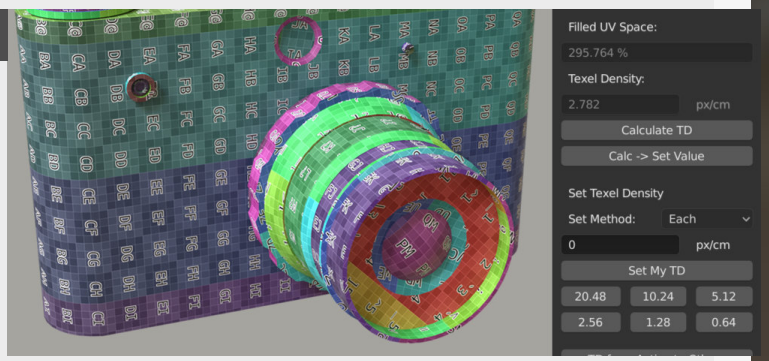
There are some exemptions to this rule though, if the asset is gameplay related (for example guns, objective assets, etc...) than it will usually be higher resolution and on the other side assets that are only used in the background (like mountains in the distance) will be slightly lower red.

3. How to measure it?

We take this 1024pixel by 1024 pixel texture that we have on the 4m wall for this we need to divide the texture size (1024) by the size of the wall in cm (400) giving us **1024 / 400 = 2.56px/cm**.

There are a couple of tools that we can use to measure this inside of all the different modeling packages, in some cases its build in like in Maya for example.

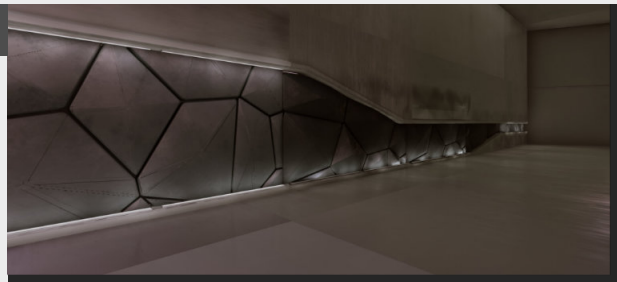
In other cases like Blender, you can find some sweet addons that do this trick for you, and example addon for this is **"Textools"** that has a texel density checker build in as well as other useful tools for unwrapping. This addon is available for Max too.



4. Debugging and comparing

When working within Unreal Engine you can't really measure the texel density directly using a viewmode or something like that because the texel density is going to be dependent on the size of the textures. So the way we get visual consistency is by using the tools In your favorite software before importing them into the engine.

A thing we could do is use make a checker material for each of the texture sizes we intend to use and then apply those according to the texture size onto the assets, we can then check if there is any major outlier within these textures and then adjust these if needed.



Anchor point normals

Introduction

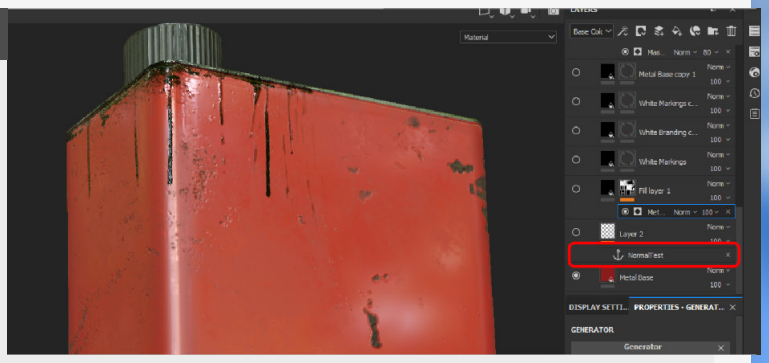
Ever wondered why those normals that you add on top of your mesh never pick up the changes you add after the bake? Well this entry will seek to rectify that straight away and let the smart materials pickup all the information that you add later on through the use of anchor points.

1. What are they exactly

Anchor points allow you to reference other layers inside of substance painter.

A really useful use case for this is adding normal details on top of the mesh after you baked the normal map from a highpoly onto a lowpoly mesh, because if you do this, it will not factor in the elements you paint into any smart materials you have applied

So by adding an anchor point to the layer that you use to add details to the material it and linking this anchor to the smart material it will have access to all the wear and tear information from that layer.



2. Adding normal details

Next we will be adding some normal details to the model, for this let's go to the "shelf" and look for "hard surfaces". In here there are normal maps that you can add.
To add these to your mesh you need to add a new layer, go to your brush and add it to the normal slot and disable the alpha. Because if we don't remove the alpha we are going to get an alpha falloff.

Now we can start adding details to the mesh that we have, but the wear and tear is not going to show just yet.

Which brings us to the Anchor points!

3. Anchor point setup

First we will add the anchor point to a new layer that contains the previous details. In the anchor point we can see that it is currently empty of any references so let's add one. For this example we will be using just a simple white fill layer with a metal wear and if we go into that generator we can add the anchor point in the micro normal section and change the reference channel to normal for it to pickup the correct information.

Last thing to see the changes in action is to put the Micro Details "on" in the Generator Parameters.

4. Settings and remarks

Now once this is all setup, we can start digging into the settings in the generator "micro details" section of the metal wear generator.

Inside this section we can change the a bunch of different parameters to give you more control over the spreading of the actual effect that you have anchored.

Keep in mind that you can only reference things in the same texture set and can only be used on fill layers, fill effects and substance filters.

Woven patterns in Substance Designer

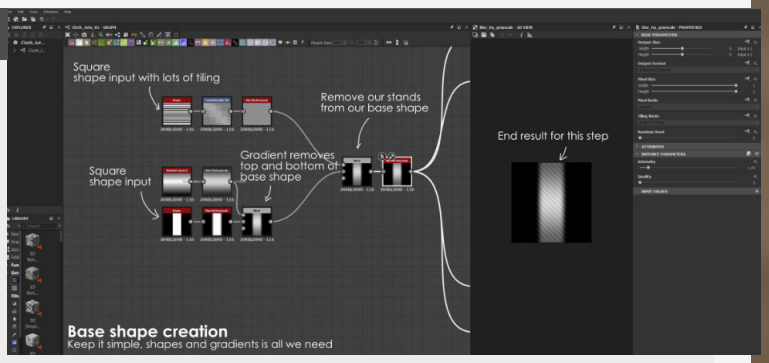
Introduction

Something I've done recently for my own little scene I've been working on some materials and Jute was one of them, but you can use this technique for anything with a woven pattern.

1. Base shapes

First off let's start with the base shape that we will be using here, this will start from a simple square pattern that we then squeeze down, and then we multiply this with a gradient to make it falloff at the top and bottom (which we will use for making the pattern later)

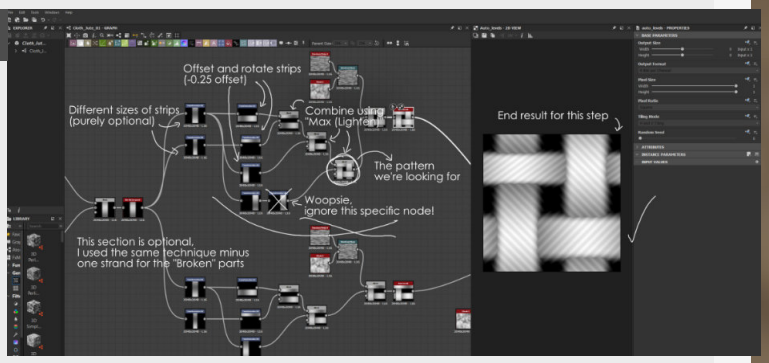
I also wanted to get a bit of a feel here that they are made of strands so we subtract some slightly blurred lines from it to make it feel as such.



2. Woven Pattern

We have our base shapes now, it's time to make a pattern out of them, for this we will be using a couple of simple transforms and blend them together. For this we want to move the darker part of the shape into the corners and then overlap the white part of the next one with the dark part, this will create the overlapping nature we're looking for.

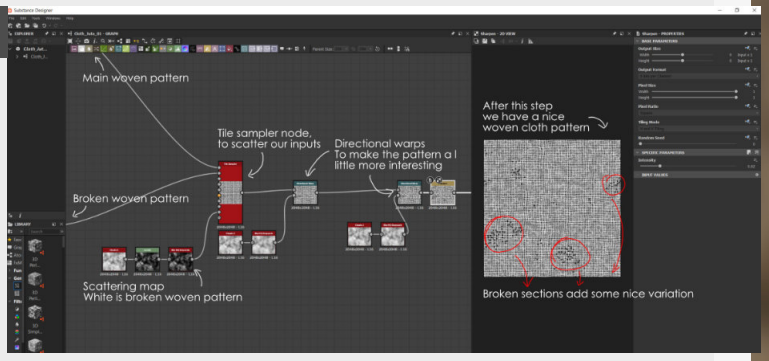
Then we just have to repeat this process for the other corners of the pattern, all blending them together using a blend node with **"Max lighten"** and finishing it off with an auto levels to push the black and white values until they are maxed out.



3. Scattering the pattern

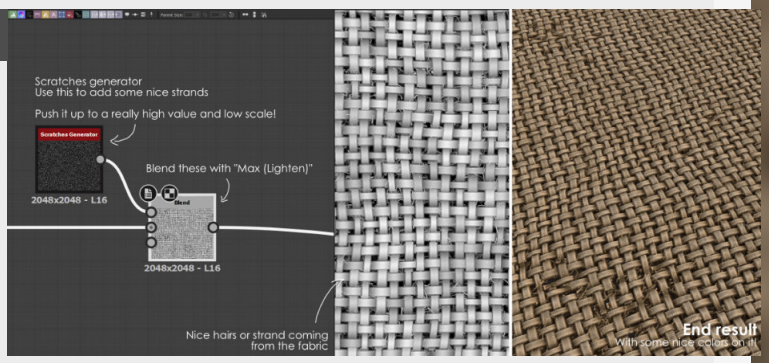
Let's scatter the pattern using a Tile sampler next, if you want to make a damaged pattern then I did that by duplicating the setup we made before and cutting out one section of the pattern which makes it feel like it's a little damaged, I haven't refined this step for my purposes as the scale I'm aiming for is having a lot of tiles, so you won't see that detail up close anyways.

With the tile sampler we can control the amount of tiles with the **"X"** and **"Y"** amount, if we keep them at the same values then we get a nice square pattern, which is what we want here.



4. Random strands and colors

Now the material that I choose here does have strands and little tufts sticking out of the material, so as a bonus step I've added a simple step of adding these with a scratches generator that I then add this on top of the previous by again using the "Max" mode so they only show up in the darker regions, so we won't have them spawning on top of the white.



Anchor point color variation

Introduction

Anchor points are really useful in Substance Painter for linking information between different layers in your current scene, one of my favorite ways to use these is to add them for color variation for trimsheets.

But can also be used in various other ways to add different colors on top of your favorite materials that need a little color variation.

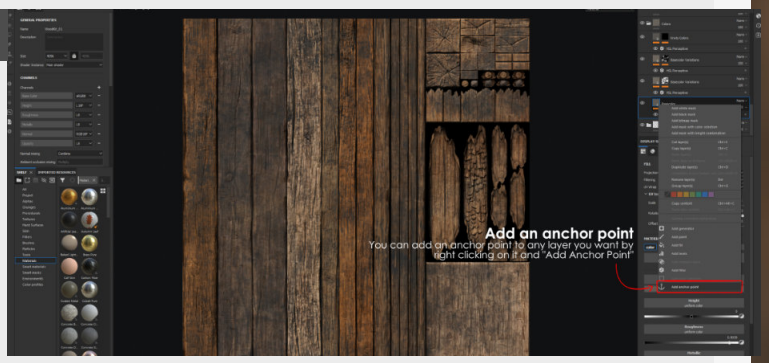
1. What are anchor points?

They allow you to sample something from one layer and use it to another layer above it. Making it a really interesting way of sharing information between different layers in your current file.

I've already written about this a long time ago where we used these anchor points to make sure that edge wear also applies to normals. **(which can be found under "Anchor Point Normals")**

2. Setting up an anchor point

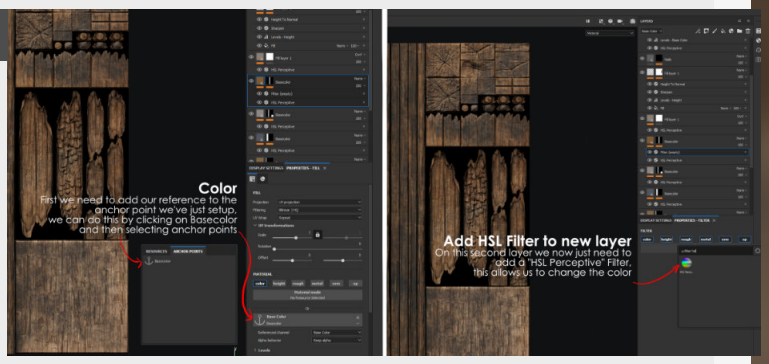
First thing we need to do is to create an anchor point itself on the layer that we want to sample, in this case for our color variations we're going to be using the **"Basecolor"** layer, which will be serving as our base color from which we will be building the other layers.



3. Linked color layers

Then on the layer above that we can sample the basecolor from the anchor point and then add a HSL adjustment later on top of it that allows you to change the color of the anchor point we're referencing.

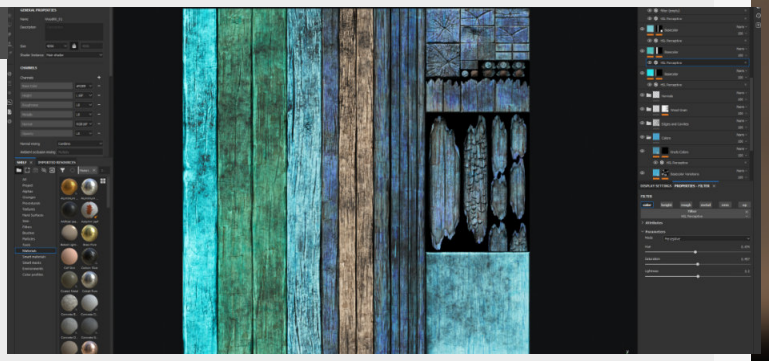
We just need to do this a couple of times for each color variation we want to have, but once it's setup we can easily go back and change the colors if needed.



4. Adjusting the color

Now that we have all the other layers setup with an HSL adjustment layer on it we can go back to the base layer and change the color of this "Basecolor" layer and change the color of this specific layer.

Changing this layer will also affect the layers on top of it as well, making it super easy to change the color of everything instead of changing them all one by one.



Hand Painting tips

by **Nikkita Racquel**

Introduction

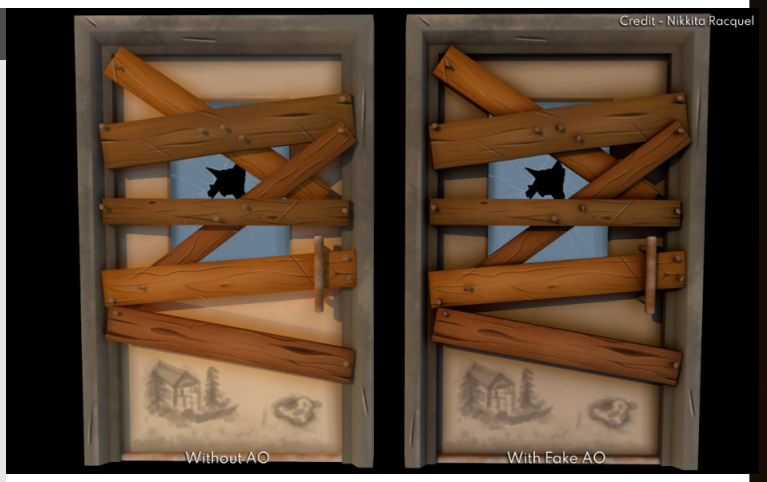
A few tips to getting that little bit extra out of your hand painting using diffuse only.

1. Fake AO

As the Ambient Occlusion information of a mesh comes from the AO map that would exist in a PBR setup, we need to "fake it till we make it". Because AO adds such a large amount of depth and shape dimension, it's important to make sure we include it or the piece will appear much flatter and less vibrant than we intend.

Well painted light does a lot of great work for a piece but if it doesn't have shadows to contrast against, the light falls short and doesn't have the same level of impact. Of course it goes without saying, there will normally be an element of overall AO due to post process, depending on how the artist chooses to render their piece.

However this overall AO while providing good depth to the piece as a whole, doesn't do as good of a job as a baked AO map to provide detail shading to each individual object. Making sure to paint in AO shadows not only help to communicate form but help to push those textures further.



2. Consider Textures

Because traditional hand painting doesn't utilize PBR maps to help portray the intended effects, considering the effects the texture may contain or even the texture itself is vital to telling the right story. When mentioning texture in this sense however, it speaks to the material the artist is emulating whether that be wood, metal, glass, meat, bone, etc.

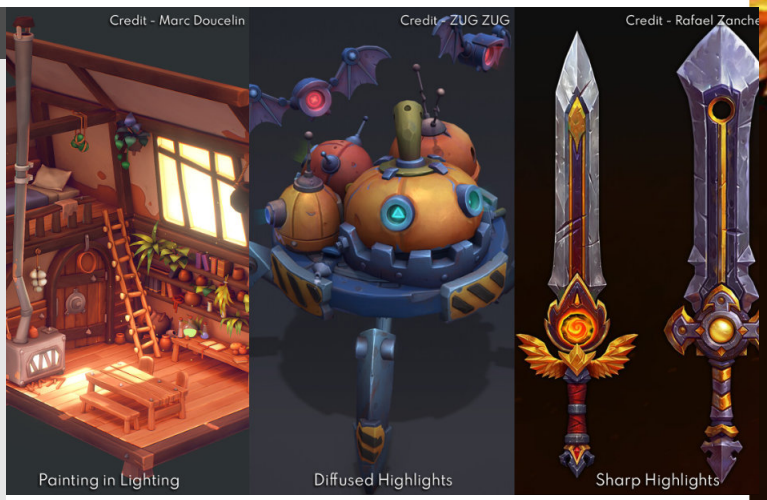
Next step is now incorporating story to consider what texture effects carry that story. This is something done in every texturing workflow of course but in another form with hand painting, all of this information needs to be included in the diffuse and convincingly carried across.

Let's use meat for example. If cooked, the outside could be crispy in some places, glossy due to oil, the inside might also be glossy due to juice, there could be some areas that are very matte due to how meat can be after being cooked.

3. Incorporate Light

This tends to be used more in personal projects by artists as it just isn't feasible to use in-game but it looks fantastic when done well and still is a very viable way to practice how light might interact with certain shapes and colouring. The way of incorporating light that is seen pretty much every time hand painted work is encountered is of course, highlights. This tends to come two fold, depending on the look the artist is going for. A much more diffused light or diffused material but still containing some sort of light reflectively, will have a softer section of highlighting over a larger surface area.

Second stronger highlights either laid over the top or used on their own come from either the material itself very reflective or the light is catching the material at such an angle, it would create a harsher highlight. The second tends to be found a lot on metal, edges of grain like wood, glass, water or high gloss materials.



What is Ambient Occlusion

Introduction

There are multiple different implementation and use cases for ambient occlusion, we have the texture version and also the in engine implementation.

1. What is it?

Ambient occlusion is a way to simulate surfaces that are close to each other and cast shadow a little shadow that really grounds elements together. Usually this small shadow is really tricky to render (we are getting closer to get good results tho!)

This was first introduced as a post processing effect by Crytek, but it can also be used as a texture to simulate the same effect but more localized.

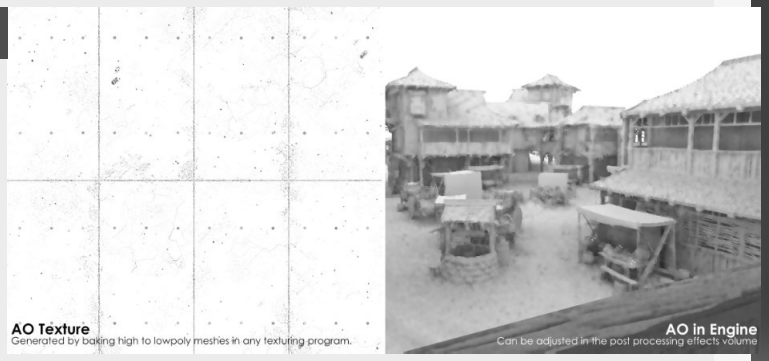
2. How do artists use it?

TEXTURES

Ambient Occlusion Textures can be baked inside of Substance Painter, Marmoset and other bakers... And then you can use them in your engine of choice to add this additional shadowing to your meshes.

IN-ENGINE

This is usually a global or post processing setting that you can adjust the size of the effect and also the intensity of it.



3. Common Engine implementations

(SSAO) Screen Space Ambient Occlusion
As in the name, this will only work for assets that are within the field of view of the camera, so assets outside of the screen will not have this effect.

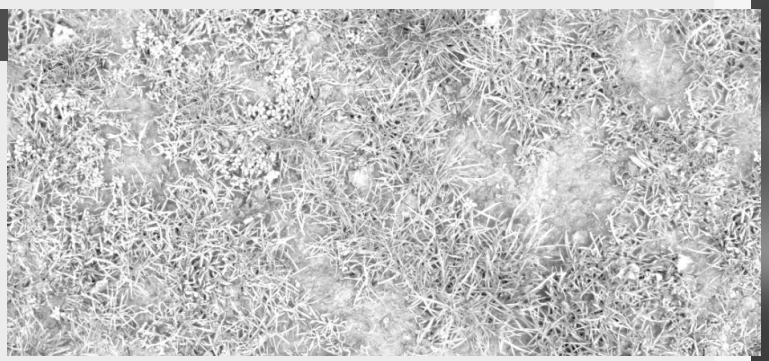
(DFAO) Distance Field Ambient Occlusion
This is specific to Unreal Engine, where it needs Distance Fields to work (have a look at the previous entry on this topic)

(RTAO) Ray Traced Ambient Occlusion
Probably the most accurate and also most intensive solution available to us at this time, and removes some of the flaws of SSAO but is still limited to screen space itself.

4. Texture implementations

If we have highly detailed objects we can also bake these Ambient Occlusion shadows down onto the textures to replicate the effect within our textures. This is especially useful for smaller details that rely on fine variations.

All engines also come with material implementations where you can add the Ambient occlusion into it should you want too.



08 MATERIALS

World Aligned Materials

Introduction

Most of the time when we deal with textures we only see them when we apply them to textures, but we can also add them on a world basis too if we want them too, allowing us not to have to worry about the texture lining up between different meshes that are close to other. But there are also other great applications of this...

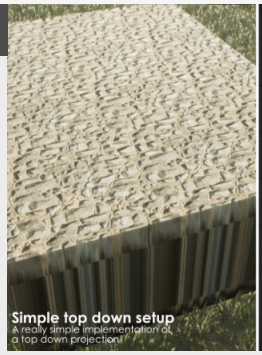
1. World space textures overview

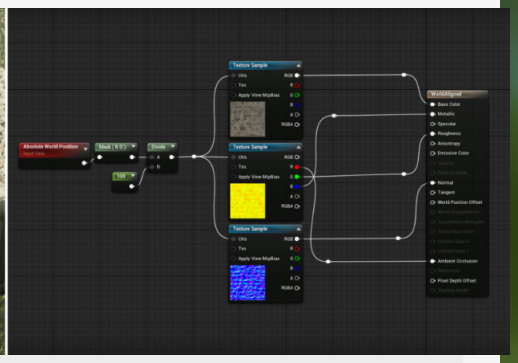
Before diving into the examples, let's start with the basics.

Let's open the material editor first and start by adding an "Absolute World Position" node, which is going to make it so that the textures we add are now going to be added on world position. Now you can quickly see that this is still stretching on the sides of our little cube we've added.

We can then divide this by our UV scale so we can control the tiling and then add this to our UV input slot on our textures.

This is more for as a top down projection, which can be very useful for different grass colors, modular floor meshes, etc...



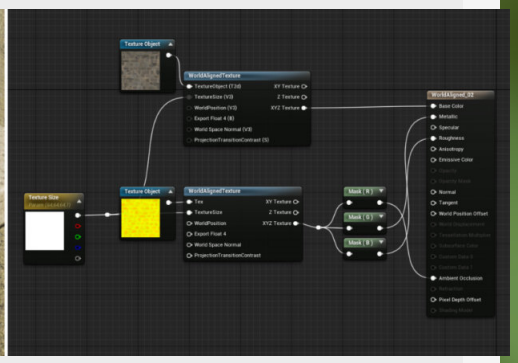


2. World Aligned Texture Node

Now what if we want to add the textures on walls for example?

This is where the **"WorldAlignedTexture"** comes into play, this more advanced node let's us change choose how it will be projected onto the meshes we add this material onto, so in our case we can add a XYZ texture to our material and we now have the texture applied to all sides of our mesh.

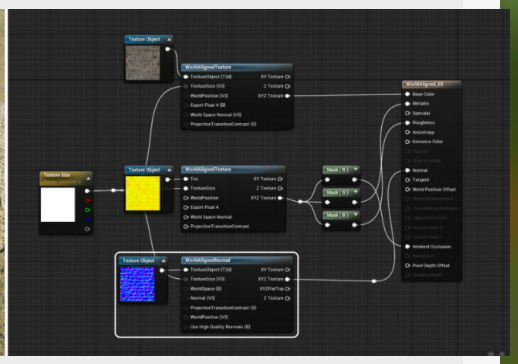
Keep in mind that this doesn't use your normal **"Texture Sampler"** node but rather a **"Texture Object"** node, which doesn't have all the inputs and outputs but you can still just add your own texture in it like you normally would.



3. World Aligned Normal

Now, this works for everything except for a normal map, because a normal map influences how light warps on top of that mesh we need to add in some extra math in there to make that work. We're not doing the math our self tho.

For this we can add a **"World Aligned Normal"** node, which gives us a different looking node that's specifically for our normal textures, there are a bunch of options on these nodes, but usually we can keep them pretty simple.



4. Some examples

Some example use cases where you can find these tricks are when you want to have a nice gradient of color going over your grass, have a bunch of cliffs or walls that you can add onto a spline or over multiple meshes without having to worry about your tileable textures or UV's linking between different meshes.

Don't let that limit your imagination, you can use this for your textures but also for your masks like in the grass example, think outside of the box.

Master Material Basics

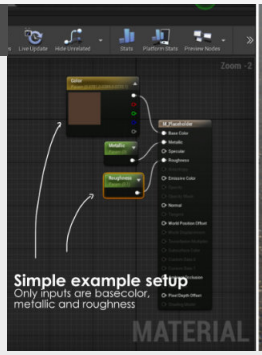
Introduction

Master materials are a powerful way to create a reusable material for multiple assets in your scene. These are also more performant and quicker to make then to setup a new material for every asset in your scene. So let's have a look at how we can set these up for our scene.

1. What are master materials?

They allow us to reuse the same graph over and over and change material parameters on the fly and independently from each other through material Parameters. Allowing you to change them on the fly through Material Instances (in next episode).

As a simple example, you could have a Master material for your placeholders, which in it's most simple form is just a Color, Roughness and Metalness adjustment, if we turn these into parameters, we can then change them in the Material instance, independently from other material instances or the master material. And we can always go back to the main material to do adjustments.



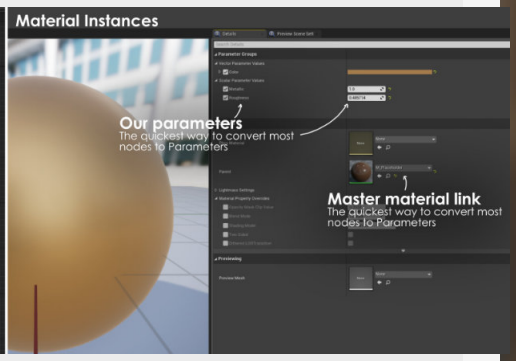
2. Creating parameters

We can create them by in most cases right clicking on the node and selecting "Convert to Parameter".

Name: Pretty self explanatory, just put the name of the Parameter here, this is also how it will show up in the Material Instance.

Group settings: These are pretty useful, allowing you to separate all the parameters in different sections. Trust me this is going to be really useful making more elaborate materials.

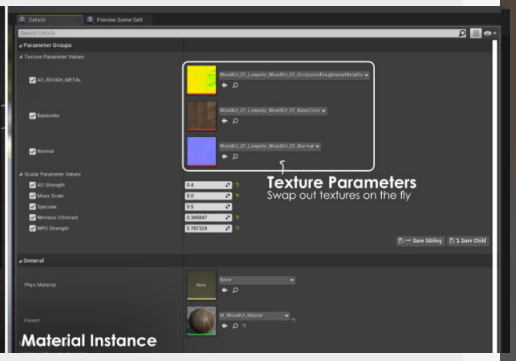
Values: Allows you to set the default value and the min/max values for the given parameter, this limits the use of the parameter.



3. Texture Parameters

Textures can also be turned into parameters, opening up a whole new side to this, you can have a master material that allows you to control the damage on your assets. Then have different texture maps for different assets, and so you don't have to create new material, just material instances.

Then once you have these different material instances, all you need to do is drag in your new texture maps slots and you're good to go!

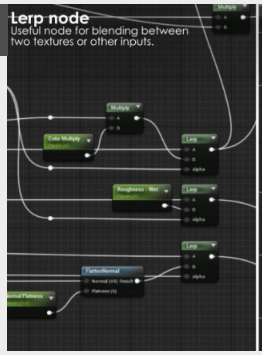


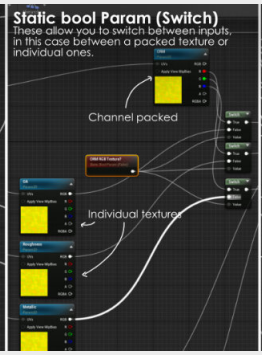
4. Other interesting nodes

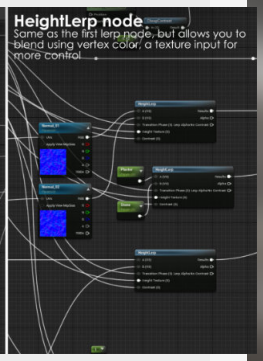
Switches
Allow you to switch on/off section of the material when needed, so for example you could set this up where you don't want to adjust the damage, switching it off, or even add/remove additional controls.

Lerp
Technically not really a parameter itself but allows you to hook up an "Alpha" parameter that goes from 0-1, allowing you to adjust a blend of the inputs A and B.

Heightlerp
The same as the lerp above but it uses a heightmap for blending, making it more detailed and interesting to use. So give it a try!







Material instances and how to use them

Introduction

This is building on the previous Tip of the week on "Master material Basics" so check that one out before reading into this one.

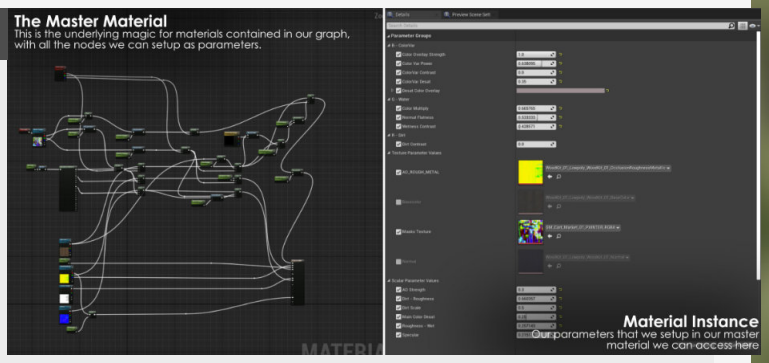
Material instances are really useful for creating quick variations without making the same material over and over again, we can setup parameters for everything that you don't want to make over and over again.

1. Material instances

Material instances allow you to access all the parameters setup in the master materials, they also act independently from each other.

This means you can have two material instances using the same master material but have completely different parameter values. And if you need to change any parameter in the master material this will automatically propagate to these instances too.

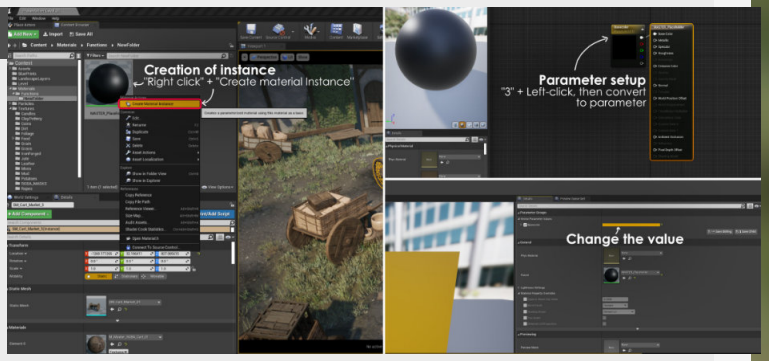
This allows for a ton of creative freedom and ease of use in any scene. Cutting down on the boring redo work that you most of the time have to do with any scene.



2. How to use them?

If this is your first time setting something like this up, start with a simple setup. An example of this could be using a main material for all your placeholder colors for your different assets, so it's pretty easy just hold down **"3"** and click (this will create a vector3) and then we can convert this to a Parameter, and then we can adjust this color in our material instances and create new material instances for each new placeholder color.

Then you can set them up by right clicking on the main material and selecting **"Make material instance"**. Then we can access all the parameters we've setup.



3. Material per object/shader type

One thing I usually try to do is make one master material for one type of setup, so for example one shader for foliage and one shader for my typical assets that don't require Subsurface scattering and alpha.

You could setup a master material for all the assets in your scene but the amount of balancing and tweaking you have to do to make it work for all your assets (Let alone the amount of instructions etc you need, making the material really expensive).

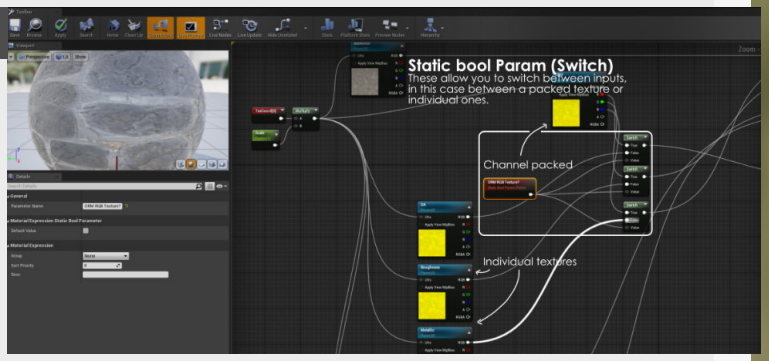
However, you can also switch the shader type in the material instance if you need too in the general tab of the material instance.

4. Using Switches to disable sections

To optimize the material more we can also use **"Static booleans"** and **"Switches"** to disable parts of the material to make the material cheaper to render.

We can add these by looking for **"Static Bool Parameter"** and a **"Switch"** and hook the Parameter to the Switch. Then you just need to add the node setup to both **"On"** and **"Off"** to control the nodes that will be activated when you have either of them selected.

However it's still cheaper to not have any of these nodes in your material that you might not use anyways, so that's also why I make Master Materials per shader/object type.



Material hotkeys

Introduction

This entry will focus on some of the hotkeys that can be used inside of the Materials editor inside of Unreal Engine. This will really help you speed up the material creation process inside of Unreal Engine.

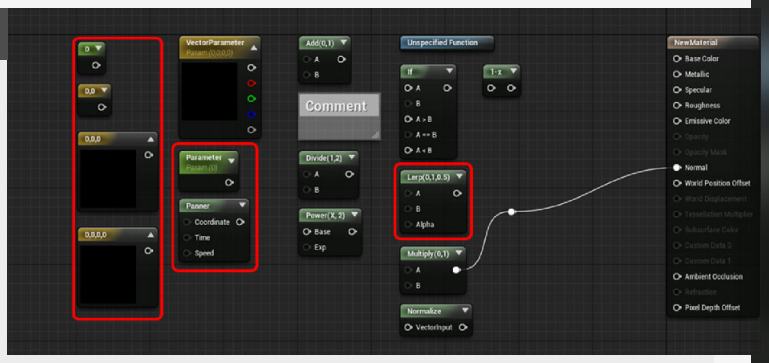
1. Section 1

Parameter (S)
Probably my most used hotkey, this allows you to add a simple integer value as a parameter.

Linear Interpolate (Lerp - L)
Allows you to blend between two values based on a 0 to 1 value.

Vector 1-4 (1 - 4)
Adds different vector values, so 1 is an integer and 2 to 4 are vector values going from a vector 2 to 4.

Panner (P)
A panner node allows you to scroll different textures, this can be used for things such as rivers,etc...



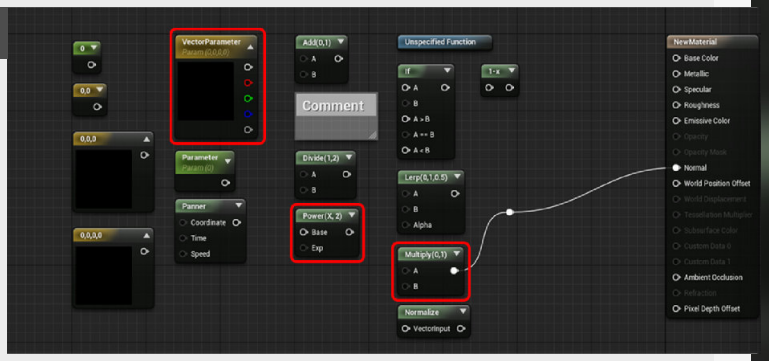
2. Section 2

VectorParameter (V)
This is just a quick way to get access to a VectorParameter instead of making a vector first and then right-clicking to convert it.

TextureCoordinate (U)
This allows you to link to a texture that you have in the content browser.

Multiply (M)
Quick access to the multiply node, which allows you to blend two values together.

Power
This will take the first input and multiply it with the "Exp" input.



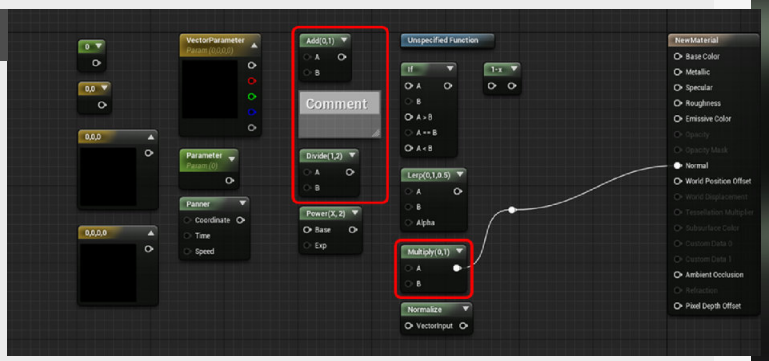
3. Section 3

Add (A)
Blend between two values, but adding instead of just blending them.

BumpOffset (B)
Also known as Parallax mapping, this is essential to give a fake depth to your material.

Comment (C)
A nice little comment box to clean up your material graphs.

Divide (D)
Takes the first input and then divides it by the second input.



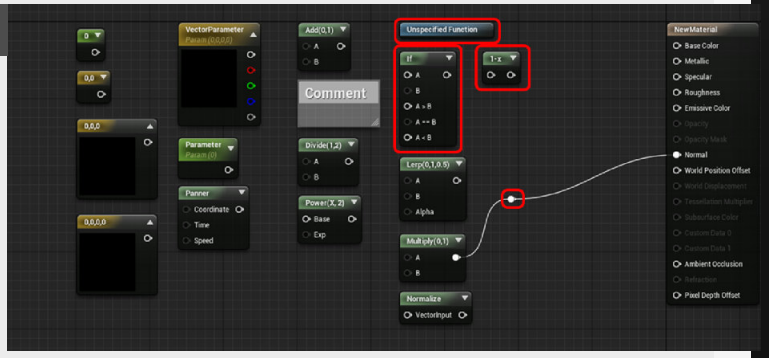
4. Section 4

Oneminus (O)
Inverts the given input so for a simple example if you feed this 5 it is going to put out -5.

If (I)
Checks if the inputs are of a certain comparative value like "Smaller" ,"Equal" and "Greater"

MaterialFunctionCall (F)
This is to call the Material function that you might have made previously, if you have added one.

Reroute node (Doubleclick on connection)
Reroutes the connection, makes it easy to clean up those material graphs.



Vertex Painting - 01

Introduction

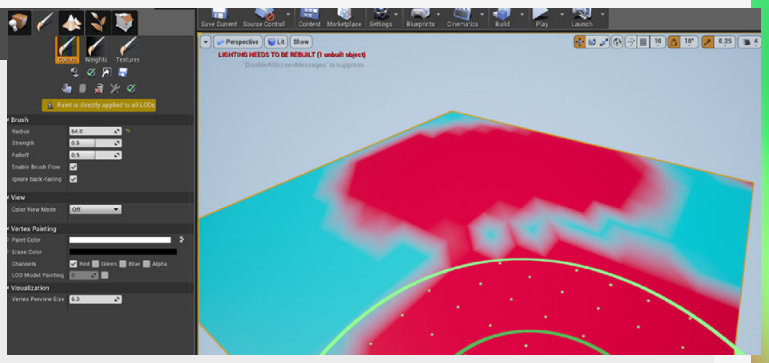
Let's have a look at vertex painting, this is one piece of a multiple edition talking about multiple aspects of vertex painting, this one will talk about setting up the materials and some more general information.

1. Getting started

So, Let's talk about vertex painting, super useful for blending 2 or more materials together, commonly used on terrain, but can be used on meshes as well.

It allows you for blending of values based on painted layers that uses the vertices to store it's color value.

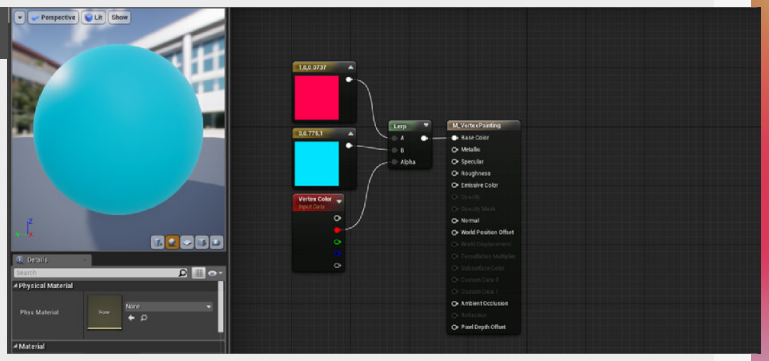
We can then use these color values to assign materials to them through the use of some nodes in the material editor.



2. Basic material

So let's dig into the material itself, starting out with a simple blend. So let's add two colors and a "vertex color" node which will allow us to pick we are going to use for blending between the two colors.

If we then use this as the alpha value for a Lerp node and have the two colors as inputs in the lerp, this will then allow us to blend between the two using the vertex painting channel we chose (Red in the example).

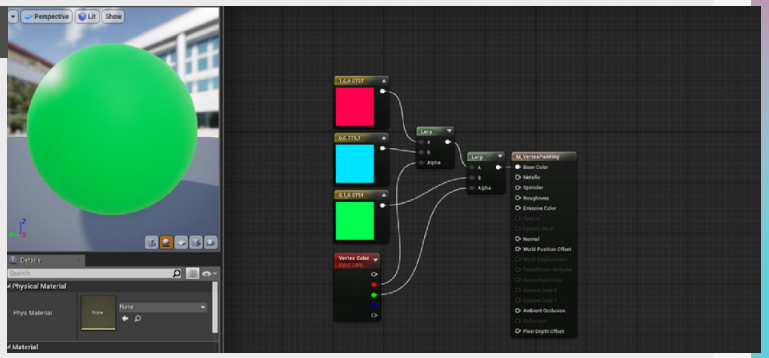


3. More layers

Now what if we need to have more than two layers? We can just add another Lerp with another color as input.

Now that we have three colors and two lerps we need another channel from the vertex color node to allow us to blend between them all. you can use these inputs to control other material attributes.

When we go back to test the vertex painting we can now see that we need to use both Red and Green channel instead of only the Red channel.

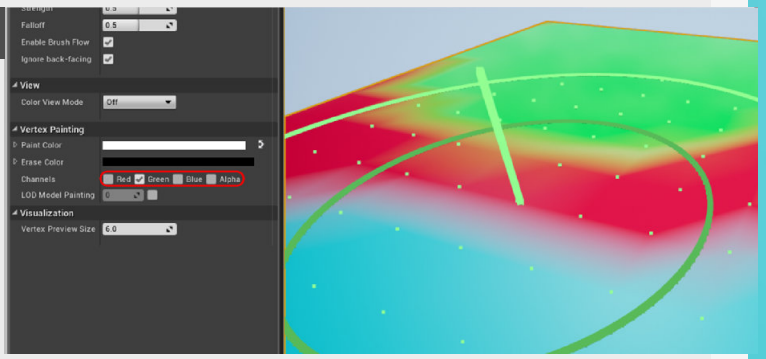


4. More detailed control

The channels of the tool correspond to the channels of the Vertex Color node inside of the material we have just created.

Another good thing to know, as you probably already noticed is that when you are using vertexpaint, it uses the vertices of the mesh you are painting on, this then means that if you need more control over the blending between materials you need to add more vertices so that you get more resolution.

Vertexpaint uses the vertices to store data, vertices = "Resolution".



Vertex Painting - 02

Introduction

To continue this little multipager on vertex painting, we are going to continue with an example on how to use textures and blend some puddles on the basic cobblestones, which should give you a good example to show what you can do with vertex painting.

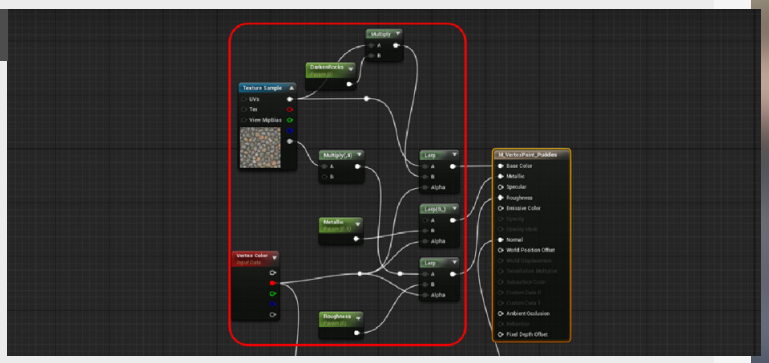
The previous one can be found here

1. Continue on part 01

As discussed in the previous section where we went through the introduction on how to setup and use the basics of Vertex Paint.

So to take this initial introduction further, we are now going to look at how to expand on this with some actual texture instead of colors, and for this we are going to use some of the existing content already in Unreal Engine.

2. Material setup

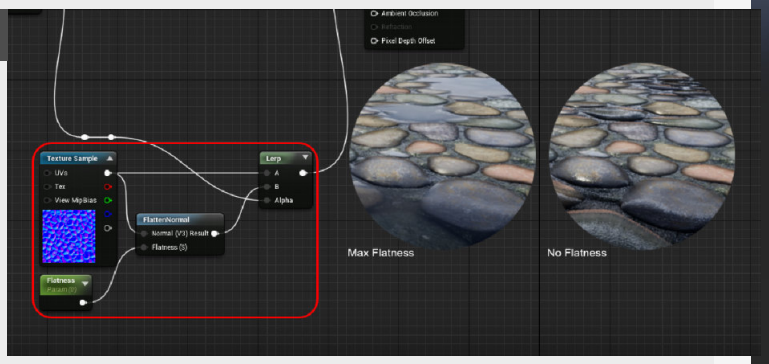


Albedo: Lerp between the normal version of the pebbles ("T-CobbleStone-Pebble-D") and a darkened version of the pebbles (to give it a look that they are under water)

Metallic: Blend between 0 (no input) and 0.5 for the water value.
Roughness: This is a temporary value using the Alpha map and a parameter for the roughness value for the water.

For the normal of the texture we are going to move to the next section.

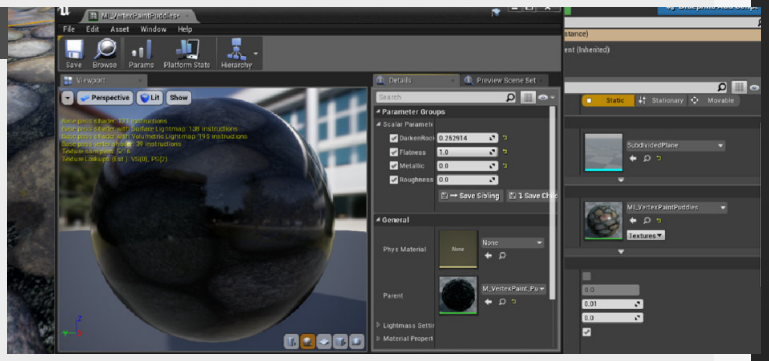
3. Flatten the water



Next we need to flatten the Normal, luckily for us there is a node that does all the heavy lifting, also we can let the height texture influence this as well, but I didn't do that.

This part of the material is just a blend between the normal of the pebble texture and the same texture but then flattened using the "FlattenNormal" as a parameter (which I called Flatness) to control how flat the normal of the water is going to be, allowing you to fake the depth of the puddles.

4. Parameters



So then to help with quick iteration, we can go through the material and change all the values to parameters which will allow for quicker iteration time if you instance the material.

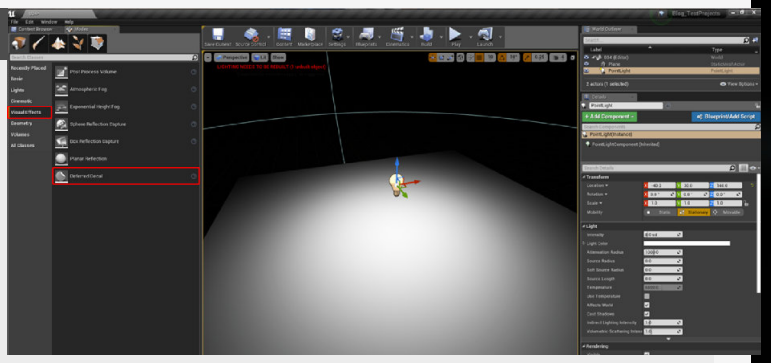
Now when we paint using vertex paint with the selected vertex color of the "VertexColor" node, we can see that we can paint in puddles on the texture on the mesh (make sure that the mesh has enough vertices to work with for the VertexPaint to work)

Decals - 01

Introduction

In this entry we will take a look at decals, how to set them up and how to use them in your environment. Decals are a useful way of adding unique details to modular sets that rely heavily on tileable textures and cannot have build in unique details.

1. Why do we use them?

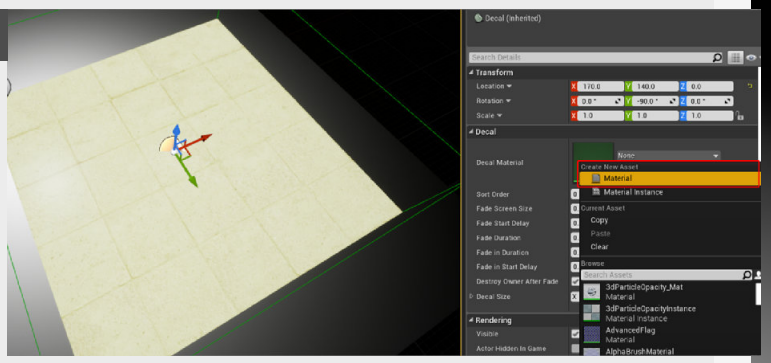


Decals come in many forms and shapes, there are a ton of different usecases that we can use them for as well. In this Mini-Series we will be going through some of the major usecases.

Some of the use-cases can be adding damage to walls, graffiti, smaller dirt cover on floors, edge damage etc...

But before diving into some of these examples it is important to start out with the basics first and see how they are setup in Unreal Engine.

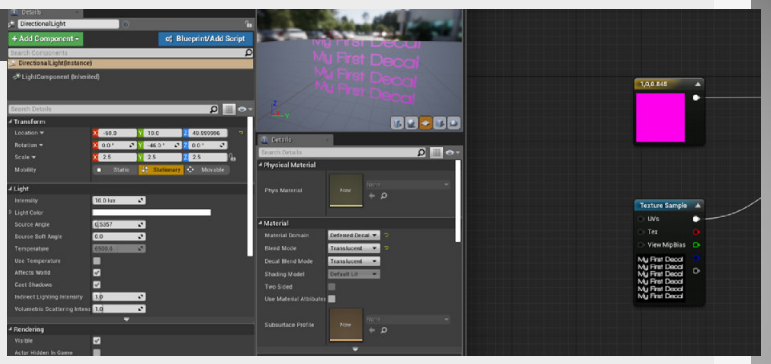
2. Starting out



So let's have a look on how we can get these working in our projects, first we need to grab a "Deferred Decal" asset from the "Visual Effects" tab in the "Modes" menu.

If we drag this into our level and we look into the details menu of that Decal we can see that the "Decal Material" is empty, so let's get to fixing that, if you click the dropdown menu you can see that there is a section that is called "Create New Asset" so let's create a new Material that we will use for this reason.

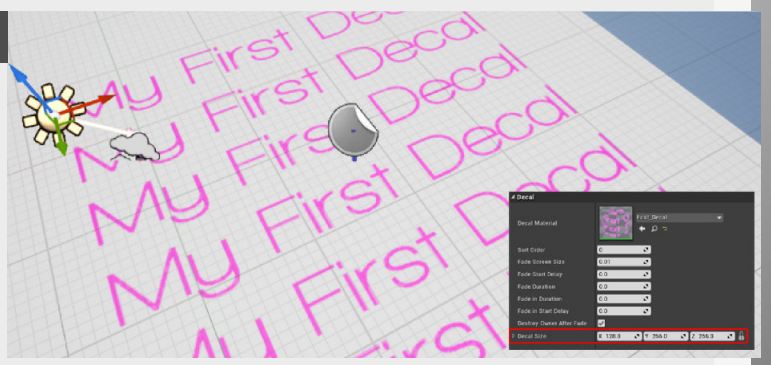
3. The setup



For this particular basic example we are not going to take a deep dive into materials, we are going to stick to the basics for this one.

When creating a Differed Decal material, you will need to change things like the "Material Domain" to "Deferred Decal" and the "Blend Mode" to "Translucent" (which is going to pop up when changing the material domain). I just added a quickly made Alpha Mask Texture for the opacity slot and a simple color for the Basecolor input.

4. Usage



Now you noticed something else, that the Decal is basically is a projection inside of the "Box", so scaling this box influences the way the decal is going to be drawn.

This means that if you have blood splatters or any thing else that splatters on the environment the box is probably going to be bigger than graffiti for instance.

To adjust this you can just scale the decal like you normally would or you can change the "decal size" inside of the decals details menu.

Decals - 02

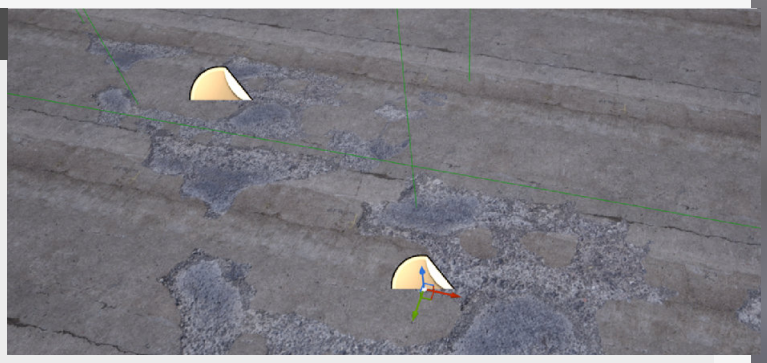
Introduction

In the second part of us looking at decals and how to use them Inside of Unreal Engine, this time we will look at an example use case of decals.

1. Why do we use them?

Damage decals are a really nice way to add details to some flat textures, that can quickly become super tileable looking or look flat otherwise. This is also a nice way to add some nice story or texture to the Environment.

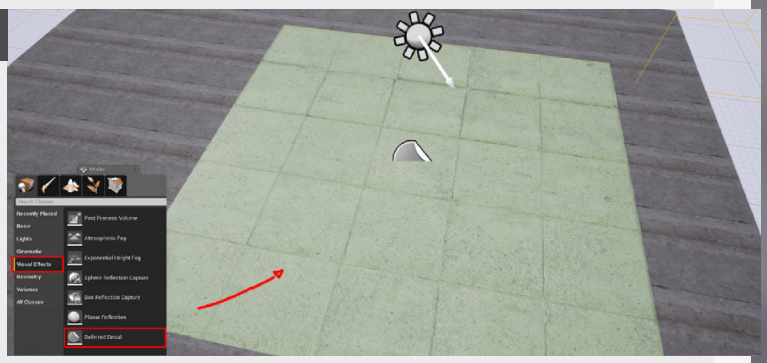
In this example I will be using some textures that I have found on the texture website www.textures.com so the examples are going to be a bit different to what you would normally do yourself as I only have the diffuse to work with for these textures but the main idea remains the same for these.



2. Adding the decal

First off I added a texture to the normal plane that is in the default level when you load up a new scene in Unreal Engine, this will be the basis for a floor texture which we can then overlay a decal over.

So then let's add a decal to the scene like we did in the base example from last time.



3. Setup and options

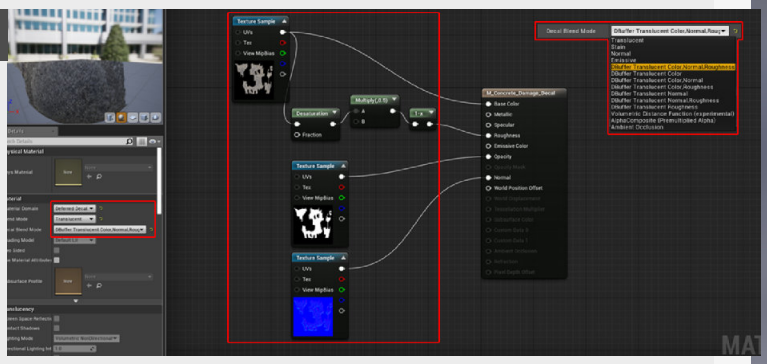
As mentioned before, this is just the setup using the textures from www.textures.com.

The roughness in the material is just the de-saturated version of the basecolor and then inverted, and a quick normal map, but just as examples.

"Decal blend mode":

This is the most important option, this dropdown allows you to choose which material outputs are going to be used by the Decal in the level.

So for this instance I used "DBuffer Translucent,Color,Normal,Roughness". So it uses all your typical inputs.



4. Usage & Experimentation

Once everything is plugged in you can now see that the result is quiet nice, it had it's own Normal, Roughness and Basecolor map.
We can then use the same decal in multiple ways to create some nice variation in the damage that has been done to the floor.

Ofcourse for a better and more varied result you can use multiple different damage decals on one decal sheet to save memory and both make them more useable.
There are a lot of options in the decal section, for cheaper decals you can only use the normal blend mode, but feel free to experiment yourself.

Master Material Creation - Part 1

Introduction

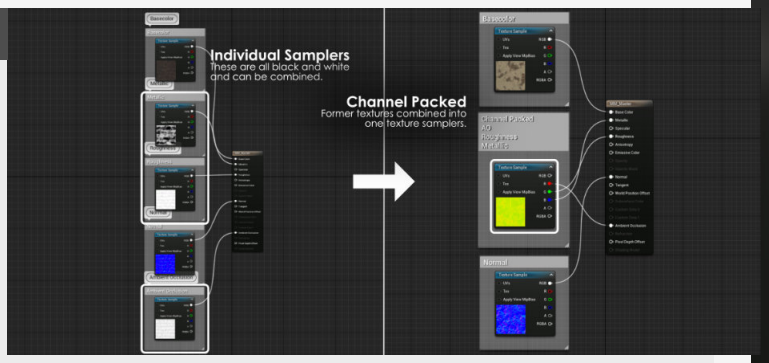
Let's take a deeper dive into material creation in Unreal Engine, in this one we're starting with a simple section of a master material and then working our way up for it to be more complex.

Keep in mind that master materials and the nodes you use in them depend on the outcome you are working towards, so this is more used as an example on how to set one up yourself and you should tweak it for yourself.

1. Texture samplers

Keeping it simple with this one, we're going to just grab all the textures and drag them into the material editor, we can then connect them to the main material output, that honestly the most basic for a material setup in Unreal.

Even in this most basic form, we can already do some optimizations, by channel packing texture together to optimism the sampler count of our materials. So instead of having a separate input for metallic, roughness and Ambient Occlusion, since they are all black and white, we can pack them together in one single texture. More on that can be found in **"RGBA - Channel Packing"**.

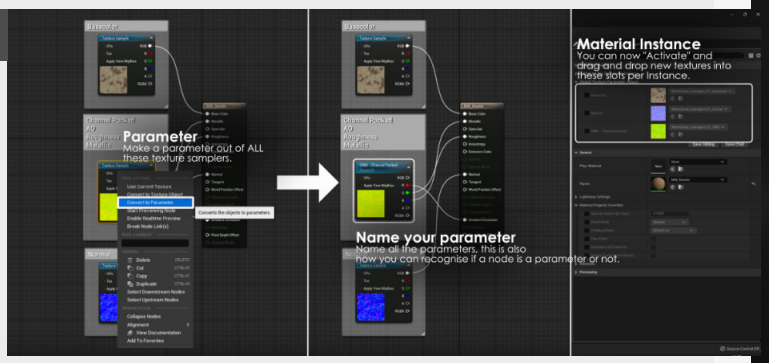


2. Making Parameters

At this point, our material doesn't offer any options for customization, this is where parameters come in.

We can turn a lot of nodes into parameters, which makes them adjustable once we make a Material Instance from this Master Material.

So just simple right click on the texture samplers we added before and then **"Convert to Parameter"** give them a suitable name. This will then allow you to drag in replacement textures in the instance. Making this material usable for more than one instance. We just need to make a new instance for each new version of the material we need.

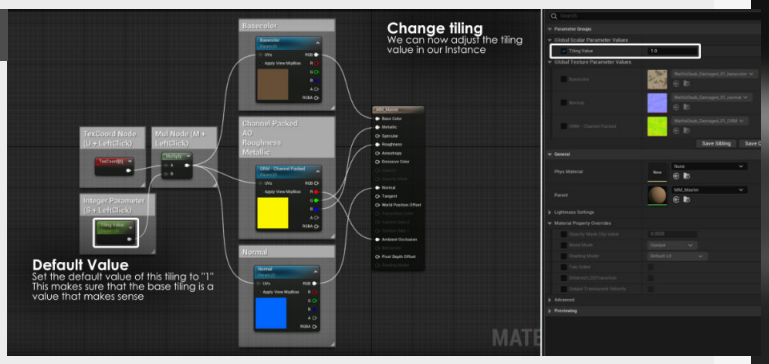


3. Change tiling option

Let's add some additional power to the material by allowing us to change the tiling of the material, which is useful for tileable materials.

To do this we need to add a couple of nodes before our texture inputs, firs off a UV node, to direct which UV we want to adjust, then multiply this with the value we want to control (Make this into a parameter) and then connecting this into all the inputs for each texture input.

This gives us both control over the texture we can use and also control the tiling of this material.

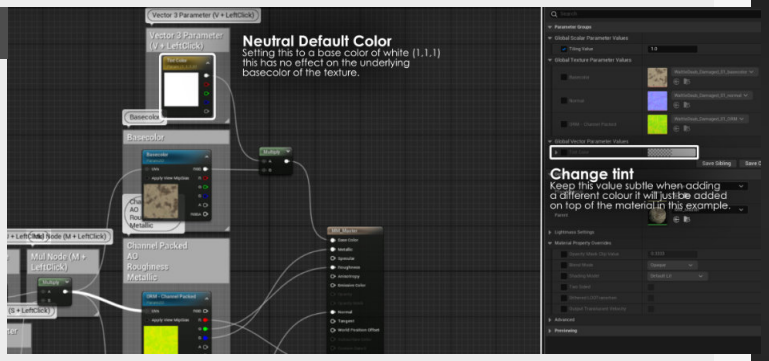


4. Adding a Tint

Now to round this section off we can add one more bonus thing to our material and that is to add a tint to our material, this allows us to add differences between materials even if they use the same texture.

For this we need to go to the basecolor section of the material and simply add a multiply before it goes out towards our main basecolor output. And then we can use a Vector 3 (Which will just give us a color) as a color on top of the main color.

However, this is a simple approach and can look weird because it affects the entire material, so best to keep this subtle.



Master Material Creation - Part 2

Introduction

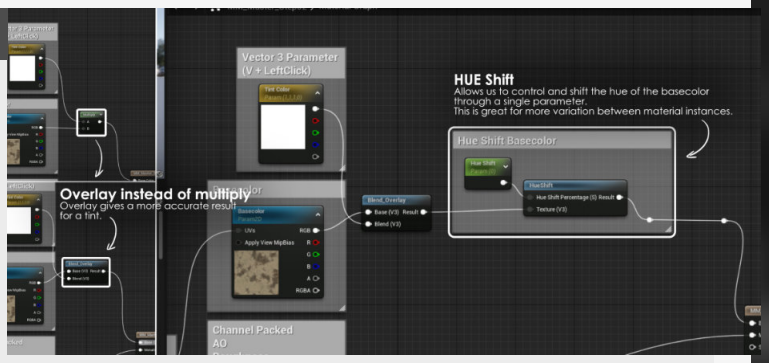
Let's take a deeper dive into material creation in Unreal Engine, in this continuation of the series we're going to be diving a little bit deeper into the master material creation process by adding a hue slider, adding normal strength controls, detail normals and look into switches which can help you optimize your materials.

1. Color Hue Shift

First Let's correct something from last week. Go back to our Basecolor sections of our material we created last week and start with a slight adjustment first, after Helder Pinto mentioned to use Blend overlay instead of a multiply.

Once we've done that we can also add the option to change the hue of the material, this is very useful to create additional color variation between assets if needed.

The node for this is actually already built in and is called **"Hue Shift"** and we can just add this after the Blend overlay node.



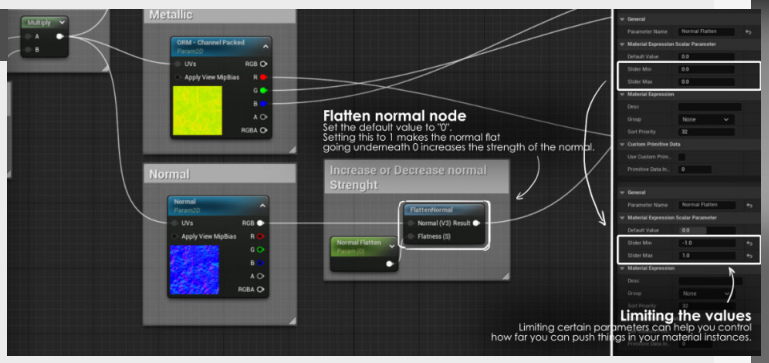
2. Normals Strength

Let's move over to normals again and add two new things to this section. First let's add the option to reduce or increase the strength of the normal map we've added.

Look for **"Flatten normal"** node and add this behind the normal map we've added.

However, you can see that if we turn this up too much then it becomes weird, so let's limit the range of this effect by limiting the inputs to -1 to 1 in the input.

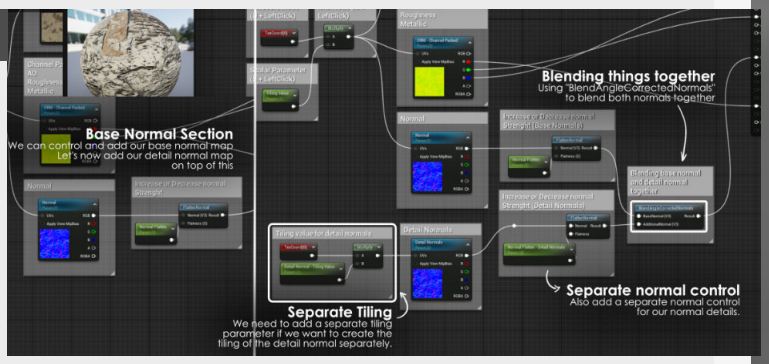
Now we can't go higher or lower than these numbers. Useful for limiting any effect you need to limit.



3. Detail Normals

A detail normal map is used to add an additional normal map to our mesh to give it the look of it being more detailed than it actually is by adding smaller details. If we're blending normals together we need to be aware we can't just use a normal blend, as the colors need to be accurately be blending together to retain the normal information the lighting needs for it's calculations.

Instead of using a normal blend we looks for **"Normal Blends"** for our case we're going to be using **"BlendAngleCorrectedNormals"**, this combines the information of the two different normals together.

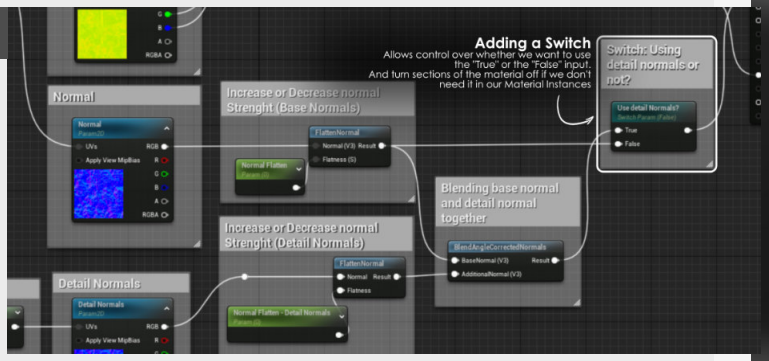


4. Switches

They allow us to control sections of our material by being able to set it to active or inactive through a checkbox we control as a parameter.

So let's take the previous section for example, by adding a **"Static Switch Parameter"** node here and then using the base normal map for the first input and then the base and detail normal combined as a second input we then get control over which section of code we want to use through our new **"Use detail normal?"** parameter.

Which also highlights that it's good to set these parameters up as a question so you know what it will do if you tick the checkbox.



Master Material Creation - Part 3

Introduction

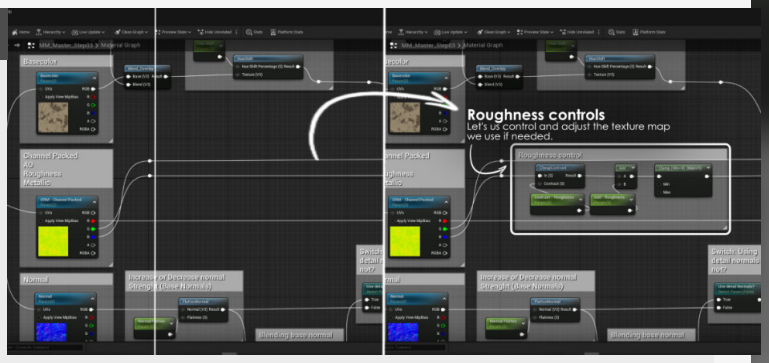
Let's take a deeper dive into material creation in Unreal Engine, in this continuation of the series we're going to be diving a little bit deeper into the master material creation process by more control for our roughness section, dive into material functions and apply some parallax occlusion mapping.

1. Roughness Controls

I love to be able to control the roughness with some manual inputs.

So for this let's go to our roughness section and add a contrast node with a parameter for control, this gives us control over the contrast. Then we add an "Add" node, this gives us control over the values if we want to add or subtract from them. Lastly, let's make sure these values stay within reasonable bounds by clamping the values between 0 and 1.

Making sure that no matter how far we push the parameters they will be within reasonable values.

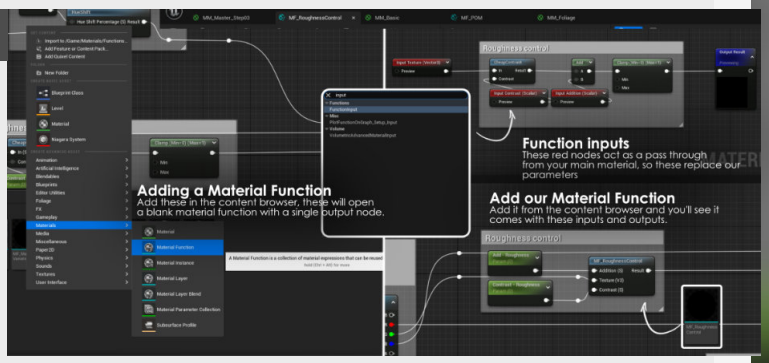


2. Material Functions

We can see that all of this becomes a little unwieldy, so fix that with Material Functions. These are groups of nodes that we can reuse between materials and condense mess.

So for this example let's take a look a the setup we just made for example, we can combine all the nodes we made for our roughness control into a function as an example.

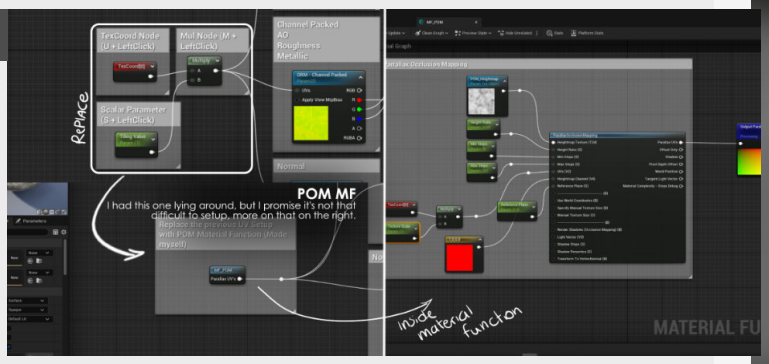
These material functions become a separate item in your content browser which we can drag into other materials if needed. If you don't want to reuse these material functions you can also select all the nodes and **"collapse nodes"** turning all of them into one node.



3. POM

I still had a parallax material function lying around so I can just easily drag that into this material and connect it to the UV's for the object. Now all I have to do is to change the input texture parameter to the map I actually want to use and I'm good to go. As for the parallax node itself, this effect will fake depth bake on a heightmap, so you will need to add a Heightmap TEXTURE OBJECT (These are different from normal textures). Then we can control how "deep" we want to effect to be with the Height Ratio and how accurate it needs to be with min/max steps.

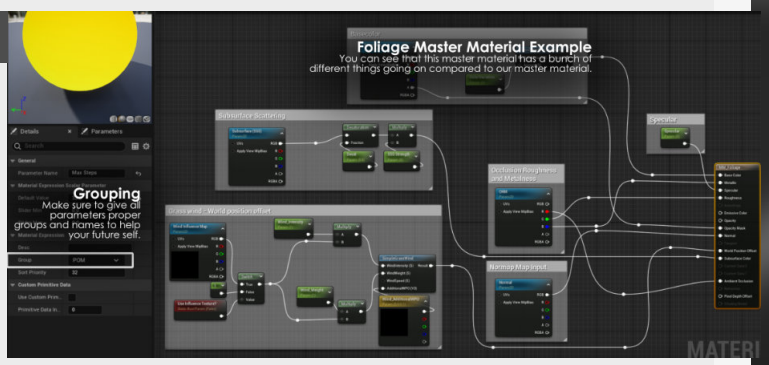
More in-depth information can be found in the tips and tricks PDF that has a page all about this setup.



4. Organization

Two final things that I wanted to mention for this series is that you can continue organizing this materials and it's parameters into proper groups (seen on the left if you select a node). This makes it way easier to find all the parameters you need by being able to look for the groups first.

And the last thing, This is a good example of a generic case for a master material, however, you will probably want to create other master materials for different applications. Such as one for Foliage, one for rocks, terrain and more. This is where you can pick and choose what you need. So start building your own!



09 UNREAL ENGINE

Unreal Camera Tricks

Introduction

Some great things can be done to improve the work you are doing inside of Unreal Engine, so for this little breakdown we will be looking into the different things you can do to improve the camera. Set some really smart hotkeys and even help you out with your compositions.

1. Locking the preview

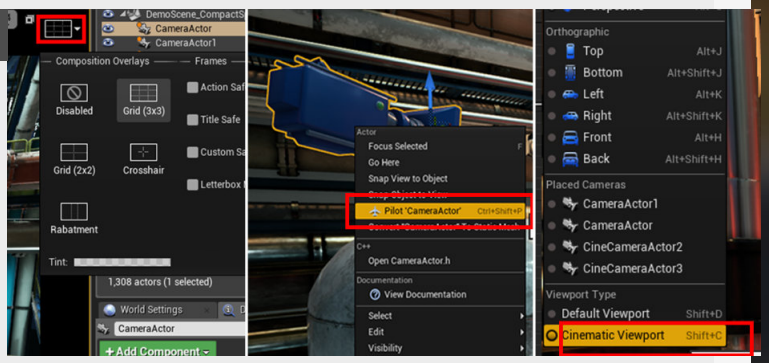
When creating and selecting a camera you can lock the preview screen using the little pin in the left hand corner of the camera preview screen, so it always shows that camera's perspective.

An alternative to this can be creating a second viewport and setting that to a camera you want.

2. Pilot camera & Composition overlays

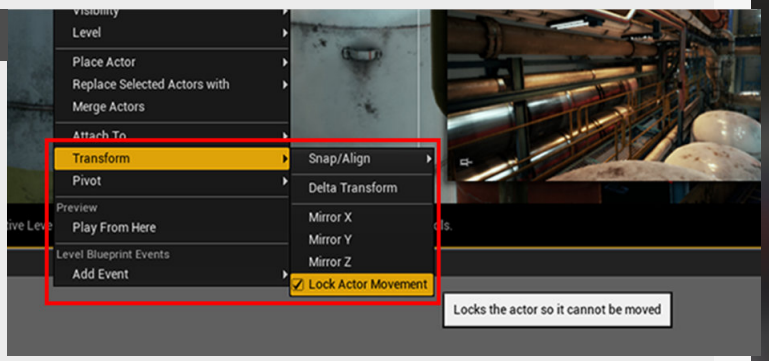
If you want to be really accurate about camera movement there is always the option to control the camera in your current viewport by right-clicking and selecting "pilot camera" or Ctrl+Shift+P.

This is super useful for finding a good camera angle, especially with the camera in cinematic mode and selecting one of the composition overlays in the top-right corner of the viewport.



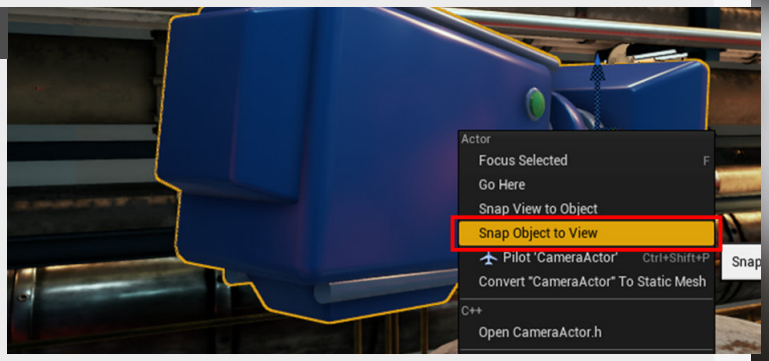
3. Locking the actor movement

When using camera's and always going back and forth it may happen that you incidentally move it from time to time, to avoid this, right-click on the camera, go to "transform" and tick the box "Lock Camera Movement"



4. Snapping objects to view

When you found the perfect angle for a shot, and you don't want to move out of it but you need a camera, you can create a camera and snap it to the current view using "Snap object to view" in the right-click menu.



Channel packed SRGB options

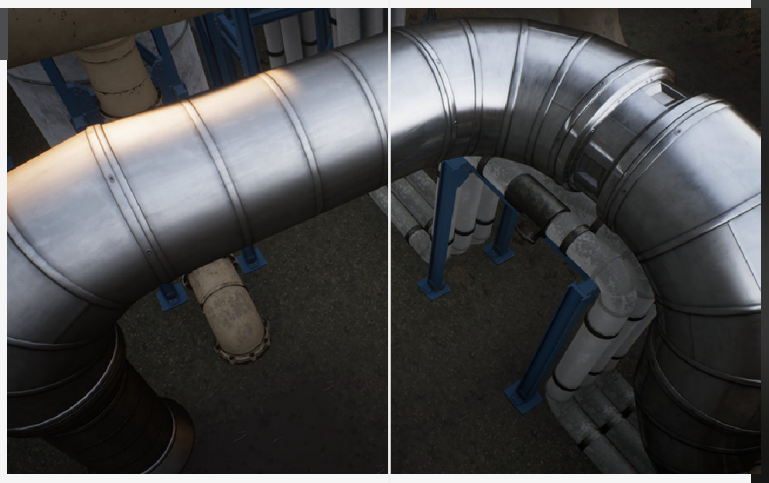
Introduction

Let's have a look at a seemingly small little checkbox that might seem harmless, but this one cost me a couple of hours of balancing and tweaking channel packed textures because they were looking different then I expected, let's dive straight in!

1. Why is this important?

This is a tiny option that is really important if you are working with channel packing textures or masks that do not need colors in their individual channels to function.

This is also an option that is often overlooked by a lot of people, including me for a certain while.

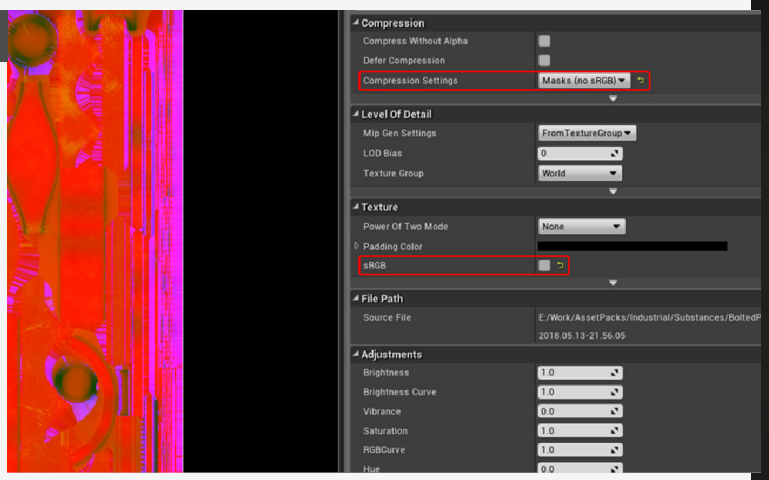


2. The difference

So you can clearly see the difference between this option being on and off in the comparison above. The image on the left is with sRGB "enabled" and the right is with sRGB "disabled".

The option is basically a tickbox to make the texture gamma corrected or not, and obviously this is not something you want on masks, etc...

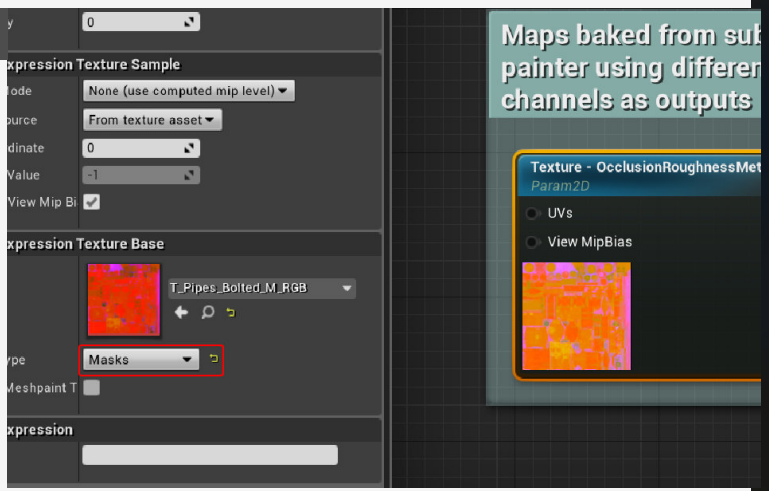
A good rule of thumb is that whenever you have a texture that doesn't need any color information at all, then you need to turn off sRGB or change the texture compression settings to "Masks" (Which will do just the same).



3. Changing settings

Knowing that we can store useful materials and blueprints in this manner I have found it useful to create a mini-library for bits of blueprints and materials that I use often.

There are obvious advantages for this ofcourse, but do keep in mind that it will break all the texture links that are embedded in the project.



Bulk Edit via Property Matrix

Introduction

This is basically one of the first tips that I did but I never took the time to translate it to the new format as the format back then was so different, but it remains a handy tool if you want to adjust multiple parameters on assets, so let's dive in!

1. When to use this

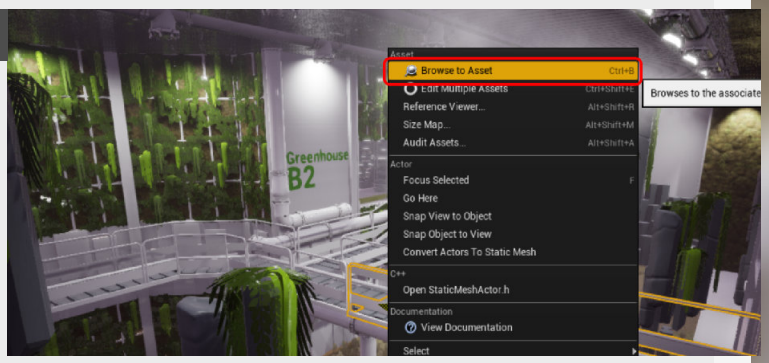
The property matrix comes in handy when trying to adjust multiple setting on a bunch of different assets that are scattered around in your scene.

So as an example this is really helpful if you want to have a quick overlook on the different lightmap resolution settings for different assets and change all of them on the go. Another example could be for instance changing the LOD group the assets belong to.

2. Select multiple assets

The tool really works best when selecting multiple assets, you will have to select them in the asset browser though, not in the level.

A nice little trick to help you select the assets that you want to adjust using the matrix is using (CTRL+B) to select the current asset or multiple assets.

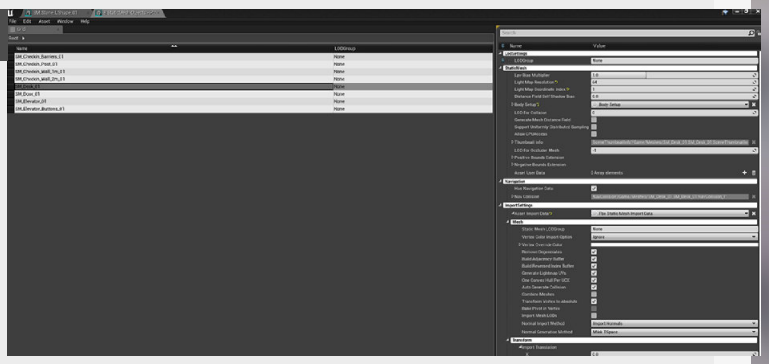


3. Property matrix

Lets dive into the tool itself, you can see that the interface itself doesn't look pretty, but it doesn't need to be because it's powerful the way it is.

Some of the things we can see in here are options such as Lodding groups, Lightmap Resolutions, if you want to import it with an offset, import it with a different normal method, combine different meshes and more...

The thing that up until this point I have found the most interesting is the adjustments of the lightmap resolutions.

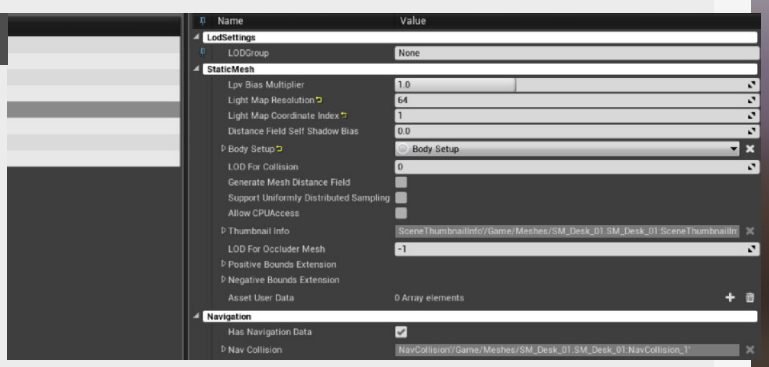


4. Limitations

However, the thing that seems to be the most interesting to adjust is missing in this list and that is the material that the assets are currently using.

But I can see why that is not in this list, because it might differ greatly from asset to asset and is probably hard to make easily adjustable because of that.

And there might be some other more specific stuff that is missing form this list but I haven't personally encountered others yet!



Material parameters

Introduction

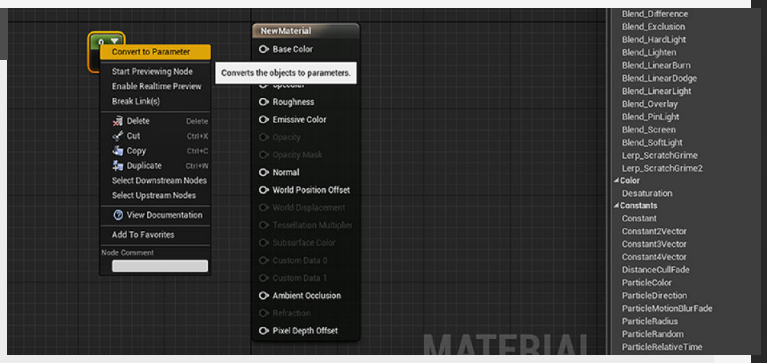
An essential step in the production of environments in Unreal Engine is the use of quick iteration and reuseability.
Material parameters help in both aspects and will greatly allow you to speed up the iteration time of materials because you can always see changes on the fly.

1. Converting a parameter

Go to the content browser and create a new material. Once inside of the material, create an constant value (1 + left click) and right click the node to "convert to parameter".

This can obviously be done not only to constant values, but also to other values you want to turn into parameters,

parameters,
for instance if you do this for a texture input, you get a texture selection menu as a parameters in the instanced material (see step 3 & 4)

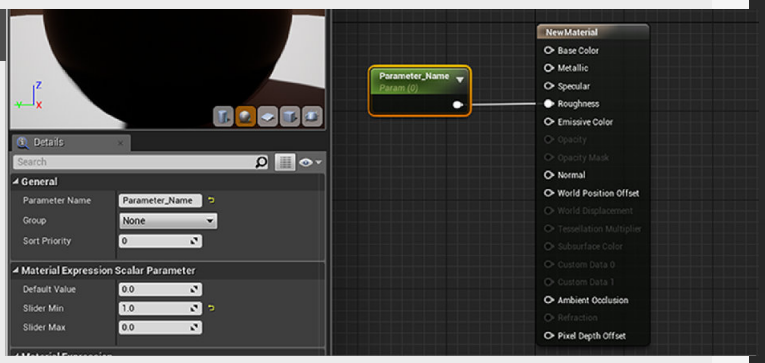


2. Names, groups and Values

You can also create a constant parameter through (S + Left Click). When selecting the node you can go into the details panel and look at the parameters to adjust such as the name and different values.

Another handy tip when you have multiple parameters is to assign them to groups, which

will allow you to
divide them in groups and make it easier to fin them in the instance.

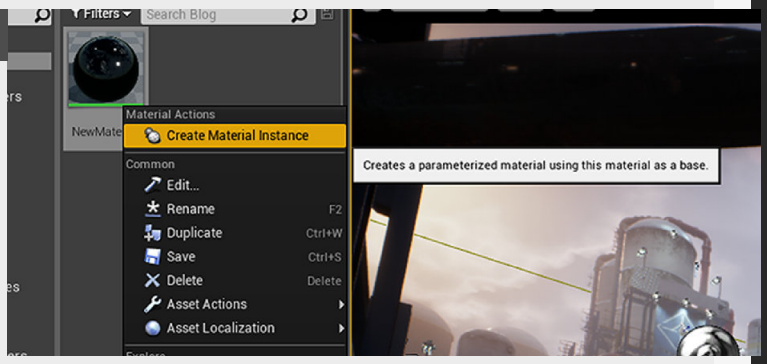


3. Creating a material instance

Once you setup the material it's time to put it into action and test it, go back to the content browser and create a material instance by right clicking on the material.

Creating a material Instance is not like creating a normal material, it does not look the same and doesn't affect the

main material nodes at all.
it does do is that it just allows you to adjust the parameters in a stand-alone version of the material which will update in real-time and thus speeds up iteration.



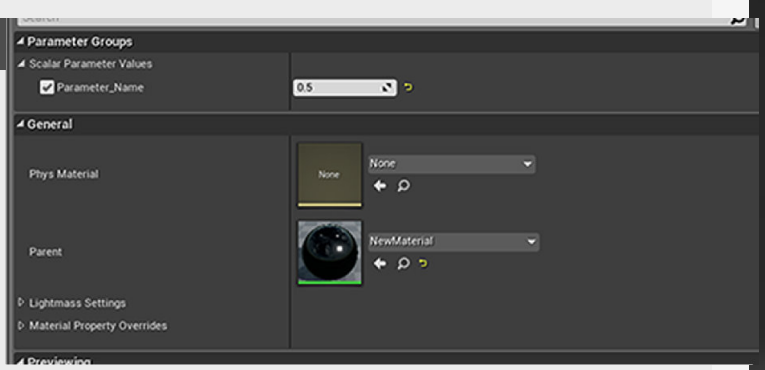
4. Finding your parameters

This material only stores the values for the parameters you have set in the main material.

This can be used on more than just constants though, you can change a lot of things on the fly if you think about it, such as a blend controlled through a linear interpolate (blending between 2 values using an

alpha value) node for instance.

A really useful trick for a lot of stuff, changing material parameters on the fly without going back into tha main material, setting up variations of different colors, etc.



Texture Tiling

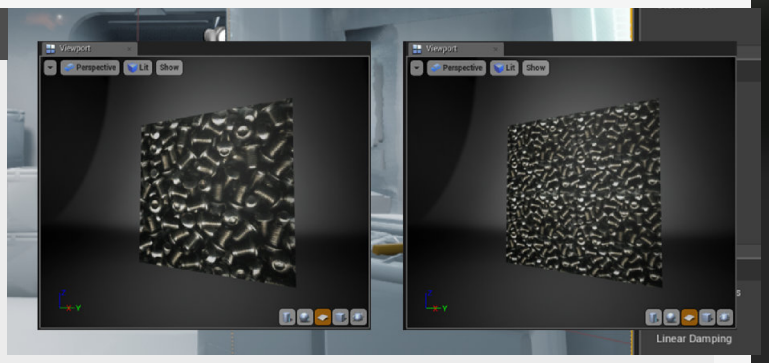
Introduction

Texture tiling is something that is super simple to setup but allows you to adjust the tiling of tileable materials on the fly through the use of this setup and exposing the values as parameters, let's have a look!

1. Let's get started

Let's have a look at a simple setup to increase the tiling of some tileable textures, super useful for Environment Art in general.

This tip allows you to change the tiling of a texture on the fly and can be useful for a lot of reasons.

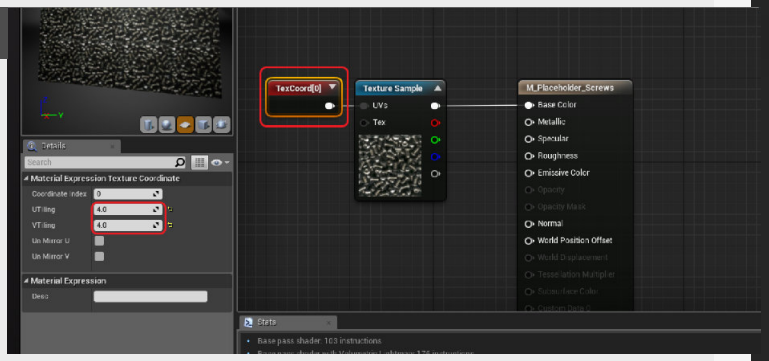


2. The simple setup

Let's have a look at a simple setup to increase the tiling of some tileable textures, super useful for Environment Art in general.

This tip allows you to change the tiling of a texture on the fly and can be useful for a lot of reasons.
Next we need a "TexCoordinates" node into the UV's input of the texture map as seen in the picture, allowing us to change the tiling.

This does feel a little bit cumbersome and cannot be quickly adjusted.

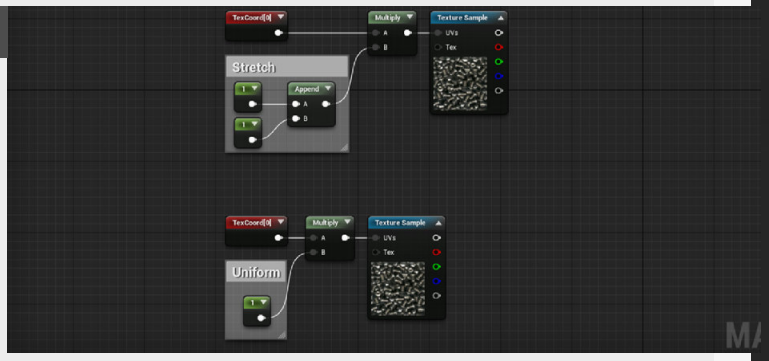


3. Uniform or stretched

As mentioned the previous setup doesn't allow for easy access or adjustability.

So what we are going to do is add a multiply node between the Texturecoordinate and the texture sample itself, in this way we can change the tiling if we adjust the value.
We can do this in multiple ways though, if we just add the value as before we get a uniform scale.

But if we add two values, append them to each other we can adjust them independently of each other and have the texture stretch if we want.

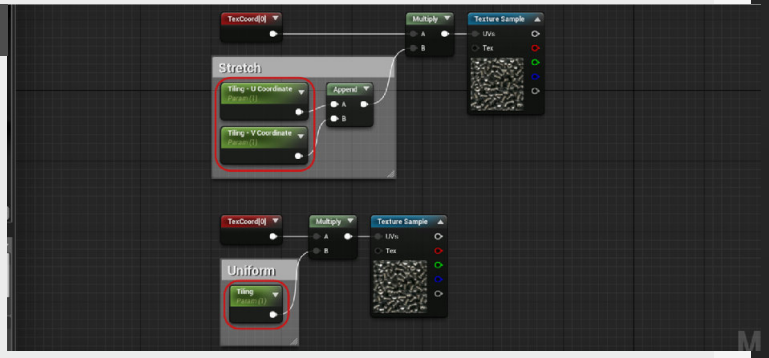


4. Parameter setup

This setup is nice but still feels a little bit of a burden and not that friendly in it's use.

So all the integer values that we added before we can convert them to Parameters, this will allow us to change them on the fly.

For this setup to work though, we need to make a material instance so that we can get access to these parameters.



Parallax texture setup

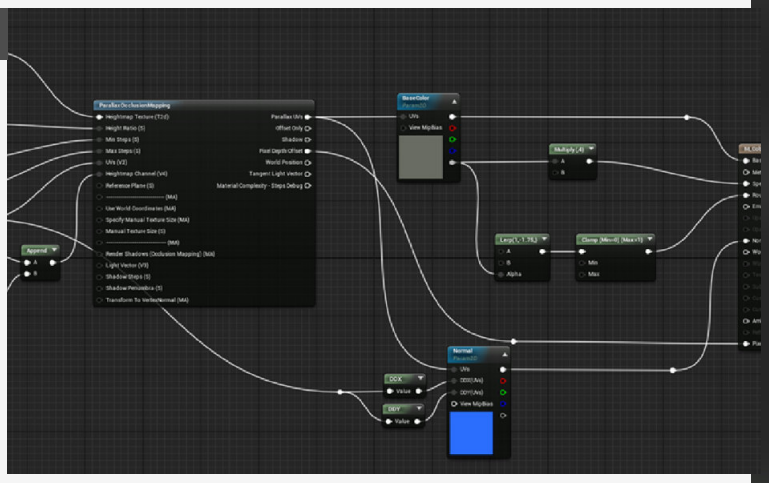
Introduction

Let's have a look at parallax textures and materials inside of unreal engine, these are really useful for simulating depth but do not work all that well when used on assets you can see from all angles.

1. The difference

You can see the difference between a parallax texture (on the right) and a normal texure (on the left). The main different is going to be the added depth that the texture has, which is what a parallax material does.

So let's have a look at the material node breakdown.

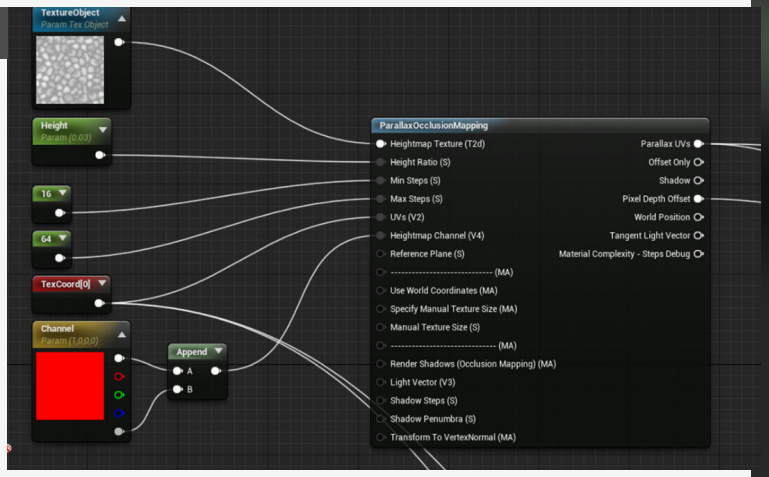


2. Parallax occlusion mapping node

Heightmap texture <u>OBJECT</u>
Make sure that this is a TEXTURE OBJECT, we can't convert this later on.

Height ratio
The height ratio determines the height of the Parallax offset itself.

Min/Max steps
This is the layers the effect will have, more stepping means the layers are less visible.
Heightmap Channel
The channel of the texture object used in step 1.
Pixel depth offset
Will offset the pixels so it accurately simulate the depth and intersection with other meshes.
Normal map adjustments.
DDY and DDX nodes are added to a normal map texture that is set to use "derivative" MipValueMode, so that the normals get accurately displayed as well on top of the parallax effect.



3. Performance and limitations

Sed vitae arcu. Aliquam erat volutpat. Praesent odio nisl, suscipit at, rhoncus sit amet, porttitor sit amet, leo. Aenean hendrerit est. Etiam ac augue. Morbi tincidunt neque ut lacus. Duis vulputate cursus orci. Mauris justo lorem, scelerisque sit amet, placerat sed, condimentum in, leo. Donec urna est, semper quis, auctor eget, ultrices in, purus. Etiam rutrum. Aliquam blandit dui a libero. Praesent tortor tortor, bibendum vehicula, accumsan sed, adipiscing a, pede. Nullam et tortor. Suspendisse tempor leo quis nunc fringilla volutpat. Donec rutrum ullamcorper lorem. Nunc tincidunt sagittis augue. Quisque lacinia. Phasellus sollicitudin. Mauris purus. Donec est nunc, ornare non, aliquet non, tempus vel, dolor. Integer sapien nibh, egestas ut, cursus sit amet, faucibus a, sapien. Vestibulum purus purus, elementum ac, luctus ullamcorper, ornare vitae, massa. Nullam posuere sem ut mauris. Nullam velit. Quisque

Unreal Engine Auto Screenshotter

Introduction

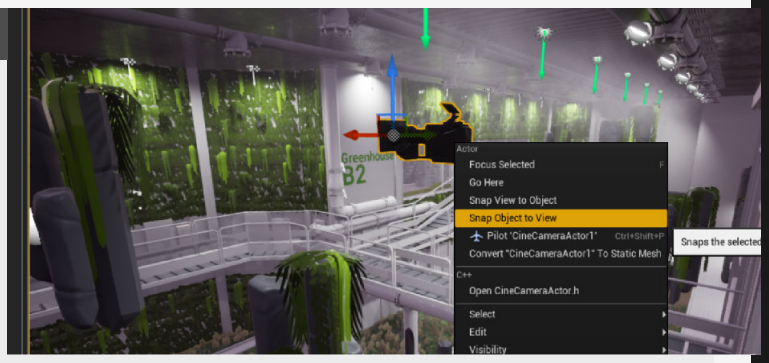
This week we are looking into a neat tool that will allow you to setup your own little automatic screenshot tool. I found this when I was looking around for some cool tricks to share on a blog by Thomas Harle which can be found here and will be linked in the blog post itself http://www.tharlevfx.com/unreal-tips-and-tricks/

1. Setting up camera's

First, we need some camera's to actually render out the images and angles that we want.

What I always use is to save the different angles to different presets that are available through CTRL+(1 to 0) to save the current viewport to that hotkey, you can then recall this by pressing (1 to 0).

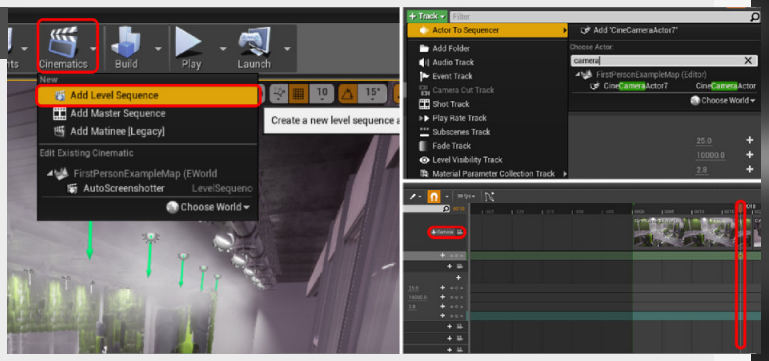
Once we have these setup, we can then add camera's to the scene go to the angle you want with the hotkey and then "snap object to view".



2. Level sequencer

Time to put them into action by adding a new level sequence, which is a timeline where you can add things like camera's, animate their position and render out video's.

Then in the sequencer we can add a track in the top left corner, select "actor to sequencer" and add the camera you would want to add. The camera is now in the sequence, but we still need to add it to the track, do this by clicking the little plus in the camera cut section.
Make sure that before you do this that the red line is on the right frame because it will add it here.

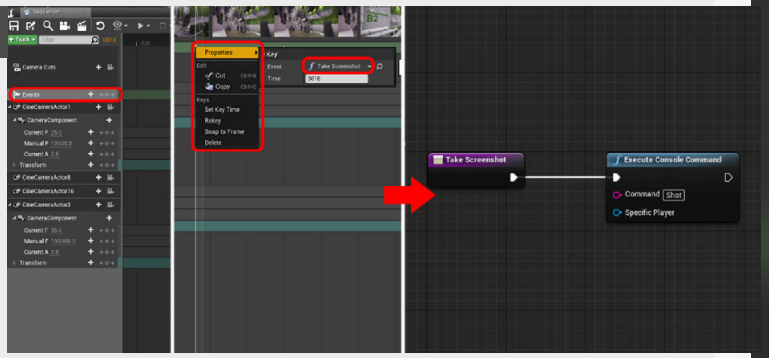


3. Screenshot event

For this we need to add an event track first (top left corner "+Track"), this will allow us to assign events to keyframes in the animation. Adding the event can be done through setting the current frame and clicking the + in the event track.

Now that we have a little event keyframe we can rightclick, go to properties and start adding our event (which are called endpoints).

This will open a new window, all we need to do here is link the input node that's there to an "execute console command" with "Shot" in the command.

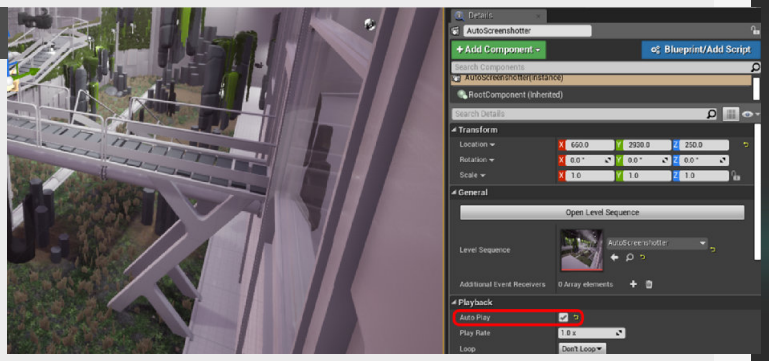


4. Running the tool

Now all the things we need to do, is let the tool auto-play when playing the game. This can be done, when selecting the sequencer and going to the details panel and ticking the tick-box "autoplay". And you should be good to go!

Some notes, when I was trying this out I had some issues with the first screenshot, so you need to first screenshot, so you need to give the sequence enough frames to render out the different screenshots.

Also, make sure to enter fullscreen (F11) and remove all game helpers (G).



Snap & Transform tools

Introduction

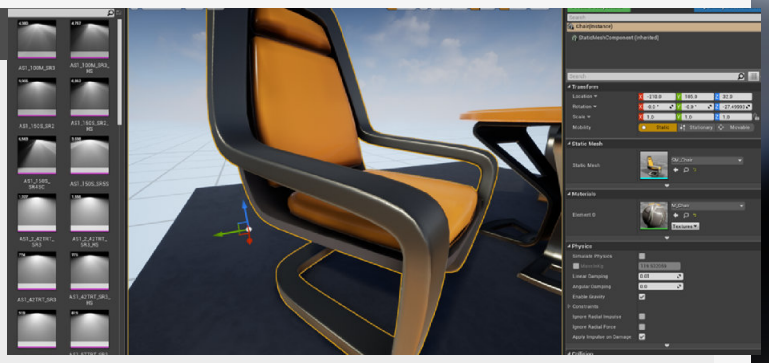
This week we will have a look at some of the transformation and snapping tools that are available inside of Unreal Engine. This will help you get the most out of making modular kits and speeding up building levels with them as well.

1. Moving the pivot

You can move the Pivot Point temporarily using Middle Mouse in the center of the Transformation widget.

This is only a temporary move of the pivot point though, if you want to adjust the pivot point in a permanent matter you need to adjust it in the 3D package of your choice.

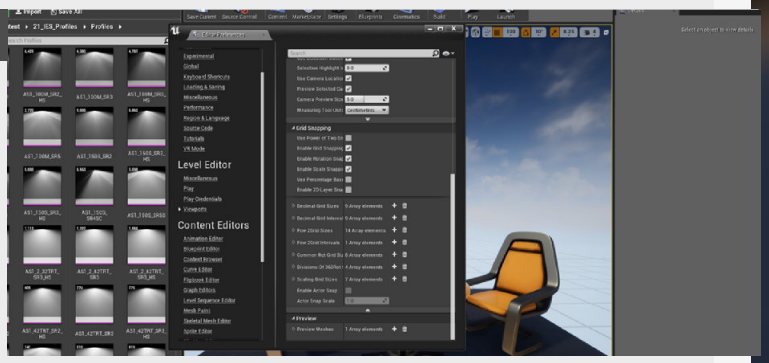
Only if it is temporary it is super nice if you would want to snap this chair to another object in the scene.



2. Custom snapping increments

You can add your own increments of snapping in the snapping options under: Edit > Editor Preferences > Viewports under the "Snapping" section.

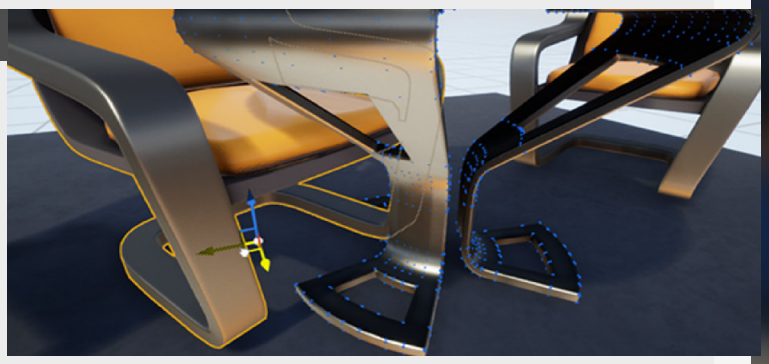
There was never a reason for me personally to do this, but I can see it being really useful if the scene or game are making requires some specific snapping settings.



3. Vertex Snapping

We can also snap using "Vertex Snapping" which then shows up with blue dots across the mesh, so vertices are needed to snap to.

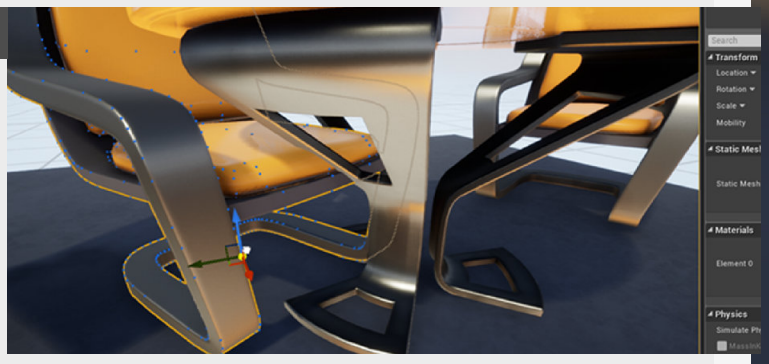
To activate this option you can simple hold "V" and then select the vertex you want to snap to



4. Combining these options

If you need to be really accurate, you can use previous tricks in combination, so set the pivot to a vertex and then "vertex snap" this to another one of your choice.

This can be really useful if you really need the accuracy of snapping one vertex to another for instance, this is super useful for modular buildings.



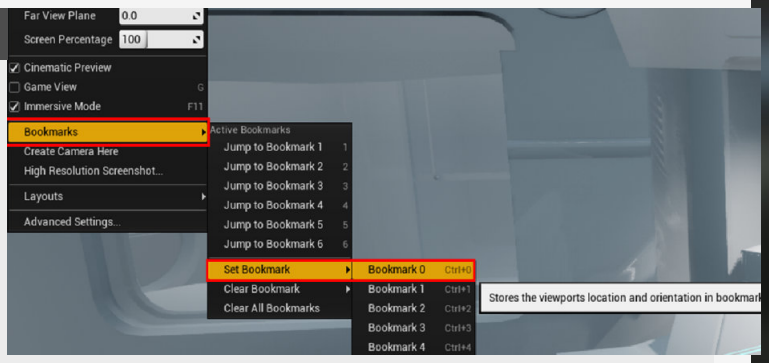
Unreal Engine Hotkeys

Introduction

This entry will focus on some of the hotkeys that can be used inside of the Materials editor inside of Unreal Engine. This will really help you speed up the material creation process inside of Unreal Engine.

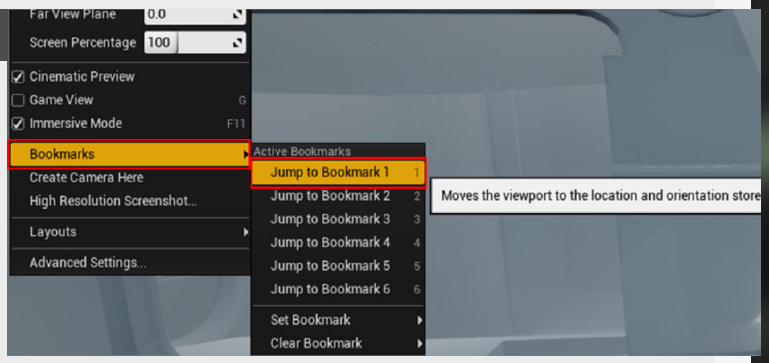
1. Bookmarks

If you have that nice camera angle and don't want to lose it, just press Ctrl + 1 (up until 0) as your hotkeys to save the current camera angle under the current number you selected.



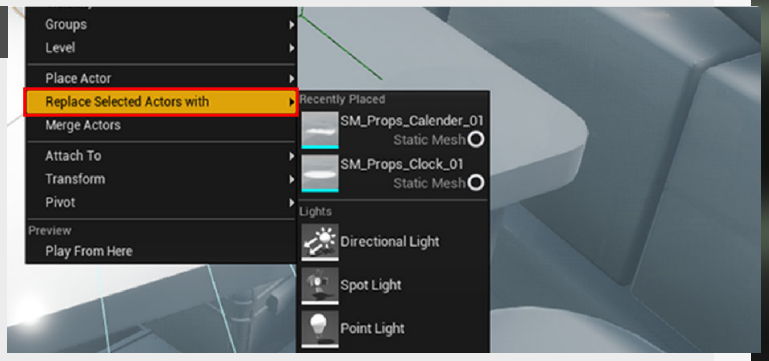
2. Switching bookmarks

You want to switch between them using the corresponding keys to where you saved them so that means 1-key is going to correspond to the one you saved using Ctrl + 1, etc...



3. Replacing actors

Having a nicely placed asset but need that different mesh in exactly the same place? Use the "replace selected actor with" function to swap out the current asset to the new one, the only thing is that it only has the most recent assets in this list.



4. Gameview and fullscreen

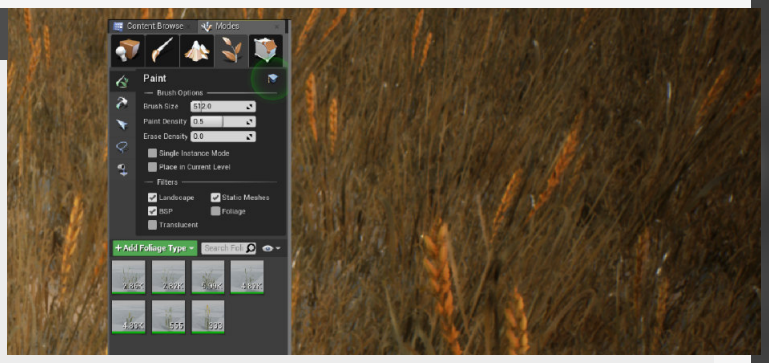
Fullscreen and Gameview press "G" to hide all the editor icons and then press "F11" to go fullscreen and go and explore without any distractions in the scene and get some nice screenshots!

Foliage editor tool tips

Introduction

We will have a look at some of the Foliage tool tips that will be useful when using the tools to spread foliage or other smaller meshes around in a scene, can also be really nice to spread other natural stuff around, like rocks and other stuff as well.

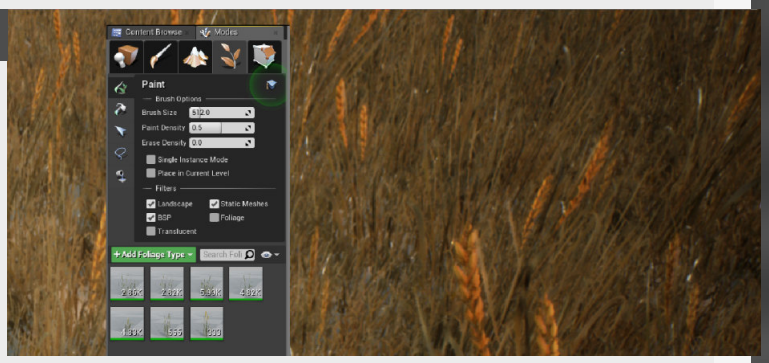
1. Foliage mode



Let's have a look at the Unreal Engine Foliage tool, which can be accessed through the "Modes" panel and then clicking on the 4th tab.

One of the first things we want to do before we can adjust any of the values, is add a piece of foliage.
This can be done just by dragging and dropping an asset from the content browser.

2. General settings



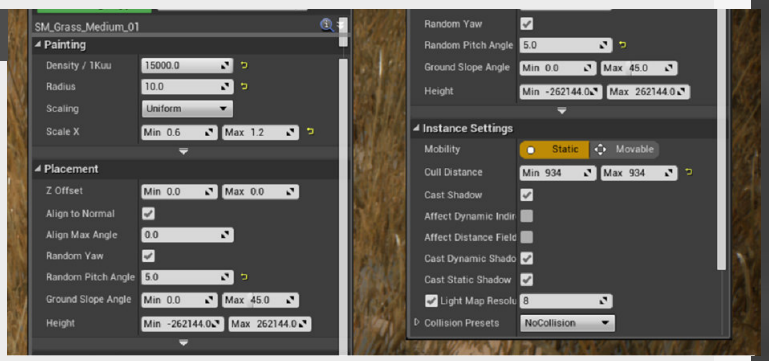
Now start to adjust all the different pain settings, the filters specifically are really useful, allowing you to only paint on a certain type of mesh.

You can also see that there are other types of different modes, such as Reapply (allows you to adjust parameters on the already painted area.)

Select and Select Lasso, which allow you to select specific areas of foliage.

Paint bucket, which is really useful for laying the base level of foliage and then moving in to add specific kinds of foliage.

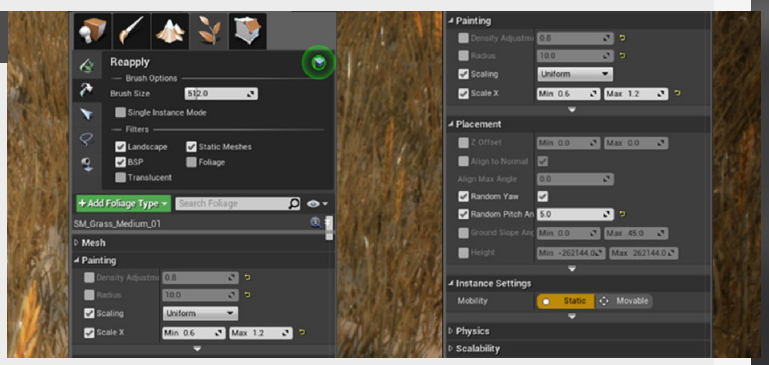
3. Individual Settings



Let's have a look at the specific settings, these can be accessed by clicking one of the assets that you added in the little preview menu below. A good way to start is to give each asset variation in scale (Scale X) and alignment (Random Pitch Angle) but keep it simple.
There are a ton of interesting settings that can still be adjusted, so experimentation is key in this step.

Additionally, you can influence the performance of the foliage by setting a cull distance and removing dynamic and static shadows.

4. Reapply settings



Made yourself a nice little patch of foliage and want to adjust the scale or density? Just use the "Reapply" tab.

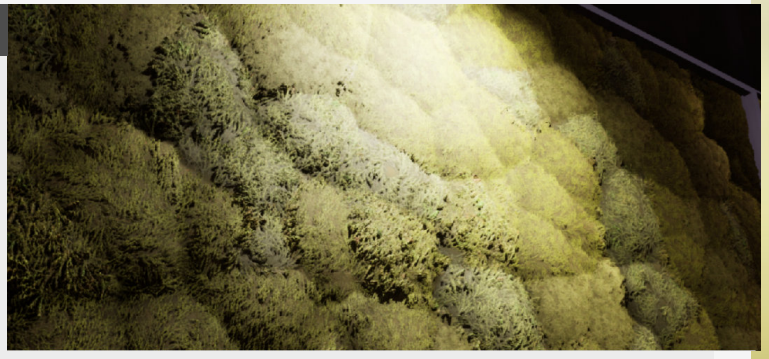
In this tab you can notice that you have the same settings as you had in the paint section, the nice thing about this section is that it allows you to isolate specific settings to reapply.

Unreal Engine Contact Shadows

Introduction

Contact shadows have been here for a while now (since version 4.14), but not that much people know about it or use them. They are a great way to add visual fidelity to a scene by adding a more accurate approximation to shadows with dynamic lights.

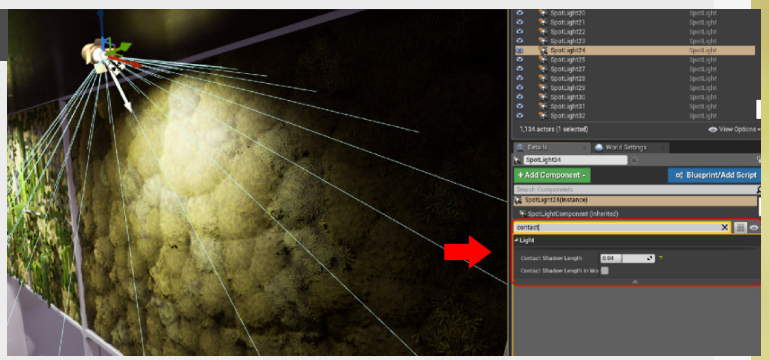
1. How and when to use?



There are some specific usecases that apply for contact shadows, they are not going to do that much if you have a lot of smaller, real fine detail on a texture for instance.

It will need some slightly bigger chunks of geometry to work with. So make sure that you have bigger chunks that the contact shadows really love to use.

2. Usage and location



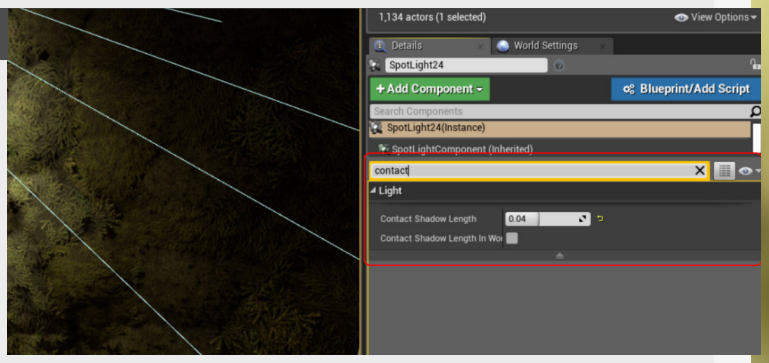
These contact shadows are available on all types of light, except for skylights.

Contact shadows and it's settings can be found inside the light panel's hidden drop down menu.

There is no checkbox, it's just a slider for the length of the shadow.

A quick way to find it is searching for it in the search bar.

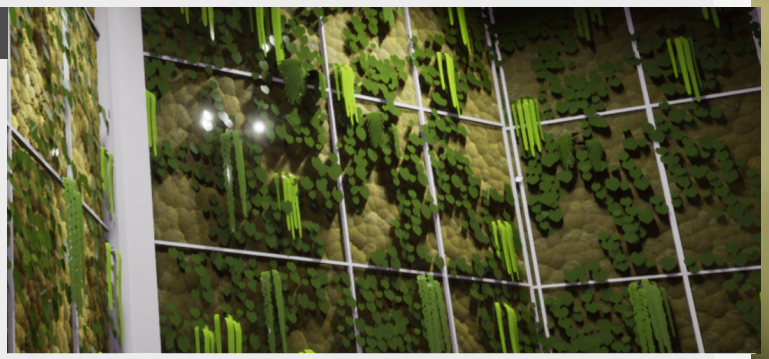
3. Shadow settings



The settings for these kinds of shadows are super simple though, you can just set the distance of the shadows. The only option that you have is changed it from screen spaced to world unit based.

The standard way of adjusting this is based on the screen, so meaning that putting this to 1 is going to cast a ray across the entire screen, where 0.5 only half and so forth.

4. Remarks



Contact shadows are really nice visually, but as most people tend to use the static lightmaps that get baked inside Unreal Engine which includes these more minor shadows.

So this is more something that is specific for people that use dynamic lights because baking shadows will be accurate as long as the lightmap resolution allows for it on static objects/ lights.

Scattering objects in Unreal Engine

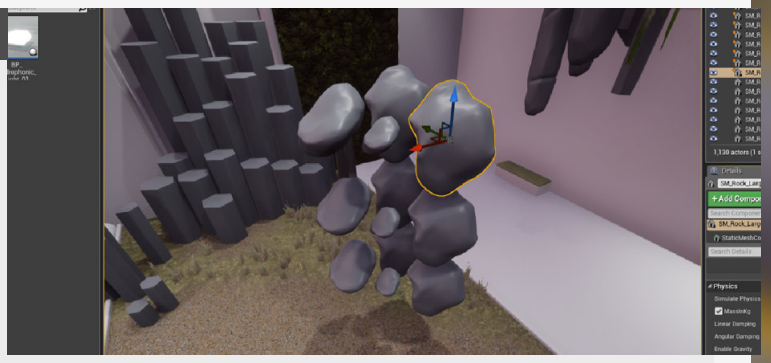
Introduction

This time we will look at an interesting way to scatter objects in your scene through the use of simulations inside of Unreal Engine.

1. Setting up the objects

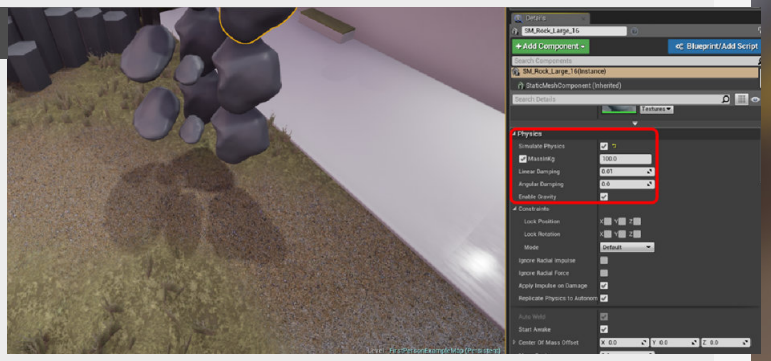
Let's dive into this neat little trick that can really help you dress up a scene in a more natural way. We will make use of simulation for this, but first we need some objects that we can scatter along the scene.

For this simple example we will use some blockouts for some rocks that I have in my current scene.



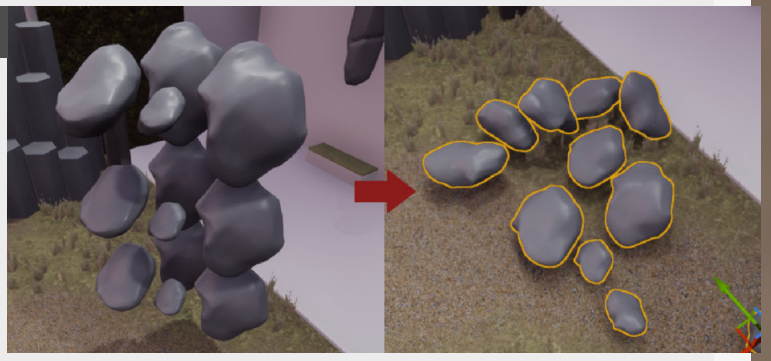
2. Checking the box

Now that we have all the meshes that we want to scatter we need to select all of them and go into the details panel. In here we will find a little checkbox under the "Physics" tab where we can "Simulate physics" once that is ticked it will behave to physics being simulated in the game The mass will already be set for you depending on the size of the object, but we can adjust this to our liking to have it simulate in different ways, but if you are looking for a quick scatter simulation, then these default options to the job just fine!



3. Running the simulation

Next up is running the simulation, in this example what we expect is that all the meshes fall down and thus get scattered around in a natural way. For this to feel more natural we can start playing around with different mass settings, like seen in the first step.



4. Converting to mesh

After the simulation is done, and all the meshes are done moving, it's time to convert them to static meshes using the "K" key or rightclick the assets and go to simulation > keep simulation changes.

Once this is done, you can enjoy the scattered meshes in your scene!

Happy scattering!

Thumbnails, Colors and favorites

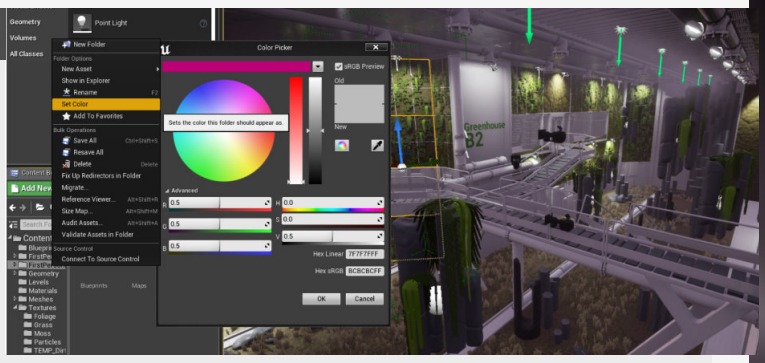
Introduction

There are a lot to learn when it comes to Unreal Engine, and customizing the way you use the program is and should be a thing that look at at a certain point. We will have a look at a smaller section of that today, talking about changing thumbnails, using colors, searching through using syntax and favoring folders.

1. Changing folder colors

You might have noticed this already, but all the assets in Unreal Engine are color coded (with a little strip at the bottom) so if we want to be consistent we can adjust the folder to follow this color convention.

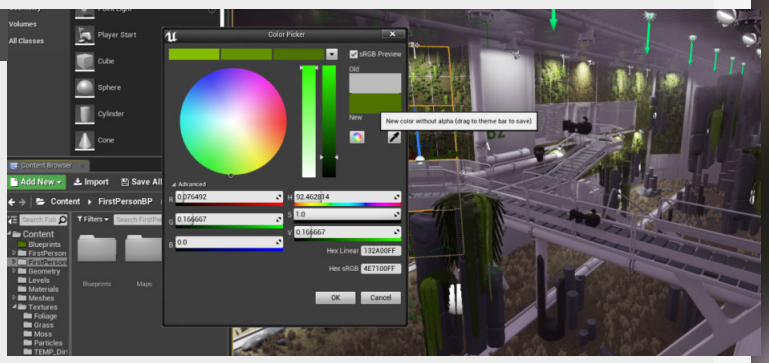
We can set the color of folders if we right click we can **"Set color"**, this is more something that is personal to you and is going to help you with quickly identifying where to look for your awesome assets.



2. Saving colors

When working with these colors using the color picker, there is a neat little trick that you can do. Dragging and dropping the color in a little bar on the top bar allows you to save the color that you have selected.

You can even create your own themes in the color picker to make it easier to pick them going forward. To do this you need to click the little dropdown menu and either duplicate or make a new theme, to add, remove or move colors within this theme you can just drag them to where you want them to be.



3. Editing thumbnails

Another neat little thing you can do is change the thumbnails from the assets. To access this you need to click the options menu and go into thumbnail edit mode first.

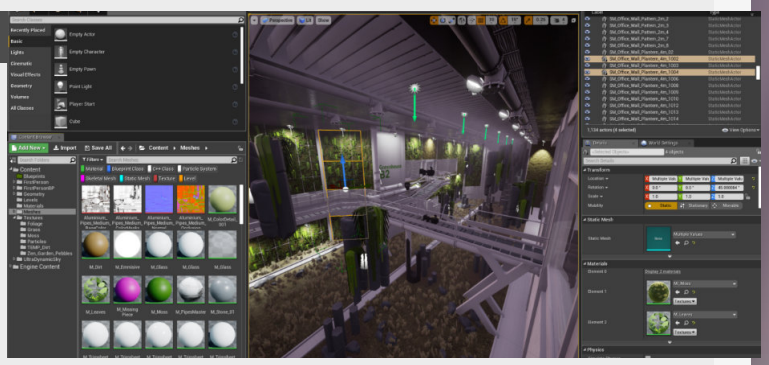
Doing this will give you access to rotate/move/zoom the thumbnails, giving you a better view of certain assets.

4. Using filters/favorites

For making searching a little bit easier you can save your searches inside of the folder structure, so you can always go back to this filter. To do this you need to fill in the stuff you want to search for so for example everything under a certain triangle count for example: Triangles >= 1500.

More commands can be found under **"Advanced Search Syntax"** on the Unreal Engine site.

Another thing that is a good trick to use is favorites, you can make any folder or even assets a favorite, doing this will make these assets or folders always pop up in the favorites tab!

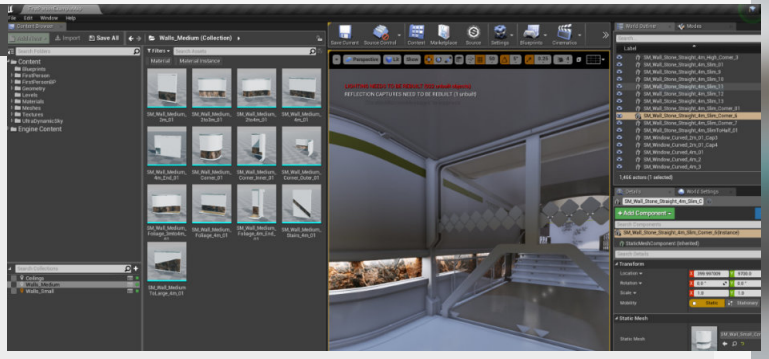


Unreal Engine Collections

Introduction

Collections are a feature that allows you to store whatever you have in your project in separate folders, making it super easy to find those assets that you are looking for, especially on larger projects.

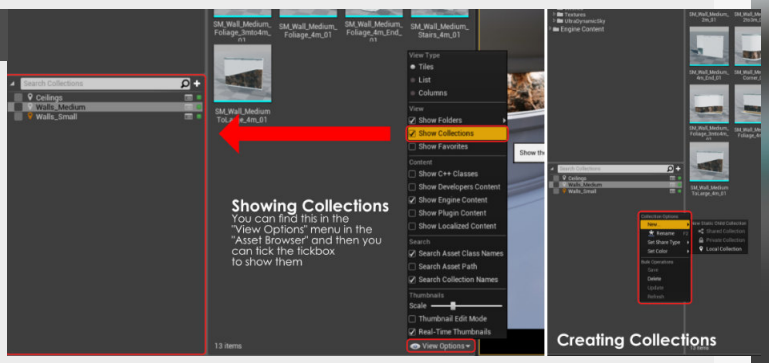
1. Collection Introductions



Collections are normally meant to be used in collaboration on big projects, especially on projects where you have a massive amount of assets to work with. But I've found them really valuable for finding assets really quickly and you don't want to change or go through your folder layout.

In bigger projects with multiple people you would put them on source control and share them with other people, I can see this being really useful even though I haven't used it that much myself.

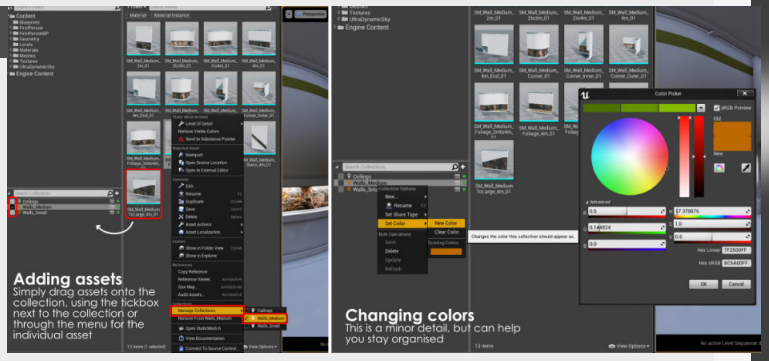
2. Showing and Creating a collection



You can find collections in the "View options" so you will first have to go into these options and enable them before you can start using them.

You can create the collection by clicking the plus button in the collection menu. We got some different options here though, Local Collections are the ones that are stored locally, which is what most people will use. Shared and Private collections are both collections saved in source control, where private being a collection that is personal and Shared being this respectively.

3.Adding Assets



When adding assets we can just simple drag and drop the ones we want into the collection or we can add them by ticking the box on the left of your newly created collection, or adding them through the right-click - Collection menu.

Additionally we can recolour the collection to any colour that you would want. This is pretty great to further personalize them, or maybe have collections of static meshes that have the same color of something, you can be creative!

You can check the status of the collection by hovering over the small colour icon to the side of the collection and it's name.

4. Removing Assets

To delete assets you need to remove the assets through the collection system menu instead of just normally removing them. If you were to remove them like you would you will delete the asset.

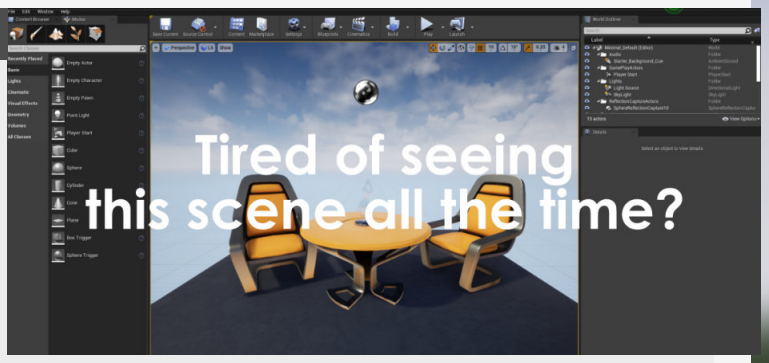
Again, if you want to remove them we can also click the tickbox left to the collections system and this will do the same as going through the menu's

Default maps in Unreal Engine

Introduction

As you might have already noticed, a lot of the time baking is just problem solving with a healthy dose of thinking ahead, so in this entry we will look at some of the common issues with baking and how we can resolve them.

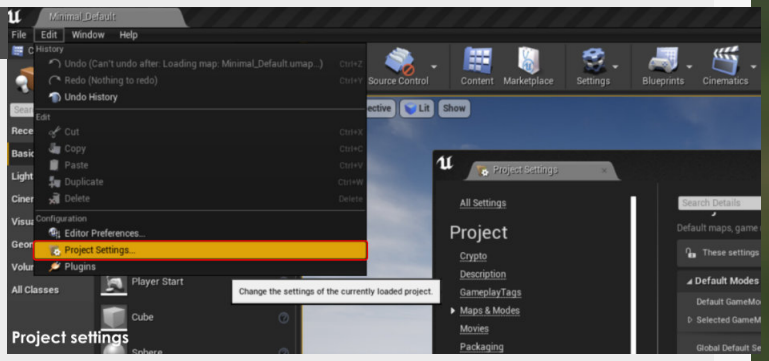
1. Tired of the standard scene?



If you have this weird issue where multiple faces are projecting onto the same UV shell and it looks all weird then this is because you probably forgot to move overlapping pieces by one UV space.

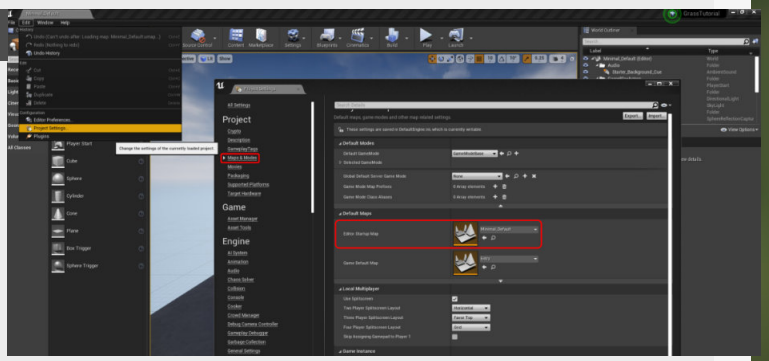
Not doing this will basically try to bake the information onto the same space twice from different surfaces on the model, where we only want this once as the UV will take care of the other side.

2. Project Settings



This seems like a smaller change to all the things you need to worry about, but it's more about the feeling you get when working on a bigger project, you just come home, sit down open Unreal and then you still have to load into a different map before you can get started. Let's remove this annoying little step!

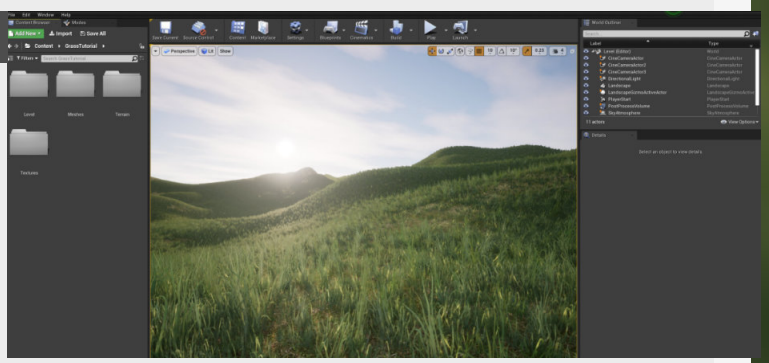
3. Picking the default map



Now for specifically changing the starting map you want to go to "Project" > "Maps & Modes" and then under Default maps you can find the option to change the "Editor Startup Map" to the level or map you actually want to load on start-up.

The option below that will allow you to control the level or map that's being loaded as soon as you run the game.

4. Other interesting sections



Another interesting section as mentioned before is the Rendering part of the Engine section, especially when trying to push certain aspects of looks in your level. Options such as the type of Anti Aliasing, if you want to use mesh distance fields or light propagation volumes can all be found here, so make sure to check it out at some point.

Saving materials to text

Introduction

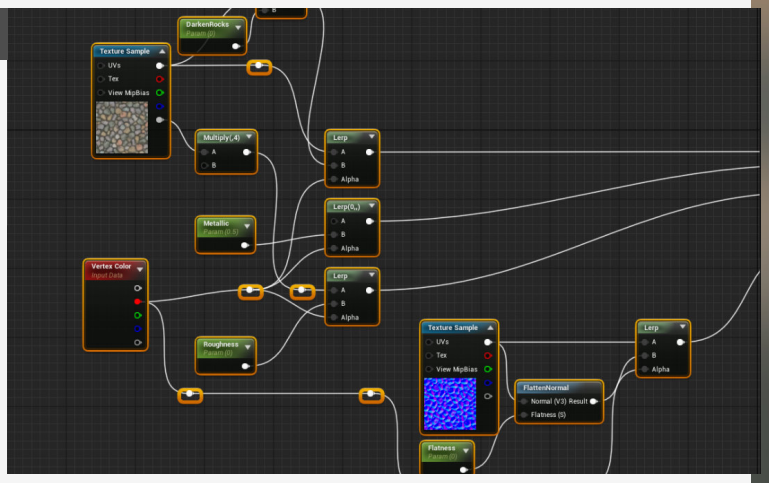
This week we look into the neat little trick that can help you send over some smaller files to people and speedup the iteration speed between people that aren't using any form of source control OR on your own projects.

1. Usefulness

This allows you to copy and past bits or entire materials to a text file. You can also do this with blueprints as well, allowing you to quickly save and share important bits of materials and code.

A really handy trick because it is so simple in it's use.

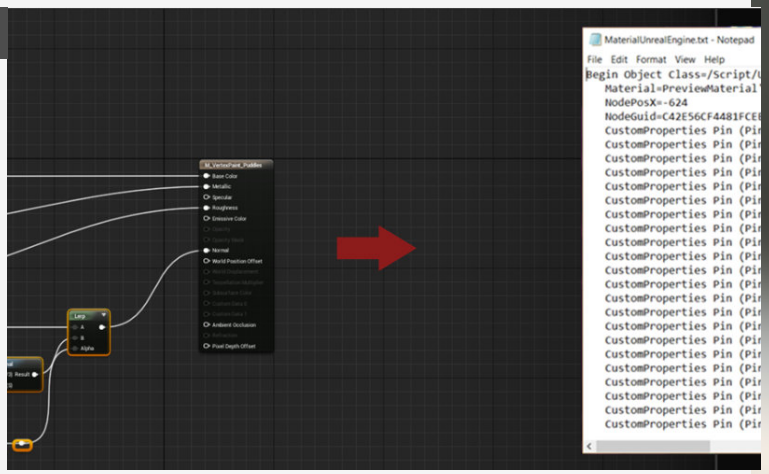
Alternatively you could also save these materials as .uassets in some place and then use them where needed in the same manner as well.



2. Nodes to textfile

When you open up a material, not a material instance, just select all the nodes and copy them over to a little text and paste in into it.

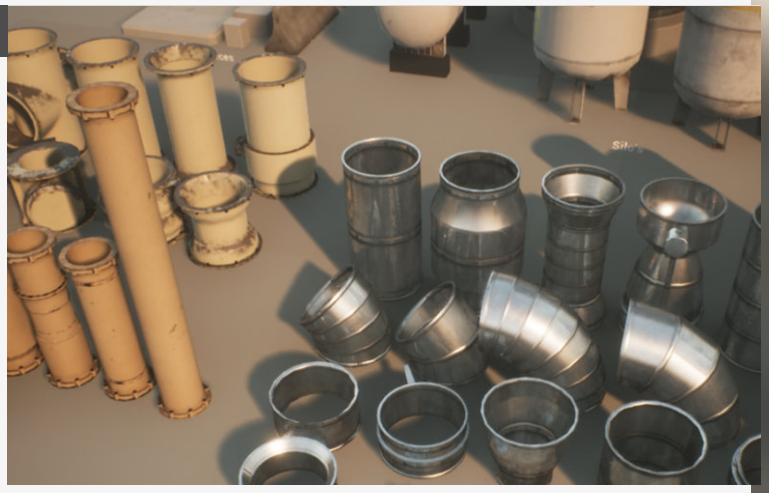
Be aware that this is not going to keep any links to local files in your project, so it's better to link them to default engine files and then replace them in the project's material instances.



3. Making an assetlibrary

Knowing that we can store useful materials and blueprints in this manner I have found it useful to create a mini-library for bits of blueprints and materials that I use often.

There are obvious advantages for this ofcourse, but do keep in mind that it will break all the texture links that are embedded in the project.



Collision overview

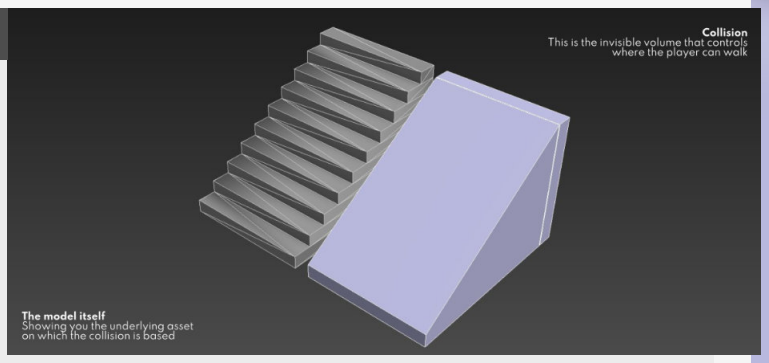
Introduction

Collisions are those invisible things that define the rules when it comes to if that object will collide with another object or not. So today we will be diving into some rules and a little on how to set them up too.

1. What is their purpose

As mentioned before their purpose is to define the Collisions are invisible volumes that get added to assets to define whether an object will collide with another. But an "Object" here has a pretty broad definition and touch many aspects in game, such as detecting where a player is allowed to walk, where a bullet hole must be spawned when a bullet collides with an object, which footstep sound will be played depending on which sound cue is added to a collision all the way to breakable or interactable objects in a game. This list can get really expansive!

But for this entry we will be keeping it to the basics and maybe at some point we'll dive into some more details.



2. Different types of collision

Simple collisions are called that because they can be a rougher aproximation for objects colliding with it such as for example moving object such as players, ai, etc... These collisions are simple representation of the geometry and should not be as detailed because they are mostly used for navigation purposes and possibly footsteps sounds affecting sound.

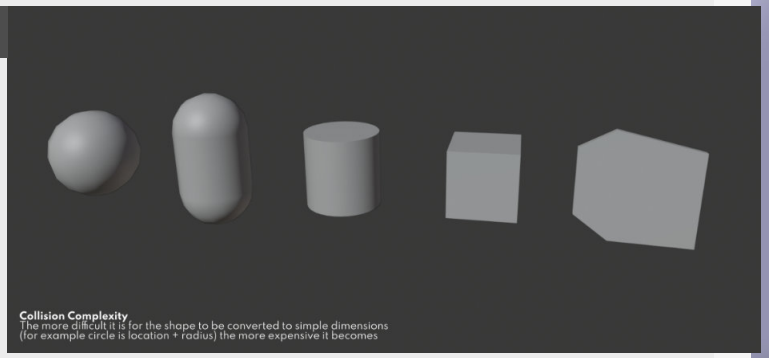
Complex collision which as the name implies is more complex, this is because it's used for more simple collision checks, such as decal spawning of bullets. This collision needs to be a bit more accurate for this reason specifically, as it will be used to spawn decals and sounds of bullets hitting it.

3. Different collision expenses

The reason why we can't just duplicate the actual geometry and just use those as collisions is because calculating collisions on these shapes is expensive as it needs to hold all the vertex information.

Think about it in terms of code, so from the cheapest object is a sphere (holding a central point and a radius) going all the way up to a full mesh.

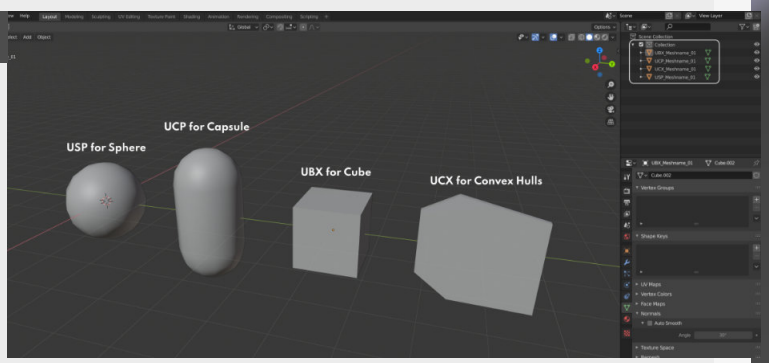
From cheapest to most expensive we get this list: Sphere, Capsule, Cylinder, Box, Convex hull(up to 16 vertices per shape), Mesh



4. Implementation in Unreal

All the assets you import into Unreal Engine will come with their own collision already setup, but for some assets this might not work for your purposes. So you can go into the asset and add your own collisions to the asset (using "Collision" and selecting the shape of the collision you want).

But we can also do this by setting it up in the modelling package you are using, by making a shape and then giving it a prefix based on the shape of the collision. Some different prefixes are UBX (box), UCP (Capsule), USP (Sphere), UCX (Convex objects)



Light debugging viewmodes

Introduction

Today we're going to have a look at something that is more relevant to an actual prodcution environment where you need to keep performance in mind, and that is what is relevant to look at when debugging lights in your own scenes.

1. Light influence viewmode

This is a neat one that I've only recently discover through Chris Murphy on Twitter, sharing a ton of useful tips and tricks! This neat little viewmode let's you check which assets are being influences by the lights in your scene.

So no more eyeballing if it is affecting it or not, just turn on the "light influences" in the "Viewmode" > "Advanced" tab, this will then show you lines between the lightsources and your selected mesh.

2. Unlit and Detail lighting mode

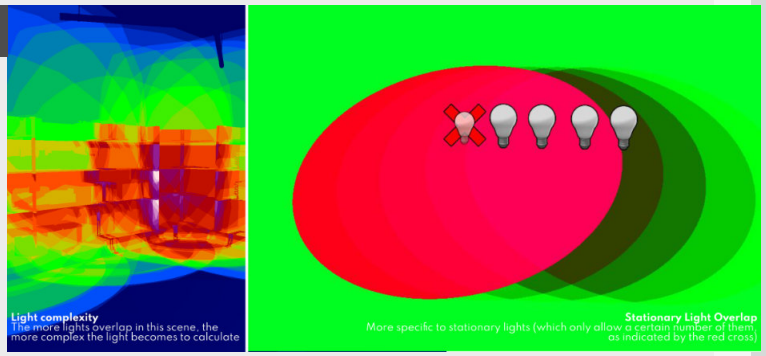
One of the most useful ones for checking the lighting without being influenced by all the other clutter that might make otherwise distract you, It fills the entire scene with a neutral material, giving you a better look if something is either of withing the values of the materials or with the lighting itself, plus it looks pretty sweet too.

The other alternatives to double check if it's your materials that are causing the issue are the "Unlit" and "Lighting only" mode, the first one will turn off all the lights allowing you to focus on the materials more then in Detail lighting. And the other one focuses purely on the lighting without any materials.

3. light overlap and complexity

Another good way of breaking down how many lights are affecting a surface at any moment is to look at both the stationary light overlap if using stationary lights, or also using the light complexity viewmode.

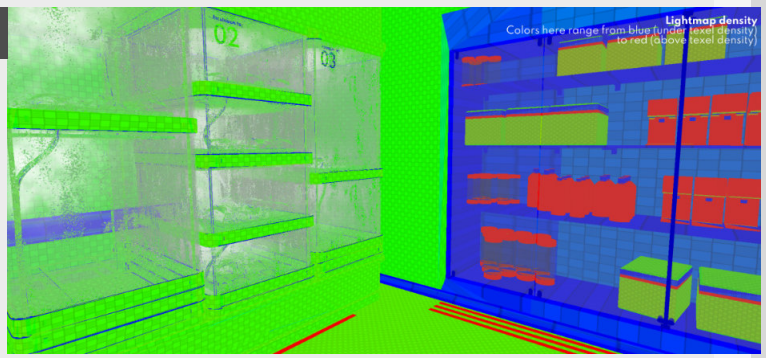
this will show all the overlapping lights in a general area by defining the overlapping sections as stronger red the more lights overlap each other, making it really easy to figure out where you can optimise your lighting.



4. Lightmap density

When working with static or stationary lights this is something you will definitely be using to calculate and doublecheck the scale of your lightmaps, on which all the lighting information will be bake down onto, so definitely need to check if they are setup right.

For this you are aiming for a nice green across the board, if the colors go towards the blue that means that the texel density is on the low side, so upping the lightmap resolution for the asset would fix this. If the colors go to the red, then you might want to lower the resolution of the lightmaps instead, saving you those long baking times and giving you clean results!



Unreal Engine Panner Node

Introduction

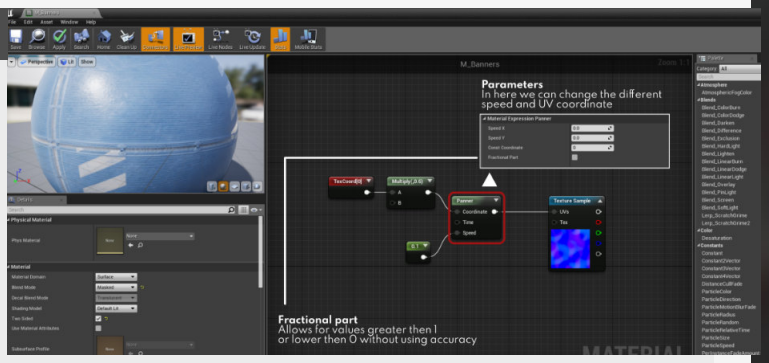
The panner node is one of the most useful nodes in Unreal Engine when it comes to creating nice movement effects inside of some materials, it's super versatile for a lot of different effects.

1. The panner node itself

The panner node is a really useful node that can add movement to your materials with a really simple setup. The panner node allows you to control the speed on the X and Y access or both at the same time with the different "Speed Coordinates". With a simple setup you can also control these through parameters allowing you to change the speed on the fly.

This panner node is really powerful for creating some interesting affects like glowing embers, scanning effects, holographics, you name it, anything with a movement effect attached to it.

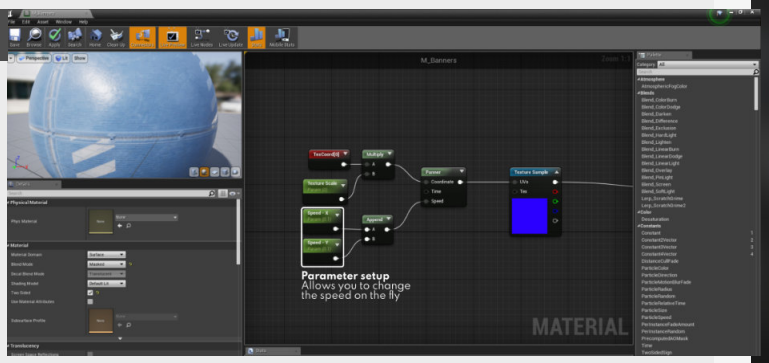
To add a panner node to you material you can just hold "P" and click to add it to your graph.



2. Basic implementation

If we look to the graph on our right you can see the panner node directly connecting to the texture sample node, we can then open up this node and change the speed of the panning effect itself. As mentioned in the previous step we can add a couple of nodes to the setup to make it work with Material instances and control the speed in real time by adding the two Parameters, connect them to an append node and then onto the panner node attached to the speed connection.

More information can be found in the material instances section in how to set them up. Or looking at the weekly tip from Blog **#012**



3. Example : Cloth movement

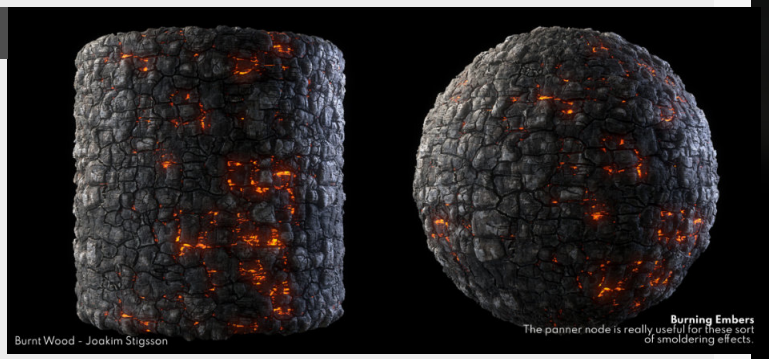
First one is an effect that I used on my last environment art piece "Last Bastion" it's pretty subtle tho. I used it there to add a detail normal on top of the cloth coverings on the fencing, so it's combining the base normal and then adding this panning normal on top of it to simulate the movement of the wind blowing against the tarp.

This effect is pretty hard to see with static images tho, I mean this goes for all movement. In addition to this we can also make a black and white copy of the texture and use it to drive the world position offset, even simulating moving vertices.

4.Example : Embers

Another different example could be the usage of this same sort of effect for glowing embers, using a cloud texture (which you can simply generate in photoshop) if we then attach a panner node to this cloud texture and use it to drive the intensity with a multiple node attached to a color like seen on the right we can get some interesting glowing ember effects.

If again, we want to push this effect further, you can use a sine node to multiple this panner emissive opacity with to create some additional flickering, obviously with some adjustments.



Material linking on Import

Introduction

This week we're diving into something that will really speed up workflows in Unreal Engine, this being good import settings. Specifically when it comes to materials, this is going to be a shorter one then some of the other ones, but it's a really important one none the less.

Plus, this more simple example also allows me to do shorter and still useful tips too :D

1. Before we start

Using FBX as our export format

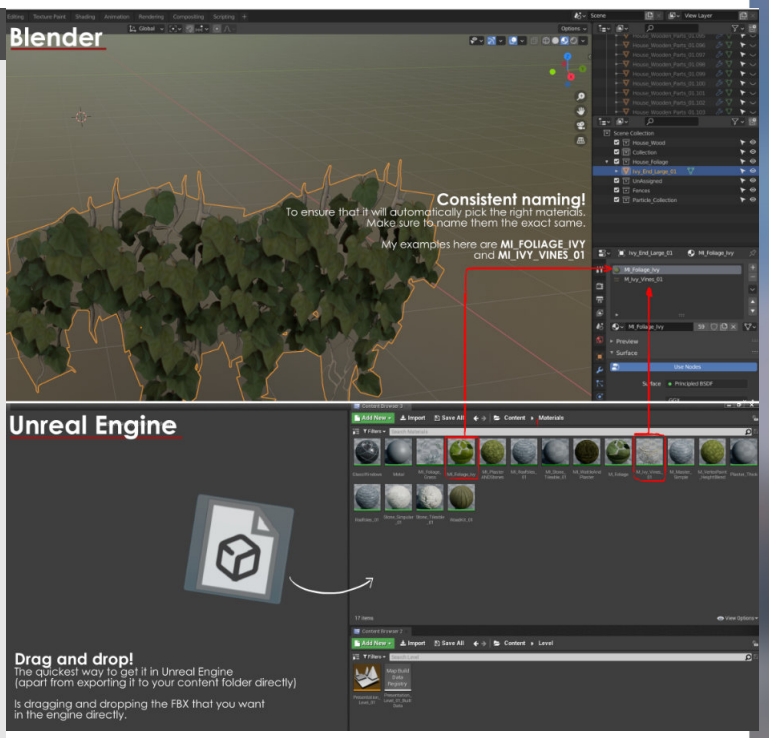
Unreal Engine uses the FBX pipeline, so this is what we will be using as our import format into the engine. Also make sure that it is FBX2018, other versions might cause incompatibilities.

Setting up materials first

For this to work, we need to have materials that the importer can pick up upon. So in this project I already have some materials setup.

Consistent naming

The main thing to check here is doublechecking that the naming of the materials is consistent between the materials that you have setup in your modeling package and how it's called in Unreal, it will look for that specific name.



2. Importer options

FBX Import options

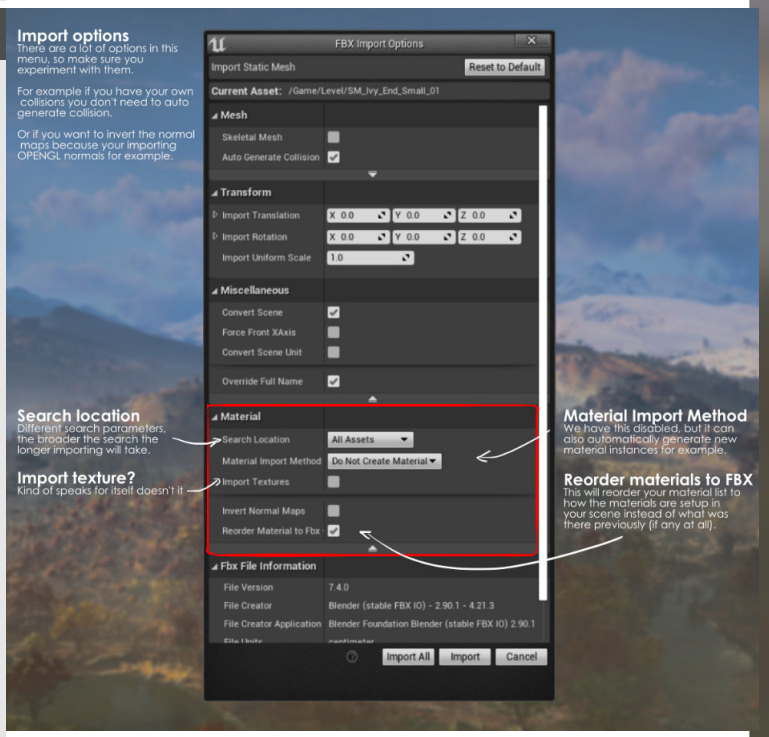
For this one we will be specifically looking at the **"Material"** section.

First off all, if we have all the materials setup already, we can disable the option to **"Import Textures"** and also set the **"Material Import Method"** to **"Do not create materials"** normally this is on and always creates a mess.

Search location

Now the last thing we need to define is the search location, which defines where the importer is going to look for materials it can add to the meshes you are importing, for ease of use I have this set to all folder and might slow down the searching process a little, but it's barely noticeable on my end

And that's it, super quick setup to really streamline your process getting meshes into the Unreal Engine.



Quick destructible meshes in Unreal

Introduction

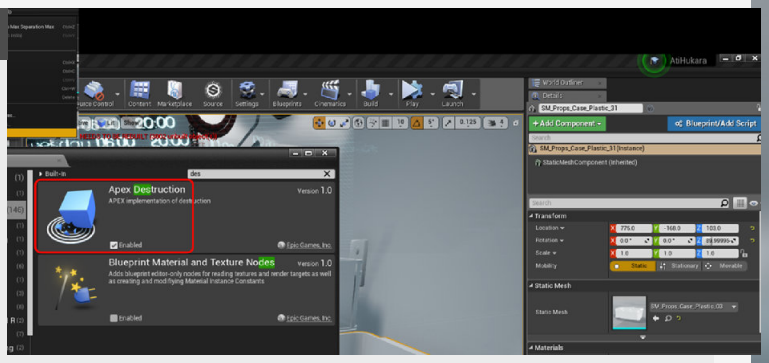
Looking to get some quick destructible meshes inside of Unreal Engine, then let's look into how you can use the fracture tool to create some destructible pieces from the meshes you feed it.

1. Activate the plugin

Let's talk about destructible meshes, normally this was an option in mesh settings, but it moved to it's own separate plugin "Apex Destruction".

To activate this plugin, we need to go to Settings > Plugins and search for "Apex Destruction" and tick the little activate tickbox in the bottom left corner.

You will probably need to restart the engine once you have done this, so that you can use the newly activated plugin.



2. Converting meshes

When restarting Unreal Engine we can can now see that the plugin is active if we go back to the plugin menu.

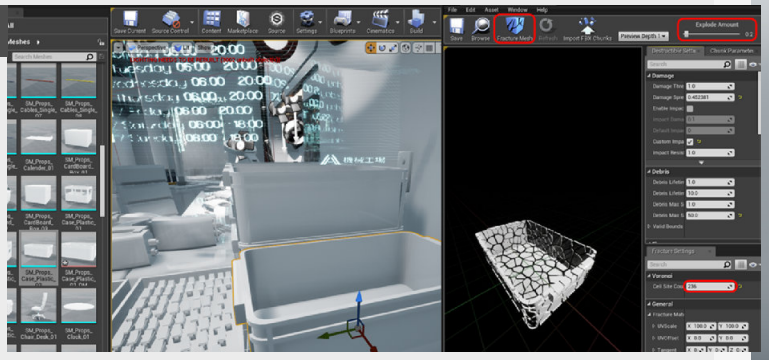
Now if we want to convert a normal mesh to a destructible mesh, we need to right click on the mesh we want to convert in the content browser and click "Create destructible mesh".

3. Fracture settings

Once we click the "Create destructible mesh" a new menu will popup.

Settings to note:
Cell site count (under voronoi settings) This changes the amount of debris chunks the fracture mesh is going to have, but you might notice that the mesh itself doesn't change, this is because we need to process it first. We do this by pressing "fracture mesh".

Other settings such as importing your own chunks can be found right beside the fracture mesh button as "Import FBX chunks"



4. Need more control

This is a quick overview for creating some destructible meshes for testing purposes, but if you need more control over the way you destroy your mesh, that is where you need to go into your favorite 3D package and model it as separate chunks and import them through the previously mentioned way with "Import FBX Chunks".

After you added your own chunks, you get the option to go into the "Chunk parameters" menu and adjust all the separate values.

Mesh Distance Fields

Introduction

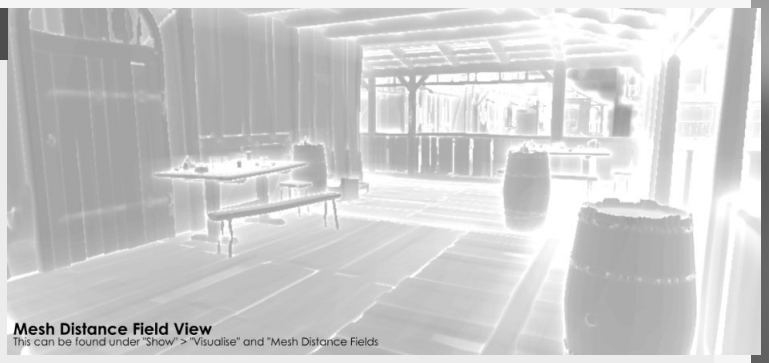
Mesh distance fields are becoming more important when creating bigger scenes where you want to have representation of dynamic AO, Global illumination and Shadows on static meshes. And have become even more important with UE5 and Lumen. So let's dive into what they do, how to get them and tweak them too.

1. How do they work?

First of all, I'm just a simple artists, so don't expect a full technical explanation here.

In it's core and what matters for us is that these are a simplified representation of your meshes that are particularly useful for real time ambient occlusion and shadowing for static actors in your scenes.

This is also why we're talking about it now, with Unreal Engine 5 coming up it will be important to understand these a bit better as they are used for the dynamic reflection and shadowing system that's used by Lumen, and now replaces SSGI and DFAO.

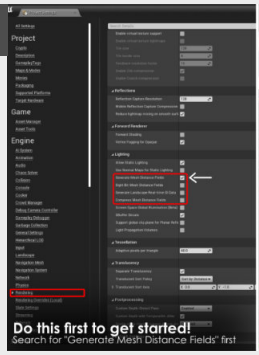


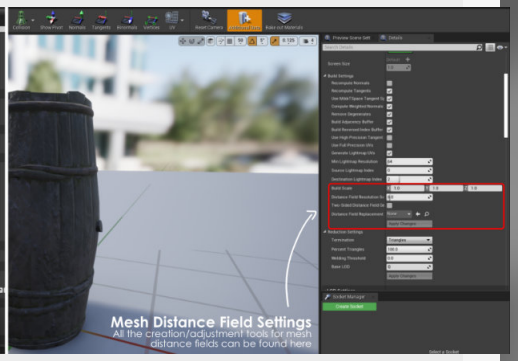
2. How do I make them?

So first of all, how do we get them in the first place?

Check "Generate mesh distance fields" in the project settings before you get started with generating them at all. Then once you've restarted the Engine it will automatically generate them per asset basis.

You can double check this by going into the debug view in your viewport [**Show > Visualize > Mesh DistanceFields.**] and also by checking per asset if there is now a little line on the top left that mentions how much memory they take up.

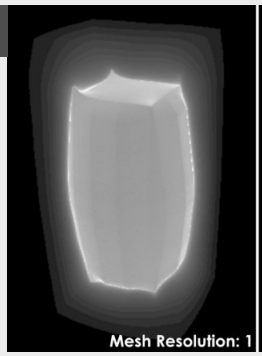


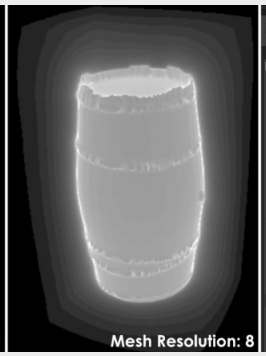


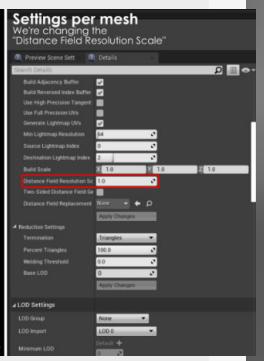
3. Tweaking them

The main tweaks we will be making here are by going into the individual mesh settings and then under **"Build Settings"** we can find **"Distance Field Resolution Scale"** which is the main thing we can adjust to make the distance field more accurate, but there is a limitation of 8mb per asset or a resolution of 128x128x128. But for most usecases that should be more then enough.

If you need a more specific mesh in there tho, you can always import your own. And for smaller or thinner surfaces you can always tick on the **"Two-Sided Distance Field Generation"** on, but this goes come at a significant performance cost.







4. Other implementations

Foliage assets can also make use of this, but they are usually more expensive compared to normal assets because foliage relies on smaller and often thin surfaces, so needs "double sided field generation" and you can also enable this "Affect Distance Field Lighting" on individual instance setting in the foliage tool too.

Global Distance Fields are also something that gets generated with the initial generation of Distance Fields which are a cheaper version and thus allowing you to use it for everything, but I will leave this for the more technical minded.

Different collision types

Introduction

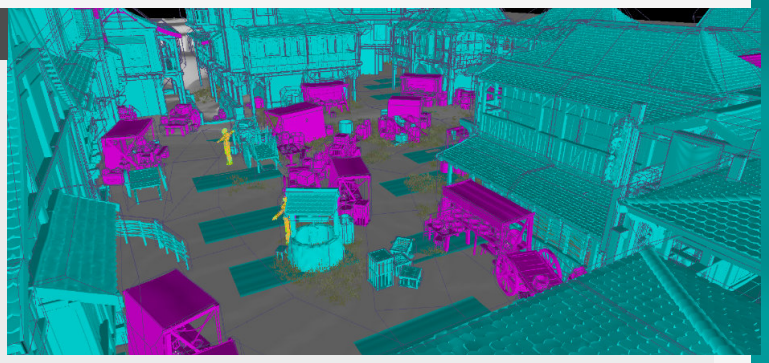
As an Environment artist you will also be in charge of creating collisions for your meshes too, but usually there are multiple purposes for when we need to do them and they all have different or even multiple usecases.

1. Why we use different collisions?

So first off, why do we use the different collisions in the first place, can't we just use the most complex collision and let it do all the interactions we need them to do?

Well, in theory we could, if we had unlimited performance bandwidth, but with all the complex calculations that are happening every millisecond already, it's really about optimizing every aspect of the game and complex collision detection can be really demanding.

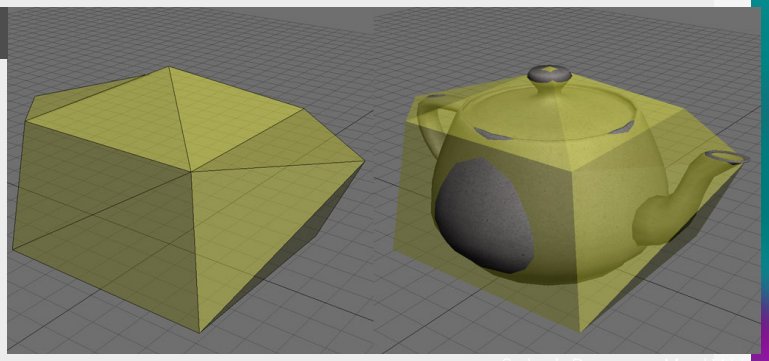
This is why we distinguish between simple and complex collisions and the meshes we use for them as well.



2. Simple Collision (SCOL)

This one is the simple collision of the shape we're creating, this is used for player and AI navigation, so it doesn't need to be super accurate as the player and AI are usually bigger objects.

The simpler the shape the less expensive the collision detection calculation will be for the object. Some simple shapes that are used frequently are a box, sphere, capsule up until a convex hull (with limited amount of vertices in some cases)



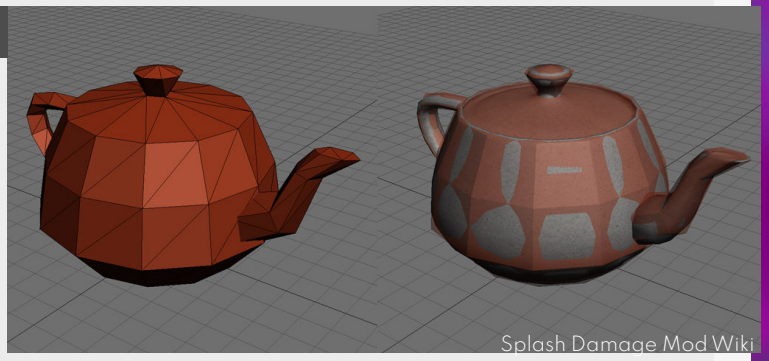
Splash Damage Mod Wiki

3. Complex Collision (CCOL)

These are used for more accurate, complex and more expensive calculations, like the spawning of bullet decals, sounds and more. You still need to optimize them tho!

A simple approach would be to literally duplicate your visual mesh and use that as collision mesh, but this would quickly cause issues with the performance of the game you would be working on, this is why we optimized version of the art mesh for these.

The collision material will usually be changed depending on the material, this controls which decal and sound it will spawn too.



Splash Damage Mod Wiki

4. Specific setups

Each game can deviate from these settings and use collisions in a different way, depending on how accurate they need to be and how close the camera will be compared to the assets that will be colliding with each other.

For example in a top down strategy game the collision setup doesn't need to be as detailed as a first person open world game.

So make sure to always test your collisions in the game itself, to get a good idea on how they will be used and might need to be tweaked as well.

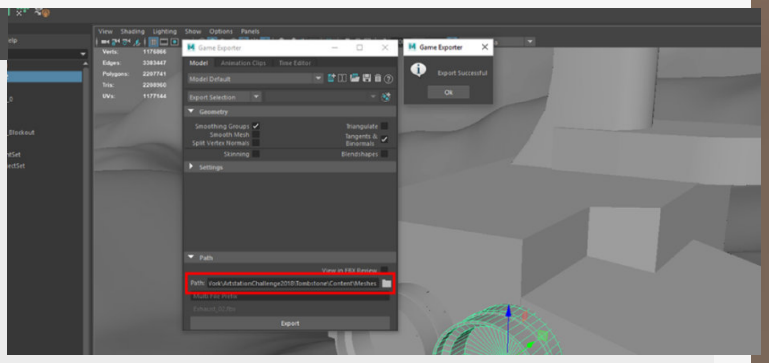
Planet Coaster in-game image

Importing & Exporting

Introduction

If you having issue with manually updating all the assets in your scene it's time we look into the following options for optimizing the workflow to allow you to do more and quicker iterations.

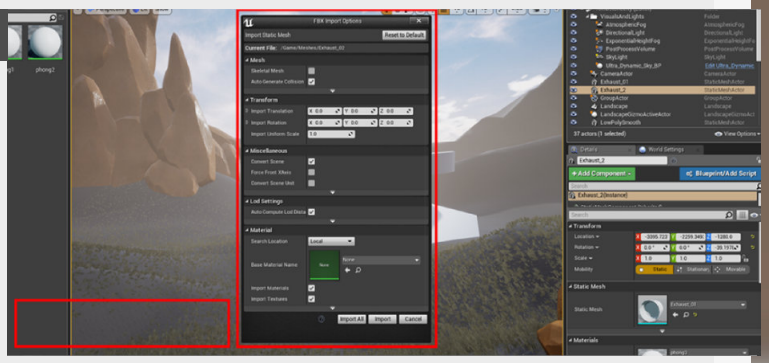
1. Maya game exporter



This is a feature I only started using recently, and for that I am a bit ashamed though because it feels that everyone is using it and I was just lagging behind.
The feature I am talking about is setting up an import folder where you export all the meshes to and then Unreal Engine Automatically imports them into the current project.

I use Maya's LT game exporter for this feature but this can be done with any exporter of your choice (so don't mind the screenshot too much, this is my exporter popup), so I just set the folder path to a folder inside of the Unreal Engine Project and export it.

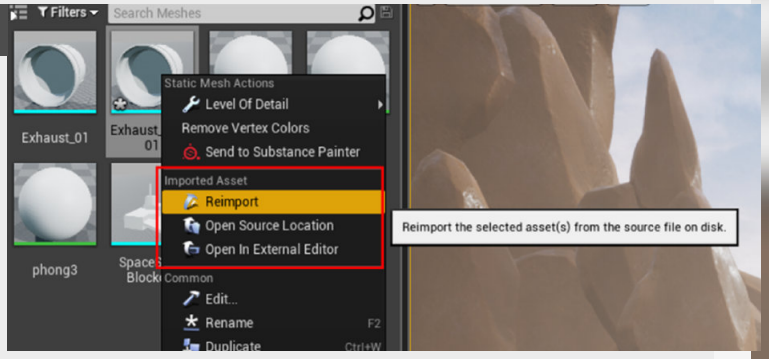
2. Import settings



If you prefer doing it the old fashioned way or have setup a scene already, there are other ways to speed up the workflow, instead of going to the folder or looking for the asset.

We can just use the reimport feature, which just updates to the newest version of the exported fbx.

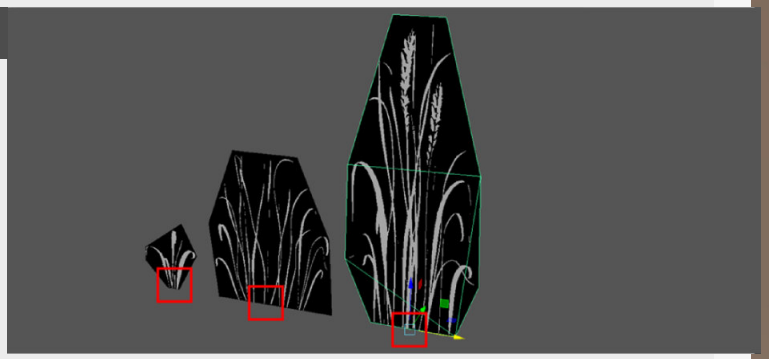
3. Reimporting



When using the features above it is important to have logical pivot placements for all the assets that you create, this helps you greatly when updating a scene or updating a blockout.

Normally when you have setup your Maya output folder for the meshes, Unreal should automatically detect these changes and update the assets accordingly.

4. Pivot setting



Another important thing is that when you adjust your mesh in your 3D program you don't change the name for the assets, if you do so it is going to create a new one instead of overwriting the previous version of the chosen asset.

A simple example is seen on the right where you can clearly see the pivots being in the center at the bottom of the grass meshes.

Depth of field in Unreal Engine

Introduction

Today we're diving into depth of field, a really good way to pull attention and focus to a prop and give a nice sense of depth to your scene in general, while specific to certain scenario's it can really enhance the feel of your renders.

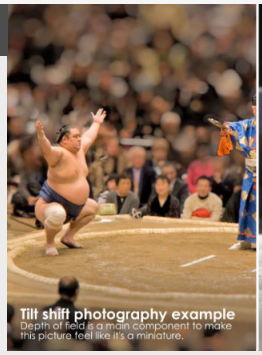
1. Overview

Depth of field is a good way of emphasizing the depth of your scenes, it will blur and reduce the noise in the area's that are out of focus, which in turn pulls extra attention to the area that is in focus.

Which is also why I personally love to use it, it's amazing for dividing your artwork in different layers. As in Foreground, Midground and Background layer.

It can also be used to give the picture a different feel, like for example if we look at tilt shift photography which gives the impression that you're looking at a miniature world.

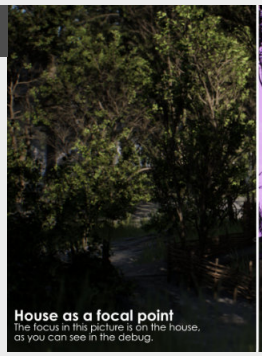
But let's dive into some of the settings first



2. Focus point

Let's start with the easy one first, the point of focus for the scene, in camera's this is usually done by focusing manually or automatically depending on the camera or lens you have. It's way easier in Unreal Engine though.

In Unreal Engine (or other 3D programs) we can just set a distance at which we want to focus and then also check the distance by pressing the "Draw debug Focus plane" so you can check if the focal point is in the right spot. To make it even easier you can also pick the asset you want to focus on by clicking the little eyedropper symbol by just clicking it.



3. Aperture

The aperture on the camera controls how strong the depth of field effect is, the camera's in Unreal Engine mimic the real world effects on camera's, where the aperture controls how open or closed the shutter on a camera is, which controls how much light can enter the lens area.

Less light on the camera lens means more camera blur (or depth of field).

However, compared to real life where you need to adjust your exposure values you don't need to do that in Unreal Engine.

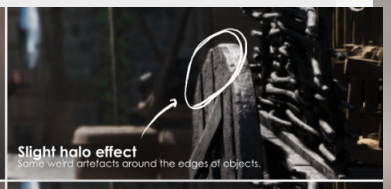
4. Some issues

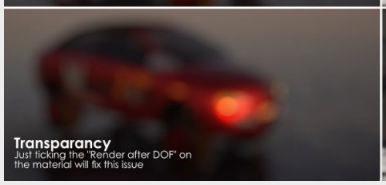
TRANSPARENCY ISSUE

You might have seen that there are issues with transparency, well it's because it renders transparency after the depth of field pass. But the fix for this is super easy, just go into your transparent material and tick "Render after DOF". Thanks @JustHannahGrace on twitter for sharing this!

HALOING ARTEFACTS

There is a massive improvement that can be done by activating a simple console command "r.TemporalAA.Upsampling 0". But I have to thank **William Faucher** on Youtube for sharing this tip with each other.





Adding content packs back in

Introduction

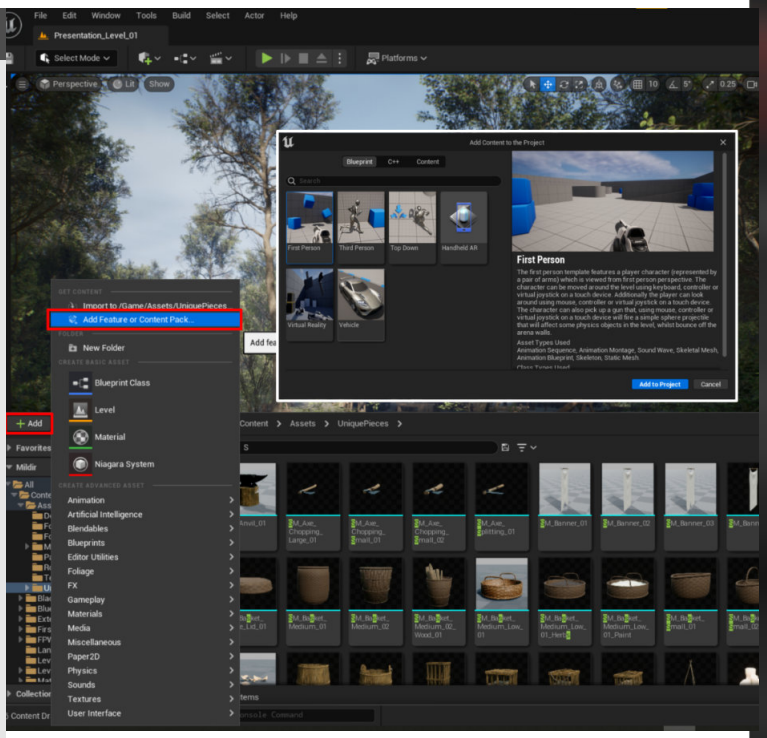
Donec mollis justo sed justo pellentesque sollicitudin. Duis bibendum adipiscing nibh. Maecenas diam risus, molestie ut, porta et, malesuada eget, nisi. In fermentum leo sed turpis. Sed lacus velit, consequat in, ultricies sit amet, malesuada et, diam. Integer mauris sem, convallis ut, consequat in, sollicitudin sed, leo. Cras purus elit, hendrerit ut, egestas eget, sagittis at, nulla. Integer justo dui, faucibus dictum, convallis sodales, accumsan id, risus. Aenean risus. Vestibulum scelerisque placerat sem.
In eget sapien vitae massa rhoncus lacinia. Nullam at leo nec metus aliquam semper. Phasellus tincidunt, ante nec lacinia ultrices,

1. Adding the content back in

You can add any additional content back into unreal Engine by pressing this green **"Add Content"** button. This will show this new menu where you can select what content you want to add to your current project.

This will then populate the folder with this content again and you now have this content available again.

But you can see that if you try to press **"Play"** it won't do anything just yet.



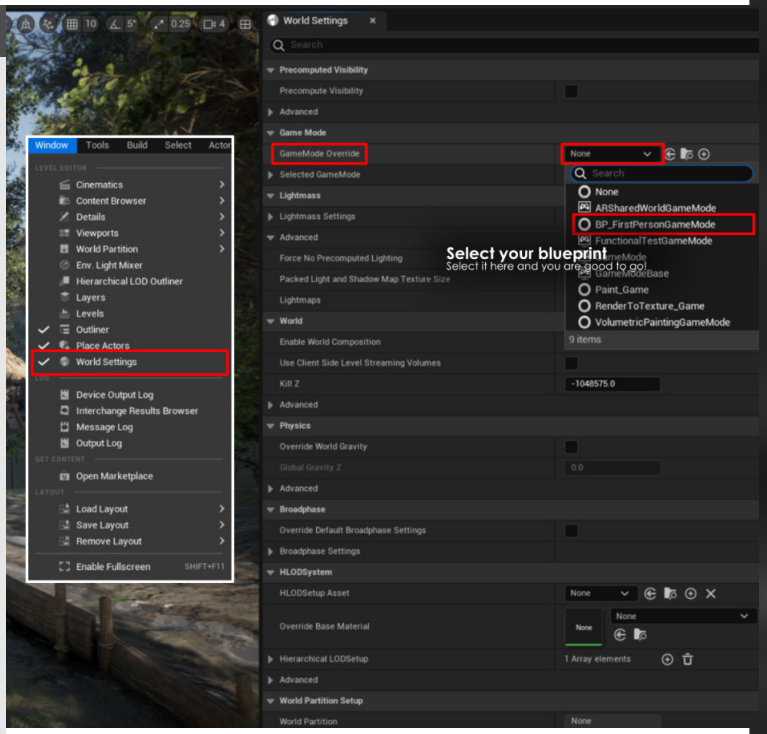
Making the Blueprint work again

To get these content packs to work again we need to link the newly added blueprints back to the game editor logic of when we start the game.

These settings can be found in the **"World Settings"** under **"Game Mode"** > **"Gamemode Override"**.

And you can either drag your BP_FirstPersonGameMode or BP_ThirdPersonGameMode in here or select it from the list.

After this you are all setup and good to go!



Easy Procedural Foliage Scattering

Introduction

Painting foliage on a landscape is a time intensive thing to do, especially when you want to make changes on the fly once you've carefully painted foliage already. So today we're looking into using procedural spawning to help you scatter foliage quick.
IMPORTANT LIMITATION This functionality doesn't support height field meshes, so it's great for smaller assets that don't need more detailed lighting, but doesn't work for larger meshes such as trees since they wont have correct Global illumination or Ambient Occlusion.

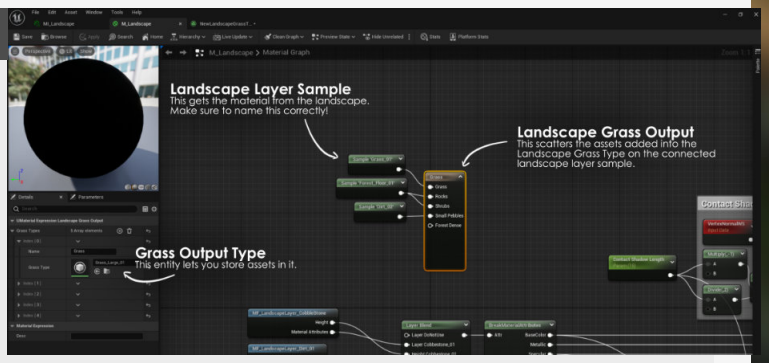
1. Overview

This technique consists out of 3 elements working together to add procedural spawning to your landscapes.

"Landscape Layer Sample" which will be the link to your landscape layers, so if you paint the one you specify it will add meshes to it.

"Landscape Grass Type" this is the container that holds all the assets which will be spawned with this system.

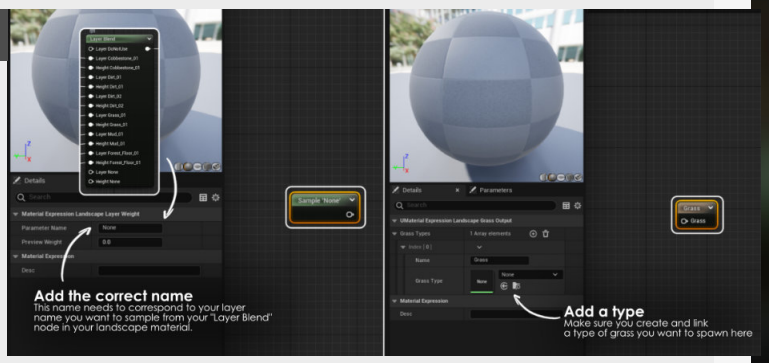
"Landscape grass output" this node acts as a link between the above, so it will take the landscape layer input and then spawn assets from within the selected "Type" on top of it.



2. Setting up "Grass output" layers

Let's go through the basics for this setup, we link the **"Landscape Layer Sample"** into the **"Landscape Grass Output"**. Make sure that the name of the **"Landscape Layer Sample"** is accurate to the name of your landscape layer blend name, if these are different then it won't work.

We can add more layers to sample and technically set one up for every variation we need, just as an example you could have a Forest floor that has dense vegetation, but create another type that has only rocks and spawn each type on a different landscape material layer.

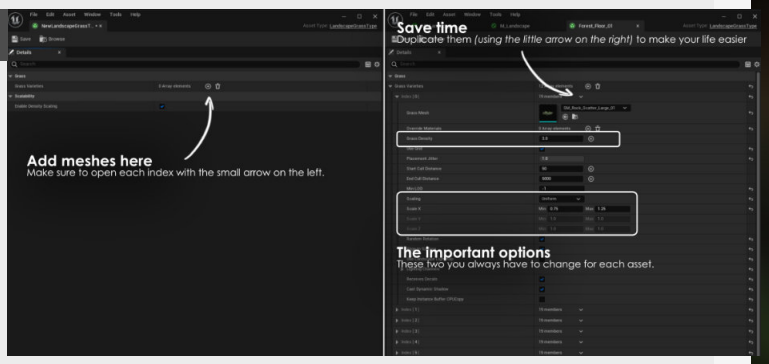


3. Adding meshes as grass output

Adding meshes to the **"Landscape Grass Output"** is quiet easy, we can do this by adding a new **"Landscape Grass Type"**.

A new interface will open where we can add additional "Array elements", each element comes with it's own separate controls on density, variations, etc... so you can customize it all in here.

Here we can change the **"Grass Density"** and the **"Scaling"**, the density controls how frequent you will see the mesh and the scaling controls how large it comes in, I always at least set the scaling to something like MIN: 0.75 to MAX: 1.25, so you can see some nice variation.



4. Playing around with it

Now the fun can begin, we can add more assets in each type and start populating them as we want and tweak them on the fly.

Sometimes it might take a bit to load in on project opening, and you might need to refresh the system trough either changing one parameter n your **"Grass Output Type"** slightly or by **"Build Grass Maps Only"** in the build menu.

Unreal Engine Camera Bookmark

Introduction

Tired of remembering camera positions, or always in fear of accidentally moving a camera? Camera bookmarks in Unreal are a life saver when it come to removing this stress from your projects.

1. Overview

These bookmarks are really easy to setup, if you want to set a bookmark you can use **"CTRL + Number"** where all the different numbers can be used as bookmarks.

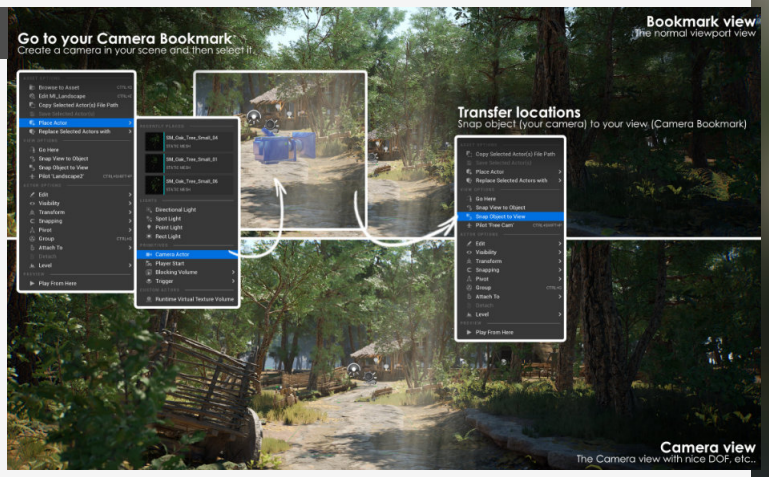
This allows for a total of 10 bookmarks in your scene.

Now if you want to go to one of these bookmarks you've set you can quickly press the corresponding number *("1" for camera bookmark one for example)* or even quickly flip through them, this is amazing if you have multiple important locations in your environment.

2. Camera to bookmark

Once you've got this set, you can create a new camera, select the camera and then selecting **"Snap actor to view"** this will then move that camera to that location on your bookmarks.

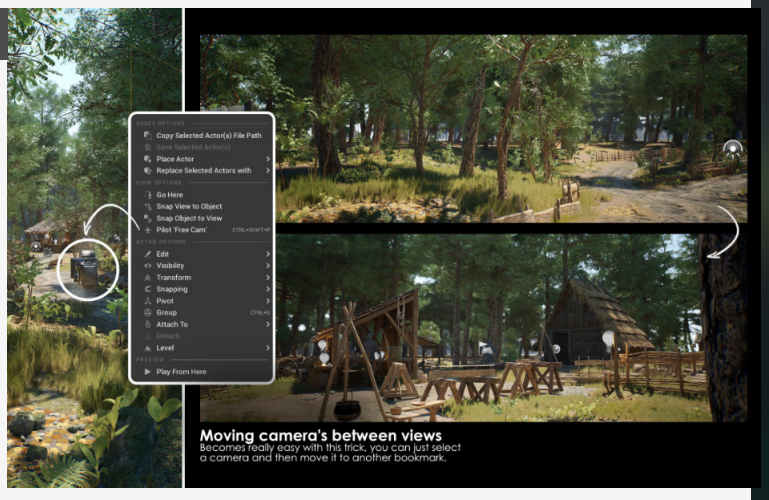
This is great for creating specific camera actors in specific places.



3. Move camera back into place

Another way of doing this is to **"pilot"** the camera and then press the bookmark you want to snap too. This is great if you found a nice shot, saved it, but then accidentally moved the camera later on.

(To avoid accidentally moving the camera you can also disable moving in the actor panel)



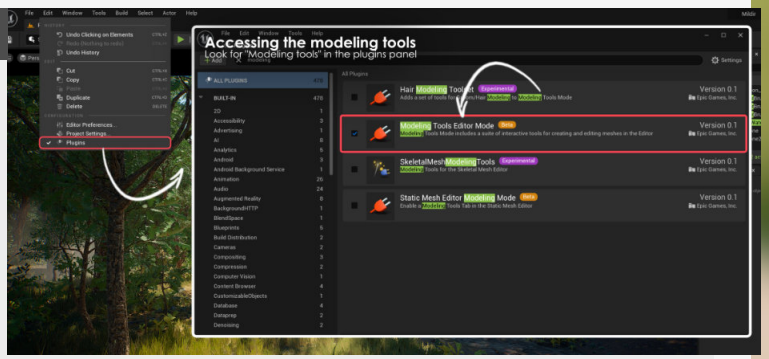
Baking Vertex Color in Unreal

Introduction

The modeling tools are a relatively new addition to Unreal Engine, and when I was experimenting with it I stumbled across this bake vertex color module. Prompting me to test it to add additional details to models that use a second UV set.

Even though it's still pretty basic on what information you can bake especially when doing it on a channel by channel basis, it's definitely worth investigating.

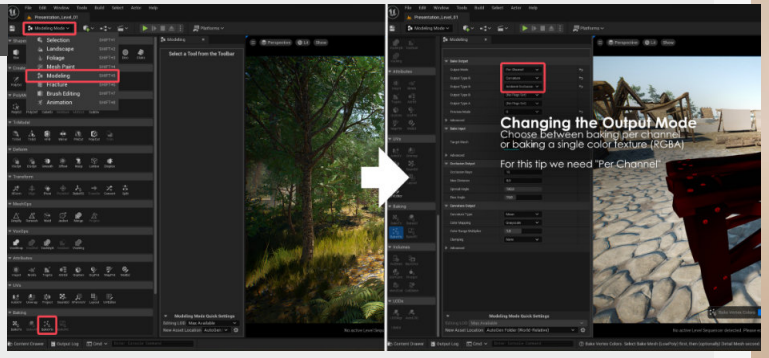
1. Accessing the tools



These tools are disabled by default, to enable them go to **"plugins"** and then look for **"modeling tools"**. Once you've checked that they are active you may need to restart the editor for this to take place.

On restart we can now find these in the **"Selection mode"** menu under **"Modeling"**. This will open up a whole new set of tools, from actual modeling tools (yes you can model stuff directly in engine) to the vertex baking tools we are going to use for this tutorial.

2. The options

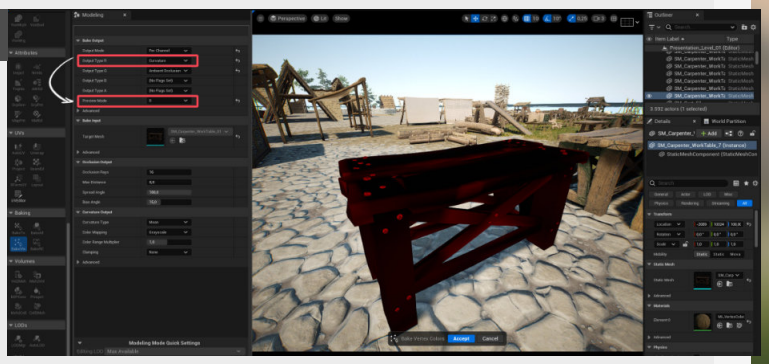


Select the **"BakeVtx"** option first, opening a new menu.

On opening **"Output mode"** is set to bake a full RGBA texture, which can be useful for if you need to bake a texture directly in engine. However, for this section we're going to set this to **"Per Channel"**. This allows us to store date in the different vertex color channels for this specific asset, so in our setup we store **"Curvature"** in the Red vertex color channel, **"Ambient Occlusion"** in the green vertex color channel, leaving blue and alpha empty.

We can then sample these different vertex color channels to use as masks for any materials that need additional layering.

3. Preview mode



In terms of the data we can bake down, it's pretty limited here, only allowing us to bake Curvature and AO, but if you then add more layers on top of each channel in the material you can get some great looking variation.

Some of the other options in this panel that are worth noting are the preview mode, and both the options for AO and Curvature, since there will allow us to change the look and feel of it.

If you are changing the Ambient occlusion parameters, put the preview mode to the "red channel" this allows you to see what you are baking and get a better feel for the end result before baking it properly.

4. How/Why I use this

I've been using trimsheets that require additional detail layers on top of them to make them look more unique. Before I was doing this with Substance Painter, then generating a quick mask for each channel , packing that into one texture and sending that to Unreal Engine.

However, you can already see the issue, if you're doing hundreds of assets that means hundreds of these textures.

This setup reduces all this to a single click per asset and storing the data on the asset itself. Combine this with only having to create material instance for a preset (clean, dirty, etc...) and it's a win!

Prefabs in Unreal Engine 5

Introduction

We are looking at a new feature they added in Unreal Engine 5 called Packed level actors or Level Instances, which are amazing if you are building something at a larger scale and want to reuse groups of assets over and over.

I've personally been using them as prefabs that don't have additional logic stored in them for now.

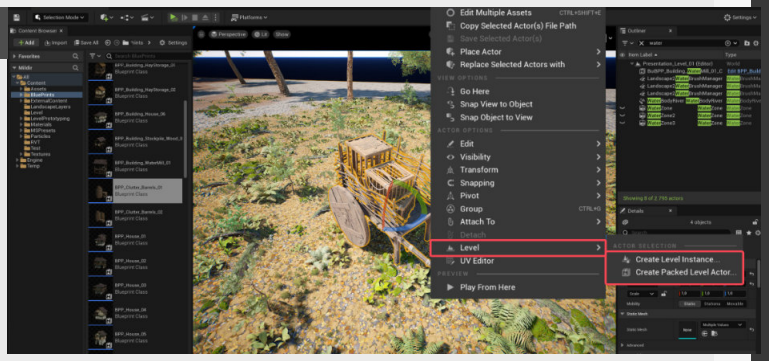
1. Overview

This new addition is amazing to create prefabs of meshes that can be used all over the place.

Another great thing is that these groups can also be broken again to get your initial assets back from it, which was also not possible with blueprints, so amazing stuff all around!

To create them, you simply select a bunch of assets you want to combine and then right click all of them and go to "Level" and create either a "Level instance" or a "Packed level actor".

2. How to create them

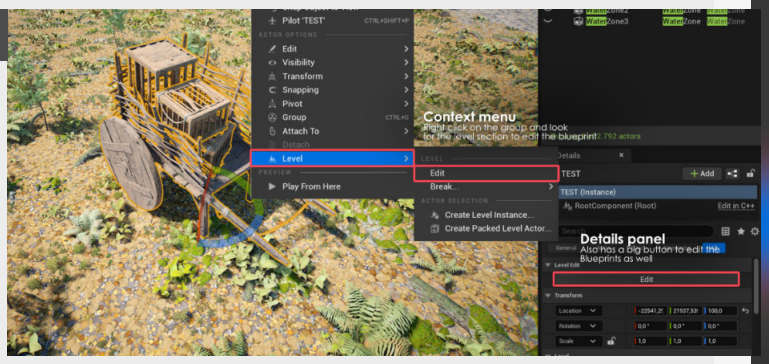


The first selection of the two here and it's the basic version. This one is perfect for groups of meshes that don't need any additional logic added to it.

To create this select your group of assets, "right click" > "Level" > "Create Level Instance".

This will open up a window where you can select your pivot placement for the group of assets.
And will also create your new Blueprint meshes inside of the folder selected.

3. Changing them up

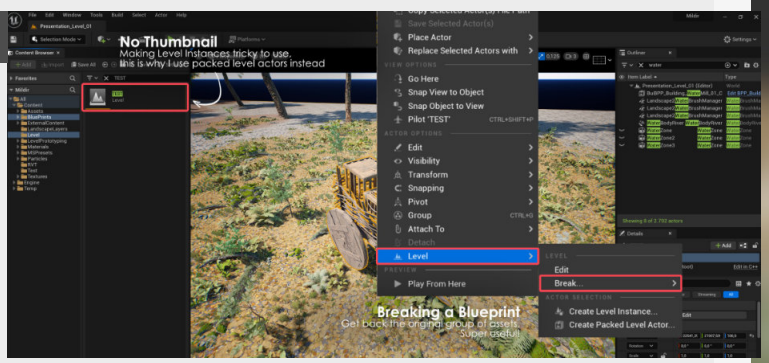


These new Blueprints also come with a new way of editing them. Instead of having to go into the individual level which is used to store the data, you can just edit them inside of the current level instead.

To do this, just select the blueprint and in the details panel you can now see a new "Edit" button pop up. This will gray out the meshes that are not in the group, making it very useful for editing.

Once you are done, you can just go back to the main group (in green) and click "Commit changes" or alternatively, "Right Click" > "Level" > "Commit" to commit these changes.

4. Some additional notes



Meshes are instanced: this means that vertex paint will propagate to other instances of this blueprint as well.
No thumbnails for Level instances: making it annoying to find a specific one inside of a collection of them. And generating your own takes time.
Breaking them apart: You can break these blueprints, making it super easy to go back to the original assets.

This technique to make quick variations for these blueprints, since you can just take a base one.

This stuff is amazing!

Volumetric fog volume

Introduction

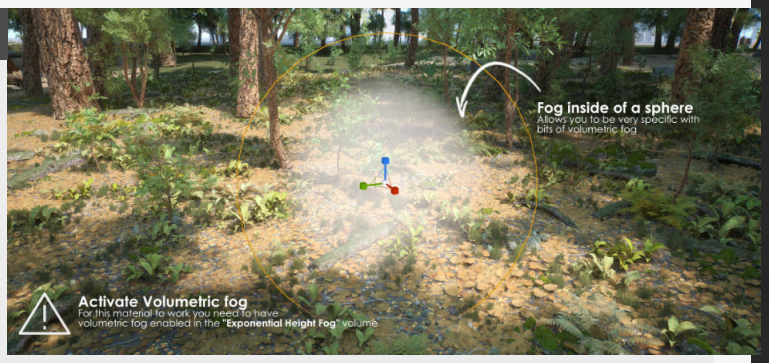
Volumetric fog can be used in a lot of ways, so it's good to have a material around that allows you to add fake sections with some fog you can scatter in your scene. And that's what we will be looking into today!

Big thanks to Chris Murphy (@HighlySpammable) for his super insightful video on 35 UE5 Features You Probably Don't Know About and his constant tips on Twitter.

1. Overview

Volumetric fog is usually done in a more generic way where it affects the whole scene in general. However, sometimes you want to get a bit more control and add specific section of fog in specific places, you can do this with these local volumes which are just meshes that are filled with a special material that will act as our volumetric fog.

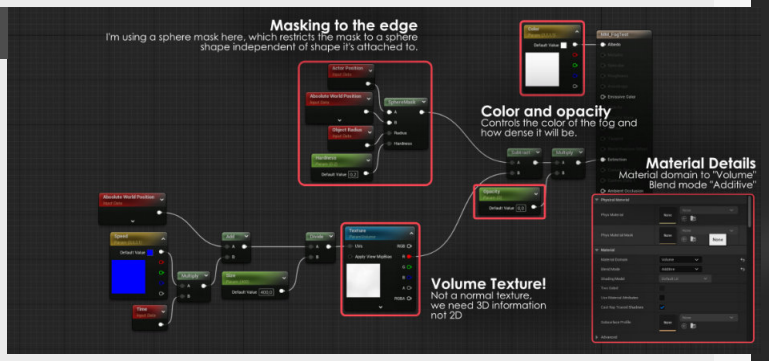
This material has a falloff that goes from the border of the mesh to the internal point, and we will setup the material to give us control over that as well.



2. Setting up the materials

For the material itself you need to make sure to set the material domain to "Volume" and set the Blend Mode to "Additive" (it will give you a warning if not set to it for volumes anyways.

The materials are pretty simple for this material you just have to pay attention to the texture sample. Because this is not your normal texture sample, it's actually a volume texture, for which I used "T_VolumeNoiseErosion32" which is part of the standard assets that come with the engine itself.

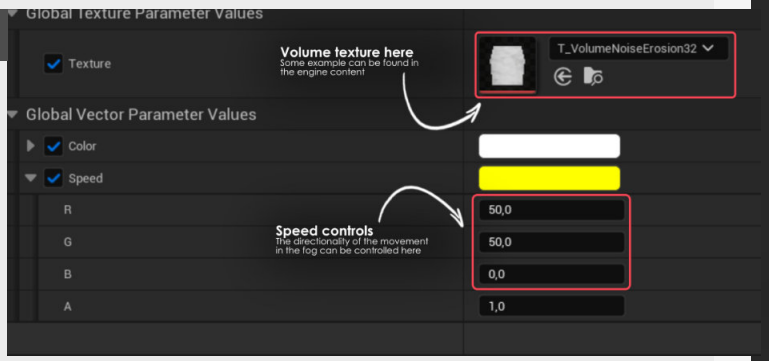


3. Tweaking parameters

First, let's add it to the scene by adding a sphere from the basic shapes into our scene and then adding the material on this asset.

Now we can start tweaking it by playing with the opacity, color, contrast and size. Size is a very interesting one as we can change how detailed the size of the volume itself it.

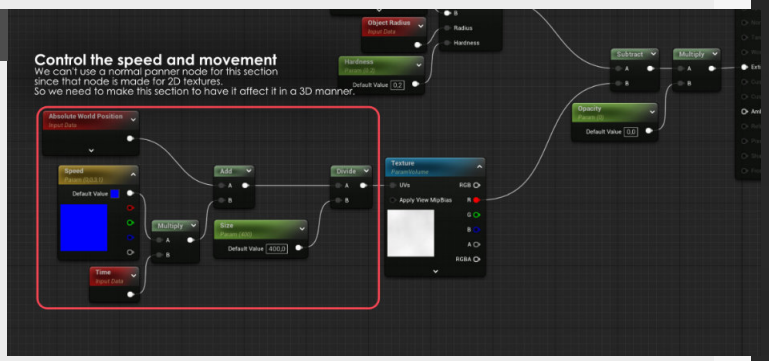
The way that I've set it up is more for a circle, since I've been using a sphere mask and that will only cut it off in a circle independent of the shape you add it onto.



4. Adding some movement

Lastly, as a nice little addition to the material, let's add a bit of movement using the Time input , world position and a vector to control the speed. (We can't use a panner, because that's made for 2D textures, so only requires a vector 2, where as a volume would require a vector3 instead)

This little setup will help you add some subtle animation to these volumes.



Arrays in Unreal Engine

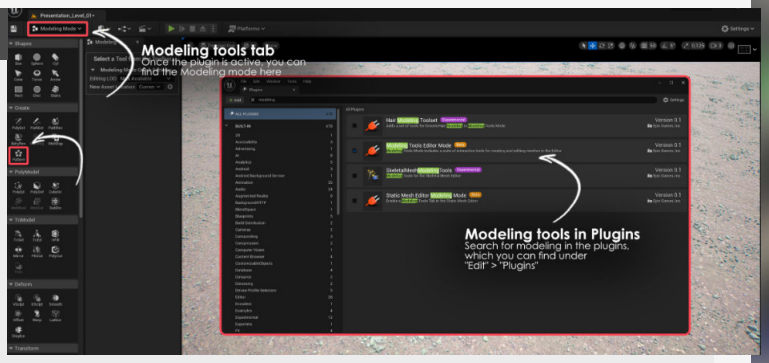
Introduction

Arrays are useful when creating repeated selections of and asset, ideally with some sort of randomization. But did you know that you can also do this straight in Unreal Engine? You can! However, it does come with some limitations.

1. Finding and accessing the tools

To get started, enable the modeling tools in the plugin menu and restart the engine if necessary. This will add a new tab under the dropdown menu located on the left side of the engine.

In this new menu we can now find the **"Pattern"** tool under the **"Create"** tab. If this is disabled then you need to make sure to select an asset first. So we need to add one asset that will be used for the scattering of the objects.



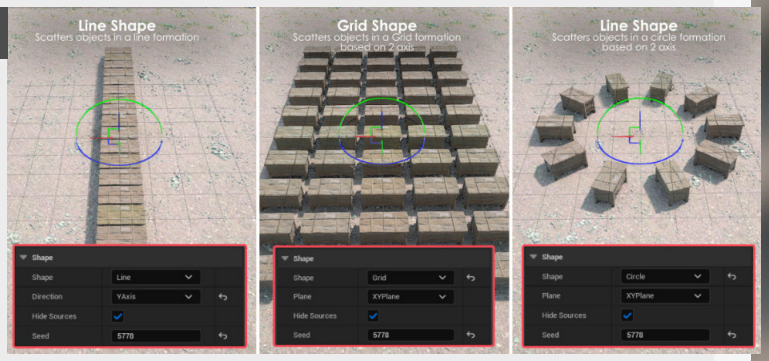
2. Shape Selection

The core workflow when working with the pattern tool is to first choose the shape of the pattern.

LINE: the base option, which will just scatter the assets you've selected on a line.

GRID: Creates a two dimensional grid of these objects, with controls of the length and the width.

CIRCLE: Instances the asset into a circle where you get to control the radius and where the circle starts and ends.



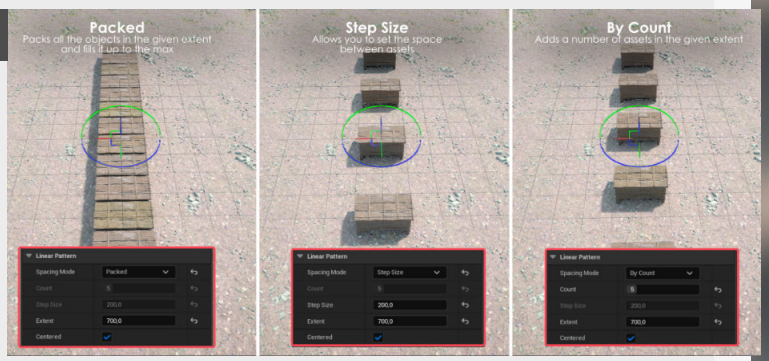
3. Ways to control it

The main way to control it is with the spacing mode first, where we have a couple of options.

By Count: Allows us to control the amount of instances of the model our self

Step Size: Adds the assets based on distance, so the smaller the step size here the more assets are going to be added

Packed: Will automatically fill the shape with as many assets as it can based on the extent of the shape itself

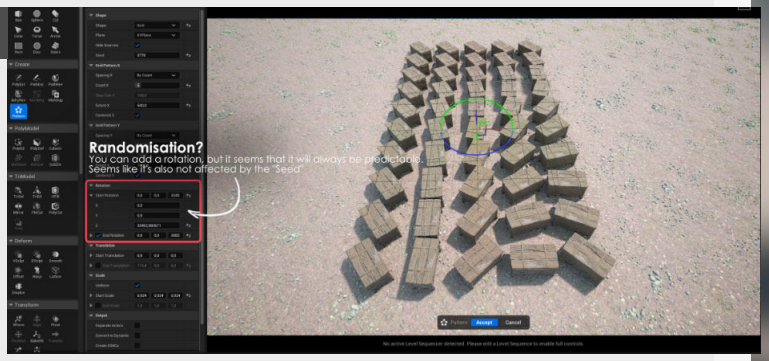


4. Limitations

When working with this I did find some limitations, the biggest one for me is that there is no randomisation between object (at least in 5.1, the version used at the time) but it does have a random seed value, which is odd. (Or I didn't figure out how to use it)

Also it doesn't allow for multiple assets at the moment, which makes it annoying to create interesting tile based meshes.

As a small side when using this tool it is destructive, you can't go back and control Z doesn't work eith4er, which is annoying! However, it does save your settings, which is good.



10 LEVEL ART

Blending shapes together (Rule of Three)

Introduction

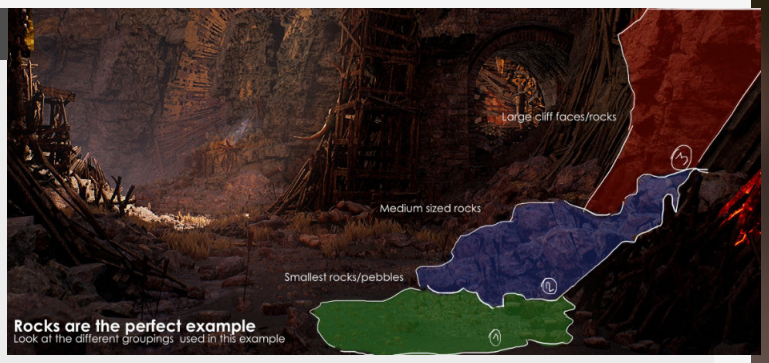
Humans have this natural connection with the rule of thirds and nice transitions between elements, so this is where we come the the rule of three's, a technique that I've been unconsciously using for a long time but have recently discovered the name for.

1. The rule of three's

A rule that can be applies and used in your own scenes and games to create more balanced compositions, from the macro to the micro. This rule tells you that you need three different sizes of volumes to create a nice and balanced composition in your images.

Starting from big blocking shapes, going down to medium shapes for a nice transition between big an medium and then small shapes to bridge the final gap between these shapes and the base terrain.

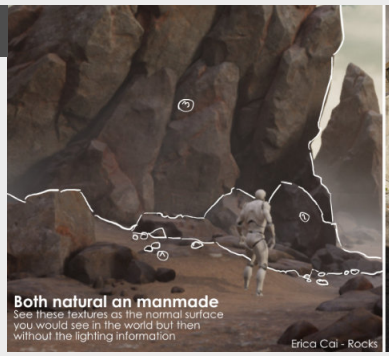
This creates for a really nice an natural feeling flow from these big to small elements.



2. Where does it apply?

Let's take it one step lower down where these shapes come as the first step as blending the big shapes with the underlying base terrain. These shapes can also be used as objects of cover in a lot of games where as the big shapes are a bit too big for that.

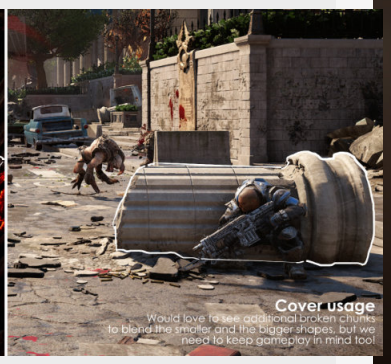
They also help to bridge the gap between the really small and the bigger shapes in your environments. Most of these come in the form of rocks or solid objects, as bushes and plants can cause issues with AI still seeing you through them (and shooting through them as well) in some games.



3. Medium shapes

Let's take it one step lower down where these shapes come as the first step as blending the big shapes with the underlying base terrain. These shapes can also be used as objects of cover in a lot of games where as the big shapes are a bit too big for that.

They also help to bridge the gap between the really small and the bigger shapes in your environments. Most of these come in the form of rocks or solid objects, as bushes and plants can cause issues with AI still seeing you through them (and shooting through them as well) in some games.



4. Small shapes

Don't underestimate these! They do make a huge difference in your environments and not only in this example for the blending with the terrain.

These can range from small decals, to pebbles, smaller plants, etc.. They might seem small, but they add a ton to the environment and can really help reinforce the story telling.

Tiretracks in the mud? Someone must have been here recently with a vehicle. A trail of blood between crushed patched of grass? This is the trail that my prey took on my hunt. These can add so much, so use them with good intentions.

Blueprints for level art (OUTDATED)

Introduction

A good way of speeding up the process of being able to dress-up large spaces is by creating groups that can be reused. But how to do so in a good way is something that might take a bit of practice.

1. Creating good groupings

First off all, what we need to do is gather some of the assets that we want to start building our prefabs with and start by building bigger groups so we can break these down later in smaller groups as an option. This also allows us to see how much we can get out of the current assets we have created.

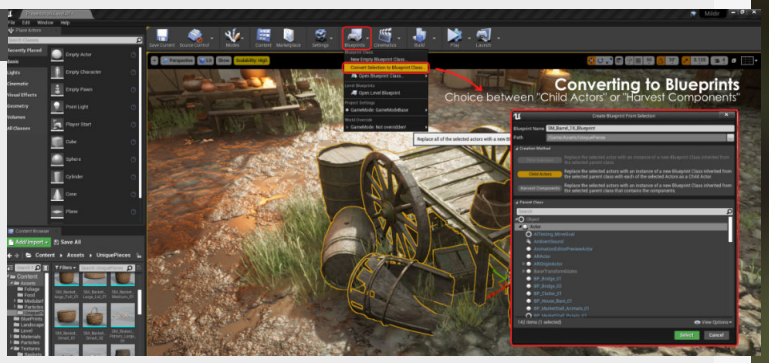
We have to be careful about the combinations we make and that we keep them generic enough so we try to reduce the visibility of the reuse as much as possible, which we can also do with some unique elements later, but first, let's see how to make blueprints first.

2. Converting to blueprints

Is actually pretty straightforward. Before digging into that tho, we need to go to the menu above on the toolbar under "Blueprints" and then select "Convert to blueprints".

Now this is where we get to option to where either we want to harvest the components or not, "Harvest Components" will strip other blueprints into it's components where the other options won't do so.

Once we have these blueprints is that once we do adjustments in the blueprint these changes will propagate live to other blueprints already in your scene.



3. Adding variation

Once we got some base blueprints setup it's also super easy to just duplicate it and do some changes to make it look different to the other ones.

When making these groups try to think about them as modular pieces, where you want to create small, medium and large groups so that you can combine these different blueprints to create some collections if needed too.

However do be careful by not making to many of these individual blueprints and keep them usable for larger set dressing, smaller set dressing can be done individually.

4. Add unique elements

So that's also exactly what we will do once we have a nice collection of these prefabs ready. We can then just add little touches of unique assets to them to make them look different from the prefabs.

LOD's for trees in games

Introduction

Trees in games can be quite a technical and complex asset, since they are usually really dense in geometry, you usually have a whole bunch of them and you need to keep it all within budget as well. So this time we're going to be diving into a high level overview on how we fit all of them in budget.

1. The detailed up close one (LOD0)

Let's start up close, the main mesh we see is the one we get to see up close *(Only environment artists really ever look at them like this, but that's beside the point)*. This mesh needs to look the best out of all of them, so that's also why we dedicate a lot of geometry to them whenever we can.

There are still opportunities for optimization though!

The main optimization we can make here is centered around the player's viewpoint, keeping this in mind for a first person game we know that people will mainly focus on the roots, bottom trunk and the first one or two layers of branches going up. This means that we can start to reduce all the other sections like the upper part of the trunk and upper branches in a gradient going up from the most visible section of the tree, without anyone even noticing it.

2. Transition meshes (LOD 1-2-3)

We're looking at the mesh at Medium distance now, which is going to contain most of our LOD's (Level of Detail meshes). These meshes are going to act as the bridge between our most detailed and most expensive LOD towards the cheapest to render one.

The main change that is happening here is that we're reducing the amount of polygons on each branch or even replacing them with planes that have a representation of that same branch but in the form of a texture applied to them (think baked down onto a plane).

Now even with simple planes we can still create the illusion of depth by being clever with the geometry. X formations are s good examples of this and provide decent coverage. But the approach depends highly on the type of tree you're creating.

3. Billboards (LOD 4)

Now we're standing really far from the tree. At this distance we can get away with some really magical stuff.

At this range we usually bake down the texture onto a plane and let that follow the camera of the player, but you might think that it would look weird? And you're right, that's we're some of that magic comes in. We don't just bake one angle of the tree down and add that onto the plane but we take multiple angles of the same tree. Let's say 8 for example.

These 8 sides will then be baked onto one texture and when you as a player walk around the tree it will sample and blend between a different section of the texture depending the rotation of this plane is facing towards the player.

And if we want to make this more or less detailed we can capture more or less angles of a tree, it's magic!

Level Of Detail Meshes

Introduction

LOD's of (Level of Detail) are meshes that are used to improve the performance over distance by reducing the number of polygons, materials and textures the further they are away from the main gameplay camera.

There are many different applications and ways of optimally doing LOD's meshes, so we'll keep it to a more wider overview of the concept.

1.Geometry

The big change and most visually impacting if done incorrectly is the change to the geometry, this meaning that if you can see the change in the geometry happening that's when your LOD's need some tweaking, it needs to be as seamless as possible.

In most cases we don't have to do these by hand though, and we can just run them through a program like Simplygon that will do an automatic LOD creation pass that is usually pretty decent, with addition control to change it per object basis, which is usually determined per project.

2. Textures/Materials

Not only the geometry will get optimized over time though, we also need to reduced the textures/materials over distance. So usually us artists also need to take this into account (this might also be done automatically in cases).

But the way we optimize this step is by replacing the materials on the long distance meshes to a simple color.

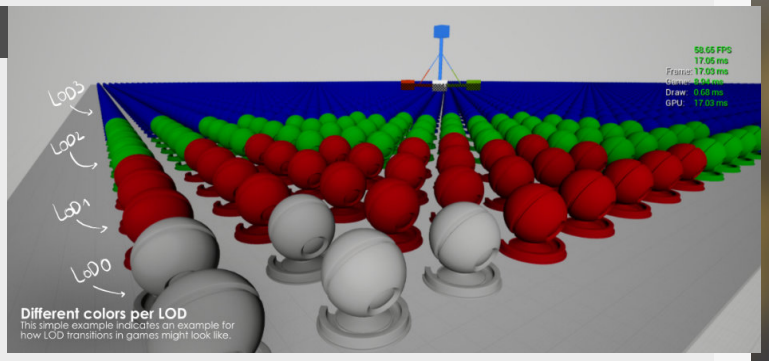
Since these meshes are only visible from such a long distance we don't need to have expensive textures be added to our memory pool and can just replace them with a color swatch and simple materials, reducing their performance budget.

3. Switching between Meshes

There are different ways of transitioning between LOD's. The most common usecases of switching is either over distance or by screenspace.

The first one speaks for itself, once you hit a certain distance from the object the object will then transition to the next LOD.

The other one is used in Unreal Engine and this one will look at an object and determine how much space it takes on your screen and base the transition on that factor, with the additional control of deciding what the ideal screen percentage is based on the type of object.



4. Vista Meshes

An interesting case for LOD's are Vista meshes, these meshes need to be visible from a larger distance to serve as a player landmark.

This also means that we can't really rely on culling them once they are behind another mesh, so we need to look for a different solution.

That's where **"Vista"** meshes come in, these are special meshes, usually generated by an automated process, that will take your final LOD and then do a quick bake of your first LOD to a new unique (but really small) material that is then used to fake the look of something intricate on the horizon. As you can see in this example from Fallout: New Vegas.

Occlusion Culling

Introduction

This time around we're having a look at what exactly Occlusion Culling is, being implemented by most engines by default to optimize your scenes/games and remove unused assets and free up resource budget.

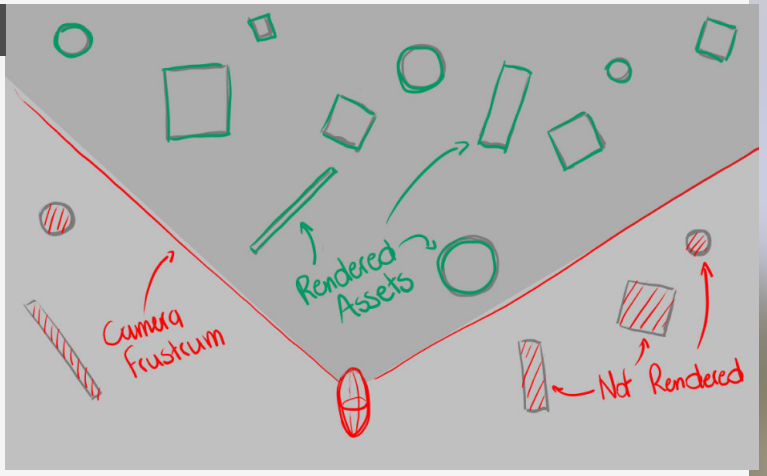
1. Visibility Culling

Often referred as **"Frustum Culling"** as it's specific to the camera's frustum do decide which assets it will cull or not.

This type of culling happens when the camera viewpoint is rotated away from meshes and they are outside of the viewable space (frustum) of the camera. Removing all these assets to lessen the amount of geometry data it needs to calculate on screen.

In unreal engine we can change the dimensions of this frustum by changing the max (far clipping plane) and minimum (near clipping plane) depth of the frustum.

We can also visualize this by going to the advanced drop-down and **"show camera frustum"**



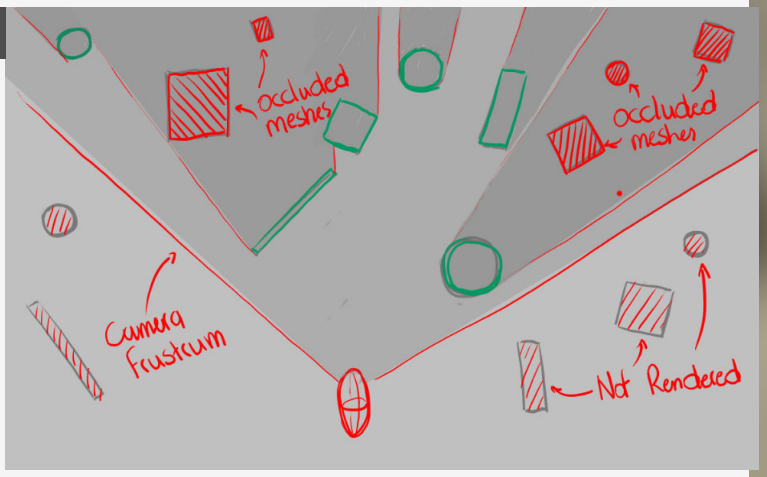
2. Object culling

Object culling is the second most important one. It depends on the visibility of meshes based on other meshes in the scene, so if one asset obscures another asset then it won't render that asset that's being hidden.

Some engines can have a specific input mesh specifically for the culling to help get some more accurate results in certain scenario's.

If you are looking to visualize this type of culling you can either so this by using the console command "FreezeRendering" this stops the culling in its current frame and let's you go around the scene looking at what's culled or not.

In addition to this you can also use "r.VisualizeOccludedPrimitives 1" to show the green bounding boxes of assets that are occluded currently.

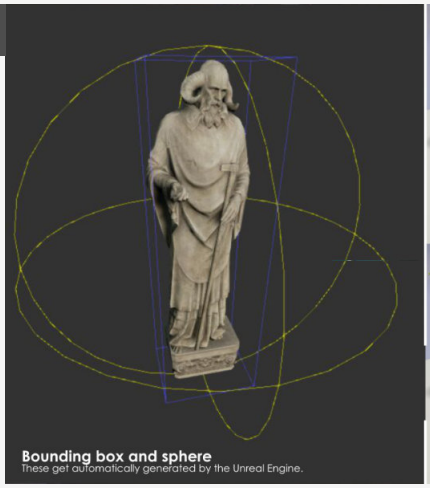


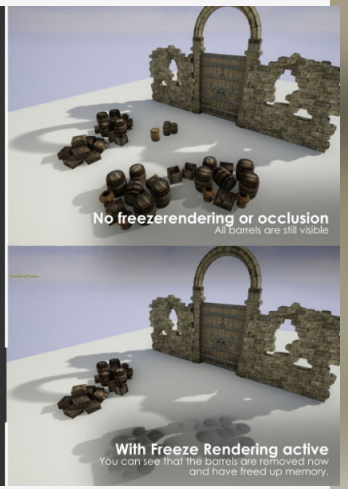
3. Changing dimensions

Unreal engine uses both a box and a sphere to do these checks within the engine, a quick check for the sphere and a more accurate test for the box. But you can change these after the fact by either going into the details panel for the asset and changing the bounding box per instance in a uniform way or going into the mesh settings and changing the positive or negative bounding box settings for the mesh.

However this should mainly be used for testing or very specific cases where you actually want a result that doesn't align with the mesh itself.

Also, we've been focusing on Unreal Engine, but some other engines offer the opportunity to change the asset culling volume to a specific mesh that can be used instead, giving you a more accurate culling result where needed.





Hiding Parallax Limitations

Introduction

This is a small trick that is really useful for when you want to hide the restrictions of parallax occlusion materials, and maintain the illusion that's it's not actual geometry.

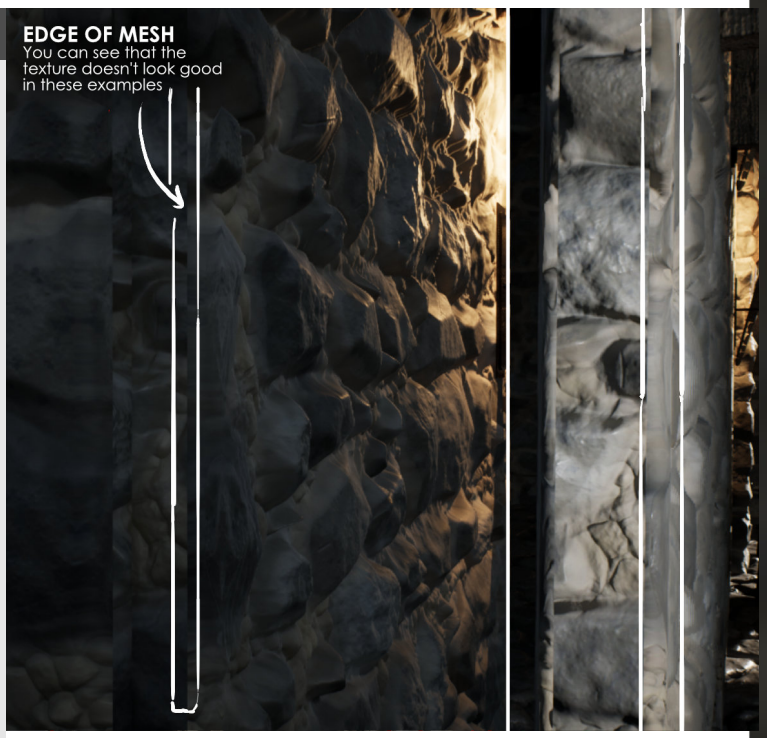
1. Limitations of POM

When looking at glancing angles at parallax you can really start to notice the limitations of the fake geometry effect you get when adding POM to objects.

So games that use this technique go through a lot of trouble to hide these areas so that you either never look at them. They do this by either using it it in places where you can really get a good look at it at shallow angles so it doesn't break the illusion or they hide it with other meshes and always have the parallax meshes obscured with other ones that don't use any parallax.

The latter is also the example that we're diving into here.

Examples of why you don't want people be looking at the edges or shallow angles of parallax materials can be seen on the right, that's when you can start to notice the stretching.



2. Keeping the illusion going

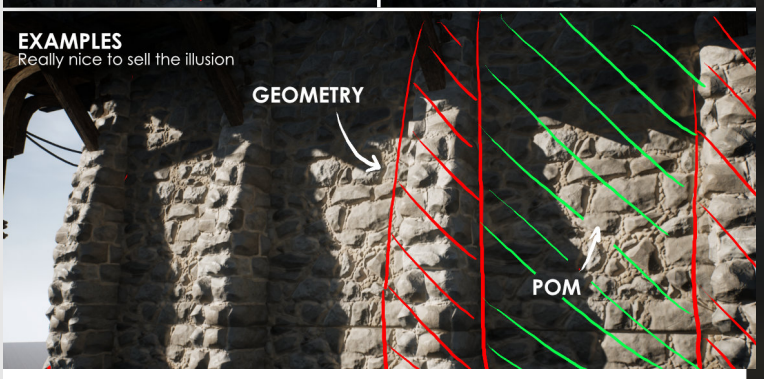
Like mentioned before, games environment artists take great care in putting meshes around area's where it might break the illusion of it being a game.

And this is no exception, we add meshes made out of more detailed geometry with interesting silhouettes to obscure the fact that the flat walls use the parallax effect.

The main goal here is to keep the illusion going so that people can't notice which is real or which thing is fakes.

If we look at the example from the last step we can cover up the seam of the stones, as this just looks plain bad.

For this example we just add a nice pillar in place surrounding the flat sections on the wall and that will be enough to keep the illusion alive!



Stacking Level Instances

Introduction

To expand on a former topic on using level instances we can expand on their functionality a little bit by stacking them inside of each other. And creating a layered structure within them.

1. Group within groups

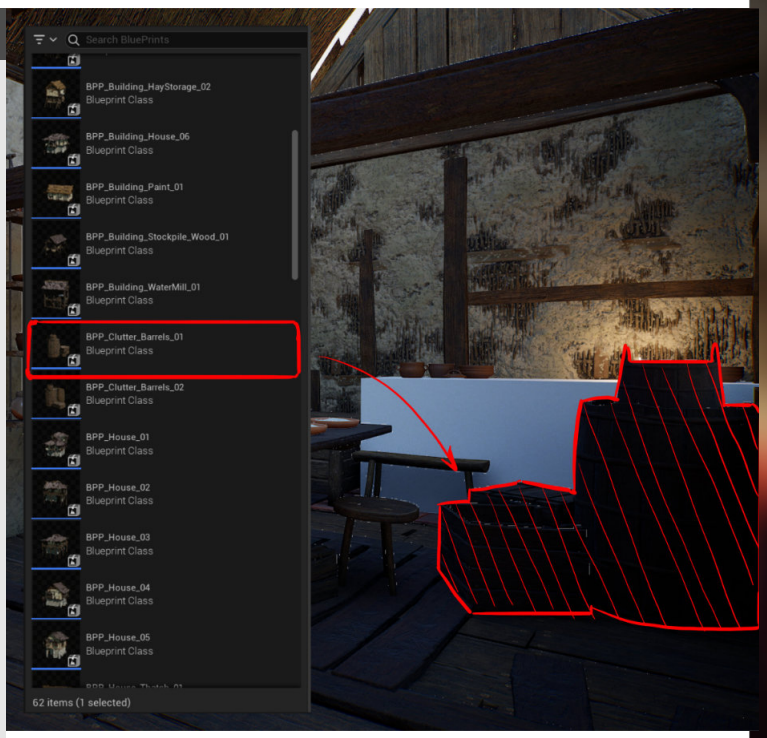
First before we dive into it we need to think about how we're going to make these groups and this will depend on the project you are doing yourself.

For example, In my case I found that making smaller collections like barrels and bits of storage and then adding them to bigger level instances like buildings or workshops or thing like that, that is the way I currently like to structure them.

To add them to each other we can go into the main level instance, go into "Edit" mode and then either copy or drag in the other level instances.

So I've tried 2 layers at maximum for the current setup, but you might need more depending on your setup. However, I would be very cautious with this and actually limit the amount of layers to make sure that you're not adding a bunch of unnecessary complexity to it.

It might also become more unstable too? But that's just speculation.



2. Multi level breaking

So last time we looked at the amazing feature of being able to break apart these level instances down into the individual assets again. Another amazing addition to this is that you can break down multiple layers and decide to only break down certain layers of the entire structure.

So this means that in my example I have a House level instance which also contains another collection of barrels level instance inside of it.

If we decide to "break" 1 level it will only break the house itself and keep the barrels as a level instance still. If we break 2 levels then it will break apart the house itself and also the collections of barrels.

It's amazing to have this amount of control with this system, it's awesome!

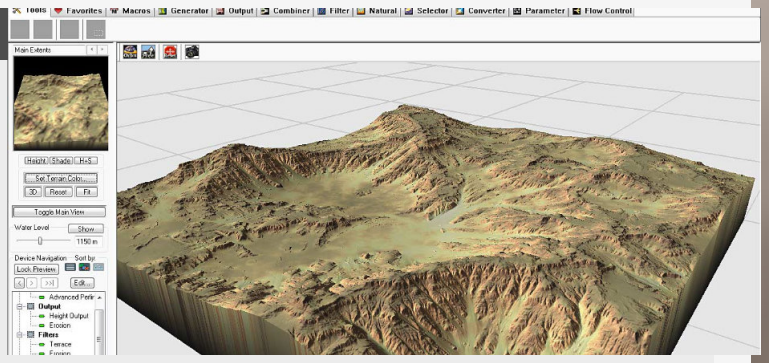
11 TERRAIN MANIPULATION

Generic Landscape Tips

Introduction

When building environments and landscapes you will often see that landscape sculpting and worldbuilding will go hand in hand. Let's dive into some stuff that will make your environments, locations and landscapes better than they were before with some tips on how to establish world logic, use scale references and more!

1. Building heightmaps



Let's dive into some more general tips on landscaped, first starting with ways to build heightmaps (black and white textures that indicate height) There are multiple different programs we can use to build up these heightmaps, such as World Machine, Terragen, Houdini, etc...
The most well know off these is World Machine, which allows you to use nodes to build a terrain texture.

This program also allows you to export all the color information which you can then use to blend different textures together in Unreal Engine for instance.

2. Scale references

Keep testing your environment and running through it. This will always give you the sense of scale you require to keep building your environment to a believable scale, because if you build your heightmap in World Machine (as seen in the example above for example)

you don't really get that sense of scale (although erosion helps).

Continuously test your levels by running around or adding some scale reference assets to your levels like trees, rocks or houses for example, these will really help you with giving you back that sense of scale.

3. Explore the real world and use it

This might speak for itself a little bit but tools like Google Earth/ Maps are super useful when it comes to planning locations and spaces in your world.

The top down view gives you the option to find some cool locations or layouts that you can use to create your own.Another thing that is useful is that you can use the measure tool (ruler on the side)

Lastly, use the streetview to your advantage, this can give you a quick view of the style of buildings, terrain elevation differences and general building metrics.

4. Building world logic

Think of it like a sculpt, you have to start with the bigger shapes and build up layer by layer. But before even starting your world you need to define the rule-set.

With this I mean asking all the questions that are needed to create a living, believable world.
Where is your region based, what is the main source for erosion of the landscape?how has it changed over time? What kind of structures would people inhabiting this region use?

It all starts with asking these questions and defining them.

Terrain features for player leading

Introduction

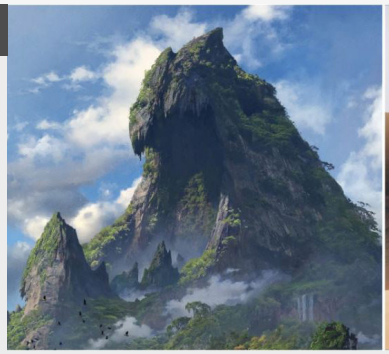
As an environment artist you have a lot of tools to help people find their ways in your environment, most of these are coming from a good composition and for this one we're going to see how we can reinforce this with some good terrain features. Let's dive into a more general one first and then go into some specifics.

1. Big Terrain Shapes

Walls and cliffs
mostly added as a first step when blocking in your shapes. These are then used to prohibit the player from going outside of the playable area.

Terrain slopes
Slopes can also be used to push the composition of your environments, their natural softness will often lead the eye for the top of them to the bottom. So you can use this to your advantage.

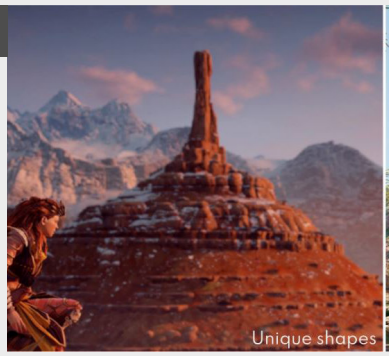
Use negative space
We have these big shapes around our entire environment, if we want to lead to player to a specific point between them, we can just create large gulley on that section, creating negative space and contrast.



2. Points of Interest

Having bigger more unique natural shapes in your environment is also a good way of attracting the players/ viewers attention to a specific point. Nature has some pretty weird structures and shapes by itself, so get inspired by it!

You might have even seen a lot of them in games you've played recently. These can range from weird mountains, to stone arches, giant waterfalls, you name it!



3. Empowering Exploration

Creating natural blockers
A trick that has been used in countless games is to tease something and increase a sense of mystery by hiding it around the corner or blocking it by natural formations such as small slopes, rocks or foliage. Straight sections of road are just really boring to traverse, especially for longer stretches of time.

Hiding and revealing locations
This trick can also be used on the vertical aspect by showing a large structure in the distance and then hiding it as you go down into the foliage or lower levels of the level. This has been used to great effect especially by Naughty god in all their games.

4. Micro Features

Paths in your environment
Subtle details can really make a big difference, pushing a well worn path down a bit into the terrain will help lead the player towards it. These paths by themself already create a contrast between the textures used.

Smaller ridges
Smaller elevation changes can also be used where the terrain allows for it, having smaller ridges can help guide the player in a certain direction with repeating shapes and again shape contrast.

Landscape Introduction

Introduction

This first part of a multi-parter on landscapes in going to dive into some of the basics of creating landcsapes inside of Unreal engine. We will have a look at some of the more important settings when creating it, where to find it and lastly how to import your own heiahtmaps!

1. Landscape mode

Let's dive into the creation of landscapes, this is something that I haven't delved into that deep myself inside of Unreal Engine.

Starting with the basics, let's look at the landscape tab in the Modes section.

At first you will notice that if you don't have any landscape created previously that the options to "Sculpt" and "Paint" are greyed out.
For a quick introduction, press the little Learning button inside of the green blinding circles, this is really great for a quick start into landscapes.

2. Landscape settings

Let's have a look at some settings next:

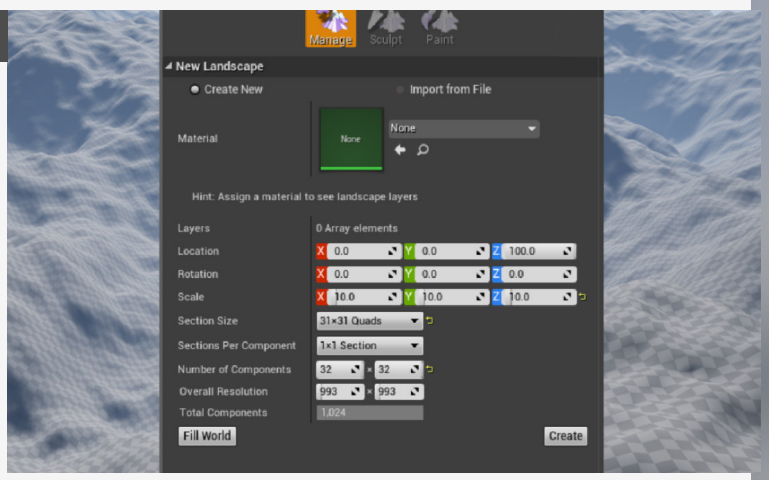
MATERIAL: This is a slot that allows us to apply a material when the landscape gets generated.

LOCATION, ROTATION & SCALE: Self explanatory really, allows you to transform the terrain.

SECTION SIZE: Used for LOD and Culling settings, smaller sections will LOD more aggressively (At higher CPU costs) Larger components are less costly.

OVERALL RESOLUTION: Controls the number of vertices the landscape will have, useful for aligning landscape to heightmaps.

What I normally do is create the landscape that I want to ave and then start cutting away at the landscape using the "Component Tools" which allows you to add and remove certain components.



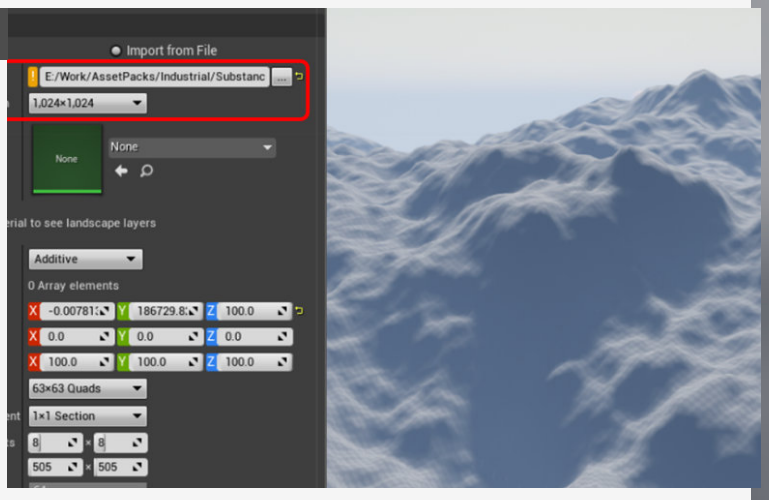
3. Importing heightmaps

When starting out your ventures into landscapes it should be super daunting to just start from an empty lot of land, this is why I suggest you start out using a heightmap.

Heightmaps are black and white textures that indicate where the altitude of the landscape is going to be placed.

So white points are going to be the highest point of your landscape where black points are the lowest points of your landscape.

You can import these when creating a new landscape as seen in the picture on the right. It's also going to automatically set the resolution for you.

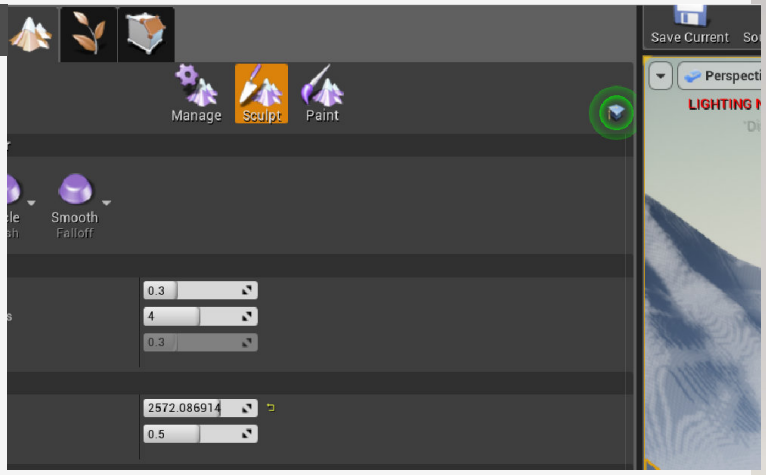


Landscape sculpting

Introduction

The second part on landscapes will look a bit deeper into some tips on how to sculpt landscapes, for this we will dive into some of the settings of the brushes and more, so let's have a look!

1. Sculpting mode

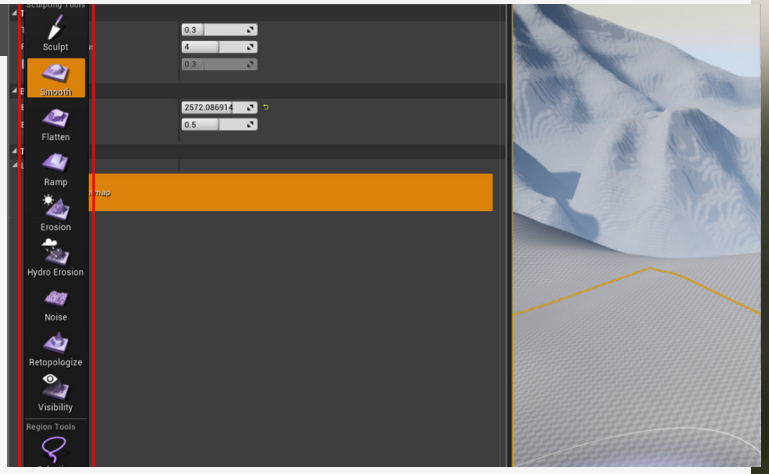


This time we will dive into sculpting terrain inside of Unreal Engine, there are some great tools that let you sculpt your own environments.These tips will help you to know which sort of thing you need to use for the correct end result. Each brush has it's own settings, but let's dive into some of the more generic settings first.

Brush size
The general size of the brush, specified in Unreal Units so 1 = 1m.

Brush falloff
The brush falloff, depicted by a number from 0 to 1 where 0 is a hard brush and 1 is a soft brush.

2. Sculpting settings



Sculpt
The basic sculpting tool, can be changed to other things than a circle using alpha's, different falloffs, etc...

Smooth
Will smoothen the transition between all the height variations.

Flatten
Really useful for making platforms and flat sections, or even make angled flat sections using the "use slope flatten" option.

Ramp
A really useful tool to create paths us mountains or straight sections between two points

Erosion
Best to use on contrasting edges, such as mountain ridges, this will simulate the transfer of soil between the highest point to the lowest point of the brush.

Hydro Erosion
Erosion based on rainfall, is going to create the feeling of pools of water collecting.

3. Brushes and falloff settings

Brushes
Circle is just the basic circle brush

Alpha is used to import your own alpha maps using the importing menu and will allow you to paint that alpha and it will follow your cursor.

Pattern does the same as alpha, only difference is that this will be world spaced and thus always uniformly applied.

Falloff
All different options for different kinds of falloff, allowing you to specifically set it to your liking.

Automatic grass spawning on Landscapes

Introduction

There's a way better way of adding grass to your landscape instead of painting it yourself, we can do so by also adding it automatically based on the landscape layers that we will be painting like grass.

So we're going to be diving into the setup this time.

1. Adding meshes to scatter on top of the landscape

Landscape material

First we need to start with the landscape material itself, while it might look like it's a lot but it's fairly straight forward if we break it down. All the individual layers have been added as "Material Functions" which are basically smaller reusable sections of Nodes, but in this case I made them to make the material a little cleaner and less looking like a spaghetti.

You can also see that I already have all the layers name setup in the "Layer Blend" node which will be blending all the inputs together depending on the materials that are painted on the landscape. The name in particular is important for the next step.

Landscape Grass Output and Landscape Layer Sample

Now that we have this we need to start by adding a **"LandscapeGrassOutput"** node, and a **"LandscapeLayerSample"** in our Landscape material and make sure that the name of the "Parameter Name" in the sample node matches the name of your layer you want to be sampling from so they are linked together.

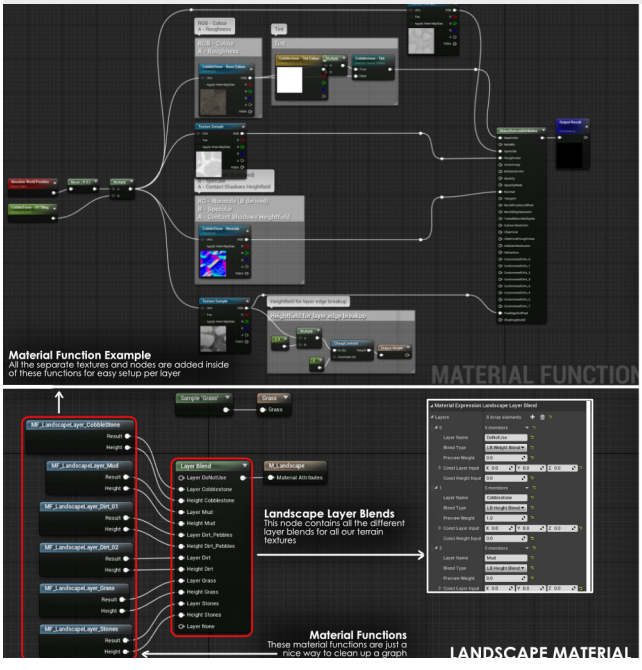
Landscape Grass Output

We still need to link the types of mesh we want to spawn inside of the **"LandscapeGrassOutput"** but for this we need to create a **"Landscape Grass Type"** this contains all the different assets that we want to scatter through this system, this also allows us to add multiple different assets and change parameters for them all separately.

If you are used to the foliage tool for the landscape itself you will recognize most of these options that we have in here already, they are a little more limited then the foliage tool, but it's great tool to have assets spawn dynamically, plus we can also change these parameters on the fly.

Happy Painting!

Now once we've set this up we can enjoy painting our grass material and it will spawn grass on top of it dynamically, not bad for this quick setup!



Unreal Landscape Material Basics

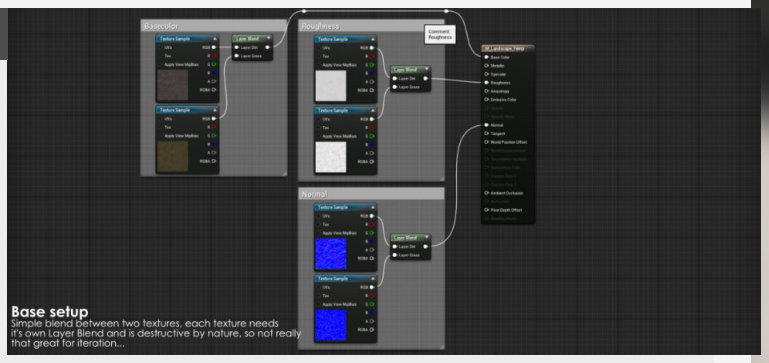
Introduction

Landscapes in games are usually quite restrictive in how many layers and textures it can use and the material itself can become quite complex really quickly, but in this one, we're sticking with the basics and making a base Landscape Material for you to use in Unreal Engine.

1. Basic overview

The basic setup is something alike Vertex Painting Material setups you might have seen where we use a node to blend the different textures together, but in this one we're actually using a **"LandscapeLayerBlend"** which allows us to do so with multiple different textures.

We can name each of these layers individually and select the type of blending we want for them individually from each other which is super nice for some nice transitions.

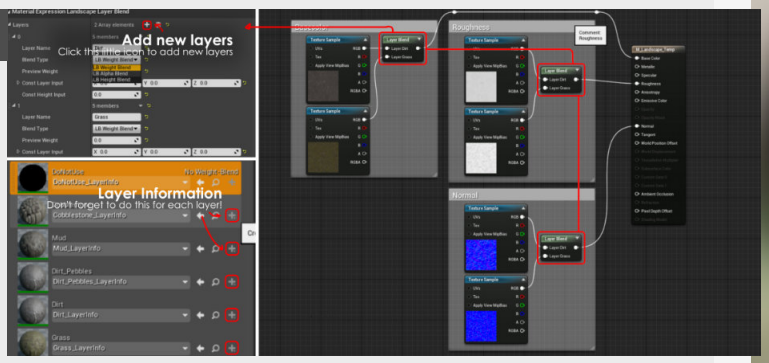


2. Layer blending and Information

The core of the material is going to be all about the **"LandscapeLayerBlend"** node, which controls the blending between all the landscape layers that will be added to it, so we can add more layers as we need to, and then depending on the type of blend (more on that in a bit) we can change how they interact with each other.

Also, for these layers we need to add some additional **"Layer Information"** which you can add through clicking the little plus icon next to each paintable material layer.

additional tip: if you're landscape material is black from the start, fill it with a random layer to give you a base.

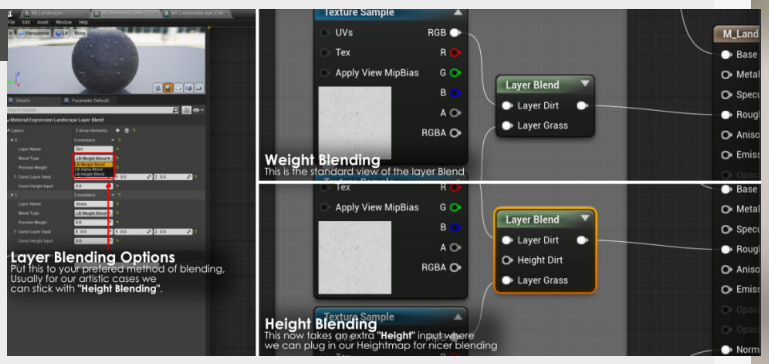


3. Different blend types

Alpha Blending
Is useful if you want a really specific layer order and need that control, this is the way to go.

Weight Blending
Weight blending allow us to independently pain all the different material layers we have, this does come with the standard way of blending, like a basic vertex paint, which works, but doesn't necessarily looks that good.

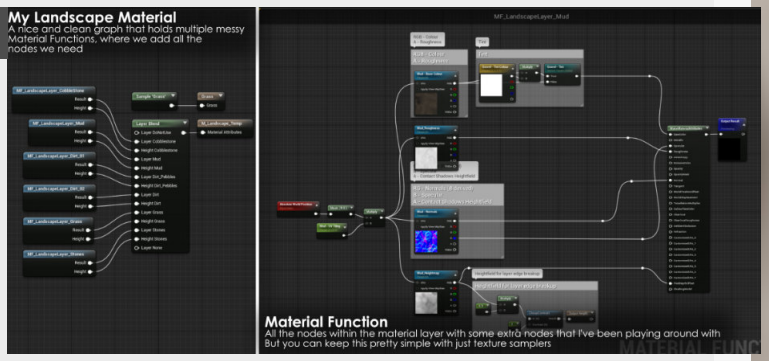
Height Blending
My preferred way of blending and allows for the most interesting look, this blends the layers together based on a heightmap. Which makes it so you can have dirt build-up between stone crevices.



4. Cleaning up the graph

A master material for landscape can become a mess of nodes and lines cris-crossing very quickly. This is where material functions come in and allow us to clean the material graph up a little.

In this case I've setup a new Material Function for every layer of the landscape material (because if you use the same material function it would be controlled by the same parameters, and we want individual control over them). Then all we need to do is make a Material Instance and then assign that to our landscape



Details in landscape sculpting

Introduction

As an environment artists you might get in contact with landscapes at some point of your professional career or personal work, so let's dive into it a little bit ad how you can make the most of your terrain sculpting ventures and how smaller additions can have a big impact.

1. Kinks and blobs

Smaller irregularities can really add a lot to how landscapes look and feel, so for our terrains to feel natural we need to make sure that we also focus on the smaller stuff. Such as avoiding straight lines and large flat spaces, these are some things that are just unnatural looking and feel out of place.

So eve a straight road will some smaller nicks in it, adding these will make them look so much more natural then straight ones. Same for the large fields, add a little bump or dip here and there and it will make it feel so much better, but you got to be subtle with both of them, making your terrain blobby is not great, make sure to think about erosion and water flow to add context.

2. Different height elevations

This applies to the micro and the macro, we've already talked about smaller irregularities but we also need to apply this on the larger scale too. We can use different height elevations for great opportunities to elevate gameplay spaces with more verticality and also just to put extra emphasis on the important of certain locations like hills where people can scout locations as an example.

Now, to take a step back slight elevation changes can also help, you can see this done in Farcry 5 for example between most fields, this is done so you can still see across them and don't have your field of view obstructed by wheat or other crops.

3. Hide things behind corners

... or hidden paths, both are a really great use of fueling a players creativity and usually they don't need that much work. Even a little trail in the sand giving you an indication might be enough to already pull you into a nice exploration route.

And to come back to the title, corners are also great to obscure upcoming area's of interest (and also help with optimization through culling, so you can become better friends with local technical artists) and thus fuel your curiosity drive.

4. Mesh and Terrain Blending

Knowing the limitation of your landscape/terrain density, especially in production environments you're not going to be able to crank everything to the max so this means that you need to find creative solutions to some of these limitations.

And that's where rocks come into play to hide some of these harsher landscape transitions, especially where you have different height transitions over short distances, these area's you just need to fill up with rock or cliff meshes. Another good example for this is the transition to a cave, where the landscape just has a hole in it and then transitions to a different mesh (in most cases). You will def need rocks there too.

Landscape displacement in Unreal Engine 5

Introduction

We're going to be looking into how to get displacement back into Unreal Engine 5, however, at the point of writing this is still an experimental feature, so be careful when using this, it was pretty unstable when I tried first setting it up on my end. Let's dive into it!

1. Project & Addon settings

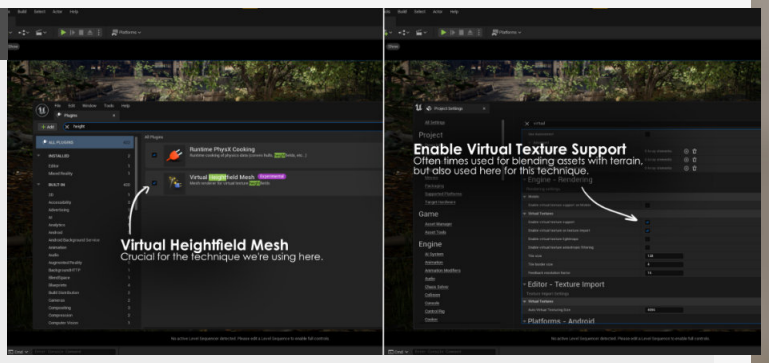
The first step is to make sure that we've got all the plugins and project settings enabled.

PROJECT SETTINGS

Look for "Virtual" in the top search bar and make sure to check **"Enable Virtual Texture Support"** to enable the support for them.

PLUGINS MENU

Next we need to enable **"Virtual Heightfield Mesh"** in the Plugins menu, this is an experimental mesh renderer to create a secondary terrain that support virtual heightfields. But more on that later in step 4.

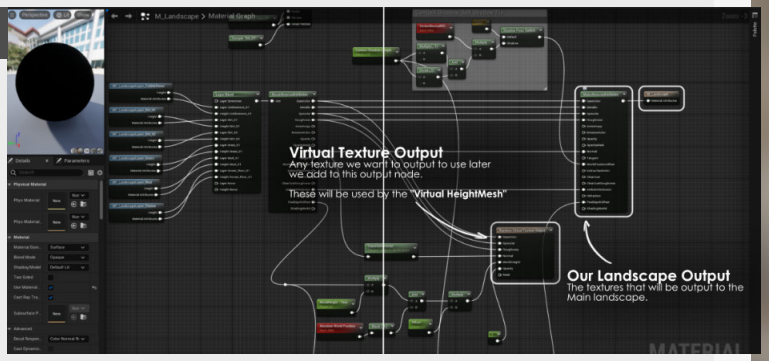


2. Output from Landscape

With the basics out of the way, we can now start putting it together, first we need a landscape with a material that we can then sample from.

In the Landscape material we can then also add a **"Runtime Virtual Texture Output"** node and pass all the information we want to use into it, so for our example here we don't only need the height but also the other texture layers too, because this Virtual Heightmesh can't sample the terrain material we use on the landscape.

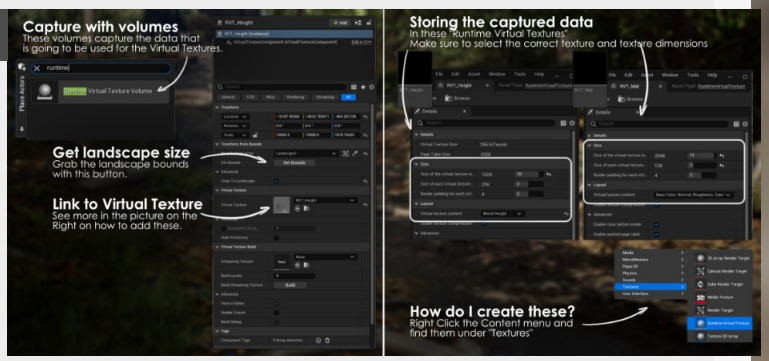
Allright, so at this point the data gets outputted from the landscape, but it still doesn't gets used anywhere else.



3. Virtual texture volumes

CAPTURE: For the capture section we need **2 "Runtime Virtual Texture Volumes"**. Add these from the content menu and make sure to copy the landscape bounds. We need one for the **"Height"** and one for the other texture maps **"Basecolor, Normal, Roughness, Specular"**

STORE: With two new **"Runtime Virtual Textures"** you can find these in the Right-Click > Texture > Runtime Virtual Texture. We need to create one that stores **"World Height"** (displacement) and another one for the other textures to store the data for the material we will assign to our virtual terrain.

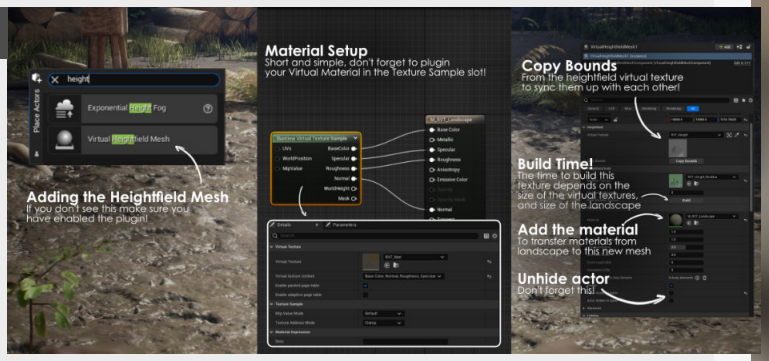


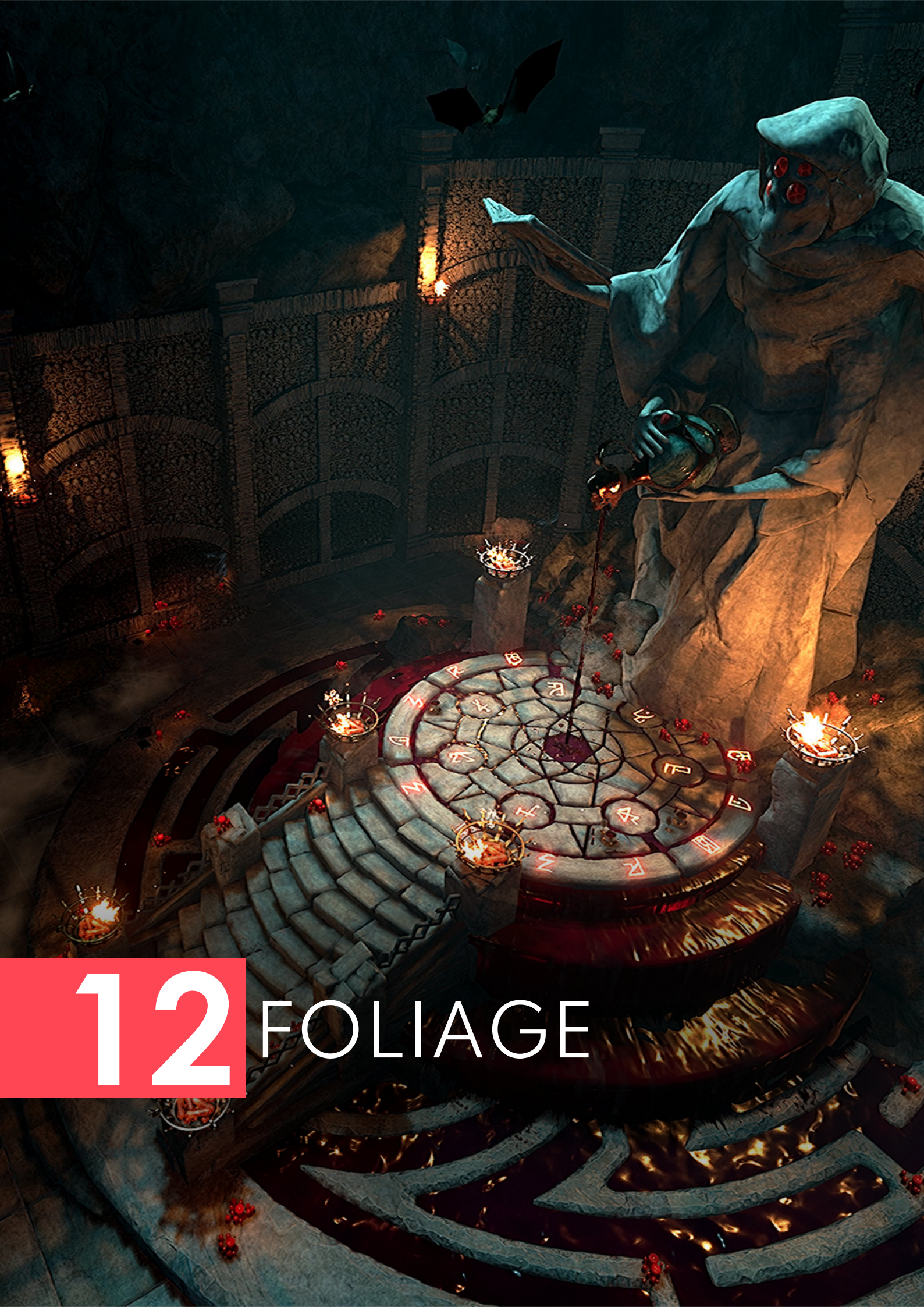
4. Use data for Virtual Height Mesh

We can add this new mesh in the **"Place Actors"** menu and look for **"Virtual heightfield Mesh"** and add this to the world.

By default this mesh is hidden in editor, so make sure you uncheck that tickbox. Then we add the **"Heightfield"** Runtime virtual texture we created before as it's input and bake out a **"Min Max texture"** which will dictate the look of displacement.

Now add a simple material to display our material on this mesh, just make a new material, add a **"Runtime Virtual Texture Sample"** and use the "Material" Runtime Virtual Texture as it's input and add this on the mesh.





12 FOLIAGE

SSS In unreal engine

Introduction

Crucial for getting good looking foliage but can also be used on different materials too, for example curtains or icicles are a really good assets to use them on too. In this one we're going to look into how to set it up and make the shader as well as an example.

1. Subsurface overview

Subsurface scattering will immitate the colours bleeding through thin objects when hit by light. This is most commonly seen on foliage, so if you look at plant leafs they let through light, so SSS will imitate this effect.

Normally when adding this effect we do this with a mask (grey scale) to drive how much of this effect will be coming through.

Then we multiply this with the strength or amount of SSS so we can influence the subsurface on the mesh.

2. A typical SSS Mask

So let's have a look at the masks that we normally use for the subsurface scattering, where the white of the mask will be full subsurface and the black will not have any SSS.

So usually the leaves will have different values of **BLACK/GREY** where the stems are **WHITE** so they don't let through light. (This does depend on the type of plant/foliage obviously.)

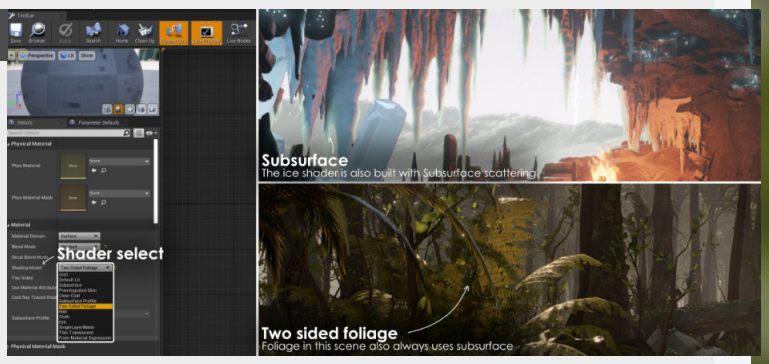
A really good start to generate SSS Masks is to start from a thickness map that is generated from the thickness from the mesh and then use that as a base to start building your SSS map from.

3. Shader setup Unreal

Before we dive into the actual material we need to look at the different shaders depending on the type of object you will use either shader model **"Subsurface"**, **"Subsurface profile"** or **"Two sided Foliage"** which also has SSS in it.

So if you're not making any foliage you are probably best to use **"Subsurface"**.

"Subsurface profile" is special in the way that is uses reusable profiles for how the subsurface reacts to light, so if you are working with different presets of subsurface that you can reuse on multiple assets, then this is the shader you will want to use.

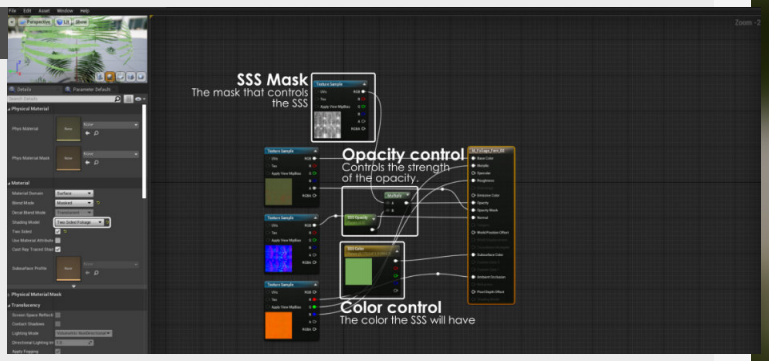


4. Shader example

For the simplest shader setup we can just use the texture into the Opacity slot and Subsurface color multiplied by the color.

Subsurface Strength: stands for how strong the overall subsurface effect is.

Subsurface opacity: This controls the opacity of the strength of the effect, you can see how this can effect the light shining through multiple layers of a mesh. We can also multiply this with a masks to let the mask influence it.



Alpha overdraw

Introduction

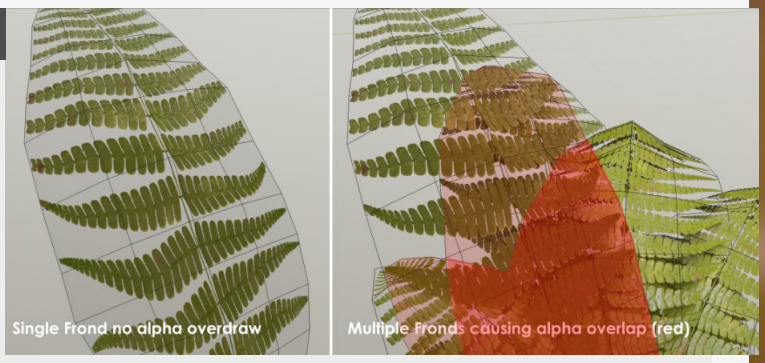
Since I have started doing some tips about foliage this is a topic that we can't avoid, so let's dive into a bottleneck when it comes to making foliage for games, this can be a major issue on performance when you have a bunch of foliage on your screen that uses alpha masks. Let's dive in!

1. What is it?

Alpha overdraw happens when the camera has to render multiple overlapping sections of alpha (opacity) textures.

Alpha mask by itself is more expensive to render because the camera has to calculate which pixel is behind the one that is currently see through, and this only gets worse when you have multiple layers of large alpha (opacity) textures like the example we are talking about, Foliage. Some other examples are VFX of any type of asset that deals with a lot of opacity.

On the right is an example of Overdraw, the brighter the red the more overdraw (and more costly it gets).

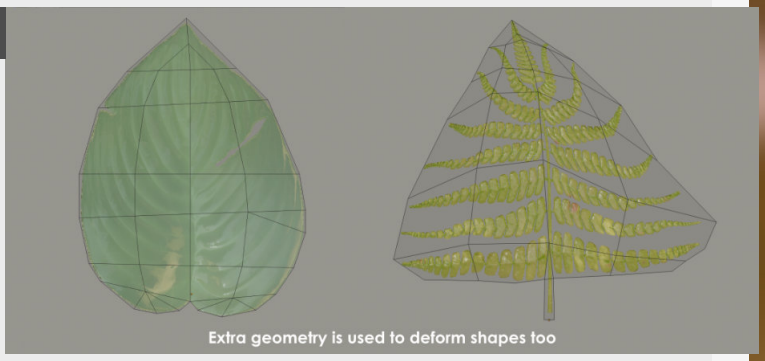


2. Optimize your opacity

For example for a hosta leaf it's better to spend bit of extra Poly's to get as close to the shape as possible because it doesn't have that much of tiny details.

Compare this to a branch full of leaves and you cannot make this entire branch into poly's. Keep in mind that branches are the worst possible because you have hundreds of them on a giant tree and then you scatter that tree in your scene, so adding or removing a couple of polygons can have a large impact.

So it's a balancing act between getting the most out of your shape for your leaf while maintaining a manageable polycount.



3. Plan your UV layout first

So if we are going to use a lot of opacity say for instance on branches, then it's not a bad idea to plan your UV layout first before you start modeling your branches.

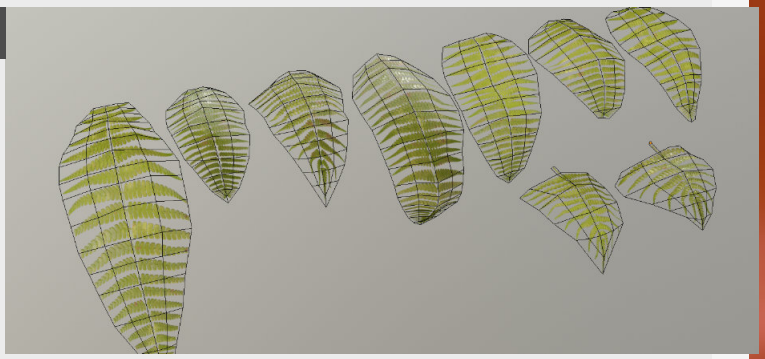
The way that I do this is looking at the shape of the tree/bush and see if I can extract the main shapes from it and getting the variation that I need to create all the different bushes/trees that we want.

So as an example on the right I have a couple of different ferns leaves to get all the shapes and stages of the fern. Keep the overdraw in mind when doing this step and try and reduce it as much as possible when making the Highpoly meshes.

4. Check it in the engine

So now you when you are out there creating your nice foliage, that why you should always check if you have this going on and if it's an issue. To be honest this is a bigger issue when building a full game instead of just your own scene, but it's still good to know that this could be an issue if you let it get out of hand.

Hope this was helpful and will let you run things smoother than my foliage did in the beginning!

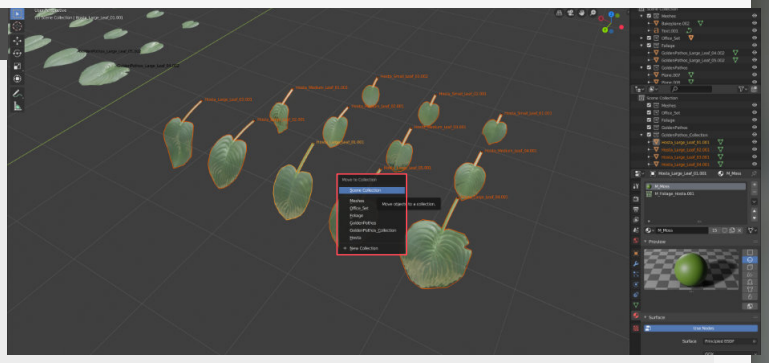


Blender foliage with particles

Introduction

Let's have a look at how you can use the particles inside of Blender to your strengths to quickly create foliage that behaves likes ferns, hosta plants and other low bushlike foliage.

1. Base leaves and collection

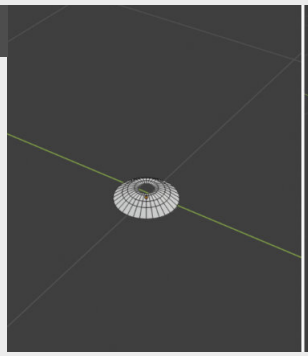


So first off all, for this little example we need some leaves (with or without branches) that we can use to scatter with the particle system.

For my example I have these Hosts leaves that have stems that go all the way to the ground.

So when we have all these different leaves what we need to do next is to put them in a collection, (you can do this with the **"m"** key and then create a new collection) this will help the particle system recognize what it needs to scatter around and this will give us some nice variation.

2. Particle spawning shape

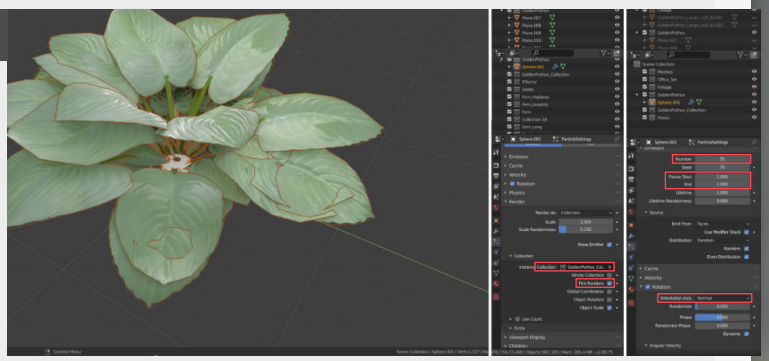


Now that the we have all the different leaves its time to have a look at the bade shape that we will use for the particles to spawn onto.

You can spawn particles in a number of different ways but for this specific example we will be using the normals from the underlying base mesh to define the direction of our meshes.

This is why we create a shape that looks like a squashed cylinder. This will give us a lower angle the closer it comes to the ground plane.

3. Particle settings

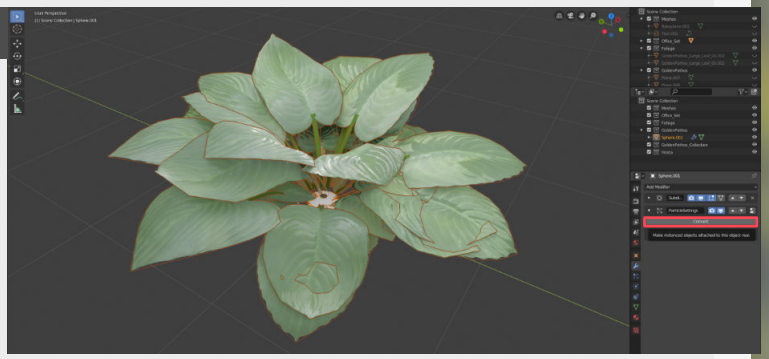


With that stuff in order, let's add all this together, first we select the base shape that we just created and add a particle system to it.

Next up we will change some of the lifetime parameters to make sure that they generate on the right frame (1 for this example, this means that your timeline needs to be at frame one if you want to see something ;D)

Now let's add the collection as our meshes to spawn. If this is acting weird you might need to double check the rotation of your leaf meshes and make sure that they are correct, for this to work properly they need to be aligned to Y+ (in my case)

4. Final tweaks and conversion



Now, there is still a thing that's missing here right? So all the leaves look really flat now, so let's make sure that we add a little more flare to them and add a little drag to them so that it seems that gravity still effects these, the nice thing about setting this up with a particle system is that you can still adjust all the individual leaves on the fly.

Now as the final thing, you are all done setting this up, it time to convert these to actual meshes, for this go into the modifier menu and press convert under the particle system tab. These meshes will still be linked though! So make sure to un-link (or create another duplicate) them first before joining them together.

Foliage in Blender using arrays

Introduction

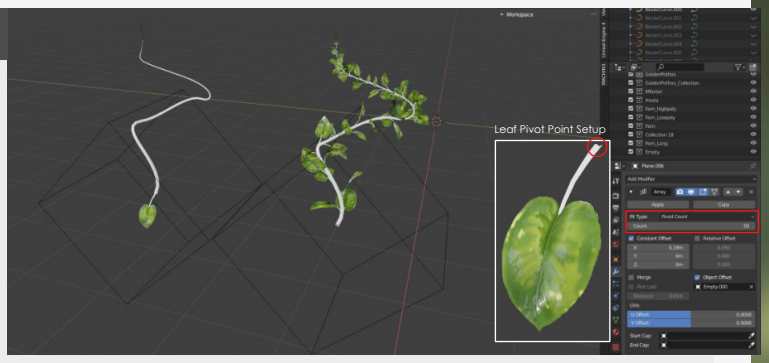
So this time we are going to dig into making foliage that conforms to curves with the help of the curve modifier and the help of arrays. This is super nice to work with hanging vines and other foliage that grows along vines or branches.

1. Pivot and settling up the array

Let's get straight into it, we will need a single leaf or branch to get started and double check that the pivot point is in the correct place or else we might get some weird results.

Then we will add the Array modifier to leaf/branch (this will duplicate a mesh a number of times with an offset of your choosing) then add the number of the amount of meshes you want in the count section.

(We can also set this from count to curve, but we are going to do some additional stuff to it later on, so it might look good now, but those changes will affect how it attaches to the curve)



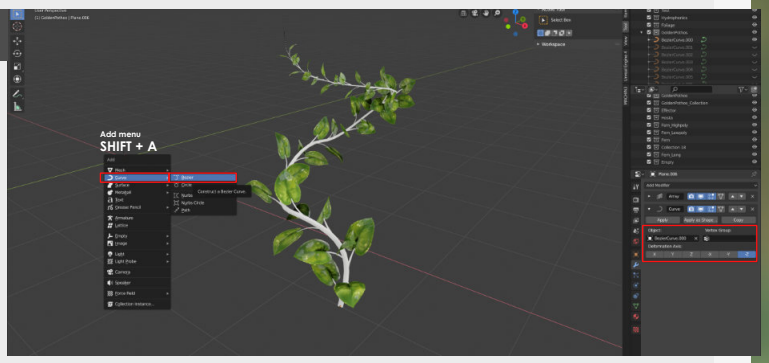
2. Attach the curve

Next up, what we want to do is attach this array to the Curve.

First let's add a **"Curve"** to our scene and move this to the same position as our leaf mesh and make sure that the pivot points are in the same place.

Next we can add a **"curve modifier"** to our leaf/branch and add this curve to the modifier, and in this modifier make sure to select the correct axis that the curve is on.

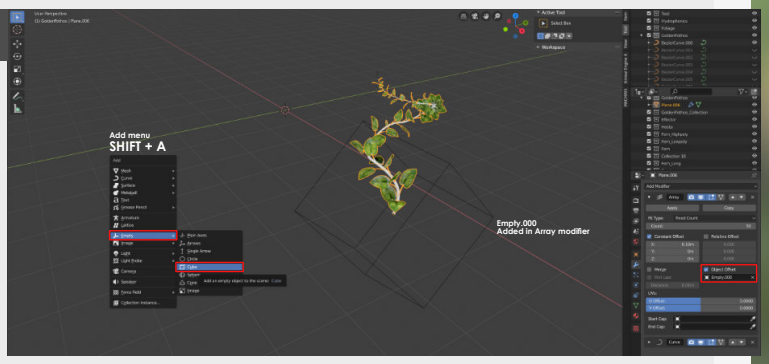
This will then snap the pivot point of the leaf meshes to the curve.



3. Manipulate using an empty

To do some of the additional transformations to the meshes on the curve we can attach an empty to act as our dummy actor for our array modifier. We can just add an empty cube or anything else in that category, doing this will attach the offset, scale and rotation from this empty and will apply this to the modifier.

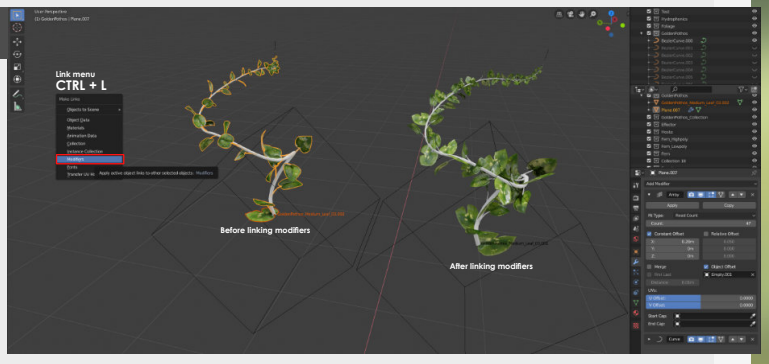
So for example if we rotate this on the Y axis then we get this nice rotation along that axis and is going to give you this nice effect.



4. Limitations

To do some of the additional transformations to the meshes on the curve we can attach an empty to act as our dummy actor for our array modifier. We can just add an empty cube or anything else in that category, doing this will attach the offset, scale and rotation from this empty and will apply this to the modifier.

So for example if we rotate this on the Y axis then we get this nice rotation along that axis and is going to give you this nice effect.



Foliage creation - Starting out

Introduction

Let's dive into some basics for modeling foliage, we will be having look at stuff like planning your UV's in early stages, what foliage to model and more. This will be useful for starting out to create foliage for any of your scenes.

1. Choosing the foliage

First up, let's have a look at which piece of foliage we want to start modeling.

It's important to do this because some foliage might require some additional pieces like stems, branches etc... Because if for example your foliage has stems sticking out from the floor up until the leaves, then are you going to make a separate texture to put these on and allows for a basic shader or are you adding it into your main material?

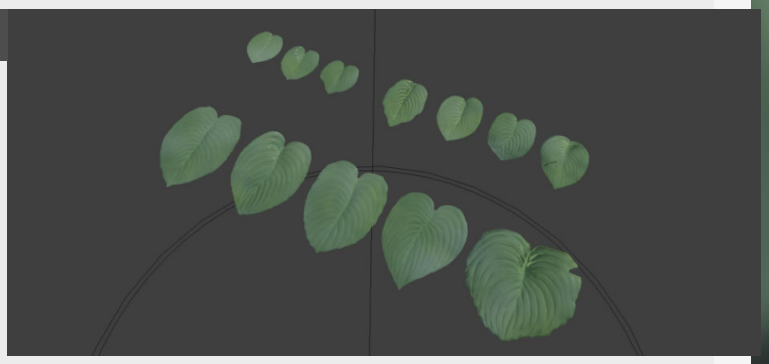
These are all things that you might want to think of and is depending on the scene in general. For personal work examples its less important bit still good to know the optimization tricks.

2. Modeling within restrictions

When starting to model the individual leaves let's make sure that we put the texture boundaries in place, we don't want to start modeling and then have issues with baking them because they don't fit on our texture sheet.

A good way to make sure that this doesn't happen is by adding a square plane underneath to represent the UV space that we will be baking it down onto. Then we always have a reference plane underneath the models that we can refer too.

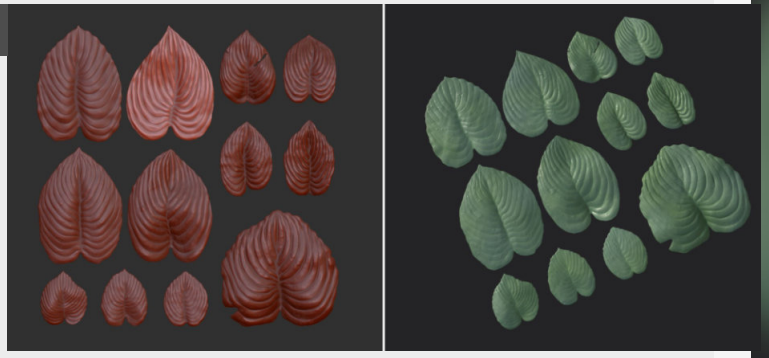
3. Focus on the silhouette



For the actual modeling we need to keep it super simple at this point, we need to be able to adjust the shape quickly if we wanted to. This will also help us to create more variations more easily later on.

Another important thing here is that we create a number of foliage pieces that are varied enough to create a number of unique looking pieces with, we we want to aim for as much reuse as possible and keeping it visually interesting and unique.

4. Adding refinements



To add some refinements we can now start sculpting on top of the mesh we created. In this step we want to focus on adding small sections of damage or larger additions like an adjustment in silhouette.

Don't focus on all the smaller details and folds though, we want to add a lot of those more micro details in the normal map texture through substance painter instead.

Grass creation - Part one (Highpoly)

Introduction

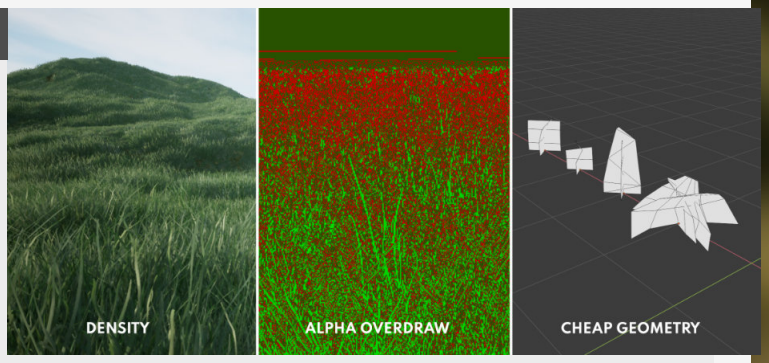
In this mini series I'll take you through the full creation process of creating grass for Unreal Engine, this is going to be a multi-parter. The first section focused on getting good highpoly meshes that we can bake down onto a plane for our lowpoly meshes.

1. Things to keep in mind with grass

Density: The density of the grass clusters you need to make for convincing looking grass. Having bigger clusters can save on the amount of grass clusters you need.

Alpha overdraw: Alpha is still pretty expensive when it comes to rendering cost, especially when stacking a lot of these alpha sections, so try to balance the amount of alpha with optimized geometry.

Cheap Geometry: For grass you need to have a lot of clusters for them to be convincing, especially in open world scenario's. Keep the geometry as low as possible while reducing the alpha,you're spawning a lot of these assets.

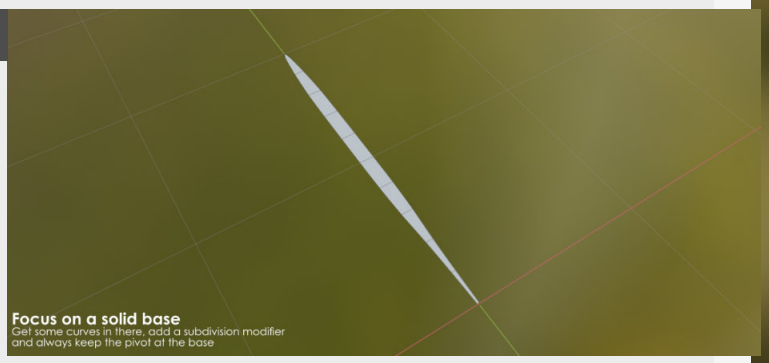


2. Base grass strand creation

Allright, let's get into it. First off we're going to be creating a base mesh for all of our grass blades. This can just be a simple rectangle at first. Add some subdivisions to the mesh so we can control these later and also add a subdivision modifier to make it a little smoother. We also want to tighten the base and the top of the blade of grass to give us a nice shape.

If you're unsure about this step, have a look at some references and investigate the shape of grass.

Always retain the initial straight shape so we can easily duplicate this one and create new shapes from this base, you always want to try and work as non-destructive as possible.

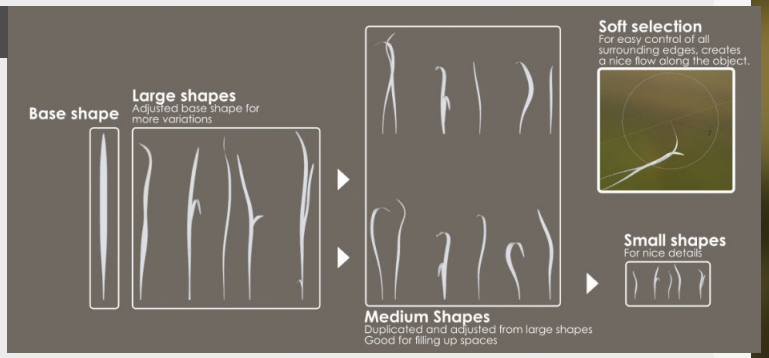


3. Different scales and rotations

For this I'll be duplicating the base shape we just created and then start adjusting the shape with a soft selection starting from the top, just rotating and moving the shape around to create some interesting shapes that we can use later on.

We're doing this step to give us a sample of a bunch of different grass blades that we can then use in the next step to start clumping them together and make some interesting collections out off.

I've found that creating a base collection of five-ish shapes and then scaling /rotating them works really well to create some variation.



4. Combining into clumps

Now that we have a lot of shapes it's time to puzzle them together into clumps that we ca then use to make combinations out off. We're also doing to same with these clumps, we don't want to have uniform looking clumps of grass. Give yourself some variations to really play around with, so in my example I have a wide, tall, small and medium section.

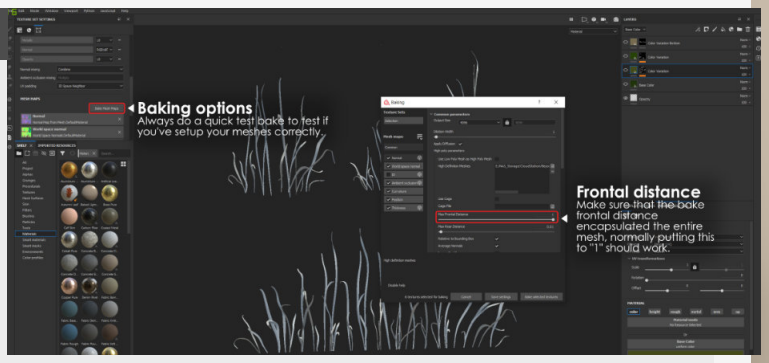
All of these assets are then laid out on top of a plane so we can see how much space each and every one of them can take up, try and maximize this space as much as possible, this will be a 1 to 1 translation of your texture.

Grass creation - Part two (Lowpoly)

Introduction

In the last entry we made the highpoly grass collections that we will be using in these sections, so if you missed those look for "Grass creation - Part One" in this collection. This time we're going to be working on creating the lowpoly meshes and preparing them to use in a game engine.

1. Baking down to a plane



Last time we had a collection of our highpoly grass meshes, one thing in addition to this, is that these meshes are now just a little flat, so first step here. Let's add a "solidify" modifier to the mesh to give it some depth, helping it to give more depth and also helping to bake a thickness map too.

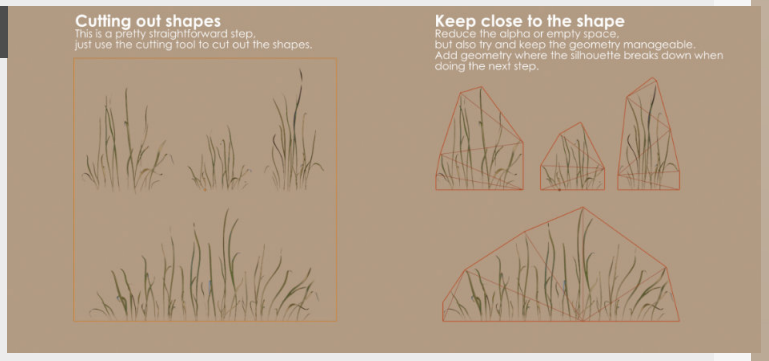
Now for baking, we've written other sections about this so if you need a refresher you can find these in the full collection under the "Baking" section. But for this section it's super simple, export both low and highpoly meshes, get them in your baker of choice and then make sure that the baking distance is long enough.

2. Texture pass

Not going to spend much time on this pass to be honest, from grass we can keep it pretty simple. The main things you need here are colors at the bottom that make sure they blend with the texture you will be putting underneath them on the landscape. and then maybe a slight gradient from the bottom to the top of the grass as well as slight color variations between the individual grass strands.

That should be more then enough for this section, you can take it as far as you personally want. But also, don't spent too much time here, throw something quick on there first, we might be adjusting the shapes later. Also make sure to export these textures for the next step.

3. Creating the Lowpoly

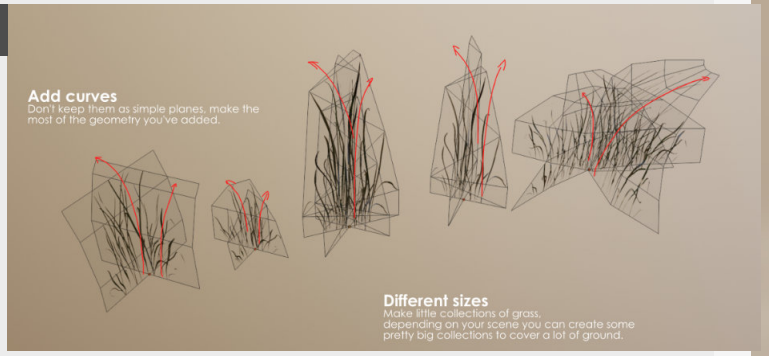


For this we just take our plane that we used for baking and apply the texture on there so we can see where we need to make the cuts to follow our grass textures.

Some things we need to keep in mind, we need to reduce the amount of alpha space as much as possible, rendering this is quite expensive, especially when these alpha sections of the textures overlap with other alpha sections (Like when spawning a lot of grass). But we can't use too much geo for this either, we're literally going to be spawning 100's if not 1000's on screen.

This is also why we made the these in a rough triangle shape, which helps with optimization.

4. Making combinations



We can't really use these flat planes with a texture on it in our final environment, we actually want them to feel as three dimensional as possible. So let's adjust them to make nice little collections of grass that are going to look nice.

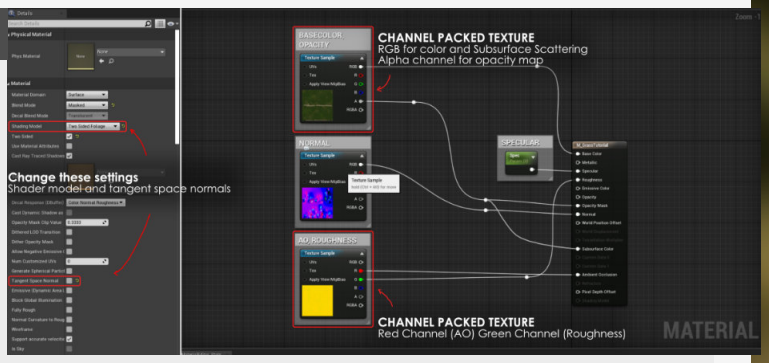
In this step it's important to make it as full as possible from every angle, also when looking down on it, so make sure you deform it along every angle.

Grass creation - Part Three (Engine setup)

Introduction

In this last part of Grass creation, we're diving into the material setup inside of Unreal Engine. Making sure that our shading looks correctly on large sections of grass as well as implementing wind and how to paint the foliage.

1. Base Material setup

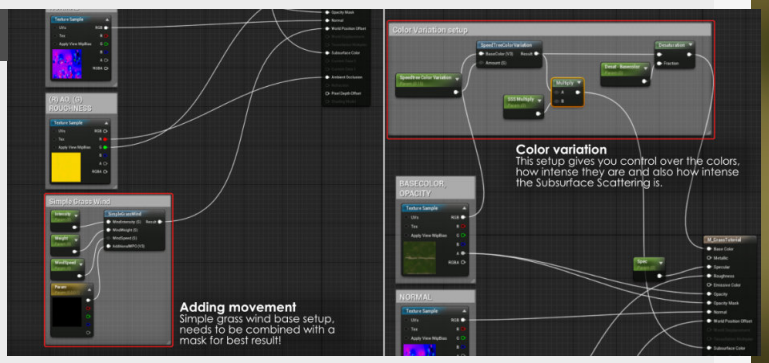


We start simply by importing our texture into the engine, which you can export from Substance Painter. For a base setup we can just set the material to be **"Double sided foliage"** under the Shader options, then we can drag in all of our textures and link them up.

Aside from this basic setup, we can also add a simple parameter for the specular, I like to tweak this option visually, especially for grass.

Last thing we need to do, specifically for grass is tick off the **"Tangent space normals"**, this will give the grass a fuller look then without it.

2. Adding some movement/variation

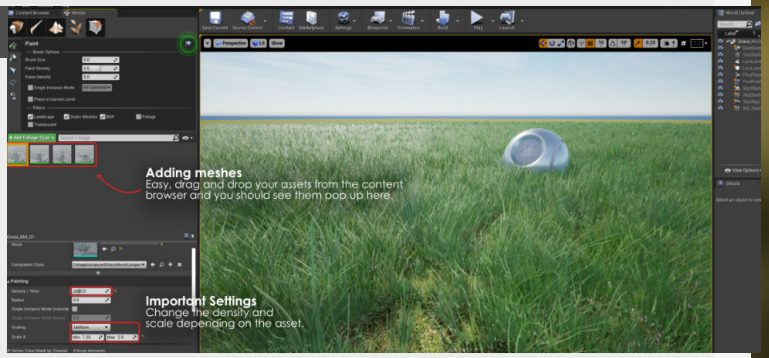


Look for the "Simple grass wind" node, hookup all the inputs and set them up as parameters so you can tweak them on the fly (more on this can be found in "Material Parameters" in our compilation). The main thing you're controlling here are the weight and the intensity.

Make sure that you paint a mask in substance for this and export it as a separate texture, this is usually a gradient from the bottom (black) to the top (white). Thus affecting the top vertices but no movement on the ones on the ground.

Also, add some variation to the color! check out **"Speedtree color variation node"** in our compilation.

3. Painting foliage



For this we use the **"Foliage"** tool, which allows us to drag in assets and then let's us paint with them with a brush, super cool!

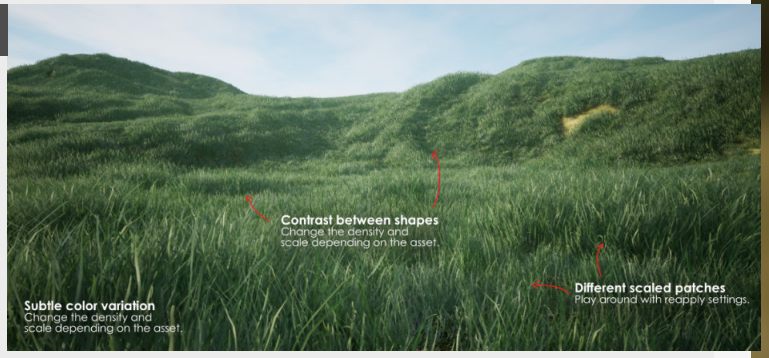
Every asset you drag into this tab will have it's own settings that you can tweak.

Density: Controls how much foliage will be spawned close together, so larger patches should be less dense and smaller patches more dense.

Scale: If you have build your foliage to scale something like Min: 0.75 to Max: 1.25 is always what I try first.

"Foliage tool tips" in our compilation can help you out more with this.

4. Paint your foliage with love



Use randomness
Use the randomness in scale but also slight rotation to your advantage, this really helps create a good chaotic feeling.

Different settings for different assets
Different assets need different settings, use smaller assets to break up the repetition of larger ones.

Reapply settings for different density settings
Use the reapply settings tab to increase/decrease the density of the foliage.

Clear up paths
If you have people walking through this grass, make sure to clear paths for them. This can also be used for player/viewer leading!

Speedtree color variation node

Introduction

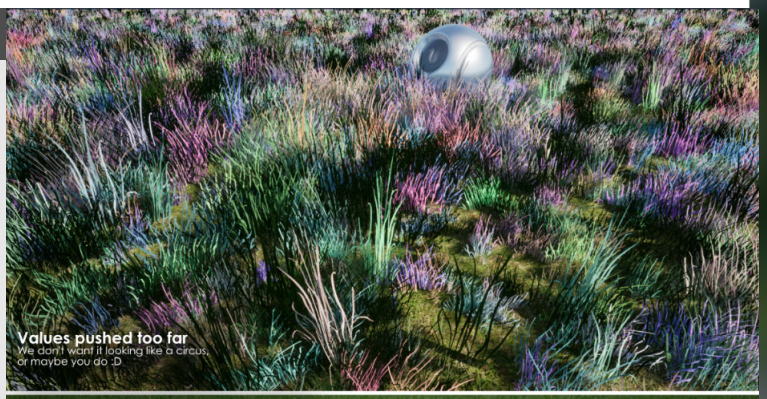
So today we're looking into how to create color variation within unreal engine with a handy node from Speedtree that allows for world based color variation. But we will also be diving into how it's normally done with foliage, maybe not on a world basis but on a more local level.

1. Easy mesh based color variation

Unreal has a pretty neat node that allows you to have color variation, it's pretty simple but it's also a bit limited in terms of control too.

The node we're talking about is called **"Speedtree Color Variation Node"**, as you can see in the example it's pretty straightforward. Like in the example we add this between the output of the basecolor and the texture we feed into it.

The main thing to keep in mind when using this node is to keep is super subtle, for this example I use a value of **0.015**.



2. Don't forget about SSS

Sometimes the end result when the light comes shining doesn't look like you we're expecting looking at the basecolor so we need to make sure that we add this to the Subsurface Scattering too.

The example on the right is just an example, again, the key to getting a good result is about zooming out and focusing on the large scale instead of the small scale.

Foliage Master Material Example

Introduction

Since we've been talking about master materials and material instances for a couple of entries, so let's dive into an example I use at the moment to do all my foliage for my current scene. There are some minor adjustments I might make to it on the fly depending on the workflow and need, so keep it that in mind.

1. Foliage shader breakdown

Texture inputs

All of the texture inputs are setup as parameters, this allows us to change the texture inputs for different types of foliage, from grass to the ivy that I'm using. The **"Basecolor"** used the RGB channels for the actual color of the texture made in substance painter, the A(lpha) channel is used for the opacity map. Always try to lessen the amount of texture samplers.

Then we have two texture inputs one for Ambient Occlusion, Roughness and Metalness. All channel packed into one texture, technically to optimize it more I could add the wind influence map you see on the left to this texture but I decided to make it a separate input so we can toggle it on/off easier. And another texture input for the normal map, which speaks for itself.

Color variation

Then adding onto the basecolor we can add world location based color variation using the **"SpeedtreeColorVariation"** node, allowing you to add more variation per instance in the world. For more information you can check out our separate blog link on that topic.

We can add some additional control here if needed, like for example overlaying a new color on top of the basecolor to change it a little or even desaturating it, but I choose to keep it pretty light here.

Subsurface scattering

For this example I'm using a simple setup for the subsurface scattering, purely driven by the basecolor and some adjustment values. However, it's nice to have a **"thickness map"** to help you give a more accurate representation on how the subsurface would react to light.

A thickness map is something that substance painter bakes and if you have a good highpoly with thinner and thicker pieces you can use that one straight out of the box, but you can also just make one based on masks too (or paint one yourself).

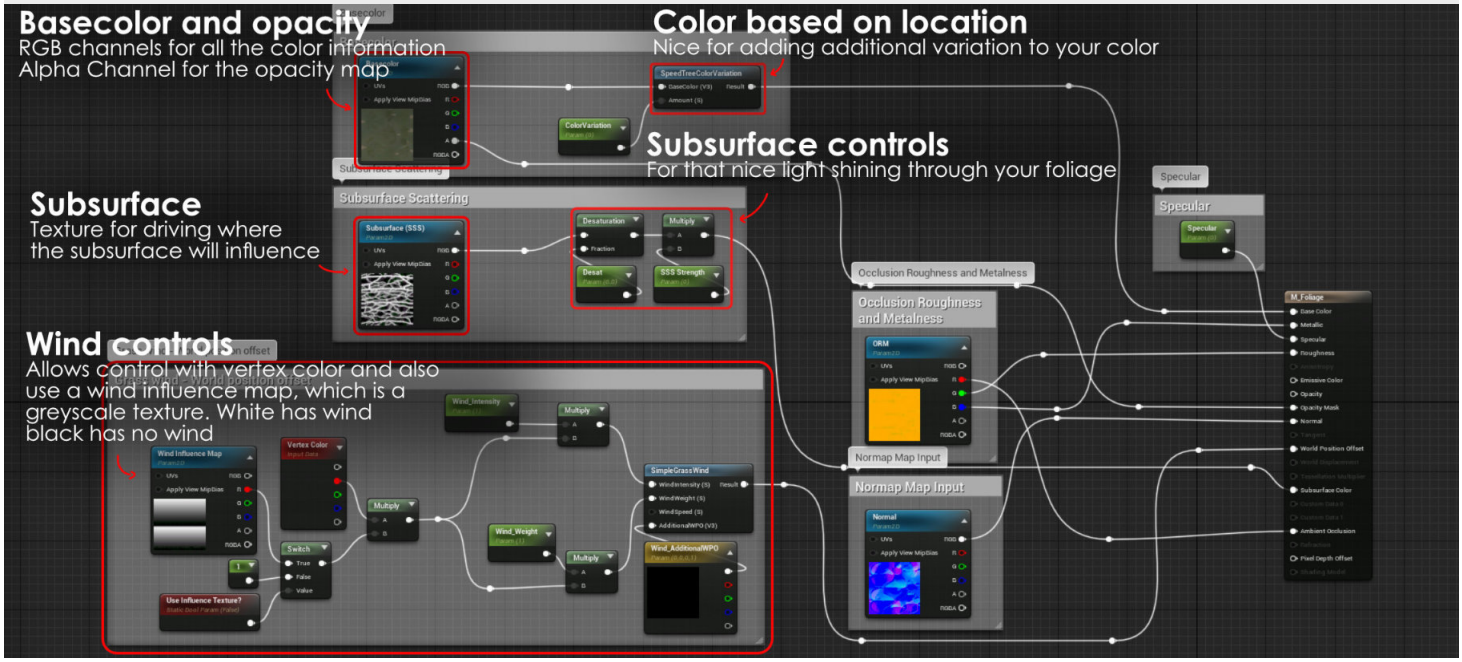
I like to built in some additional color control with a **"Desat"** node.(you can go into the minus to push the saturation too) And then finish it up with a **"multiply"** to control the intensity of the subsurface scattering.

Wind controls

Then lastly, let's get some wind to affect the foliage as well, in our case here this is driven by a mask and then multiplied by the vertex color. This gives us more specific control over where the wind will impact the foliage. Then we multiply this again with the control parameters for both the wind **"Weight"** and **"Intensity"**. Thinking about it now you could probably get away with only doing the multiply on the "Weight".

Parameter recap

All off the above gives you a lot of control over the look and feel of the bits of foliage you will need for your scene. There are a couple of expansion that you would do with adding more individual control over the color for example with some desaturation nodes if you want.



13 LIGHTING

2 point lighting setup

Introduction

Let's dive into some of the basics when it comes to setting up your first lighting setup, which will usually be something like this simple 2 point lighting system.

The scene used in this example has more lights then two, but is still a good example for this kind of setup.

1. The basics of 2 point light

This is the most basic setup to get a good presentation in terms of your props, but it's just the start. So see this more as a base to start working from and adding your own bits to it rather then the solution that always works by itself.

It works with two lights opposing the subject, the key light and the rim light in this case.

Usually when doing this sort of a lighting you want to add some slight color variation into the different lights as well, to make sure there is a nice little bit of contrast in the lights themself.

2. The main Light

This is the light that is going to put the most light in the scene and will pull the main focus of the attention.

Like in portrait photography you want this light to fill the scene the most and will be along the same side of the object as your camera will be, this will make sure that the side you are focusing on will be nicely lit.

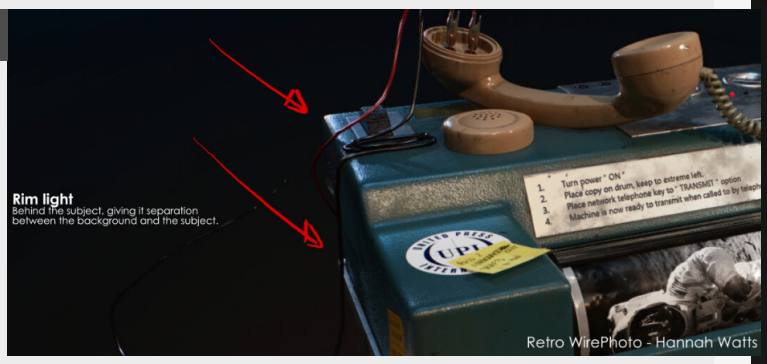
In the picture on the right you probably notice that there are multiple lights on the subject, but let's focus on the white one, which lights up the scene.

This for our example is our Main or Key light.

3. The rim light

This light serves the purpose to separate your subject from the background, which is why it's places behind it as well. This will make sure that there is a nice **"Rim"** light that catches the sides of your subject to make sure it stand out from the background itself.

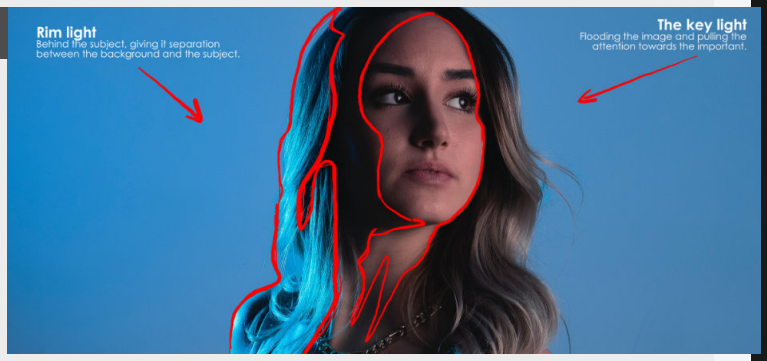
In our example on the right you can see the Rimlight on the left. Which is blue and nicely separates the subject from the black background.



4. Additional information

These setups are really common in Portrait Photography, so if you want to get more inspiration, you can always start and break down how they setup studio's for specific lighting situations, which is always a good inspirational resource for your own lighting setup.

And lastly to reiterate, use this as a base rather then the final product, each prop presentation needs it's own setup.



"Light painting"

Introduction

This is probably my favorite way of talking about lighting, where you look at your object like a canvas and you deliberately start painting with your light sources to make things stand out, so let's dive in!

1. What to highlight?

The first question you need to ask yourself when looking at your models or even your environments is what do you want to highlight? Or in other words what are you as the artist put focus on so people will put extra attention to this.

For most of your props this will be area's where you show off nice roughness variation or spots where you have nice transitions between textures, those are always nice to highlight.

Annie Nazoyan - Shotgun Basic Material / textures

2. Putting in a good base

Before we start adding in those nice little spots that will pull the attention, we will need to start with the base itself, we did do a tips like this one on a two point lighting setup that you can check out as well.

But basically we build a setup that fills the space with enough light to make it visible to the viewer, which you can do through an HDRI image or add in your light manually. The HDRI has the added benefit of also giving you a nice environment to reflect your materials back into.

Georgian Avasilcutei - Real time boots

3. How do we highlight it then?

So you kind of know what you need to highlight, but HOW do you go about this?

It will take some going back and forth to figure this step out so make sure to experiment a little.

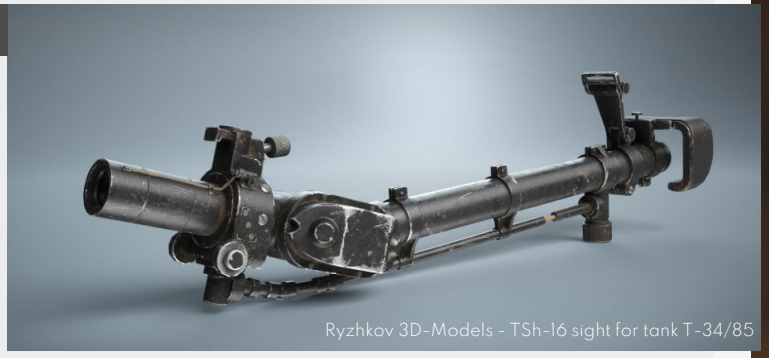
Usually the main thing I try first is putting a light right across the other side of the object so that your light will bounce of the object and roughly in the direction of the camera.

Jarad Lewin - Crypt of the Raven King

4. Don't go overboard

There is definitely a drop off point where the more lights you add the less they contribute to the lighting, you don't want to add so many lights to your environment that it's completely filled with light. You need some contrast in there!

So if you got your fill lights (or HDRI) then try to stick to maybe one or two lights to really pull the focus towards places where you want people to focus on.



Ryzhkov 3D-Models - TSh-16 sight for tank T-34/85

Tonal values

Introduction

Today's tip will dive into a little bit of an expansion onto the topic of your tonal values in your art, which are the black and white values of a piece of art, really distilling the information down to the core and just focusing on how tonally different they are.

Another section on Tonal Values can be found in "different contrast uses"

1. Overview

Differences in values are an amazing tool in the artist's toolkit that we can use to guide the player or the viewer to points we intend to focus or move towards. This usually works in tandem with other types of contrast but for this one we're focusing purely on Values.

Tonal Values determine how dark or how light something is, which also is a great way of separating different foreground midground an background layers.

2. When do we check it?

These fundamental rules (of which tonal values are a part) are usually something to keep in mind at all points when working on any artistic piece. But if you want to have some solid "Checkpoints" when it's good to check on it are when you're doing the blockout of your environment or more towards the end once you are preparing your final presentation.

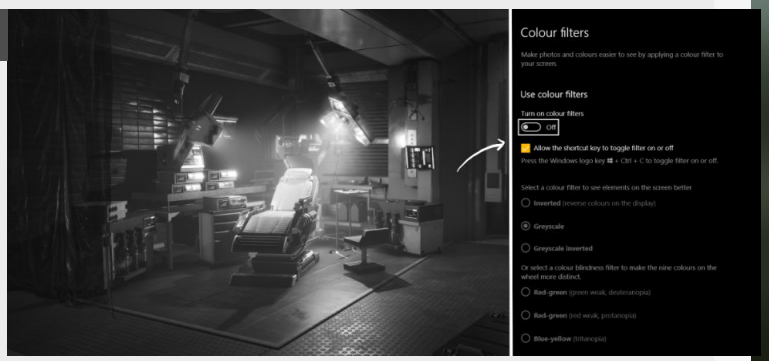
It's something that you will get more and more used to the more you apply these rules, you want to get to a point where you don't even think about it anymore and it becomes a part of your artistic habits.

3. How do we check it?

The most used way is to check it in Photoshop and adding a black and white filter on top of it or removing all the saturation in the image. An additional step to really help you see things clear is to also add a blur to the image.

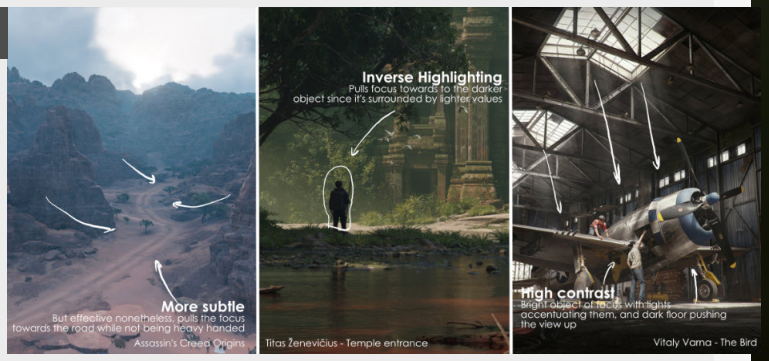
BUT the easier way to do it to go to your windows settings and under **"Colour Filters"** you can find a Greyscale filter, you don't have to go to this menu all the time, just enable the hotkey for it and you can swap it on the fly.

The standard hotkey is
WindowsButton + Ctrl + "C"



4. Some examples

Let's dive in and take a look at and example here in how to make things stand out. There are multiple ways of doing so, usually the lighter object is surrounded by darker items, but you can also flip this comparison if you wanted to as seen in these examples.



Fake lightshafts in Unreal Engine

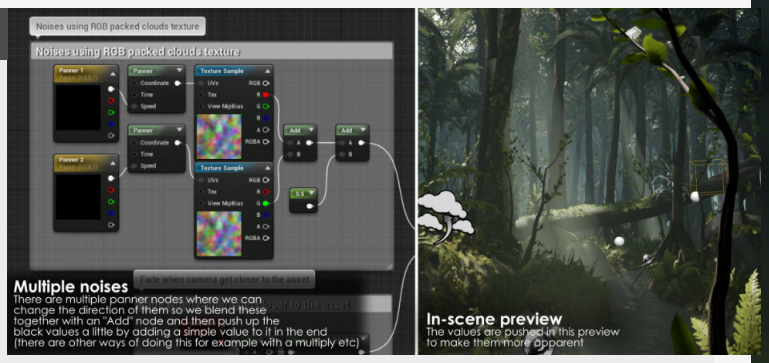
Introduction

Now that we discussed Baking it's time to dive into the next step and start talking about Texturing, but first we need to take a look at the different maps that are used in the texturing process, so here we go, let's dive in!

1.Combining noises

In this first step we take a cone shape and then start combining a couple of noises in Unreal engine material editor using an RGBA Noise texture that I made, these are 4 simple noises added in each of the individual channels for this texture that we can then use for multiple applications, but for this one I added them to create a little variation in the light, giving the impression that there are volumetrics floating in the air, to add to this we also want to simulate movement, which we do with the panners that you can see in the screenshots.

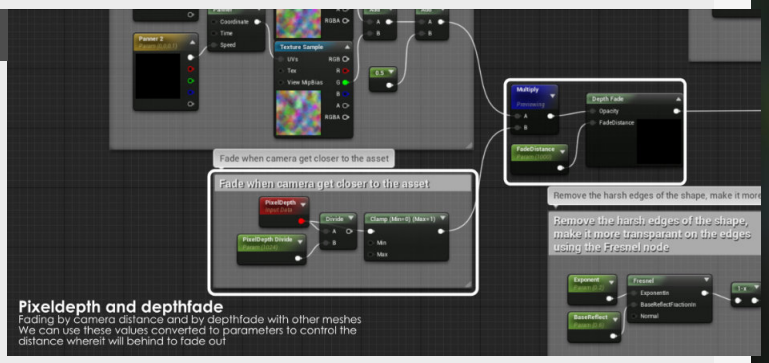
The main thing that is important for this step is keeping it subtle!



2. Fading by depth

The next step is all about cleaning up the edges, fading by camera distance and by distance of the meshes that intersect with the volume, there is a lot to unpack here so I can't really do a deep dive into all the topics here. The pixel depth will give you the depth of the scene based on the camera location so we divide this by 1024 (at first) and this will allow us to change the distance where this will blend.

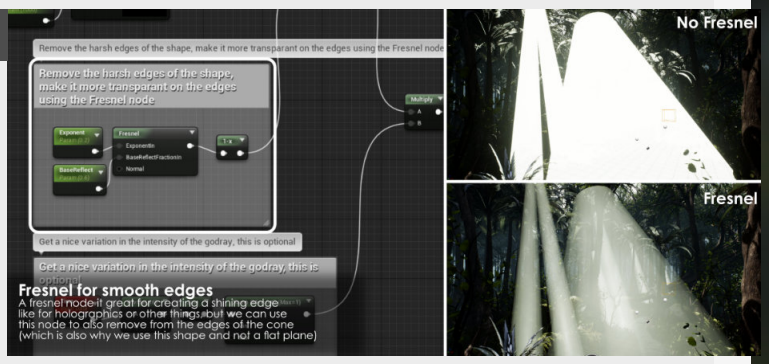
Now this is more an optional thing but it gives you a nice results with extra added depth and this is the depth fade node, this will allow you to soften up the result when other meshes are within this mesh. But before tweaking these parameters we need to add some more stuff!



3. Fading on the edges

Now let's look at smoothing these edges so we don't have this harsh volume in the middle of our scene. We can use the Fresnel node to achieve this effect, you can see what this effect does in the little screenshot, but keep in mind it's inverted with the 1-x node, because normal the Fresnel node will "highlight" the edges and make the core somewhat darker so we can perfectly use this effect to do just the opposite if we tweak the values a little, so that's exactly what where doing here.

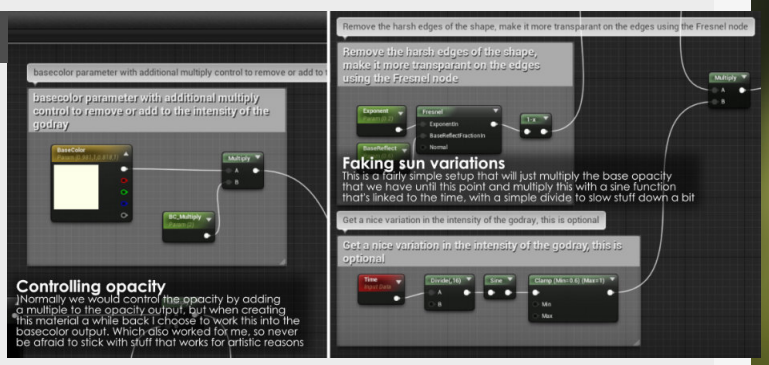
The exponent controls the power and the base reflect controls the balance between the edges and the core of the materials, turning these into parameters makes it easier to play around with.



4. Sun intensity and opacity

We are going to make the opacity tweakable and add a little fake sun intensity change to the material as a whole. Normally we would add the opacity control to the end of the Opacity chain, but in my case I decided to add it to the Basecolor for some reason, you might want to stick to adding it to the opacity chain.

To create some variation to showcase in the video I made of this scene in the end, which is why I added a Sine function linked to the time of this scene (with a little divider to control the speed of the sine function) and the clamp node at the end will allow us to control the min and max value for this material (from 0.6 to 1.0 in my case)



Unreal Engine light mobility

BeyondExtent

So when getting to lighting your environments inside of Unreal Engine there are some things you need to know about lighting itself, because there are differences in the way that you will use and set them up, so let's dive into this.

1. Static lights

So first types of lights are usually used in types of environments that need a bit more accuracy in lighting, shadows and Global Illumination. This increase in quality does come with the need to calculate where these lights will bounce around in the environment, which can take a while with bigger environments.

Also, this information gets baked down into lightmaps, which is a just an additional UV (auto generated by Unreal, unless ticked off). So this is an additional element you need to keep in mind, and need proper density for it to look good. But more on that in a future weekly tip.

2. Dynamic

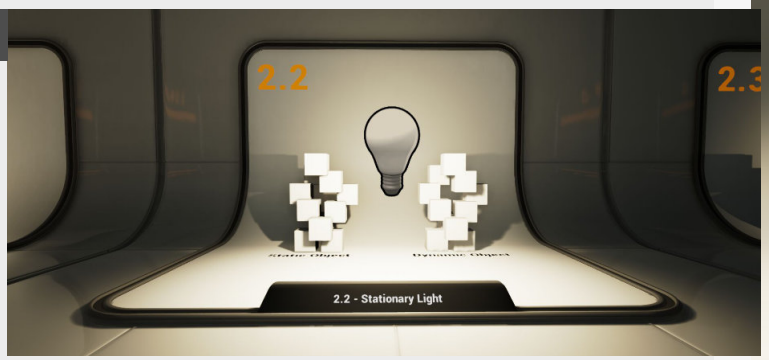
Then we make the jump to Dynamic lights, these are the opposite of static and can be moved on the fly and the "preview" is what you get. This sounds great right? but hold on because these do some with some downsides as well. First off all the limited amount of them you can use because each overlap adds additional costs, so this is where the red cross comes from if you add a lot of them in your scene.

Also there are some big limitations on how the Global illumination gets calculated through a system called "Light Propagation Volumes" which isn't the most accurate, especially compared to static lights.

3. Stationary

So now we get to the in between option, these apparently have the highest quality which is something that I didn't really know either! But what makes these light special is that like static lights they get baked to the lightmaps. However, you can still change the color and intensity of the light on the fly.

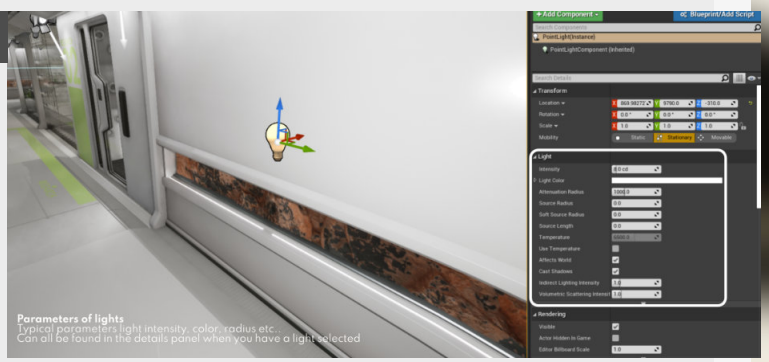
However, this will only affect the direct light and not the bounce light tho. Also this light makes use of "Distance Field Shadows" which if enabled can add to the resolution of the lightmap and make this one seem less blurry.



4. Light parameters

So let's dive into some of the parameters we can adjust on lights and add to some quick tips for this. Most of them really speak for themself, but the biggest impact on performance will definitely be the influence radius, the more these overlap with other lights the more expensive these get either to render or to calculate.

In addition to this on some types of light you can change the source radius and length, making it perfect for some sort of tube lighting for example.



IES Profiles for lights in Unreal Engine

Introduction

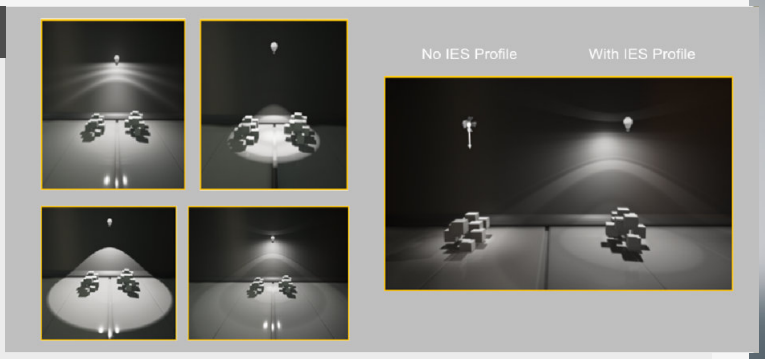
IES profiles are based on real world examples for lights and are stuff crucial in pushing the believably of the lights inside of 3D spaces, especially for architectural visualization.

1. IES Profiles

IES profiles are used to change how a light affects an environment, mostly used in Architectural Visualisation but can be useful to change the light on any light for any scene.

What they basically do is change the way that the light behaves from a normal uniform light to a more realistic light that has different falloffs and spreads the light differently than other standard lights.

These profiles are based on real world values by different manufacturers of those lights.



2. Where can you find them

You can download them for free on multiple sites and each site offers different profiles to pick from.

Some examples of these Profiles can be found on sites such as Philips or Pixar Renderman site for instance and Evermotion has a download link for a zip folder containing 58000 of them.

https://renderman.pixar.com/ies-profiles

http://www.derekjenson.com/3d-blog/ies-light-profiles

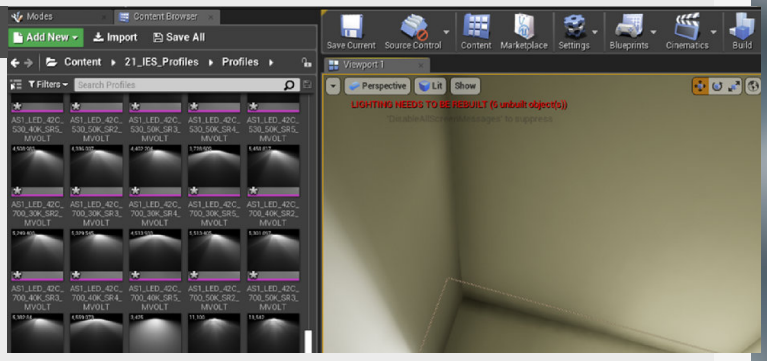
https://evermotion.org/articles/show/10234/download-ies-light-profile-collection

3. Importing them

Once you got them downloaded you can import them like you normally would do into Unreal Engine.

If you have them inside of the engine, I suggest that you don't modify them because they are based onto real world values, so using them as an anchor for all the lighting in your scene

That being said, you can control the brightness for each of the IES profiles separately in the texture properties.

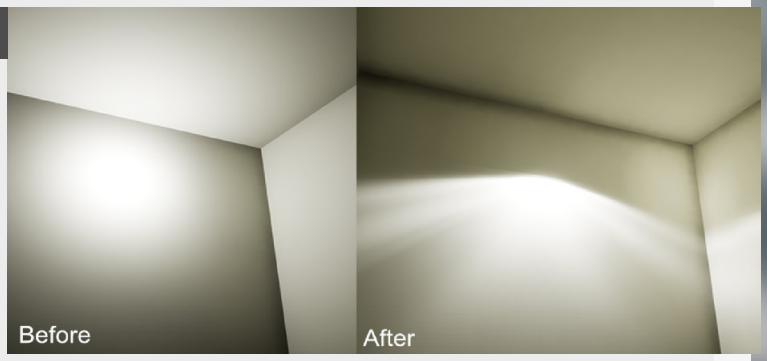


4. Putting them to action

Then apply to a Pointlight under the "light Profiles" section, and look at your newly adjusted lightsource!

You can also adjust the brightness of the light like you normally would for another light in the light panel.

Also, don't forget to tick the "use IES Intensity" in the details panel for the light itself.



Rectangular Lights

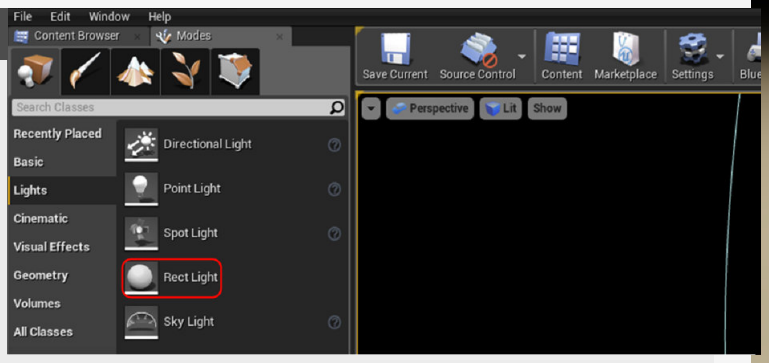
Introduction

Rectangular lights are perfect for certain use cases because they project light in a different way and are a bit more expensive because of that. They are super useful for rectangular light frames, projector screens, tv screens etc...

1. Let's get started

This week we have a look at a new sort of light introduced to Unreal Engine 4.20, the rectangular light.

This light is perfect for square lighting situations, such as TV screens, projections, Lighting fixtures, etc...

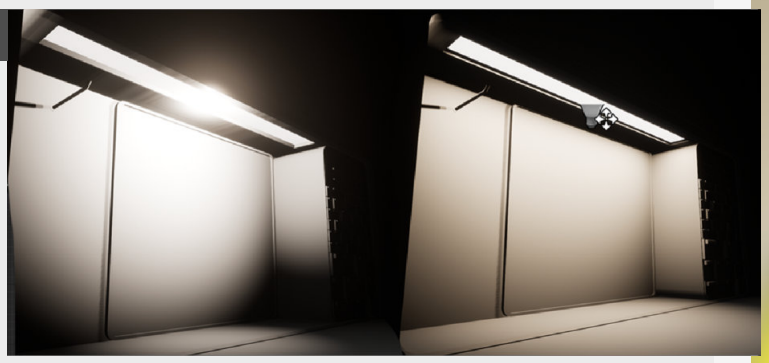


2. Comparison

Let's look at the comparison between a normal Point Light and the Rectangular light on the right.

The first thing we can see is that the rectangular light has a softer falloff, but you will also notice that it still uses a normal round falloff as a normal Point light does.

You need to be aware that the Rectangular light acts really different depending on the type of mobility you set it to, more on that in the last section.

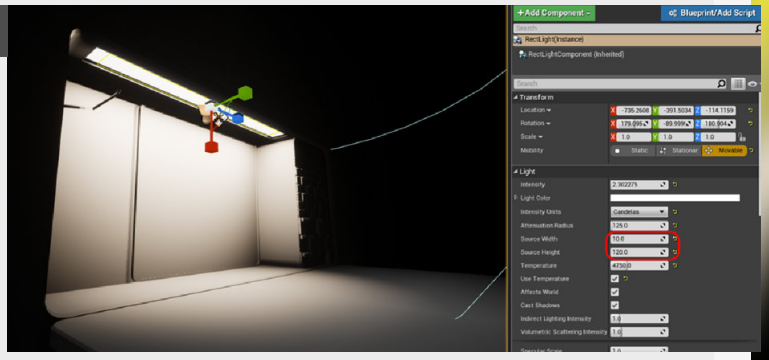


3. Settings

Let's dive into the setting for the Rectangular lights, this is really familiar to other light sources.

You have things like your normal Intensity,color,attenuation radius, etc...

The main thing that is going to be different compared to normal lights is the Source width and Source height.

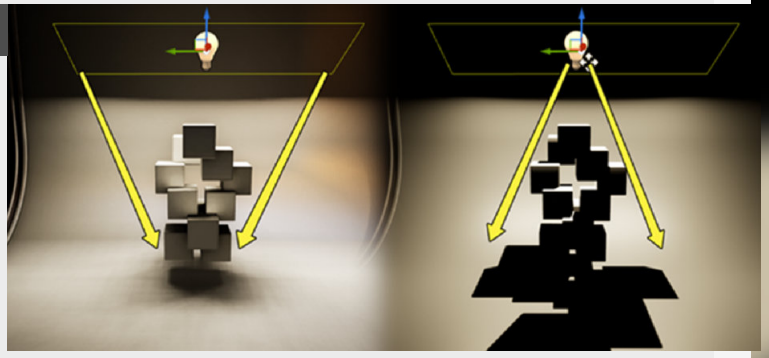


4. Light mobility

Static: Direct and indirect illumination are both baked to lightmaps when building lighting. Fastest rendering, but no realtime adjustments.

Stationary: Only the indirect illumination from the light is baked into lightmaps when building lights. GI and Shadows are static, but color and temperature is changeable.

Movable: With this setting, the light is totally dynamic. Most expensive to render, this allows for correct shadows from moving objects in the game at runtime. every frame.



Light propogation volumes (Outdated)

Introduction

If you are looking to boost your lighting and especially the bouncing of the lighting in the scene, you can do this by using static lights, but what if you are looking for some options do do it realtime? There are some options such as Voxel based global Illumination (VXGI) but today we will be having a look at Light Propagation Volumes.

EDIT: This is now replaced by Lumen in Unreal Engine 5

1. System overview

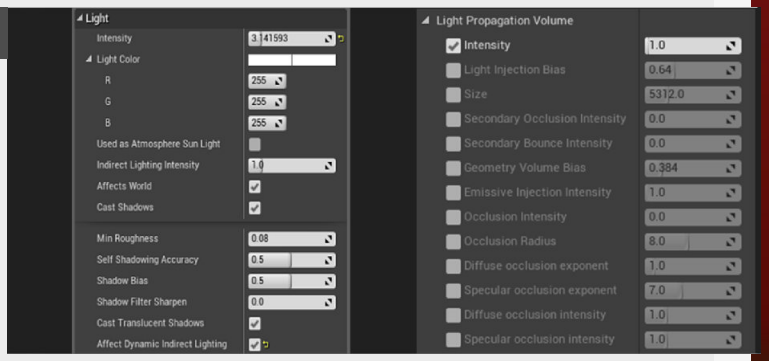
Allright, a lot of people have been wondering about this feature which is in Unity as Enlighten and is better known as Dynamic global illumination. So, if you go to your Unreal Engine folder and look for the "ConsoleVariables.ini" file which holds features which can be enabled.
In this file at location (C:\Program Files\Epic Games\UE_4.16\Engine\Config) look for the "r.LightPropagationVolume " and set this to 1 and restart the engine.

2. The settings

A couple of important settings that you need to keep in mind is that you need to set the affecting Directional light's mobility to "movable" and also tick the "Affect Dynamic Indirect Lighting" tickbox before you can see the changes affecting your level. Post processing volume holds more settings to further adjust the Light Propagation Volume, including the size, intensity, etc...

The Size and Intensity might be the most useful settings here.

Also, the LPV bias can be adjusted per mesh under "LPV Bias Multiplier"



3. Performance and visuals

Due to the experimental nature of the feature itself, it can quickly become cost heavy if not properly balanced.

Firstly we need to disable the Lightmass GI, because if we are going to use them at the same time, they result in a too bright indirect lighting.

4. Debug and materials

Firstly we need to disable the Lightmass GI, because if we are going to use them at the same time, they result in a too bright indirect lighting.

Additionally if we want to change the color that the global illumination is going to bounce of one of the materials we can switch this using the "GIReplace" node in a "Material Expression"

14 POST PROCESSING

Color correction: LUT

Introduction

LUT's are a great way to do color correction on your scene, you can add these to a post processing volume or multiple volumes in your scene and control them all separately if you need. But for the most part I will talk about only using one post processing volume with a LUT attached and setting this one to be unbound.

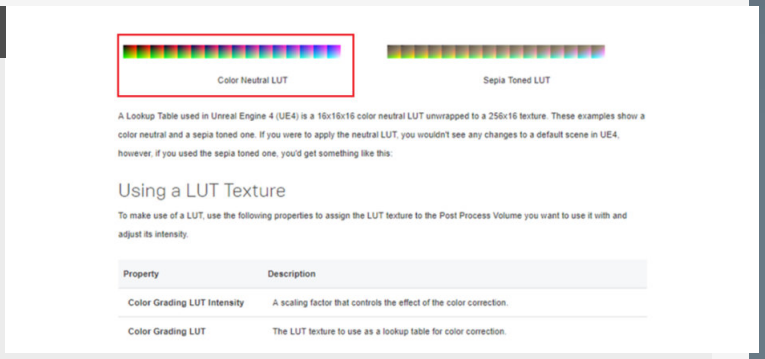
1. Getting the base LUT Texture

Download the base LUT from the unreal engine site. Look for "LUT Unreal engine" and save the left image on the first

available link. Make sure to "Save As"

You can find this on the Unreal Engine documentation here:

https://docs.unrealengine.com/en-us/Engine/Rendering/PostProcessEffects/UsingLUTs



2. Taking a render and prep

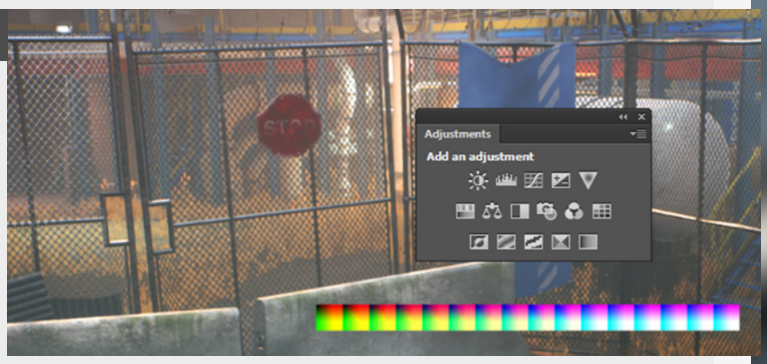
Take a screenshot from your scene (using the high resolution screenshot in the little drop down in the left upper corner) and overlay the downloaded LUT texture in Photoshop.

Make sure that you overlay the non-adjusted freshly downloaded texture over the screenshot that you took.

3. Adjustment layers

In this step you need to adjust the screenshot as you would like it to look inside of Unreal Engine, we are going to do this using adjustment layers.

Some good examples are Levels, Hue/Saturation, Photo filters (for quick adjustments), etc...
Feel free to experiment in this step, this is not going to take that much time but is going to make a big difference in the final presentation.

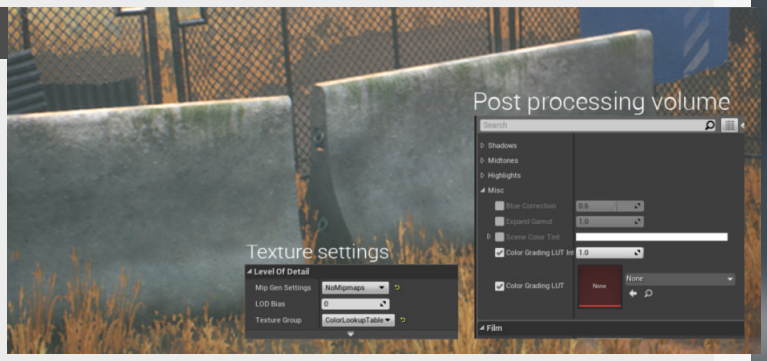


4. Unreal Engine setup

Now to take it back into Unreal Engine we need to isolate the LUT texture again, crop it to the correct dimensions (Ctrl+Click on the layer to select all the pixels on that layer, then press C to crop it).

Once you have done this save it somewhere inside of the Unreal Engine project folder to have it auto-import to your current projects

On import make sure to put the texture group settings to "ColorLookupTable" and plug it into the LUT slot in the post processing volume.



What is anti-aliasing?

Introduction

Anti Aliasing is something that all engines use or have access to, and something we as artist usually not have to think about but it can still impact your textures and visuals in a way, so let's dive into it.

1. What it's purpose?

Anti aliasing is a technique that's used to break up the diagonal pixels that cause jagged edges that are caused by pixels trying to create a diagonal line but are limited by the screen resolution.

Anti aliasing will soften these sections in a variation of ways depending on the technique used.

Also, this technique is also used in baking to make bakes more detailed and crisp especially where different parts of the mesh connect.

2. Two types of Anti Aliasing

Super Sampling: take a higher resolution image (super sampling) and then use that information to make the edges look better. Examples are SuperSamplingAA **(SSAA)** and MultiSamplingAA **(MSAA)**

Post Processing: The others way happens in post processing and will blur or contrast edges using the information already available and are cheaper for realtime rendering. One example is Fast Approximate Anti Aliasing **(FXAA)** that compared depth of adjacent pixels and blends them accordingly.

Another example is Temporal Anti Aliasing **(TXAA)** combines downsampling and blurring to create nice smooth edges.

3. Impact on visuals

The impact on the visuals will be depend on your method of choice, but if you have no anti aliasing on at the moment jagged edges will be really visible on the silhouette of objects, alpha cutouts on foliage, an also on busy textures.

One example you can see in the comparison on the right.

4. DLSS, FSR and TSR

These new techniques from both NVIDIA and AMD are there to use new techniques such as AI to give you more fidelity and make your image crisper than before. Which sort of leads us back to the first example SSAA which uses super sampling.

But in the case of Deep Learning Super Sampling (and FSR or TSR, the algorithm Unreal Engine uses) is will take your current resolution and sharpen and upscale the image using dedicated AI cores to be rendered at a higher resolution. So you're not losing precious GPU power on rendering a higher resolution natively.

15 PORTFOLIO PRESENTATION

Portfolio Thumbnail Tips

Introduction

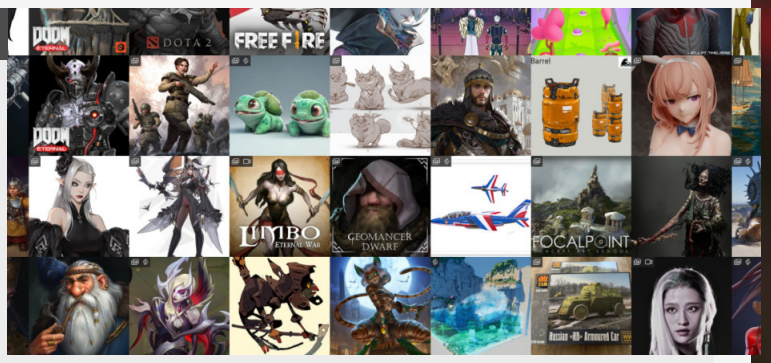
Portfolio thumbnails are meant as the doors to your artwork, so the least we can do is make that door as exciting as possible to open!

1. Catch attention in an artistic way

Your thumbnail is there to stand out from the crowd and grab attention but to do so in a way that aligns with your artistic "Brand" or beliefs, what do I mean by that?

Don't put clickbait or female bodyparts on your thumbnails if it has nothing to do with the work (this is important, if it aligns with your work and how you want to be known then that's fine) or just to attract more attention or likes to your work when it doesn't align with your intentions.

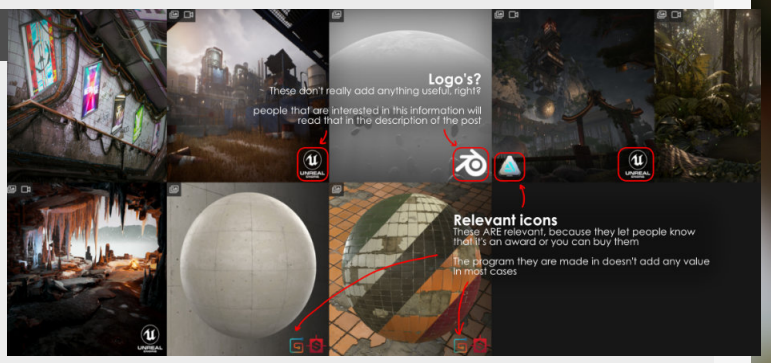
I feel that as an artist you build your own audience and clientbase, so you want people to honestly look at your work and how that relates to you.



2. Let your work to the talking

You have seen this before, I'm sure of it. Some people (including myself ;P) thought it would be a good idea to start adding logo's on top of your thumbnails, but it doesn't really add anything to the thumbnail, especially now that you have the different filters (Unreal Engine etc...) taking away the only reason why you wanted to add that logo on it in the first place.

Having your own logo or name or a personalisation that doesn't take away from the picture is totally different and can be a way to curate your personal portfolio and make you stand out a bit



3. Making it pop

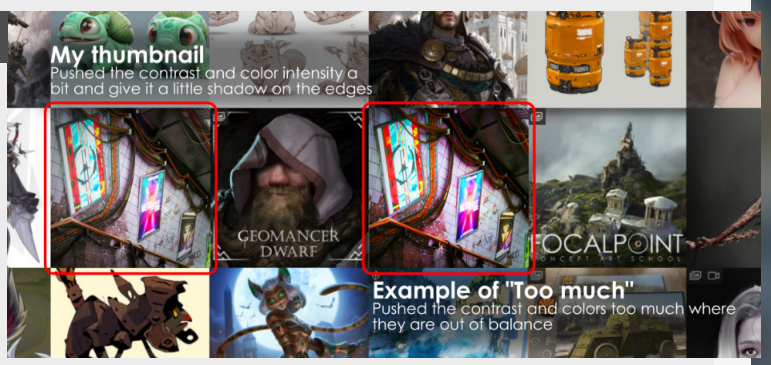
Contrast and intensity of colors are a good thing to start looking at for a good thumbnail creation, especially because we want your image to jump out from the crowd. So if your work already has really popping colors it will jump out of the crowd by itself.

This is obviously not the only thing that matters, there are things like subject matter, if there are faces or patters that people can relate with easily, etc... so it's an intricate subject and really aligned with Art Fundamentals

4. Test it on the front-page

Which also brings us to the next point, to give your thumbnail a quick test on the front-page between all the pieces of art that other people have created, this gives you a good indication if you need to push the colors and contrast a bit.

Usually I have a separate file specifically for the thumbnail where I can push the values a little bit without deviating from the original too much.



Portfolio tips - Part 01

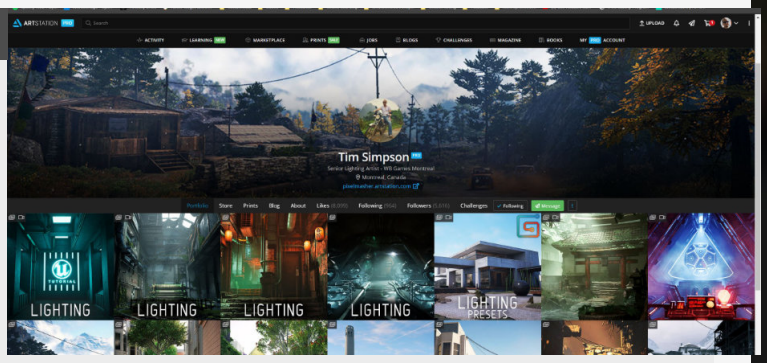
Introduction

So with this one we will dive into some the topics of building and designing your own portfolio and how this will help you with the next job and how to grow towards what you want to achieve with your career.

1. Specify your focus

The first thing people want to see when they visit is what you are good at, really focus on the elements that sell you and your skillset. So for my example this is making Environments in their entirety, this means making every aspect of it, so that's why I put my personal work on the frontpage because that shows off that there is a love and passion there to create the complete world, the story and not just a single element of it so that is something that I love to do, so this is something that I bring to the forefront.

This doesn't mean that there is nothing else that can be on show here that you are working on or that you are proud off, but it's good to have your focus in the spotlight.

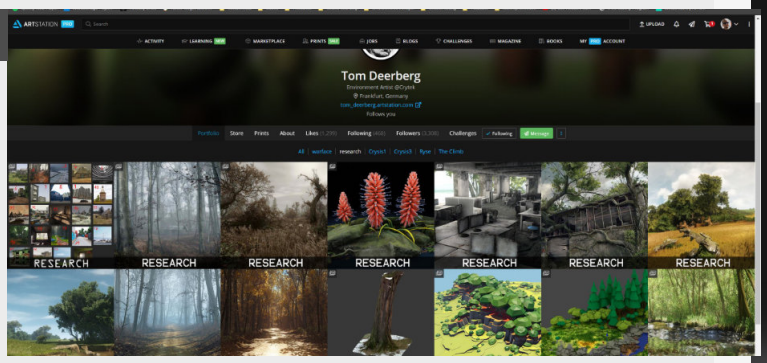


2. Showing additional interests

There is no "Correct" way to do this but I feel that a good way of doing is ,is to clearly mark it in a way that makes other people.

A good way to do this for example is to put stuff into different folders or maps in a neat way to display your different skills. Another could be is making your main landing page the stuff you want to wow the viewer and then show the additional stuff on an additional page.

But eventually the choice is yours, it's still your Portfolio and your personal space to show what you can do and how you want to be seen, just keep it clean and simple.



3. Show what you want to do!

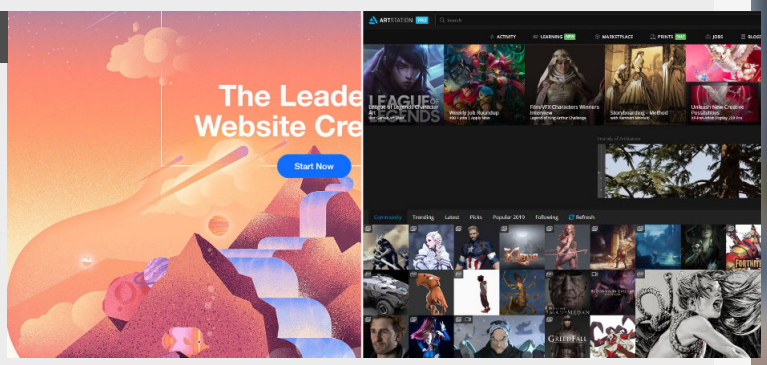
So you have been doing this thing for a while and you want to transition into a new discipline or role? Once you have work to show that you want to transition you might want to look into making this the focus of your portfolio instead thing that seems to lean into the first point but is slightly different is that you want to try and to is to give your portfolio some direction, showing what you want to as this might differ from the thing you are currently doing already.

For me personally this means putting full environments onto the forefront and including all the aspects that come with Environment Creation.

4. Artstation or others?

So this is a highly discussed topic, this is because Artstation is so widely used and has become the new standard for portfolio's because interviewers and people from studios can easily recognize where to look and how to browse your portfolio at a glance because they all use the same layout, which saves so much time when browsing your work.

However, there is also the time that is saves you and seems to be somewhat less discussed, because if you need to go through a more elaborate process to upload your work and do some minor coding you might be spending time that you could spend on creating art. The recruiter is not focused on how well designed your portfolio is designed especially when looking for gameart.



Portfolio tips - 02

Introduction

So with this one we will dive into some the topics of building and designing your own portfolio and how this will help you with the next job and how to grow towards what you want to achieve with your career.

1. Let art do the talking

This might be the main thing that is so straight forward and you need to focus on. Which is to let your art do all the talking instead of over explaining yourself.

Opening up with your best shot and then doing all the breakdowns or even doing the breakdown as a separate entry on your portfolio might be the best way to deal with this.

Don't overwhelm or obscure the thumbnails that are meant to pull the attention and pull people to your art, so the more you add to thumbnails that makes it more difficult to see the actual hard work that went into the art itself.

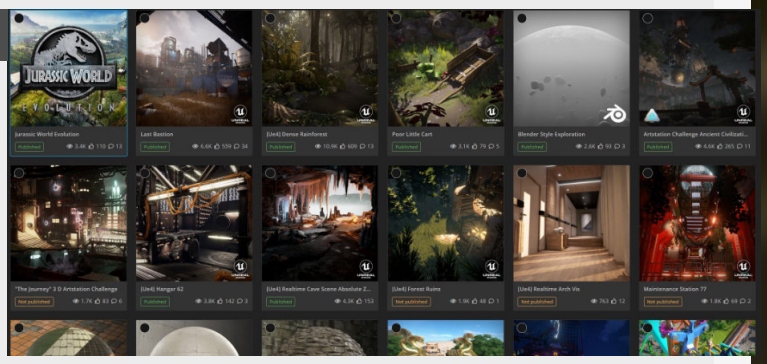
2. Showing game art

An important thing if you want to make it into the game industry is showing off all the more technical stuff that comes with going through the entire pipeline, people that Interview you or have a look at your portfolio love these things and will definitely get a better insight into how you think and how you go about creating your art.

How you go about this is more up to you but the main thing is to keep the focus on the breakdowns itself instead of again, trying to over explain, using bright distracting colors, weird typography, and so on...

Take your time when presenting this, you will be discussing this during the interview.

3. Adding too much?

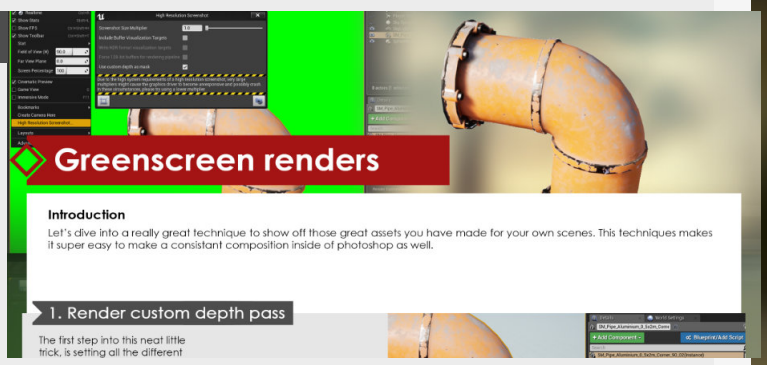


There is a point that it can become too much and will add too much clutter. But there obviously isn't a set number that you can hold yourself too but I would say to in most cases keep it under 10, if you are adding more than 10 it will quickly become overwhelming, pick and choose your best!

Make sure to pick a couple of good angles (if you have multiple ones) and then try and make your showcase for individual props as concise as possible.

I think a good format for this is having one pager per asset or group of assets that hold all the information you need, make sure to pick prime examples.

4. Take your time when presenting



This one sounds super easy but it is going to sting if you are doing project that is super long and you have to push it those final 5 percent. So make sure to do some research and take your time preparing the presentation for your art. Make sure to keep it simple and once again keep the focus on the art (or if you are doing a breakdown, focus on that) you can do this in multiple ways if you are doing single assets with a slightly blurry or de-saturated colors.

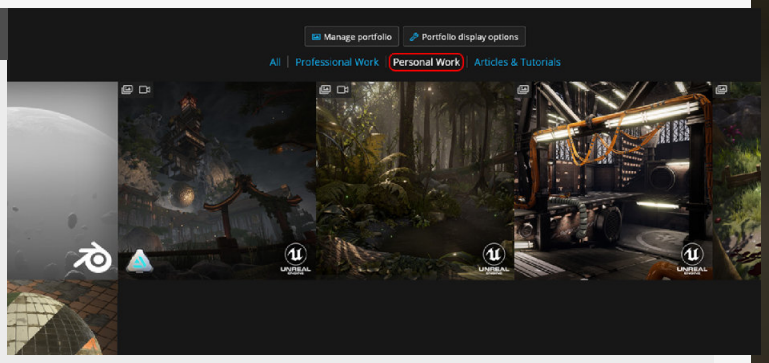
Or in the case of environments you need to pick a good angle that has a great composition and lighting that makes it look like the complete picture.

Optimizing your portfolio - 01

Introduction

You're getting ready to finally send out your portfolio and get jobhunting for that position you have been keeping an eye on for quiet a long time now, but are you ready? And more importantly, is your portfolio? Your portfolio is your ticket to that job so this entry will focus on getting the most out of your portfolio as possible.

1. Define yourself



You really need to find the thing that you love, and this takes time.

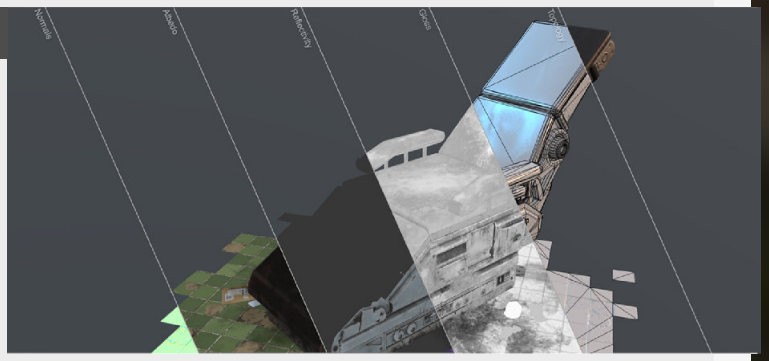
Once you know what you want to do, it's time to start looking at your portfolio and make sure that it reflects that.

So for instance if the title says Environment Artist but have characters, don't have 95% of your portfolio being characters.

For myself as an environment artist I put the emphasis on my personal work first.

Also, specializing now, doesn't mean specializing forever.

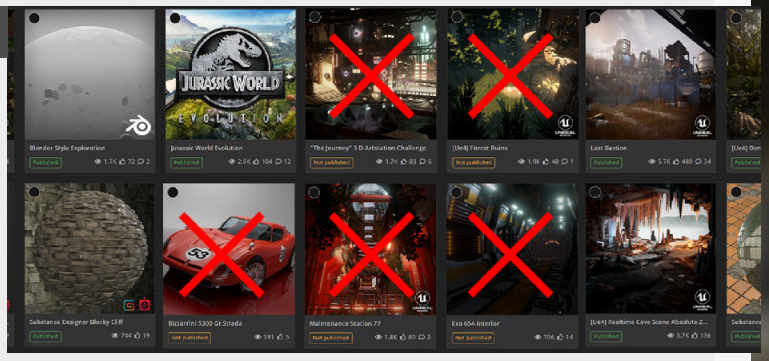
2. Present game art



Show breakdowns, there are some nice examples out there of gifs comparing the lowpoly/ highpoly for bakes, wireframes and polycounts are still useful to see as well.

Another thing that is really awesome to show (and to look ot for recruiters) are marmoset viewer files, allowing people to have a look at all the information and details on your beautifully made assets. Ofcourse, this is a little bit trickier for full environments, but still take the time to present the assets individually.

3. Optimize over time



You have to be critical about your own artwork.

The tricky thing is when to remove stuff and there are no real general rules when it comes to this.

The most important thing is to really have a look at your pieces and see if it's really the thing that sells you as an artist and if it's not, then it's time to remove it!

The best saying when it comes to this is "your portfolio is as strong as your weakest piece"

4. Old work

Doesn't need to be removed, just make sure that it's clear you think it also old work? You can make a separate folder or entry for these types of post.

This will then act as a roadmap to see where you have come from and can help people get a grasp for the kind of improvement you got out of your older work.

It's all about quality not quantity.

Optimizing your portfolio - 02

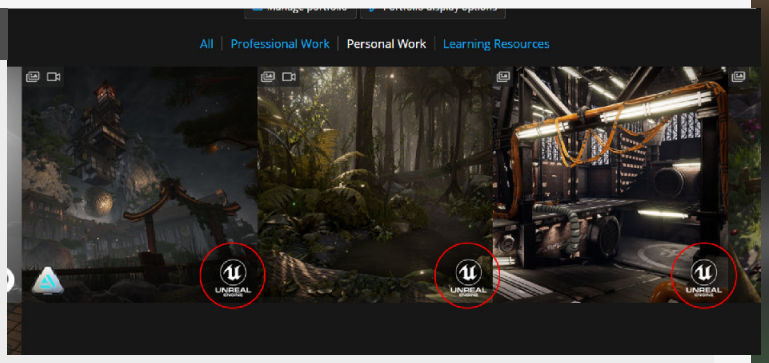
Introduction

This time we will dive into some more tips about how to get a better more well-rounded and optimized portfolio, we will dive into some topics as portfolio consistency, how to make your work stand out and more!

1. Consistency

A small one to really dot the I's and will help you standout from the others, is creating consistency within the work. This can obviously mean multiple things including the consistency of the presentation for the thumbnails or the pieces of your work itself.
A great one to make use of is a logo, or if you don't want to design a logo then consistency in the type of font you use is also a nice one to create consistency and making with your work recognizable.

You basically want people to know straight away that the artwork they just found is yours.

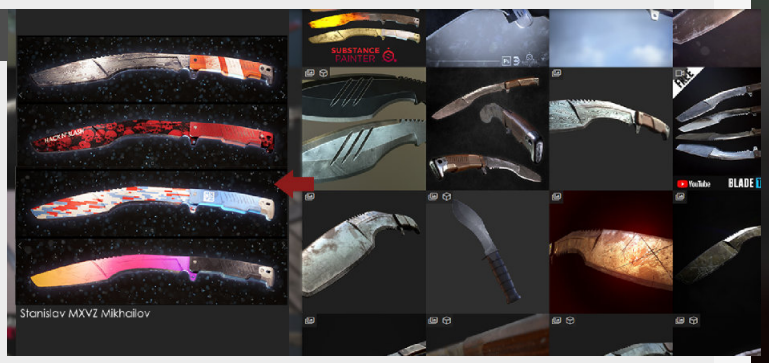


2. Make it yours

Not everything needs to be a shader sphere?!

Look for things that can really make you stand out from the crowd and make you more special.

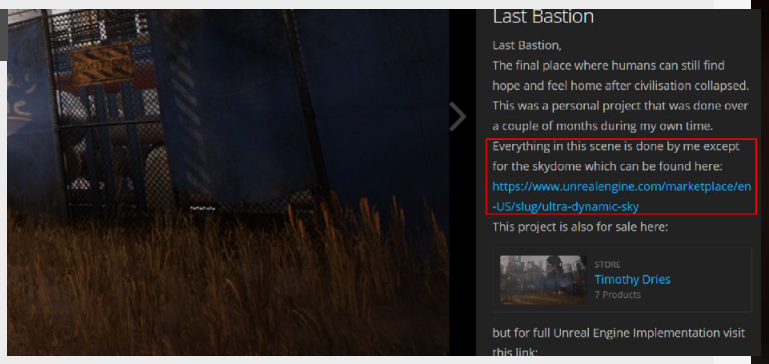
Ofcourse it's not just those material spheres, this can be applied to a lot of stuff. For example if you are doing a tutorial, do the tutorial first and then take those lessons and make them yours. After a while it becomes clear that people just followed a tutorial step by step because there are so many of them out there.



3. Not done by you?

Environments get really big and doing all the aspect on your own can be a real pain and can cost you months!

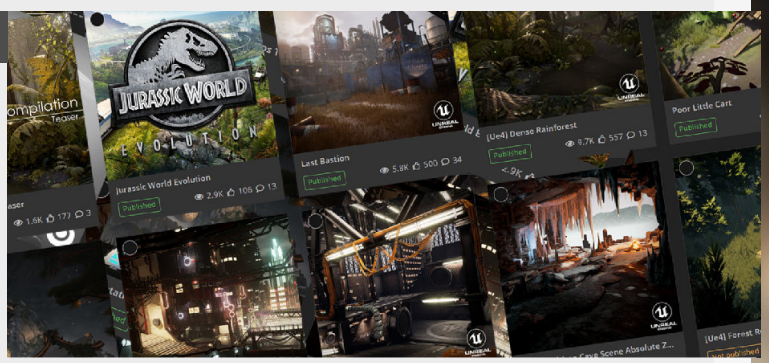
This is why it makes sense to look for other thing that you might use in your scenes, like for instance lighting setup for Unreal Engine, as long as you mention it in the artwork so people know that that particular part is not yours. This way there is no confusion about the parts you were responsible for and you help another artist out!



4. Quality over quantity

This an old saying that is still true! If people are coming over to your portfolio they want to be blown away by that killer piece(s) that you've made.This also counts for the amount of screenshots that are inside a project, don't add a ton of the same angles, focus on making the absolute best and keep pushing it!

Obviously mastering this required you to know how to balance your time and to make sure that you don't burn yourself out on this project, because the worst project is an unfinished one!



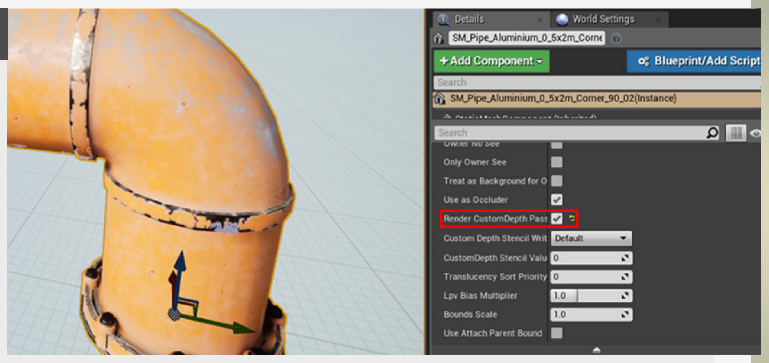
Greenscreen Renders

Introduction

Let's dive into a really great technique to show off those great assets you have made for your own scenes. This techniques makes it super easy to make a consistant composition inside of photoshop as well.

1. Render custom depth pass

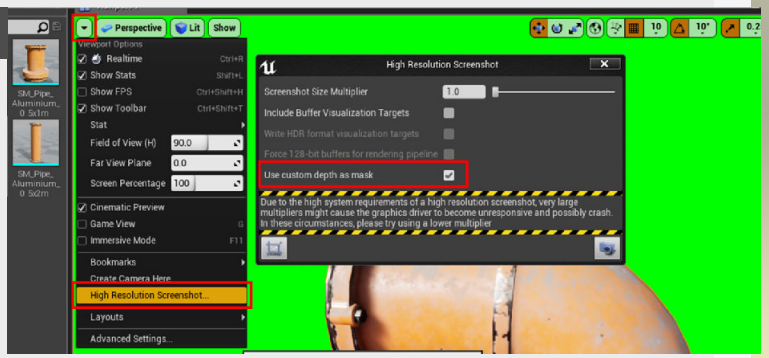
The first step into this neat little trick, is setting all the different assets that you want to render to "Render CustomDepth Pass" under the "Rendering" settings.



2. High resolution screenshots

The next step is taking the screenshot using the "High resolution screenshot" tool under the little arrow menu on the top left of the view menu.

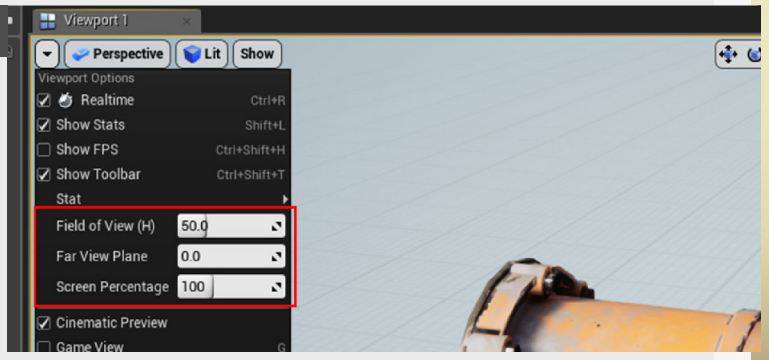
Keep in mind that higher size multipliers can lead to a crash when trying to capture it but it does give you a cleaner result, so I always use 2 for all the screenshots I do.



3. Camera settings

This is an optional part of the process depending on how you render your assets, so for instance it is going to be important if you are using HDRI images for your background because the reflections on the mesh need to roughly match the HDRI image.
As Elvis Posa mentioned at this point you can use the starter content and look for the scene called "advanced_lightning" map.
Which just let's you just swap the HDRI in that scene.

Some other options can be found under the little arrow menu in the top left corner.



4. Composing

When you got all the assets composed in the same manner, and you got a background ready, you can get to composing them in Photoshop, I just used a blurred picture that I had lying around.

This is just the render seen on the right with a field of view set to 50, on top of a background layer which is blurred, with a simple adjustment layer for a tad more contrast.

Thanks so much Elvis Posa for sharing this knowledge.

Improving your renders

Introduction

Let's talk a little about the tricks you can add to your finale image to push your presentation a bit further then usual, so let's dive into some tips!

1. Greenscreen renders

So you have all your presentation shots for your environment done, but now you want to show off that hard work you did for your props too. This is where this section can help, you can use this trick to make everything else then the props you select into a greenscreen, without it affecting the reflections etc.

So achieve this effect you select your prop and in the details panel you select "Render CustomDepth Pass" then in the screenshot tool you just select the "Use Custom Depth as Mask" and it will only render out all the meshes that have this checked.

More information can be found in a Previous Tip

2. Sharpening your images

Another smaller tip is adding a little sharpness to really define the edges and just make your screenshots a lot sharper (hence the name), there are multiple ways to add this to your work and the most used way is adding this directly on the screenshots by adding a sharpening filter.

However you can also add this directly in Unreal for example, for this there is a command that you can use "r.Tonemapper.Sharpen X" where "X" is the amount of sharpness that you want to have from 0 to 4.

3. Adding a vignette

This is a trick that comes from photography but can easily be translated to your screenshots, adding this darkened border on top of your image, this will have the effect of focusing the viewer on the center of the frame and the focal points that you have so carefully crafted.

However, this can easily be overdone too, so be careful that you don't overdo this effect, keep it subtle. My recommendation would be a multiply layer in photoshop set to 15%, but this depends on the environment itself.

4. Don't skip post processing

This is more of a generic topic, but don't underestimate the power that post processing has on your environment, a lot of people tend to skip over this step really quickly. You can really finetune the mood using the "LUT" for example, which allow you to take a screenshot, adjust the screenshot in photoshop and then bring those changes directly back into Unreal Engine for example.

However, you can also do a lot of these changes inside of the Engine itself, there are a ton of options inside of the "Post Processing Volume"

Vignette's

Introduction

Vignette's are another technique that comes from photography, which is a technique that helps you pull the focus towards the center of the image.

1. What is a vignette?

A vignette is a border around the image that reduces the brightness of these edges.

This helps bring out the contrast and brightness of the core of the image, helping people who are looking at your amazing art to stay inside of the image and not wonder outside of it or get distracted by things that are on the border of the image.

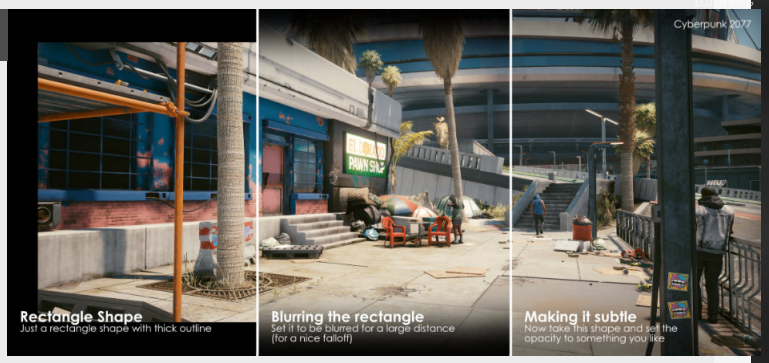
It's also a great way to add some extra depth to your thumbnails as well.

2. Setting up a vignette

The easiest way of adding one is by doing it after the fact, in photoshop or any other program you use for example. This also gives you more flexibility on how intense or subtle you want to the vignette to be.

However you can also set one up directly in Unreal Engine. You can find this option in the **"Post Processsing Volume"** and look for **"Vignette"**

When I setup my personal Vignette's I usually setups a thick border around my image, blur it a lot and then change the opacity to be around 15%, but how strong depends on your personal preference and the colors of the image.



3. Keep it subtle

The most important thing with a vignette is always to be subtle, you don't want it to be really visible so people realize you are trying to direct their attention. You can kind of compare it to what you want to achieve with your composition when making an Environment.

You need it to be subconscious enough to guide people but not something they notice.

4. Natural vignette's

Vignette's don't always have to be done in post or as an afterthought. They can already be a part of the scene your constructing by default, removing the need to add one later on.

This can be done by placing multiple layers of assets in such a way that the vignette and more directed focal point is built into the scene itself.

Some examples can be seen on the right, these don't necessarily need to have an additional vignette on top of their natural vignette.

Crisp Unreal Engine Renders

Introduction

I've been diving into some personal work again over the last couple of weeks and have been having some fun with Unreal Engine, and because of that I wanted to share some things you can do to make your Unreal Engine look better and more cinematic.

1. Sharpening your image

One of my personal pet peeves is that Unreal Engine looks a little too soft for my taste. So my first step is always sharpening the engine screen with a tonemapper or post processing option.

You can directly go into your Console Commands at the bottom of your viewport and then type in **r.Tonemapper.Sharpen "X"** (where X is the value you want to sharpen it by).

Don't overdo this setting either, because you can really overdue this setting pretty quick. I personally love to have this setting around 1.5 - 2.

2. Turning up the Screen Resolution

We can also increase the number of pixels the viewport will render. This is called super sampling the image, where you render the image at a higher resolution than your screen and then have it sampled down to your screen resolution again, giving you a more detailed image.

Artists do the same thing with some of their textures too. Bake a 4k map, to then size it down to a 2k map to get more details out of the texture map.

You can find this option in Unreal Engine by going into the viewport menu and turning up the **"Screen percentage"** to a value that is higher than 100 (as 100 is your screen resolution)

3. Depth of field

Often overlooked, but depth of field will emphasize the focus on your subject matter by mimicking the aperture you get from a real world camera. The lower this aperture is the stronger this affect is and the more the area's in front and after the main subject get blurred.

All these settings can be found in the Camera menu or again, in the post processing volume as well.

I personally prefer to do this in the camera settings to have them localized to that camera instead of a general post processing settings that effects the entire scene.

4. Focal length

Again in lens settings we find this other important section, the focal length or the angle of view that's going to be captured by the camera. It will also add to the cinematic feel by condensing the depth of the scene, thus putting more focus on the main subject once again.

However, there are limits to this and not every scene might be built for it, because the higher we put the focal length the further back we need to move the camera to compensate and not lose the framing we had before, so it's a balancing act.


16
MINI
TUTORIALS

Let's create: a roof

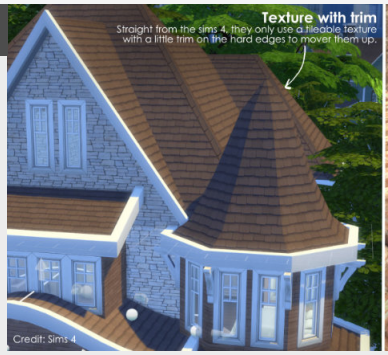
Introduction

This one was inspired by the community, since I was focused on the roofs that I'm doing currently for my personal work and wanted to highlight a couple of different ways to create a good looking roof and where they are used..

1. Texture only

Let's start with the basic version, and that is just adding a simple texture to the geometry. This is the simplest and most cost effective way of creating a roof by far, especially if it's a relatively small tileable texture that's being used.

Super simple easily adjustable and quick on creation, probably used in projects or project where you are technically more restricted and where the focus is not on rooftops.



2. Texture with edges

Now let's spice it up and build on the previous version with a small tweak that can really make out roof pop a little more and it's actually a trick I noticed when playing Kingdom Come Deliverance. for this trick we just cut in some additional geometry and make the silhouette pop a little more.

Then on the interior of the roof you can optimize all this geometry and push it up and down a little bit just to add a little life to our little rooftop.

3. Geometry + Additional pieces

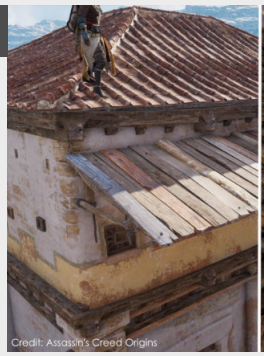
Next up we take a big jump and actually make the entire thing out of geometry, you can do this the cheap way (in terms of optimization) and just add cuts for the entire roof and offset them all.

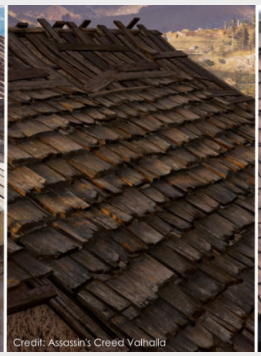
Or we can go all in and use a technique described in my other tip **"Heightmap to geometry"** to offset the entire rooftop according to a texture, which can be expensive depending on the complexity of the roof, so please take optimization into account.

You can also sprinkle individual roof panels on top as separate geometry, usually we do this trick on the edges or around damage, so the area's of importance.

4. Technique matches workflow

The technique you will be using depends heavily on the target graphical fidelity you are aiming for decided by the hardware and the type of game it is, but it's fun to see so many different practices on the same simple object and how they impact the visuals of games so much.





“The Breakdown” 01

Introduction

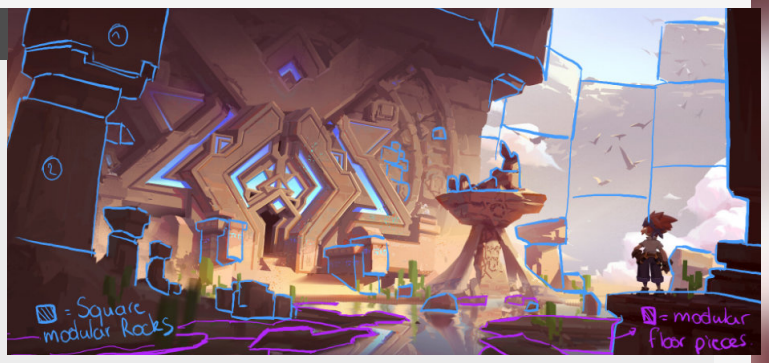
This week we'll breakdown a concept from Ruiyang Chen that looks really overwhelming from the get go but once we break it down it makes it all a bit more manageable.

1. Modular chunks

I always love to start with the biggest or the shapes that can be reused the most in the scene so we can eliminate most of the scene from the get go.

In this case it's the square looking rocks that are all over the scene, for these we can block out a rough amount of different blocks for different scales (if you wanted to do them all unique) OR use a tileable map with a detail normal on top of it for extra details.

The floor pieces can also be approached in this way and then refined with some decals.

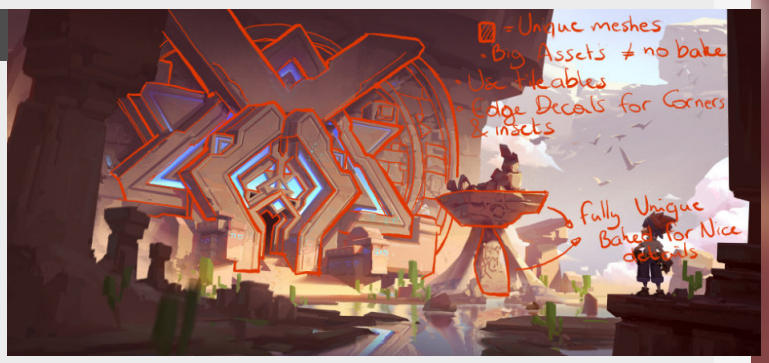


2. Unique Meshes

Next up are the focal meshes which will all be unique, but that doesn't mean that we will texture them unique. Because of their scale we can't really afford to go with unique bakes.

The textures will be tileable and we can add some nice damaged edges with "Edge Decals", these are trims of damage you can use on top of your geometry to add damaged edges.

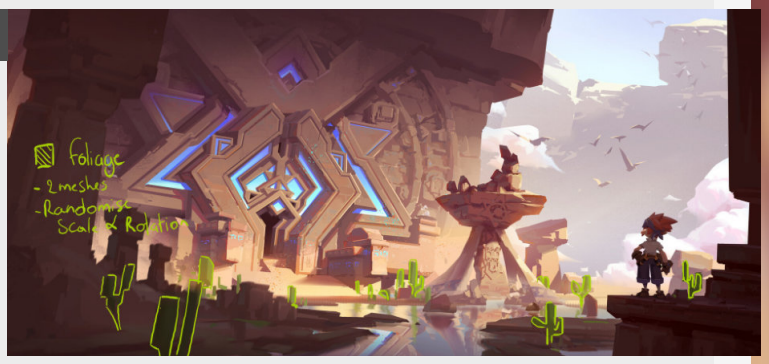
The two assets I would do fully unique are the rocks on top and underneath the pillar that is closer to the camera if you want to really add all the nice details in those.



3. Foliage

It's a fairly small part of the scene but it adds a nice pop of color, so we need to add them just for that.

But for the foliage we can keep it really simple and straightforward by just making 2 maybe 3 meshes and reusing them throughout the scene, these would also be fully unique.



4. Additional decals

Now we got to the stage where we can worry about the details of the picture, and since most of will be around the focal point (you want to draw the player in) you can add additional destroyed elements on top of the large meshes in the back to break them up.

We can also add some marking to add some storytelling in there and tie the whole image together a little more.

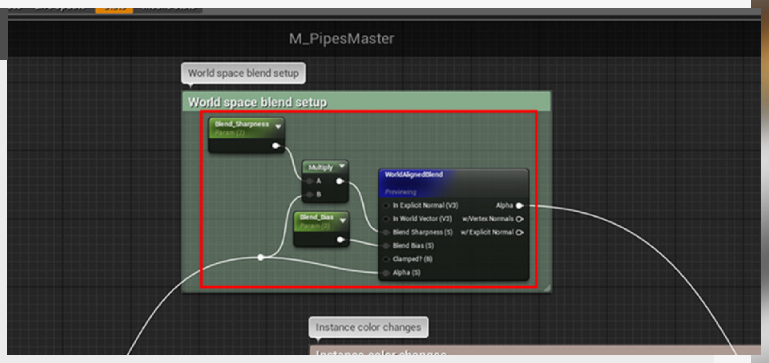
The damage decals are just deferred decals where the marking could be either deferred decals or floating geometry you add on top of your geometry.

Adjustable pipe material

Introduction

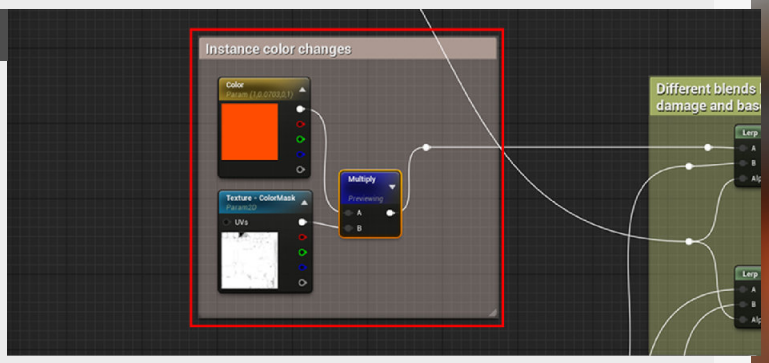
This time we will have a look at the material that I used in my personal work on the "Last Bastion", there are some really cool tricks that are embedded in the material, especially the world aligned blending which is great to use and easy to setup!

1. World space blending



The biggest part of this material is controlled through a super useful node called WorldAlignedBlend. Feeding this 2 different parameters and blending the blend sharpness with the input mask (which is a tri-planar projected cloud map from painter to remove uv artifacts).
The blend bias controls how "deep" the blend is according to world position and the Sharpness speaks for itself, the end result is a world based alpha.

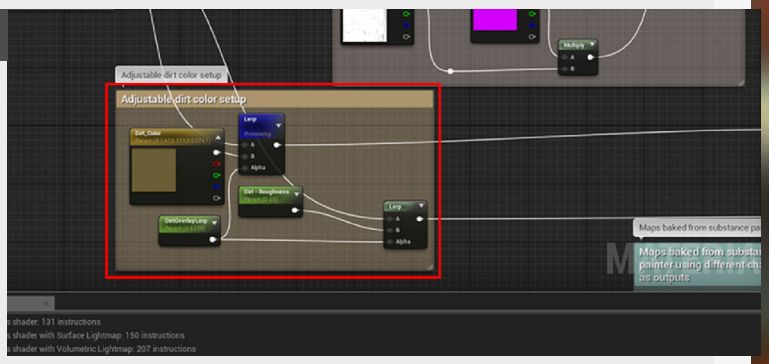
2. Instance color changes



On the painted metal pipes I opted to go for a color that I can change on the fly, so again in Substance Painter I put everything to black except for the painted metal parts and export that as a separate map.

Multiplying this map with a parameter color value allows you to change the color on the fly but still retain those dirt edges.

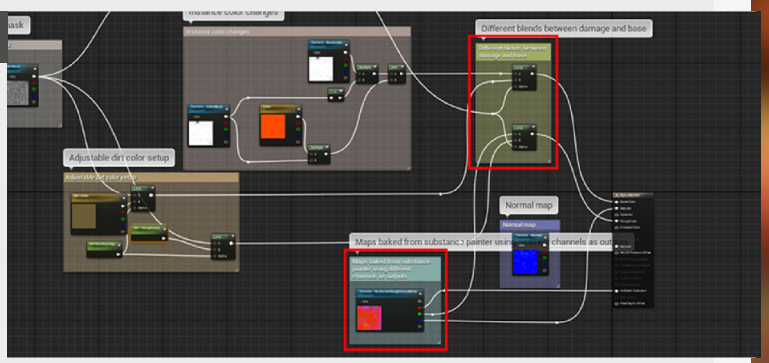
3. Adjustable dirt color



This section is a crucial one used in combination with the mask, as the damage color and roughness value are really going to show through the usage of the WorldAlignedBlend mask.

I also added in a Lerp between the Dirt mask (which is a black and white tri-planar projected mask) and the Dirt color parameter, blending these 2 in a subtle way gives the dirt some more depth instead of just being a flat color, and I basically do the same for the roughness which is a 0-1 value.

4. Adding baked maps and damage



It's time to put all of these items together, I use the output from the world aligned mask as Lerp inputs to blend between the color mask and the dirt color for the Basecolor and I blend between the roughness map that I got from Substance Painter and the Roughness value that I set as a parameter.
Additional notes: all the textures used in this material are also parameters which allows me to pull my own maps in for new sets of pipes, as well as create extra colors/dirt setups.

Fake wind effect using normals

Introduction

Let's dive into how to make things move in Unreal Engine, I personally always like to add some movement in my scenes, especially when rendering them to a video at the end for the final presentation, so let's dive into how to add your own in an example with tarp.

1. Different approaches

We can get a long way with faking this effect without having to delve into proper animation for a lot of it.

Normals only
The simplest way of doing this is by only manipulating the normal, and sometimes simple is all you need, and it's also the one we will be diving into here.

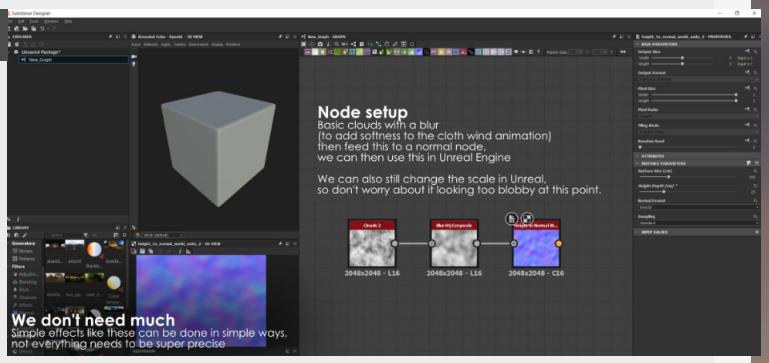
World Position offset
This will displace the vertices in world position making it really useful to create movement, but this also means that for this to work you would need to add some additional vertices so it can actually deform them. We can find this as an input in the material graph in Unreal Engine.

2. Creating a normal

First we need a normal map to start faking this effect. For this I just use a clouds texture I generate either in Photoshop (Filter > Render > Clouds) or Substance Designer. Any of the clouds textures with a blur attached to it to soften it up. (Designer might be easier as you can easily convert this to a normal map).

Then convert this to a normal map using the "Height to normal" node in Substance Painter. Now that we have our little normal map we can move over to the material setup in Unreal Engine.

Don't throw away this file just yet though, we can use the same height information for a World position.

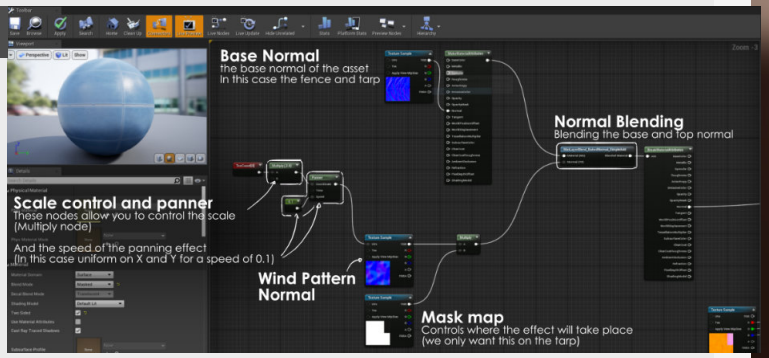


3. Setting up the material

Now that we have our normal information we can start blending this on top of the tarp normal tat already exists, or any other baked normal map really. For this we'll use the "MatLayerBlend_BakedNormal_SimpleAdd" I know... it's a mouthfull, but it's pretty simple it allaws us to "Add" one normal to the next.

We do have to make our normal into a Material for this to work tho, so for that we can use the "MakeMaterialAttributes".

It's not moving yet though, for this we can use a panner node as you see attached on the left side to our normal map we just made.



4. Defining the effect zone

Now that we have a simple setup going, we can always go in and refine the result by using a mask to mask out certain sections of the effect itself, most of the time I use a simple mask for this but you can also use vertex paint or other ways of masking for this.

You can see that because I still have some of the metal pieces of the tarp on the same texture that I mask them out with a simple black and white texture.

It's a pretty simple setup, but can give you loads of interesting results.

17 PERSONAL DEVELOPMENT

Building a good habit

Introduction

So let's dive into something more personal that a lot of people get sort of lots in, being how to build yourself a habit to make your work more consistent with these little tricks.

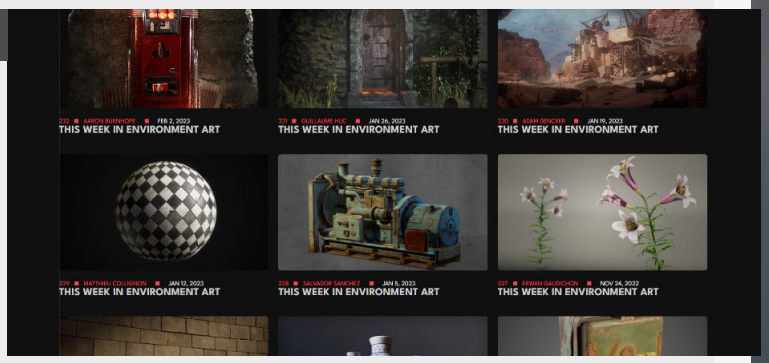
1. The beginning is the hard part

I realized that even on days that I don't really feel like it I get that feeling once I get started.

The trick here is balanced between making a schedule that works for YOU not for someone else and also not to overload yourself from the start. The best way to do this is starting small with some manageable easy wins that allow you to get excited for the other goals that you have for your work.

Once you get the habit to stick for 30mins a day for a continued period, that's when you can start looking to extending this time, build it up slowly.

2. Building some external pressure



A good way to keep yourself accountable when you feel that you are breaking your own promise one time to many is by having external pressure by other people holding you accountable, especially by pushing yourself out there among other artists.

This really helped me by doing a weekly blog, forcing my work to be seen by a lot of people out there and holding myself accountable by having something to show off every Thursday and now this blog has turned into so much more then I ever imagined it to be.

3. Make a to do list

This is something that really helps me personally, and it makes such a huge difference. I've been trying out different ways of to do lists, like digitally or manually. After doing it for so long, I can definitely say that writing stuff down in a little notepad is the best way to go.

There is something just so satisfying about writing your stuff down and then crossing it off later, there is a nice connection that happens there, so can definitely recommend you to do the same.

4. Plan your next day

Now you got your to-do list ready but actually you can get the most benefit by planning your day the day before you are going to execute on them, this will allow you to do all the "boring" planning and you use all your newly regained energy to just execute these tasks.

Also, it has been shown that if you plan the day before your mind will take this and churn through those plans during your sleep, just like some of these moments when you are stuck on something and then you just go to sleep and the next day you all of the sudden will find the solution, the same stuff will happen with your plan too.

Moving company or countries basics

Introduction

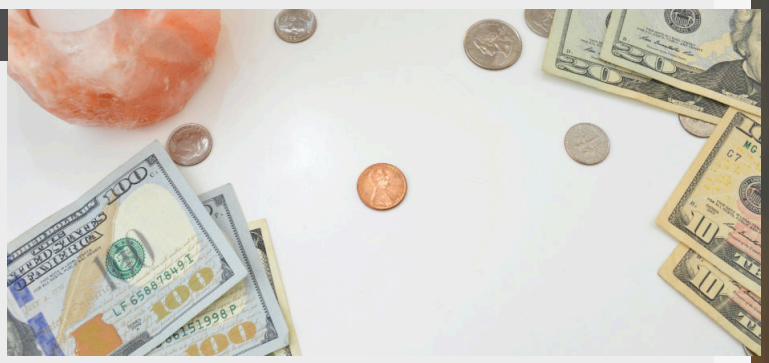
Something that is going to happen for a lot of people working in or getting into this industry is the fact that you will most likely move from the place you live to another spot, maybe even another country. So in this one I want to shine a light onto some of the lessons I've learned when moving companies.

1. Check the wages

Before you start applying to companies or at least when you get invited to a company for an interview it's always nice to check what the company can offer in terms of compensation to see if it is competitive or not.

This will allow you to at least get a rough idea of the bracket you need to be asking for when you are sitting at the negotiation table with HR. There are a lot of sources that ca help you here, most of them are pretty location specific (So make sure to check your location as well, you don't want to be like me and get all excited for 60k salaries if you're not living in the US) but one that is always available all over the place is **Glassdoor**.

2. Cost of living



Another thing to check is not only how much money you are getting in, but also how much money is leaving your pocket too, most people don't get to this step before it's too late. Make sure that if you get an offer with a number attached to take it home, do a little research before saying yes to it.

There are amazing resources out there to compare the cost of living between cities, but always add a pinch of salt and include a nice and healthy buffer.

One of these sources is **Numbeo.com**, allowing you to compare different cities ad countries to each other.

3. Ask for moving support

Especially when moving countries, as about this during your interview. Some companies have specific budgets set aside to help you with a fund called "Relocation Assistance". This budget and what it can be used for can really help you out, I had the option to make use of one when moving from the UK to Germany, and it made the trip so much easier.

It doesn't always need to be for monetary support either, this can also mean that HR will assist you with all the paperwork you need to enter a new country for example, and so much more, so ask for it.

4. Make sure you plan enough time

I've had to rush most of my moves by doing apartment viewing online (precovid), running from place to place for viewings or even booking a last minute AirBNB to bridge one month.

Do a lot of research into how easy it is to find a flat, how long it takes to get registered, what documentation you need, etc... there are a lot of factors that come into play here, so make sure to make the most of it and prepare accordingly.

Building a motivational system

Introduction

These are my personal favorites when building a routine for personal work, to help keep you on track and to keep you motivated for when you need to. These tips will apply to a large majority of people, but might vary depending on your personal routine or obligations.

1. Consistency is all

When it comes to personal work just block out a small chunk of time (let's say 30 minutes) for each day. This might seem small, but that by design, being an artist also means dealing with fluctuations in artistic motivation, so you CAN do more than those 30 minutes, but you don't HAVE to. But you will also notice that one some days you will do way more then just these 30 minutes too.

After a solid 30 days, you can up the amount of time, but usually this happens organically. Just keep to the system and it will take care of you.

2. Writing things down

Another thing that has really helped me, an artist with way too many ideas, is to write stuff down. Like all the time! There are times when you think of a cool idea when researching, just write these down.

And with writing I mean Physically writing them down, this creates a better connection in our brain then typing it out on a notepad, plus those post it's are always visible, where you can hide your notes on your Computer.

3. Make a to do list

Sort of in the same line to the last one, is to make a To do list, again, taking my post it notes in hand for this tip too. They are just great to limit yourself in the amount of tasks.

Make sure that if you write some tasks down that they are small enough so you can actually do them each day, and if you don't make it for the end of the day then just add them on the next post it.

Cross them out to once your done with a task, this gives your brain a good feeling and you can see the progress you're making in a visual way, which is amazing on projects where the visual feedback might be slower, to help you stay motivated.

4. Take time off

Last one for this week, but it's an important one. There is this current trend going on of maximizing your potential and your own time, which I fully agree with.

There should always be room for stuff that relaxes you as well, so go read your favorite book, play some games and do NOT sacrifice on sleep, this is the main driver for creativity!

Those moment that you take some time off can inspire you to do even crazier an interesting stuff when you are actually working, so see it all as recharging your batteries (Just don't try to overcharge them, that's not how this works ;P)

Motivational tips

Introduction

A lot of things going on at the same time often makes it hard to really stay into focus, so this entry will dive into some tips to keep yourself motivated and push you through some of the harder times when working on full environments.

1. Staying in focus

To me this has always been more about your personal strengths and weaknesses, for example most of the times I work the best when listening to music and can really focus, but other times I listen to podcast or even have a documentary going on my second screen.

Finding when you are in the proper mood for the one or the other, that's where the real trick is. It all starts with being honest to yourself and if you struggle to maintain focus, maybe switch it up?

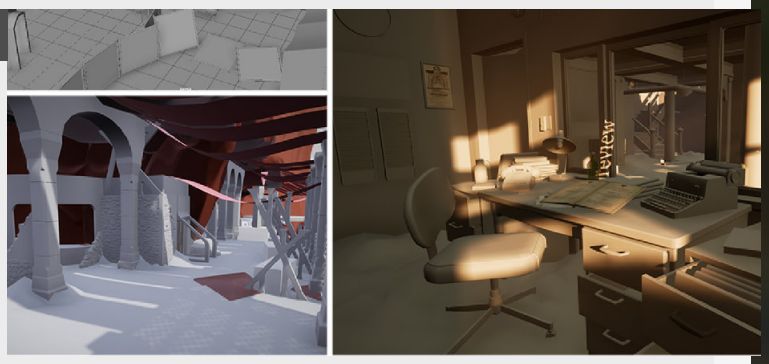
2. Topic immersion

Immersing yourself is one that I really love because it is so personal in how you do it, but the basic principle is digging in and learning about your current project topic in whichever way you feel works best for you.

I personally like to watch documentaries on the topic.

You can also achieve this by reading books, printing a reference material and hanging it on your walls surrounding your desk or just have a bunch of inspiring wallpapers as background on your pc, the possibilities are endless.

3. Taking breaks



This will stop you from burning out on the project.

What this also offer you is that you distance yourself from what you are working on and will give your brain and eyes a rest.

So if you feel like you are doing the same thing over and over or adjusting a tiny detail, it might be time to leave it for the day (or a couple of hours) and just let the image leave your mind for a bit. But most importantly finish it!

4. What do you enjoy?

You need to know what you like? what do you enjoy to work on?

Because if you get a job in the industry on a portfolio that is build on setting you up for something that you don't really like working on, then what is the point?
But how and when do you know? Well a lot of this comes from experimenting, but the shortcut lies in being true to yourself. Really try to dig into your personality.

For me, I was always taking pictures wherever we went on vacation from the buildings and locations .

Avoiding art blindness

Introduction

Ever experienced doing a days of work or a whole bunch of small adjustments without that much changing or even worse, coming back the next day and realizing that you took a step back instead of forward? This is what "Art blindness" is all about, so today we will discuss some tips on how to avoid it.

1. Asking for feedback

The first big one might also be the one that is the most obvious one, this being asking for feedback.

This will break your train of though which at this point might even be focusing on the wrong thing entirely, so asking other people will help you to go in the right direction again.

Be careful with this though, if you want clear guidance then it might be better to ask it directly in a one to one conversation instead of throwing it in a group, as you might get overwhelmed by the replies.

More tips on this topic can be found on my blog (#55) or in my Compilation.

2. Taking a break

If you are noticing these things as mentioned in the intro, like doing small tasks to little change or coming back after a time and realizing that you took a step back, it might be a time for a small break.

Doing something else really refreshes the mind, especially if this is something that take up all the attention and doesn't allow you to think about your art at all.

3. Create a varied workload

A thing that I love to do on my personal projects is working on the things that I didn't work on during the day at my professional job or even the day before on my personal work. So for example if I worked on some props I'll probably do some set dressing or terrain sculpting.

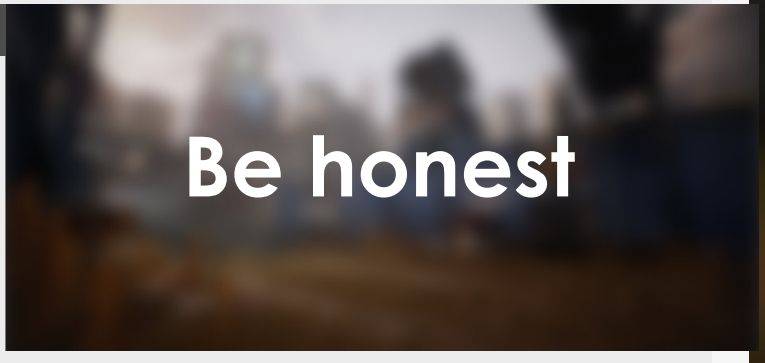
Obviously there are moments that you will just have to bite the bullet and continue doing what you are doing and is more applicable to your personal work, but still something that helps me personally in a big way.

4. Being honest with yourself

It might be obvious at this point, but the most important thing I can say is to find your own way in tackling these thing, try stuff and if it makes you feel better or it helps your art, then try and do this more often.

And if you are still struggling with getting some art done, it might be because you rushed into this project and are no longer motivated to see it through or already thinking about the next thing?

So really be honest yourself and push for something that you love to do, not something that is temporarily interesting.



Asking for feedback

Introduction

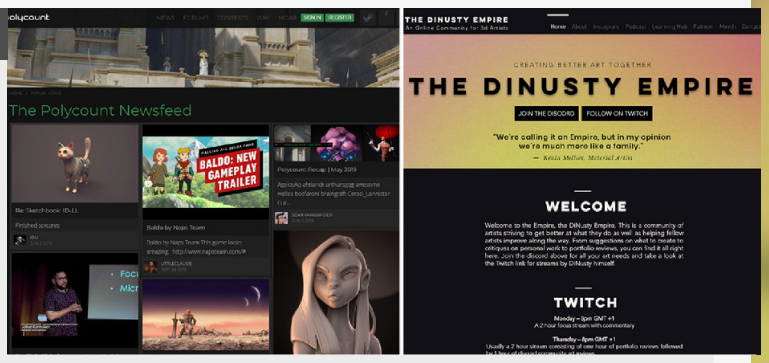
Asking for feedback is one of the more important things you do, but there are so many ways and places you can do so. So today we will look into some tips and tricks that can help you into the feedback that you receive and the way that you reach out and ask for it.

1. Where to ask

There are a lot of places to ask for feedback.

Some of the more typical places to ask for 3D / Environment Artists are probably still some facebook groups such as “Ten thousand hours” and still Polycount, if you are looking to get some good feedback then Polycount is personally still the way to go when it comes to showing your stuff to a crowd of people.

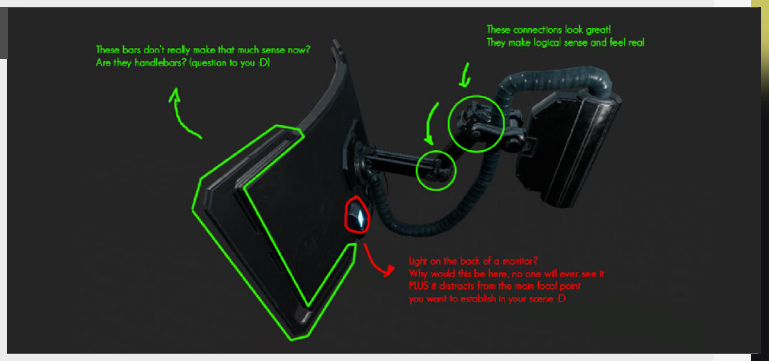
These days it’s easier to find a dedicated Discord groups such as “The Dinusty Empire”, “Experience points” and our community ofcourse!



2. What to take away

You decide what to take away from all the feedback you get. People will give you advice but not all advice is equal. So how do you pick what to take on? My thing has always been looking at the background that this person has and what kind of work that they produce and if they have any professional experience.

Listen to everyone, but in the end you need to pick and choose which point to apply to your project.

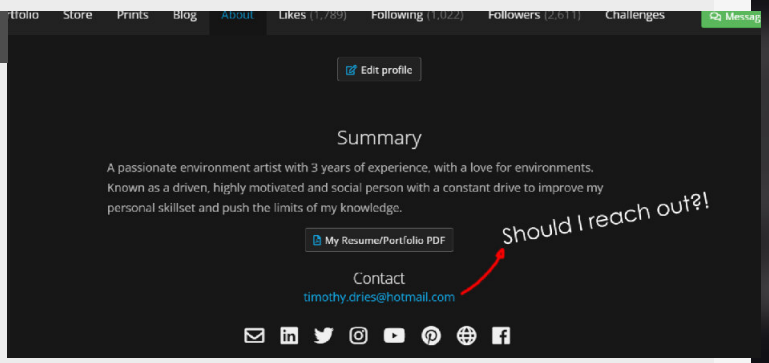


3. Personal feedback

Asking for personal feedback can be super daunting, but I have to say, many developers that I have met in the last couple of years are super open and love to talk to people and are interesting in helping others out.

This being said, getting a reply will highly depend on the way that you interact with that person.

Always be polite, Introduce yourself first and remain friendly and patient.



4. Things to factor in

Don’t feel discouraged if you don’t get an instant reply, we all have things we do on the side as well. Just give it some time and send a reminder in a week or two if you really want their insight.

Lastly though, NEVER be afraid to reach out, we are just people like you and most of us like talking to you and helping you out.

So whenever you have those thoughts about not reaching out because “you are not good enough” or whatever, push that thought back and send them a message!

Self promotion for artists - part 1

Introduction

So you have got a project, some tutorials or a breakdown you want to show off. How do we go about promoting these thing? How can you start building your own brand? Today we start delving into this!

1. Does your work speak?

This title might sound a bit weird, but what I mean with this is that there is no reason for people to listen if your work has nothing to say.

An artist personal expression the reason why people will come to your work, it's the power behind your work, it will speak to people and will attract their attention.

This can be in any way you can think off but it all starts out with finding yourself and what you want to do and expressing that to the best of your abilities.

So start expressing yourself through your work and add stories in them. Don't worry, finding your way in this takes time.

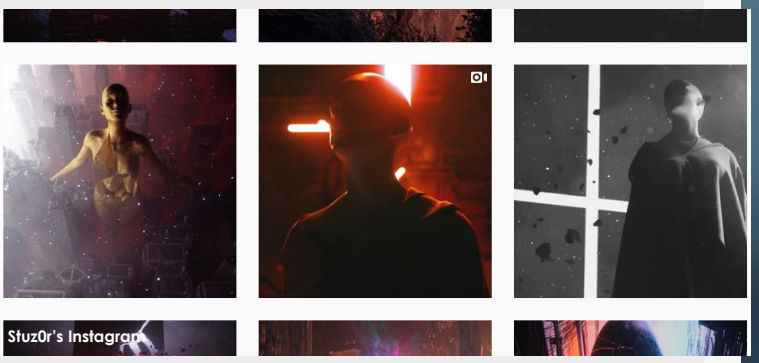
Steven Cormann - Deathloop

2. Consistency is key

If you are looking to build a following, consistency is an overgrowing factor when building your personal community, its not totally necessary but it will def boost it!

When I talk about this it's both for creating an expectation for people like when can they expect a new piece of content from you or when will you share some new piece of information.

Another thing that this will do is building up a personal schedule that will make you work on a regular basis, this is great to build up a habit that will help you boost your productivity.



Stuz0r's Instagram

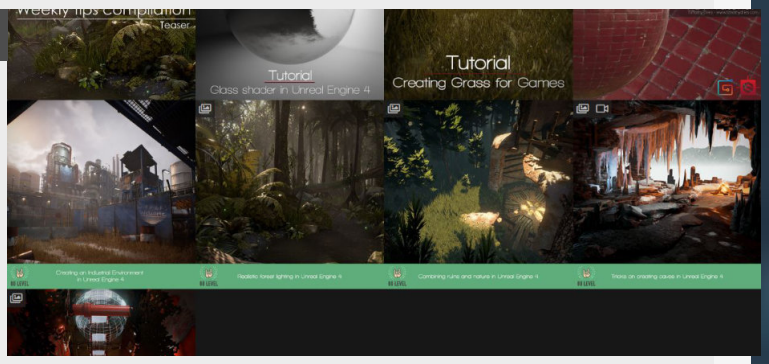
3. Offering value

If you have something awesome that you have build or experimented with a cool new workflow?

Maybe break it down and people might get insight into a whole new way of thinking and see new possibilities.

You never know if this might also open up some new doors for you in the long run where people that want to hire you saw this and know that you want to share knowledge with others and this will play a big role in becoming a solid team fit.

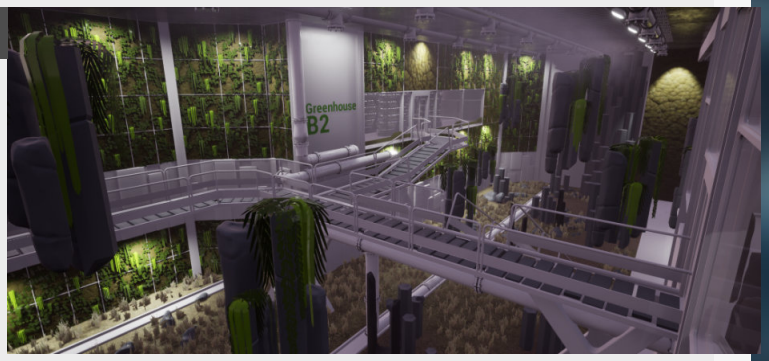
So get out there, start creating and sharing.



4. Focus on the projects

But still, we are artists and what matters most are these awesome projects that you get to create in your artistic journey, if the piece itself is amazing more people will share it and more will see it. So all the time you take away from building great projects is time you don't get to fulfill your own passion, make sure to constantly keep this balance in check.

Because a piece that isn't made well will almost never get the attention it deserves (there are exceptions to this rule...). So keep focusing on what makes your art truly yours and keep hammering away at it until it's the best that you can do.



Self promotion for artists - part 02

Introduction

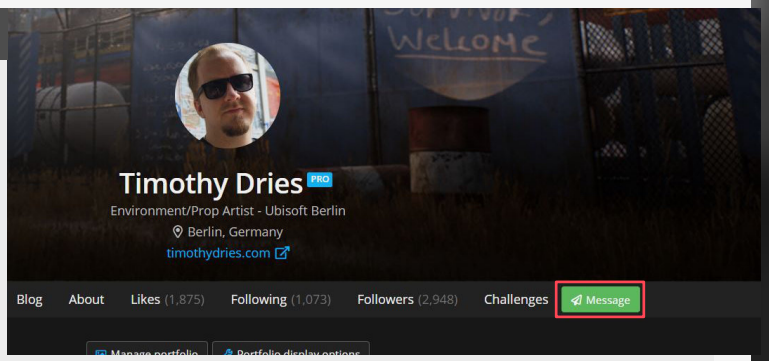
In the second part of self promotion for people we will look into some other steps that you can use to further market yourself and your art.

1. Reaching out to people

If you are trying to get into the games industry reaching out and building genuine connections can be crucial when you want to break into the industry. Building these needs to come from a genuine place though and building these relationships takes time and effort, so don't expect any short term wins here.

Maybe the only exception to this rule would be reaching out to people for business purposes, like working with them, asking about their podcast or blog.

Just make sure to keep it polite, but don't let this hold you back from reaching out because the people you are reaching out to love to have a chat!

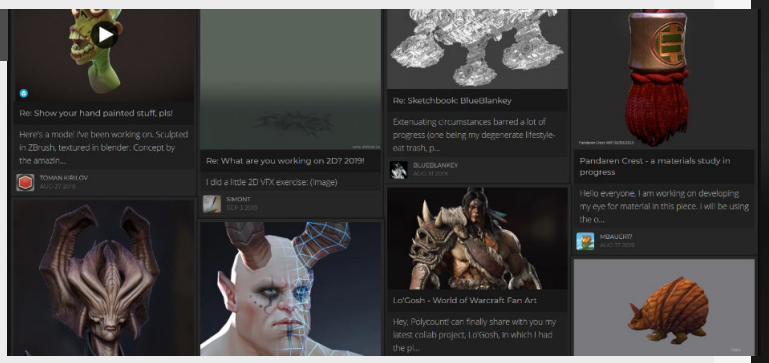


2. Places in the community

A way to find a lot of like minded people is to join servers like the Dinusty, Experience points or others if you're looking for lively places.

What really worked well for me especially when I was purely focusing on my personal work was posting stuff on forms like polycount or Unreal Engine forums for example, this would be very low maintenance from my side with the help of pre-build PhotoShop templates. So this gave me more time to focus on my work instead of doing touch-ups etc...

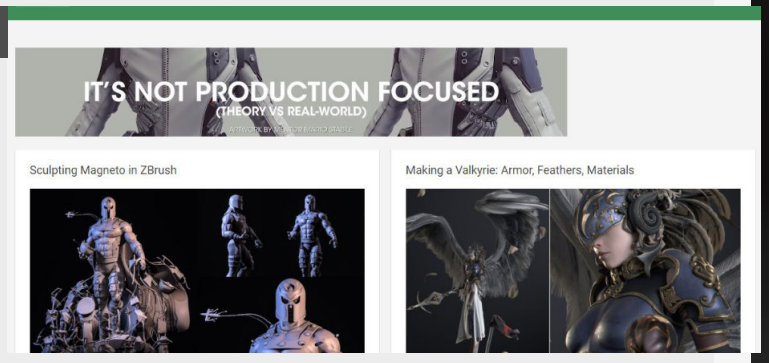
So try different things, don't be afraid to post your stuff and find what works for you.



3. External promotion

So you are doing a lot of things to get you seen all over the place, but what if you start thinking about selling products or tutorials that you want to get seen by as many people as possible? This is where looking for people to help you with getting your stuff out there comes in. There are tons of options like looking for small YouTubers, streamers that want to promote you, sites like 80.lv for your breakdowns or even more traditional ones like getting featured on Polycount if you share your work on there.

Personally the only stuff that helped me at this point is doing podcasts and articles/ breakdowns on 80.lv and it has helped me a lot!



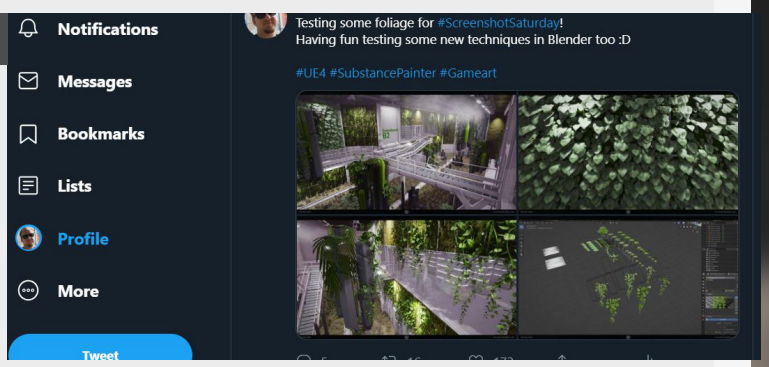
4. Social media outreach

The thing there is, social media platforms are more and more catering their algorithms towards people that are more active or towards outrage.

So organic growth is getting harder and harder to achieve as an artist just starting out.

Which is also why Artstation is really nice to build a following on, people genuinely appreciate your work and it can get seen by people that are directly in positions yo hire you, making it even more attractive.

But how you balance this is totally up to you, just don't get trapped in trying to be on all the platforms all the time.



Avoiding exposure work

Introduction

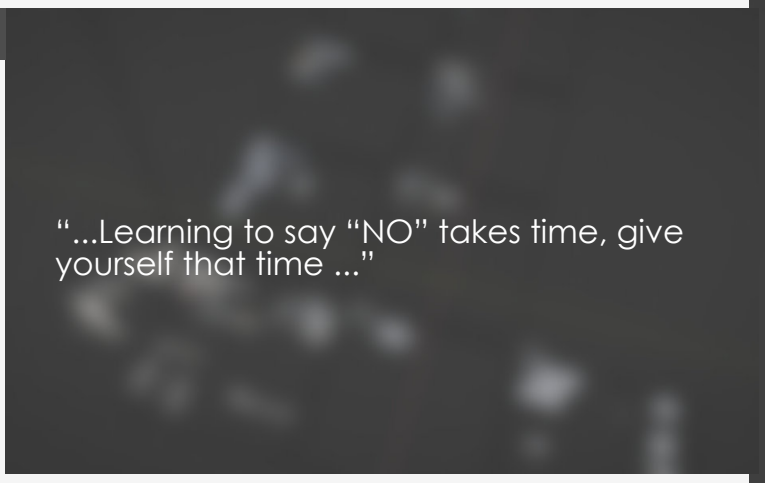
We have all heard the saying **"working for exposure"**, but you can't survive on getting exposure and you need to get some actual money on the table. Artists have the most difficult job when it comes to this because their passion for art gets seen as excessive and something that is **"just making pretty pictures"**

1. Learn to say "NO"

Let's address the elephant in the room first, we have all heard people talking about their project and why you should join it and felt little butterflies in your belly.

But that's also the time you need to take a step back and see if you are still thinking about the situation in a detached way, without getting hyped and excited till the point where all logic flies out of the window. That's the best way to get taken advantage off and get into stuff you wouldn't want to get into.

Learning to say **"no"** takes time, but it always starts in the same way, a whole bunch of stress and not knowing what the other party will say, keep it professional and polite.



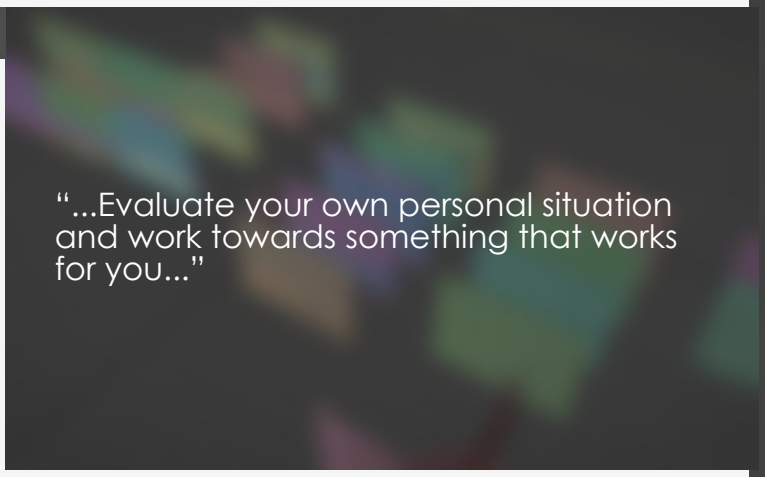
2. Your personal situation

Financial freedom has been a big goal for me in recent years and I still have a long way to go when it comes to achieving that goal, but I have a relatively stable situation so yours may differ.

For example you might still be looking for a job and be working on portfolio pieces but even in this situation you might as well spend some extra time thinking and finding ways to get some money out of your time you spend working on it.

It doesn't matter when you do it, it matters if you do it. So once you have gone through your journey and got your job it might be worth looking at all the stuff you created and see what might get you an income.

You might be surprised by the result.



3. Active versus Passive

When it comes to building an income for yourself spending some time to define what kind of income works best for you personally. There generally are two ways.

"Active" means that you put in the time once and you get the income once, this is great for the short term because you get to ask a bigger price for a more unique experience. An example of this could be a mentorship that you do on a monthly basis for instance.

"Passive" on the other hand is where you build up one thing and can sell it multiple times to multiple sources. Another awesome effect of is that the more you build these assets people are going to are the other ones and might buy them as well. This compounding effect makes it super interesting for the long term.

There are so many nuanced ways that you can really make full use of this, like starting a patreon, making YouTube videos, have a site where you sell your products and more.

Earning on the side

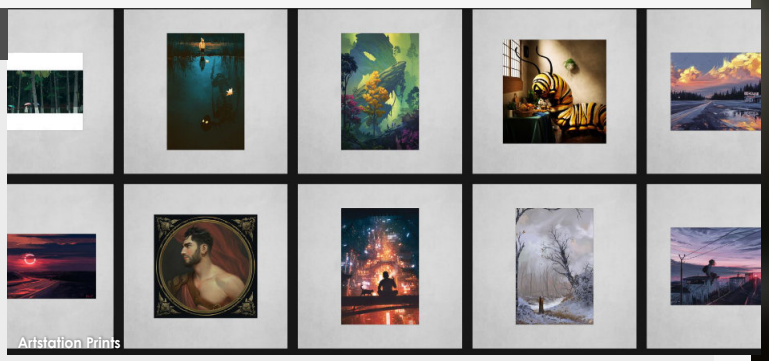
Introduction

Let's dive into some tips and tricks how to start your journey from being a "Starving artist" and making your way into the space where you can enjoy making art and getting some money from it too. Some options lend them self better to this balance than other options though so keep this in mind when thinking about these options. And as with most things on these tip sections, there are countless opportunities out there so explore these for yourself!

1. Selling prints

Probably the most easiest thing to look into is turning your artwork into prints, this can be easily done with Artstation prints, you can just upload a high resolution version of your work and then have them as a sell-able print. Obviously there are other options that allow for this and might even give you better results, but the possibility to have it on the same location as your portfolio for just a couple of extra clicks is amazing.

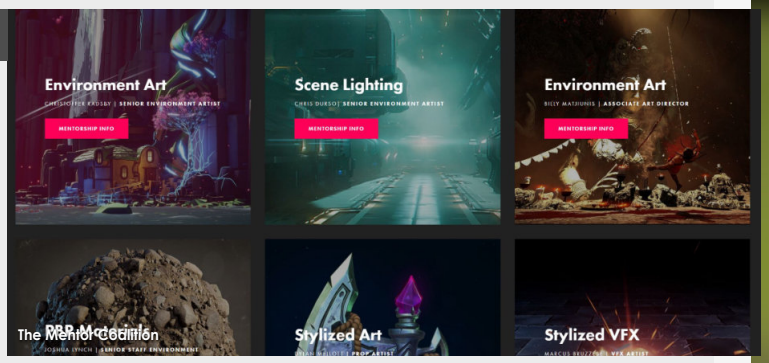
This all means that this is one of the options that doesn't impact your workflow at all, because all it takes is making higher resolution screenshots and then adding them as prints if you are using Artstation.



2. Mentorships/Teaching & Tutorials

The teaching realm is always the option to start doing mentorships or maybe start a tutorial course that will allow you to try and help/educate a large portion of people. From this point on you can decide on whether you want to release them on a platform like youtube, but as far as I know this will not bring in that much if that is what you are looking for, so you might be better off selling these tutorials on a platform like Gumroad, Cubebrush or Artstation.

Keep in mind that you need to invest a large portion of your own time into the people that you mentor if you want to do it properly (and you should).

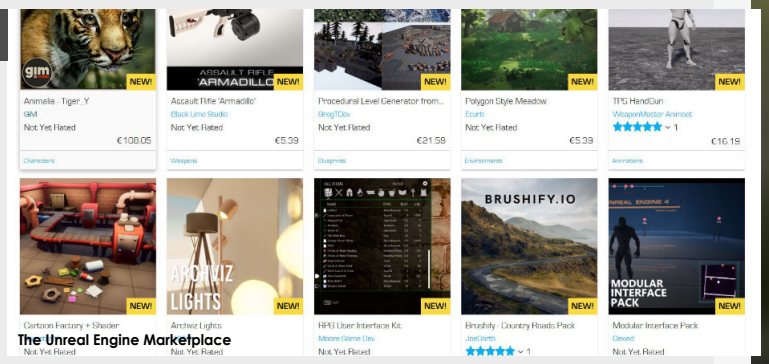


3. Assets or Asset Packages

This is probably the first thing to think about when building your own environments, when you are going through all the trouble of making all those assets, why not try and make a little bit of money out of it, it will take you a little bit more time to clean them up, name them correctly and address some of the minor issues for a potentially nice income.

Some sites you can have a look at are CG trader, Unreal Engine Marketplace and Turbosquid. Every site has it's advantages, so make sure that you do some proper research before hand and pick the one that suits your needs.

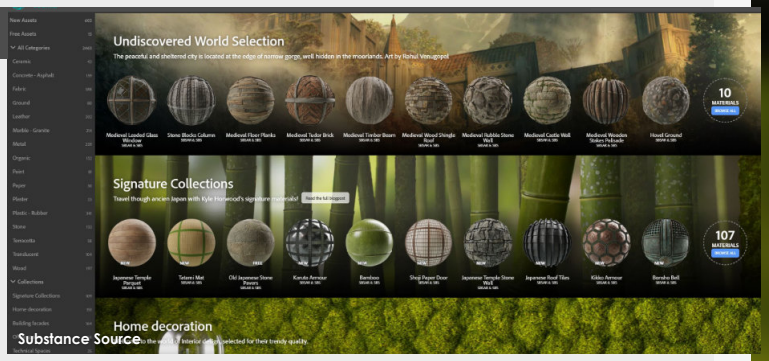
Make sure that making this doesn't take away from your own creative output!



4. Materials

This uses the same principles as the previous section but take way less time to create and allows you to do more different things in the same time you would take to make a model and definitely less to make a full environment. There are some people that really create some interesting looking materials. This also open you up for doing breakdowns and tutorials of them too that can help you boost the sales of them.

This is a really fun option if you are creating materials in Substance Designer anyways, you might as well make them available online for other people too, either through Substance Source or any other platform of your choosing.

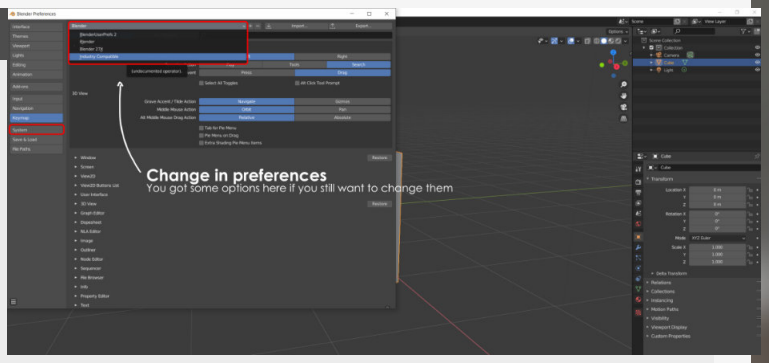


Switching to Blender - Quick Tips

Introduction

Switching programs, especially at this moment when blender becomes such a powerful, viable and most importantly, free option. This weekly tip will be all about some of the lessons I've personally learned when switching to Blender.

1. Stick to the default keybinds

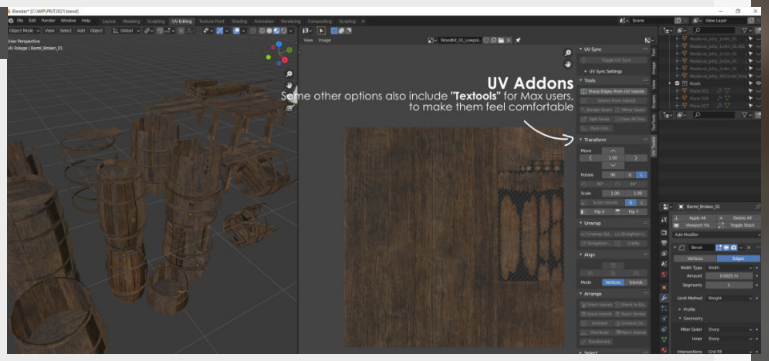


This was probably the biggest mistake I made was when I tried to force Maya into Blender in the beginning, even making my own Pie menu's at a certain point.

At this to say that I would recommend learning blender the way that it's to be used, it's super weird in the beginning, but once you get used to it, it really starts to make sense in terms of how to use it, it feels really intuitive and much quicker t use, at least it did in my case.

So give it a fair chance and stick with it in the beginning, it does take a bit off time to fully switch. I think it took me a month or something to get a bit comfortable with it.

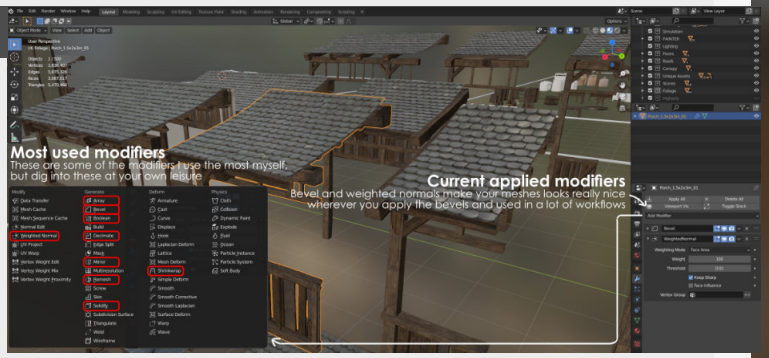
2. It's biggest flaw currently



The biggest hurdle is definitely the UV tools that are pretty basic at the point of writing this, so this is a good point to how you can make it work a little better with the use of some addons, so for UV's specifically I use **"UV Toolkit"** which I can recommend, it has all the tools I need, apart for a Texel Density checker, which is my second recommendation.

Now, you've heard that Blender is run on addons, but my advice would be not to over-rely on them, especially the paid ones might be unavailable to you once you join a company because the license doesn't cover it, so once you get a job you might have to live without those shiny addons.

3. Don't ignore the modifiers!

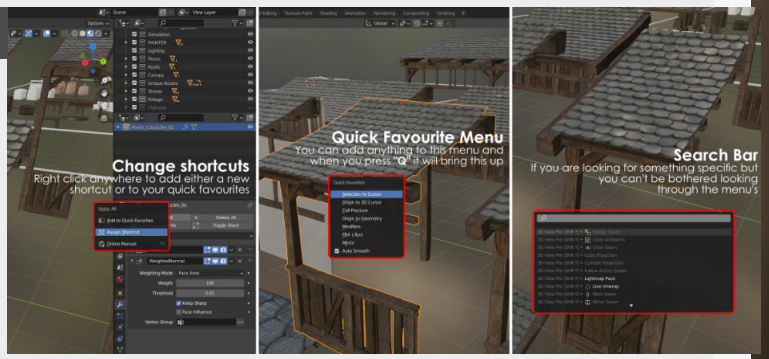


This is one of the bigger thing, when coming from Maya especially, these modifiers are amazing!

Thy allow you to add operations in a non-destructive way and give you way more playing space. No more duplicate meshes that you use as a backup or stacks of irrelevant incremental save-files.

The modifiers I use the most really depend on the workflow I use, but my favorites are the **"Bevel"**, (in combination with the **"Edge bevel weight"**,see more in **"Blender Bevel Modifier"**) **"Decimate"** (to optimize meshes) and **"Boolean"** (To cut shapes out quickly if I need too)

4. Searchbar, Hotkeys & quick access menu



This was a big gripe of me when working in older Maya versions, is that you could never easily adjust or add a new hotkey. But now it's so easy to do so in Blender, it's just a right-click away and you can set your preferred hotkey.

However, if you are worried about replacing a hotkey that is already taken you can also add it to the quick access menu, this menu is also contextual, which is amazing on itself, which is also why I use it so much myself.

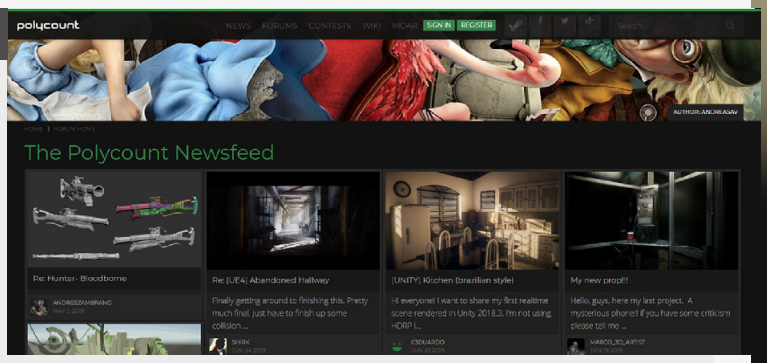
"F3" brings up a little search menu that is super useful, no more looking through all the menu's to find something, let Blender do the work for you.

Resources - 01

Introduction

There are a ton of resources out there, the issue is finding it, a lot of these you might have heard of and are more aimed towards beginners, but in future parts, we will be diving into some more specific stuff on the web to help you grow and expand your artistic skill set and yourself.

1. Polycount



This was my home for a long time, and it can still be yours too, hosting an amazing number of threads full of inspiration and feedback.

There are also others things such as challenge threads that really allow you to push your skill set together with other people working on the same concept and are an interesting way to inspire and get inspired by others.

So check them out if you are looking to get feedback or just want a new way to push yourself.

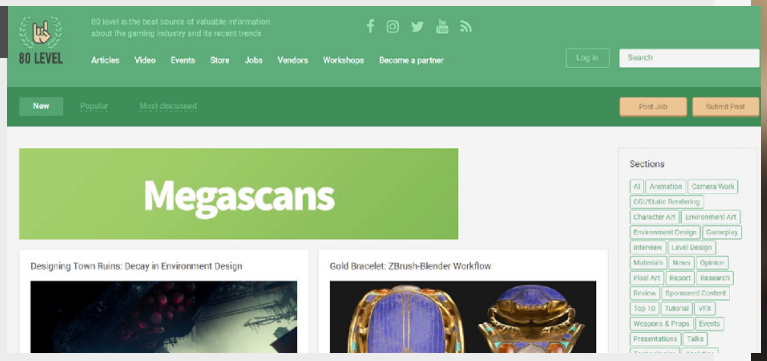
2. The Dinusty Empire

Started by Jeremy Estrellado, this is a great community to join if you are looking to improve your skillset and get feedback from other fellow artists. With people dedicated to see both the community grow and help other people out this is place to go.

A really big thing are the portfolio reviews that Jeremy does, they are super useful to get your portfolio ready to hit the games industry hard.

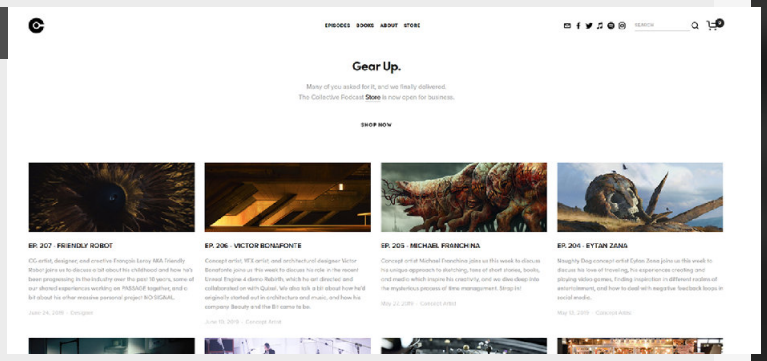
There is also a podcast attached to this community which interviews all kinds of people involved in the games industry, check it out!

3. 80.lv



Another great resource for a lot of cool tips, tools and inspiration! The reason why i find this a great resource personally is because they don't just push the top of the line work, they also share work/ breakdowns from students or people rather new to the industry.
This makes it a cool tool to gain insight on new workflows that might not know about and explore different tools.
Also if you got something to show of yourself, you can share it in the 80lv Facebook group and it might get picked up!

4. The collective podcast



My personal favorite when it comes to podcasts!

The strength of this podcast lies in the guests that Ash Thorp brings on, they are not limited to a certain field but are all really inspiring to listen to and get inspired in a way that you is not directly related to the field/ discipline you are working in.

Each episode usually covers a wide range of subjects and goes into something new and exciting every time.

Detaching from your work

Introduction

When working on art it's not always about sitting down and actually doing it, it's also about giving your brain a bit of rest to properly detach from it so you can go back at it at a later stage. But what is a good replacement, how long should the break last and how often?

1. When to take a rest?

Let's get the tricky question out of the way first, making art can be hard especially in the beginning when you are still limited more by your technical knowledge and also because everyone's limit for how long you can focus and work is different.

However, one good indication is that once you are actually doing the work, but you feel that you are getting continuously frustrated, or even feel like you are not seeing any meaningful progress anymore.

Then it might be a good time to take a little break and step away from your work, so what are the best ways of doing so?

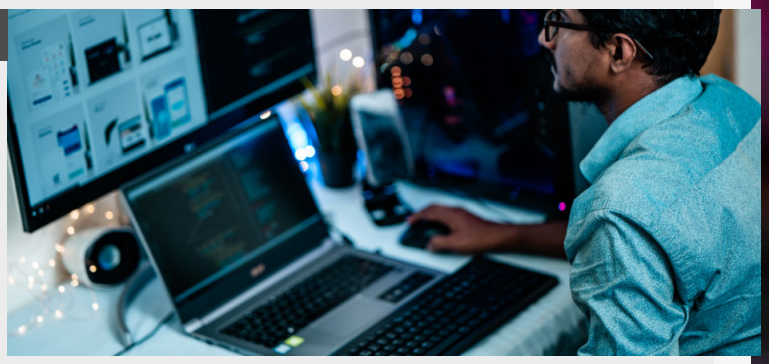
2. How to detach

GO OUT FOR A WALK: Go outside, no music, no phone, get some sun, and leave any distractions at home. I love to let my brain wander by going into a new direction and explore some unknown area's.

WORKING OUT/HOBBIES: an alternative to walking is working out. Because this requires complete focus, especially if doing hard exercises, so there is no mental capacity left for any other thoughts.

SLEEP: When working long hours it might be time to just go to bed earlier than usual. The amount of times I've went to sleep and the following morning when I wake up the problem seemed to solve itself.

3. Getting back into it



Still feeling like you need a break even after a couple of days? It might be time to try a different approach.

For me what really helps me a lot is surrounding myself with people that are passionate with what they do for their own work. There is honestly nothing better to spark passion than a passionate person doing what they love.

Some other recommendations:

- Rediscover yourself, try to go back to what made you start with art and explore.
- Starting up a second project and work on it in tandem, once you get bored of one, you switch to the other.
- Looking around on Artstation is also a good way to get the spark going too.

4. Long term breaks

Not designed to give up on your dreams but to ignite the passion you once had.

Usually I try to go traveling at least once a year. Exploring a new culture, discovering new influences and doing photography really drives em to want to create again once we get home, plus it helps to get away from the screen as well.

Making games and their art is mentally exhausting, it takes up a lot of mental capacity to get to a place where you can execute what you have in your head, so don't give up when the going gets though, it's worth sticking with it!

Being Objective about your work

Introduction

Being objective about your work is important for artists, from shielding yourself from critique and turning defensive to improving how you give feedback to yourself by looking it in a new way

1. Benefits of objectivity

Being objective is something that might be tricky especially if your just starting out, you're heavily attached to your work and it becomes who you are for a long time. However, we must also realize that this puts up a wall that makes it harder for people to give you honest critique and help you grow because you're not willing to genuinely listen.

So let's break down those barriers, starting with the trickiest one.

2. Your work is not all you are

When spending hours, days, weeks or longer on your art, trying to improve every aspect of it and getting properly immersed in becoming a better artist is that at a certain point you fade the borders between yourself and your work.

Tying all of what you are to you projects sets you up for failure, so make sure you let go of that attachment and be ready to receive critique and feedback from others and start growing.

There is no real "Strategy" here either, at least not from my side, it's realizing that you are becoming defensive and choosing to not do so next time (or currently)

3. Third Person Perspective

If you don't necessarily want to reach out to others at this point or don't feel like doing so it's time to train yourself to do. By thinking about your work in Third person.

This is a great way to partially rid yourself from your own biases and your own weight attached to the project.

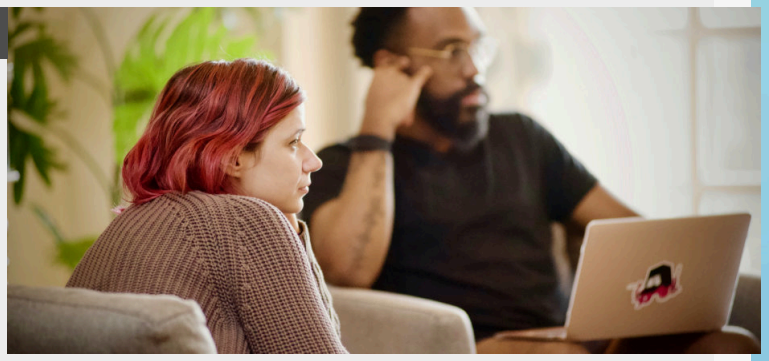
If you feel like this doesn't work, it might work to train giving feedback in Discord Groups or to fellow artists. Seeing improvements comes with experience, so the more you expose yourself to others giving you feedback or expose yourself to giving it to others. The better you will get at doing it to yourself.

4. Critique is learning

This kind of relates to the second topic and is also important. It's the realization that anyone out there giving you feedback is not there to bring you down (generally) but is there to genuinely help you on your artistic journey.

So allow people to give them the feedback, listen carefully and make sure to make notes.

Don't interrupt them by trying to defend your work, as much as you want too. Just take all the feedback in and put a value on it during your own time. As we said in the first section, they are not there to break your work down, and trying to instantly defend yourself will make it seem like you're not listening and just getting ready for battle.



Tips of giving feedback

Introduction

Giving feedback is complicated and should be seen as a gift, that also mean that you need to take into account of the person you give it to, so let me give you a couple of examples of what that means to me.

1. Be constructive

Helping someone improve their art and helping them grow are the main reasons why you want to give feedback and the best way to do this is by being constructive.

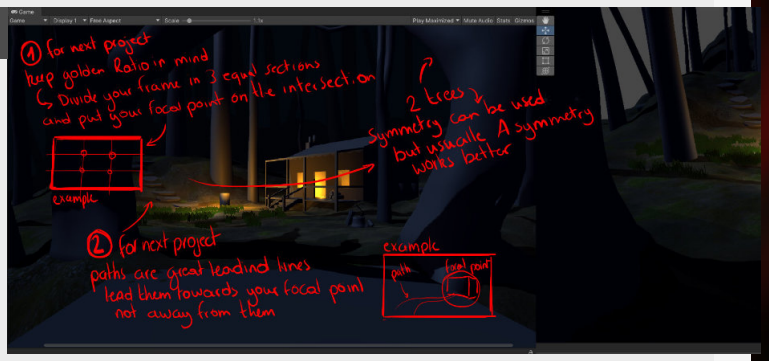
This means that a balance between being critical while mixing it in with praise and things you like about their work is crucial.

2. Be specific and use examples

Visual examples are the best form of giving feedback, this is why overpaints and images added on top of them can add so much for visual feedback, it gives us artists something to look at.

Stay away from non concrete images and words, for example: *"Let's make this shape more fun"* Fun in this sentence doesn't contain anything concrete and leave too much room for interpretation.

A better example could be "Let's make the top of the shape more square and have a pear like form for the rest of the shape" this is actionable.



3. Involve the person into the conversation

Instead of saying what or how they need to do something, ask what they would do and let them explain it first, this always opens op the conversation and allows you to highlight the good ideas they bring to the table.

4. Support their vision

Really important for artists, we always have a base idea for the work we do, and it's horrible for someone to come in and stomp all over it and enforce their own idea on it. Instead of doing this try to build onto it instead, add onto the base idea the other person already has and make it a real collaborative brainstorming session.

If you don't like some of the ideas that are there then always be ready to suggest multiple alternatives, especially visual example ones are always great to inspire each other.

It's their work and their time they will put into it.

18 BREAKING INTO THE INDUSTRY

Art test tips - 01

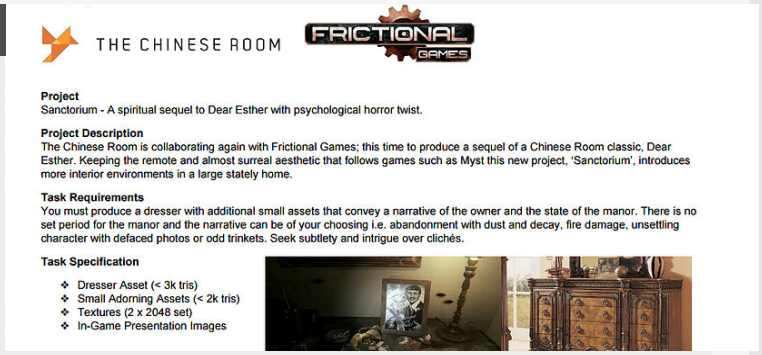
Introduction

Something most of us will have to go through (unfortunately in most cases...) when looking for a job, especially in the beginning of your career are art tests, making this the final challenge before you get your hard worked for job. In this we will discuss some tips on how to not let these overwhelm you.

1. Understanding the brief

The first one might seem super obvious bit there is a difference between reading and understanding the Brief. These Briefs are setup to test specific skills and not only just to make something look pretty. The closer you can get your work to match with the brief's intend the more likely it is you will get that job!

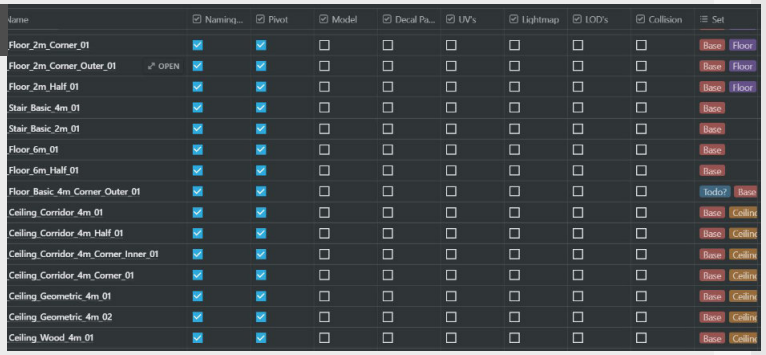
So read through it carefully and understand it before you dive in, this will really help you get the clearest path to your end result.



2. Planning it out

Now we know what the goal is we need to make a proper plan, with this I don't want to force you to write stuff down and break it down into multiple chunks in written form before getting started (although this is the best way to create a meaningful link between you and your task) everyone has their own way of breaking these down, so just use what suits you best.

The most important thing here is to break the brief down into multiple chunks that you can easily digest, helping you not to get overwhelmed.



3. Execution time

When getting to work on your tasks the focus needs to be on the tasks that are most important or will have the biggest impact on your goals. Avoid doing the smaller less important tasks.

Another thing to be aware of is to give yourself some breaks, stretch your legs and drink water. No matter how daunting the task itself is, those breaks will help you clear your head and look at things in a new way, I know that might be difficult if you are in the zone, but don't burn yourself out on this, there might be other tests that you have to do in the same period.

4. Don't get overwhelmed

Most of the time these test are really big in scope, sometimes even done on purpose tot see what you will focus on and how you manage your time.

This is also why you can't be paralyzed when you read a brief that is super daunting and seems like way to much to handle. Go back to step 2, build a plan, prioritize and just go through it step by step.

Keep taking it step by step and keep pushing, you got this.

Art test tips - 02

Introduction

Something most of us will have to go through (unfortunately in most cases...) when looking for a job, especially in the beginning of your career are art tests, making this the final challenge before you get your hard worked for job. In this we will discuss some tips on how to not let these overwhelm you. (PS, enjoy the generic photos! :D)

1. Don't doubt yourself

As you are working on your art test and you are getting to the end of this entire experience it might be easy to start doubting yourself and thinking about all the stuff you would do differently if you had more time. Don't let this hold you back from handing your test in though, let others be the judge of your work and how well you performed, you are always going to be extra critical about your own work.

Just give it all you got and send it over on time.

2. Ask questions

Another important one, which a lot of people, including myself, seem to forget is that if there is something that's unclear about the brief don't be afraid to ask questions about it.

Especially when people set up briefs they use the wording they would use in the industry, if there are things that are unclear, just ask about it, it doesn't show lack of knowledge, but the willingness to learn.

Additionally, asking the right questions is a large part of game development with so many people involved, so see it as another skill that can be useful for working within a team.

3. Add your own touch

You are an artist and that's also a thing that they want to see, so make sure that you add some personal bits of creativity doing this will make you stand out from all the rest of the artists that are potentially doing the same test.

However, make sure that you treat this as an extra though and don't kill yourself over this.

Keep focused and make sure that you focus on what the brief is trying to get you to focus on and what they want to see is there and add this if you have time left or want to spend extra time on it. (For more look at Blog #85 or "Art Test Tips - Part One").

4. Be on time!

And finally, most important of all, make sure that you are on time, there is nothing worse than getting a test that is really good but isn't within the deadline.

Deadlines are a big part of the industry too, so respect them. However, life comes first, if there is something urgent or personal that will be causing you to miss that deadline, then just let them know and discuss if you can more the deadline or do the test another time perhaps.

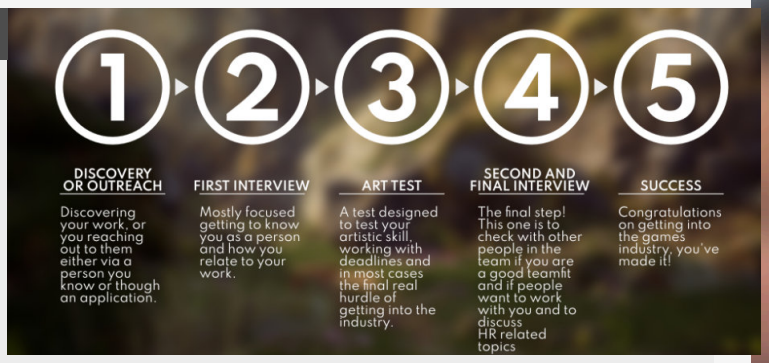
Keep at it, you got this!

What to expect from your first interview?

Introduction

So you're getting to that point where you are ready to get into the industry, but what can you expect from the interviews stage? Let's dive into that today

1. There are multiple stages



Stages of the interview may vary and how many their are might differ from company to company but usually their are at least two.

First interview: this is more just to get to know you and what person you are, at this point they are interested in our work.

Art test: maybe you get asked to do a small test for the company, which is totally normal and just indicates that they are interested in your work, but need to test specific often workflow crucial items

Second Interview: Often time the final stage, where they know all they need to know about your hard skills

2. You will be discussing your work

Usually these interviews put you on the spot with an introduction yourself and your work. So keep your portfolio at the ready.

Also make sure that you understand the things you have done, you will be asked questions about how you went about your work. And if you don't know about elements just mentioned where you got them from or what inspired you, that's fine, just don't hide it. It can look really bad if you try to hide or made stuff up, you don't have to have answers to everything.

"How and what would you improve about this project" is a frequently asked question here.

3. Difference between hard and soft skills

Hard skills: Technical skills such as your program knowledge, your artistic insight into art fundamentals and pipeline knowledge. This is important to how you will perform on creating art.

Soft skills: Personal skills, like how you deal with stress, your general demeanor, how open you are to sharing, how you talk to other people, etc... This dictates how you work within a team, if you can manage deadlines and stress.

Often even putting more focus on the soft skills their final decision because a mediocre artist who works well within a team is better then a rockstar artist who doesn't.

4. Difficult situation roleplay

We sort of mentioned it a little before, but you will get a lot of questions, from how you have done things to how you would tackle difficult situations and you will probably get put into one of these situations yourself.

There is nothing you can do to prepare for these types of questions because they are so varied and come in all shapes and sizes. So just be open and honest if you such a question comes up, being honest about these (Even if you don't know!) are totally fine, they can show you that you are open to learn new things and aren't afraid to admit it either.

Your first week in the industry

Introduction

So you've finally made it, you're starting your first position inside a studio now. But you're not really sure what to expect, let's go over some things you might encounter in your first weeks at a games studio.

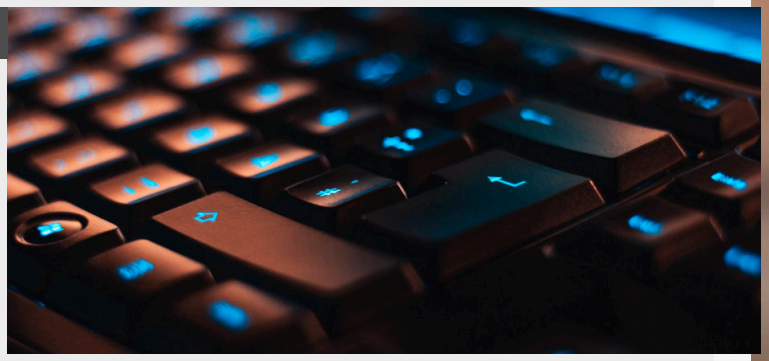
1. Getting a tour

First off all, when you're actually going to be allowed into an office again (this was written in 2021, during Covid-19) you will be getting a full tour with some other new starters to get the full tour of the premises.

Getting a full tour of all the neat rooms for taking breaks, where to get lunch or awesome gamerooms, you've made it in, so enjoy this moment and if you've got any questions, just fire away!

Lastly you will be guided to your own desk, most companies even provide you with a nice welcome package to get you started on your day and celebrate you joining the industry.

2. Onboarding onto a project



Once the tour is over (this can take up the entire day, in some cases even more!) it's time to get started.

Starting with just familiarizing yourself with your desk, changing passwords, creating your account, double checking if you have all the software you need. For this, most of the information you can find on the internal wiki pages or by asking some of the people around you.

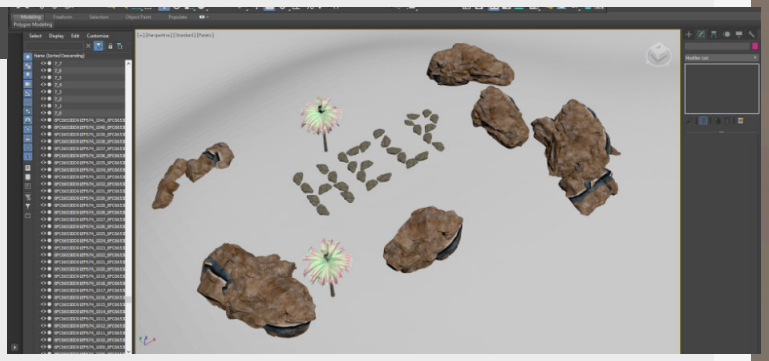
You might also test the game you will be working, most of the time this comes with an art bible as well as playing the game in it's current state to see how it's currently looking and how you can fit in to make it even better!

3. Getting to know the team

Now while you are doing this you're going to be surrounded by a ton of new people as well, so take this as an opportunity to introduce yourself and get to know the other members of your team.

Some companies they add a little treat into the mix and they have larger company gatherings and they have this initial meeting to allow all the newcomers to present them self shortly, so this might be scary, but not all companies do this, and I can tell you it was also a surprise for me when they told me. You will also be introduced to your mentor and or lead at a certain point, they will most likely be your point of contact, so make sure to ask them questions if you have them.

4. Discovering the pipeline



Now It's time to try your way around the actual art pipeline the company uses. You will highly likely get the time to test some stuff yourself, check out some documentation on how the engine works and how to get started in it yourself.

If at any point there are some issues that you can't resolve on your own, don't be afraid to ask any question about it to the people around you, they can definitely help you out with anything. Make sure to take your time here, that time invested now can save you massive amounts of time later on.

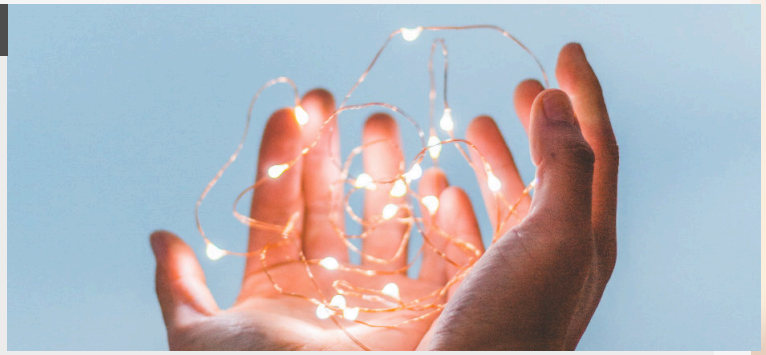
After this, you're all set for your first task!

Reaching out to professionals

Introduction

This is a bit of a different kind of tip, but it's important for people to understand some ground rules when reaching out to professionals and getting feedback on their work or some other sort of help.

1. Open the door yourself



It might feel like you need to wait for permission to reach out to professionals, but don't wait on that. Open that door yourself.

There are multiple benefits to this, firs up really good practice to just get more comfortable with reaching out to other professionals first place. It's also a nice way to maybe even get to know them better and a stepping stone to create some connections too.

So don't let that ever stop you from reaching out, just follow the next couple of steps and you should be good to go.

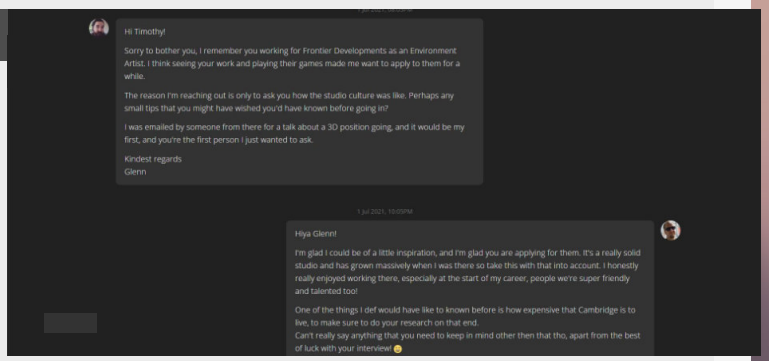
2. Be polite

I personally always felt that it's always better to go with something in the middle when it comes to your approach.

Also, don't complain about not getting an answer immediately, don't add "Bro" in your message, just introduce yourself in a pleasant way so we get to relate with who you are as a person. A person that we want to help.

Because at the end of the day, that's what this is. You are reaching out for help and advice, the more respectful and polite you do this the better the end result is to be expected, because if you are respectful to others, they will most of the time return the favor.

3. Keep it simple

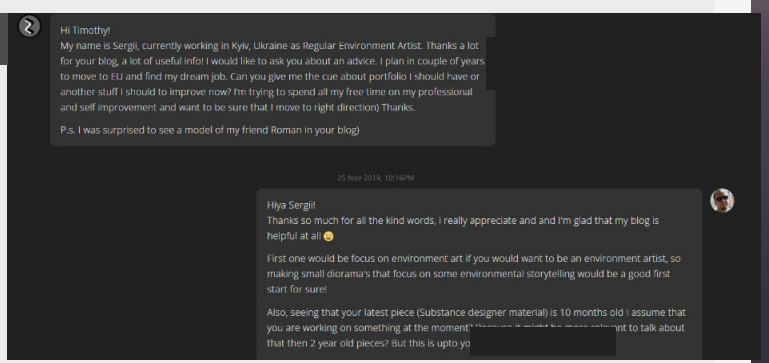


It's no secret that game developers can lead busy lives and I know a lot of them personally that still do a lot of work outside of their normal job, so think about this when reaching out to them.

Make it easy for them to read your message and reply without needing to write and entire book.

An example of this could be
- Short personal introduction, name, discipline (or interest), currently learning at a school? etc..
- Your question, as specific as you can be
- Polite ending such as "Thanks for your time"

4. Be as specific as you can



Do a little research before approaching anyone in the field "How do I get into games, I want to work on games", just doesn't work as a question and is probably better spent googling.

Try something like "... I want to become and environment artist, how do you suggest to get started?" is just way better to give a reply to.

Specific questions get specific answers and can help you more than a broad summary.

And if you ever wanted to ask a question, message me on Twitter (or Discord if you are part of the Beyond Extent community) I would love to help whenever I can.

Environment Art Tests

Introduction

Art tests are really common as the last big hurdle for you to get a job in the games industry, they are more prevalent when you don't that much professional experience just yet, but some companies let everyone do an art test as well, so what are they, and is there anything you can do to prepare for them?

1. What are they?

During the course of your application process you might be ask to do an Art test, these usually done because during your talks with the company they found that there we're still some questions in their head that could be solved during those talks.

These tests can last from really short ones of a couple of hours to multiple week tests. However, in my opinion the multiple weeks ones should be avoided, unless the company is really well know and it would be your personal goal to work for them.


Credit - Scott Homer

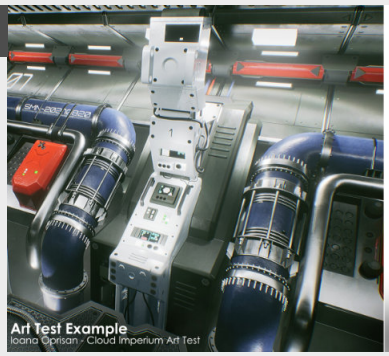

Credit - Nicola VanHemelRyck

2. How can I prepare for them?

Anticipate the workflow used or the subject matter you might be working on, and what better way to check that is by looking online if there are already test available that have been finished for the company.

If you cant find any, you can check the projects the company did before and try to find out their workflow,maybe there is some podcast or breakdown that you can track down. Also check the application for any details that might help.

IMPORTANT
This doesn't mean that this is the test they will hand to you as well.

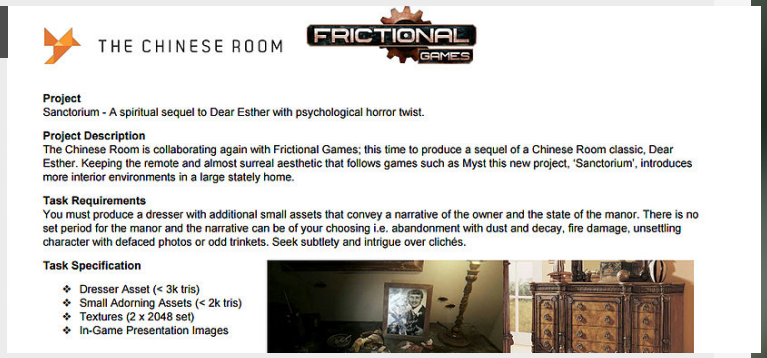

Art Test Example
Ioana Oprisan - Cloud Imperium Art Test


Application Example
Senior Environment Artist Opening at Ubisoft Berlin

3. Getting the briefing

Take your time to read the brief multiple times so you completely understand the criteria you're working towards. This is super important, because that is your goal and you should make your plan based on this goal.

Which brings us to the next step, which is planning ahead. Now how you do planning that's totally up to you, whether it's mentally putting things in line, writing it on post it notes or using an app. You need to be able to focus on the things that have the biggest impact on your scene first and then move down that list.


THE CHINESE ROOM

FRICTIONAL GAMES

Project
Sanctorium - A spiritual sequel to Dear Esther with psychological horror twist.

Project Description
The Chinese Room is collaborating again with Frictional Games; this time to produce a sequel of a Chinese Room classic, Dear Esther. Keeping the remote and almost surreal aesthetic that follows games such as Myst this new project, 'Sanctorium', introduces more interior environments in a large stately home.

Task Requirements
You must produce a dresser with additional small assets that convey a narrative of the owner and the state of the manor. There is no set period for the manor and the narrative can be of your choosing i.e. abandonment with dust and decay, fire damage, unsettling character with defaced photos or odd trinkets. Seek subtlety and intrigue over clichés.

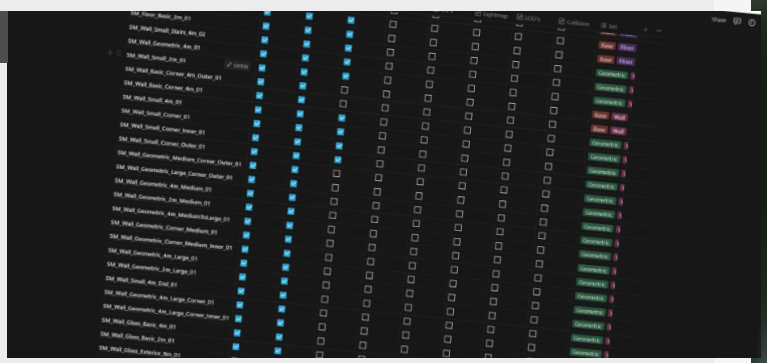
Task Specification

- Dresser Asset (< 3k tris)
- Small Adorning Assets (< 2k tris)
- Textures (2 x 2048 set)
- In-Game Presentation Images

4. During the test itself

Because we have build a plan in the previous step we can now use this as a foundation for when we actually get to work on the test itself, this is important as a large section of a test is also to see how you do under stress. So whenever you feel like your lost take a step back and go back to the plan you've created.

Focus from big to small, the things that have the biggest impact will be the most important, don't get hung up with smaller things that have minimal impact, keep them for when the bigger things are out of the way.



19 INDUSTRY INSIGHT

Your day as an Open world Environment Artist

Introduction

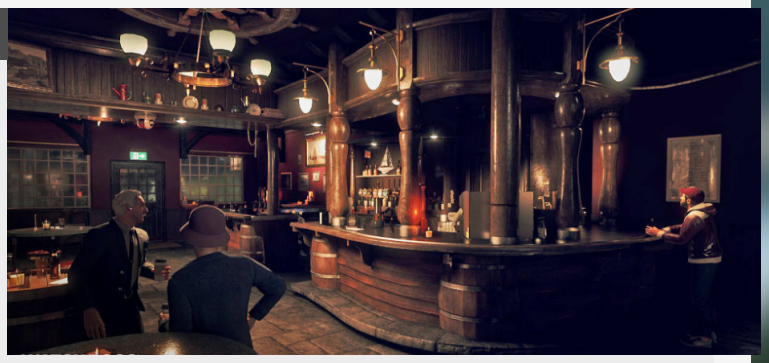
As an environment artist you might be responsible for a lot of aspects of the visuals for a game, and **"Environment artist"** is sort of the catch all for a lot of stuff when it comes to environment creation in games. In this one we're diving into some responsibilities in your daily work with a focus on open world games for this one.

1. Set dressing/Level art

Bigger and more dense, living, worlds come people who actually need to build that and tell the smaller stories that spark the interest of the players. More specifically know as level artists, they are responsible for dressing up the locations and level design blockouts within our worlds.

They achieve this by using assets that are created and are given to you over time when they come back from outsource or you are responsible for creating these levels/locations yourself.

This is also where your love for environmental storytelling comes in, using the assets available to really make an interesting environment.



Lilian Chow

2. Landscape sculpting

As an environment artist you might be responsible for chunks of terrain, both within the locations your working on and between them in the open world.

A big thing you are keeping in mind for this is definitely how the flow or wind and water can shape terrain, same goes for the movement of tectonic plates and how natural elements like rivers and waterfalls are formed, all crucial knowledge in your shaping of a world.

Another aspect is to balance the technical aspects (such as occlusion) and player direction (with impressive vista's and player leading in mind).

Stephane Guadelli

3. Prop/asset modeling

Depending on the studio, this might be a part of the same banner (see my official title of Environment/Prop artist) or a separate role. Even as a full time environment artist you might find yourself helping out on the prop creation side.

One of these examples can be working with outsource to get them blockouts that they can build upon with the help of briefs and concept art too.

The creation process will happen either through the high-to-lowpoly pipeline (less frequent for open world games, and reserved for important assets, such as guns for example) or making assets with textures and materials already available.

Hannah Watts

4. Foliage and organic asset creation

A world also needs to be populated with a lot of natural assets as well (in a lot of cases).

As a specialized foliage/ vegetation artist that's where you come in working in programs like Speedtree to create the tree's that are seen all over the place. These artists can also be responsible for other organic assets such as rocks for example too. When working on these you will have to be really strict in the amount of poly's and alpha overdraw because these assets will be scattered all over the place in huge quantities.

So you will be constantly testing the foliage you make for performance.

Adrien Paguet-Brunella

Disciplines within Environment art

Introduction

When it comes to environment art, there is a lot of confusion among people starting their environment art journey because they think that Environment art is a singular thing, but it's actually split up into multiple different disciplines, so today we will dive into some of the individual disciplines involved. Keep in mind that different studios might have different setups, and can change going into the future with changing technology as well.

1. Prop art

Prop artist build the content and assets for populating your environments. They are responsible for creating the smaller props all the way to big buildings and hero props.

This discipline is more isolated But that doesn't make it less crucial for environments in total. This can also been seen somewhat as the introduction to environment art, because everyone needs to get through the full High to Lowpoly pipeline.

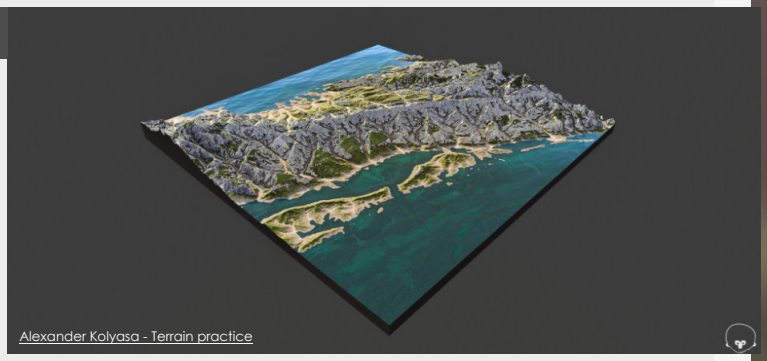
The more standard programs that are used in the pipeline are Maya, Max, or others for modeling, Substance Designer and Painter for baking/ texturing and a bunch of other programs for all various needs.

Marta "Carabea" Niemczynska - Biohacker - Biochair prop

2. Terrain art

Responsible for building the foundations of the world taking care of the underlying terrain. Other responsibilities include planning where mountain ridges, valley, lakes, rivers and more are going to be. They focus mostly on sculpting terrain, creating natural formations that look realistic, using real world examples as reference to create an immersive as possible/level for the players to get immersed in.

Depending on the stage of the project they will use World Machine, Terragen, Substance Designer or others for creating a good base that they then can iterate on inside of the Game Engine.



Alexander Kolyasa - Terrain practice

3. Level/world art

These are the people responsible for the setdressing within the world and they take the props that the prop artists make and in most cases combine them with blockouts that Level designers have prepared.

These are also the people that are responsible for introducing the storytelling that has been setup by Narrative designers and use this to tell stories while keeping in mind the gameplay space that the Level Designers have build.

Some of the programs that they use are Unreal Engine, Unity or any other in-house engine and embedded tools.

Billy Mathuris - Far Cry 5 F.A.N.G. Center

4. Foliage/Natural art

The people in charge of all the natural assets such as trees, bushes, flowers ,rocks, etc...

It's different because of the technical restrictions to other props because they are used way more in a level and are heavier to render. This means that the artist that removes 3 poly's from a grass mesh can have a large impact of thousands of poly's just because it's used so much. A big factor when making foliage is Quad overdraw, where multiple layers of masked alpha are overlapping.

Programs used are Zbrush, Speedtree, Houdini and other programs that lend to a non-destructive workflow for more iteration, variation and speed.

Jobyé-Kyle Karmaker - Middle-Earth: Shadow of War - Vegetation | Asset Shots

seniority levels in the games industry

Introduction

In the industry there are many different disciplines and levels of seniority, so in this one we're going to get into them and what they actually mean for your career and responsibilities within the industry.

1. Differences in Seniority

Junior

As a junior environment artists you will mostly be focusing on getting to grips with the tools and pipelines of the industry, learning from the senior members of the team.

Your responsibilities will be fairly minimal and will be supported and supervised by a mentor on the same project.

You will also slowly but surely be challenged with more complex and bigger tasks the more experience and time you build up working with the given tools. This will also prepare you for the next step.

Intermediate

When it's time to move up the ladder you will see that you get more ownership over the props or levels you will be working on. This also means that you will have to become more proactive and trusting in your skillset and be ready for a bit more ownership.

You will also be moving more into the role of supporting the senior a bit more instead of relying on their help to do things.

Later on you might get your first experiences with giving talks, mentoring people occasionally and also even managing your first person if you choose to go that way if you want to go that route.

Senior

Full ownership over specific sections of the game including props and also more control over the vision over the props and levels.

At this point you are also becoming more and more of an example and role model for other artists, maybe even managing some of them full time.

You also become more involved in the pipelines and workflows and will also have responsibilities in putting these together with other seniors of different teams.

Giving talks to the art team, organizing meeting and leading them will also become more and more part of your responsibilities (if you choose to do so in most companies)

Lead/Principal

So this is usually where things diverge in most studios, focusing on either art (Principal/ Expert) or management (Lead).

A **principal** will dedicate their time define workflows, remaining on the cutting edge with their art and really setting the bar when it comes to the art.

A **lead** on the other hand will dedicate most of their time to organizing the team, thinking about and planning the future of each person of the team, communicating with all the seniors, and maybe occasionally working on art themself.

Relocation Assistance

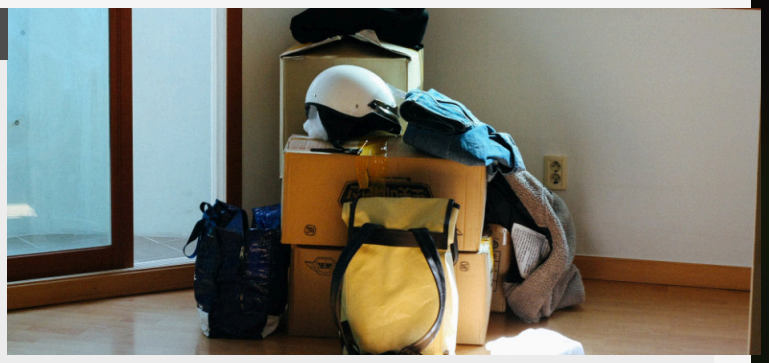
Introduction

It's super important if you are moving abroad and usually companies don't tend to highlight if they even do it. But what is it and how do you make sure you get the most out of it? Let's discuss it!

1. What is it?

Relocation assistance is a way for companies to help you make the move to another country way easier and make it more attractive for people to do so as well. Usually this means that the company will give you money to cover certain costs that are normally involved with the moving process.

However, it can also include way more than just that, so let's explore that a bit.



2. What can it include?

Monetary compensation: the company offer money to cover the full or partial costs of travel/relocation for yourself and your furniture.
Company housing/apartment: some companies will also offer temporary housing or even permanent housing in cases, sometimes even partly paid for a limited time. There are also companies who will do the house hunting for you and offer you the best selection of what's currently on offer.
Visa Procedure costs: If you are moving from abroad and require this then this can also apply to you and the company might/has to offer this when you are joining them.
Storage costs: If you are moving storage the company might even pay for storing it for a while as well.

3. How and when to bring it up

So now that you know this exists, how do you bring this up?

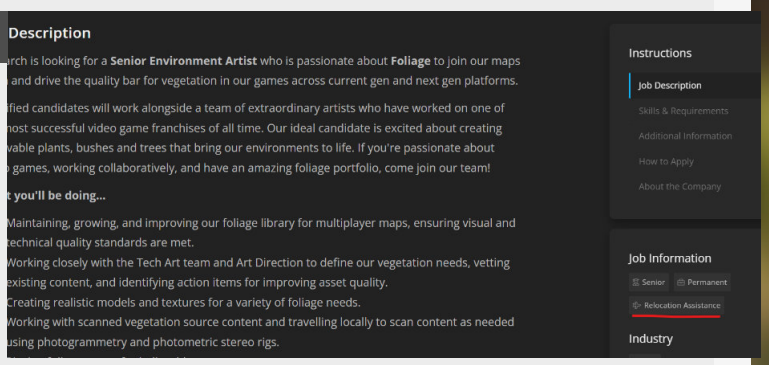
Normally during the interview there will be a stage where you will discuss your personal needs and contract. Just ask them if they offer relocation assistance and then discuss what you would need to be able to make the move yourself.

Usually the company either has a base line already in mind for this, so it doesn't hurt to ask!

4. Checking applications

So it's important to also check in the application when you are applying for the company that they indeed offer this, you don't want to suddenly realize that they didn't offer anything like that at all.

So if this is an important topic for you, and you don't see anything in the application, make sure to ask that company as soon as possible, so you know if they do offer it or not.



Seniority levels and Responsibilities

Introduction

Giving yourself a goal to work towards is part of building your own career, so having a title to work towards can give you great job satisfaction. However, with the titles always in flux and companies adding different ones, where do the differences between these levels lie?

1. Overview

Seniority levels seem to be set in stone in the games industry, however, every company can define them in a different manner since they perform different task or have different responsibilities.

It's also for this reason that I tried staying away from all the very specific responsibilities and pretty generic.

Also transitioning between seniority levels will also mean you will be fulfilling a lot of the of the responsibilities already for the next seniority and wonder why you are not getting that sweet promotion.

Never let that stop you from discussing a potential promotion with your current manager though.

2. Junior/Graduate

As a Junior or Graduate artist you will just be getting to grips with how the industry works.

So this means that you will be getting a lot of support from the Intermediate or Seniors in your team to build up all those skills. You will be exposed to a lot of new skills and be growing a lot in this period once you're getting into the industry.

This also means that usually your won't have important responsibilities and will be working to support the work of Intermediates of Seniors as well, so that they can then in turn support the art director above them.

3. Intermediate/Mid

An Intermediate/Mid or "just" Environment artist you have a solid grip of the workflow used, can work pretty independently with occasional support from seniors or leads above them.

This also means that you get more ownership as your skill set grows and you get to work on bigger sections of the game or more important props.

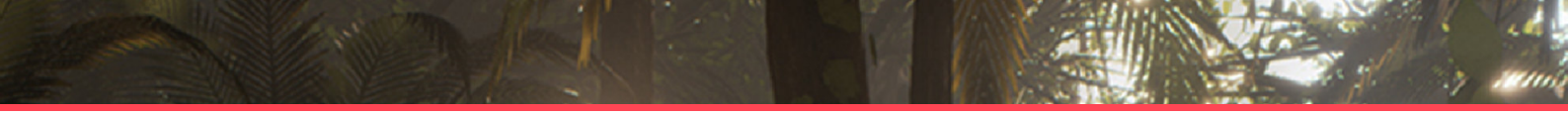
As a Mid environment artist might also start helping out with managing an on-boarding other people getting into the industry, like new Juniors or Graduates joining the company.

4. Senior

As a Senior Environment Artist you have a strong grip on all things environment art. You will also be responsible for setting the bar for the rest of the team in close collaboration with the lead artist, and you are expected to do so fully independently.

This means that you are going to be responsible for bigger sections of the game. As well as being responsible for mentoring/managing multiply people and taking care of their career progression as well.

Important note: *Managing is becoming less mandatory for people who are not great managers so managing or mentoring people might be optional for companies.*



Thanks to all our Patrons for all the support

Emmm Santiago Medrano
Antoine Dej
Gianni Carpinelli
Alvaro Miranda Tarjei Skårsmoen
Alex Blair
XBA
Sara myclay
Jason Coleman
Pieter Cornelius
A A
Alex Bellotti
Lukas Kuosa
Brandon Miletta
Steven Primrose
Andreas Lohse
Ben Black
835013563
Jonadab Oomon Cosche
aman mundy
Simttex
Jack Chatterton
shubham tripathi
edik sidorov
天航 王
Alik9999
Keith Ebanks
Leaf Baimbridge
Kyriakos Gerostathis
Ben Palmer
Emilie Lefevre
Tacker .
Cameron Fraser
NotSoGrimReeper
Rajeev Nattam
Eric
im sus
Yushinka Chen
Paul Browne
André Castman
Jorge Antão
Lee Miller
Paolo Ringegni
Fabiana Inu
ryanis3d
Gabriella Tångfelt
Alexander Nguyen
mxxrpheus
Dan Lavigne
DZoned
Riley Gravatt
Nunya
Guillem Pavon
Kostas CH
Strannik
Dmitrij Kugut
Mahaweilo
Lewis Seed
Chris Suki
ephbea
张 群良
Ryan Lopez
ZIMYENG
Karalina Shaibak
James Ward
Le Albi
Paul Elder
Daniel de la Torre
Stan Brown
Soph
Milan Van Damme
sion vialva
Lily L
Anes Sabanovic
Hiram Rodriguez
Nicolas Fernando Mendoza
Tkail
Samir Riecke
Pamela Dominguez Cairo
Goodbrand
Tristan Voulelis
Nicolas Jaud
Erick Lim
Lloyd James
David Bullock
ejcook
Brendan Amiel
Leon K
Marco
Dani C
Miha Maruško
POPCheck
Tjeu Pubben
Georg Marta Niemczynska
Danielle V
Roger Homedes Rius
Ashleigh Dale
Paul McGovern
Araki_Sensei
Pere Albacar
Petar Culafic
Влад Шевченко
Sean Love
Justin Virtualdreamz
Yolanda Cruz Girona
Rin_Chan308
simone mirandola
shanee g
Moïse Dos Santos
Edoardo Tiziani
Yoshito Wakatsuki
Nikkita Racquel
拓也 平田
Kevin Buck
Daniel Siemieniaco
Yuchen
Netraraj Pun
QiaoLiang
Ryan Levac
ALLISON LI
lux xun
Helena Hansen
Patreon Man
Mathew McLaughlin
gaudichon
순정 이
Kyran Taylor
Paula Micheletti Tavares
Krzysztof Knysak
Kain AC
Jamie
Nicolas Rocque
Tim Gautier
matt
Mohamed Elbouhy
輝 和田
Lea Kronenberger
Yuriy3783
Tyler La Belle
Ryan Shearer
Eanna O'Brien
AF
Kathleen Harwood
Salvador Sánchez
Manjot Singh
Omar Souissi
Hicham Ouchan
Justin Raymond Ramos
Freedie Chan
Bartosz Buszman
Freyja Houghton
Gonzalo De Francisco Hernando
Adam Plechatý
Scott Onions
fredqin
Alex Cheung
Leon Hatsume
Jason Yi Kai Cheung
Robert Shaer
Armando Panciotto
Andrew King
J1SH
Alberto Moreno Sagarasuu
jayxle
Alex Blaney
Alexander Örn Magnússon
Qi Han
Samuel Thibault
Infinity Vector
Nathan Rose
Rasmus B Kristensen
Alexis Washington
Rosario
Chris Hunter
Miszla
Marcello Pellegrino
Johan Gunneröd
Johan Ernstsson
Ahmet Enes Yurdakul
Chloe Tibets
Joseph H Edwards
Thomas Dayan
Anh Quân
Nguyễn Phạm
BrunoNM
MastRcheese
Joel Ferris
SomNados
Matthew Small
Doodled_lynx
Lucas Rodrigues
Matthew Cane
jamie cooper
Retroxiity
Jose Eduardo Remiro Bonatelli
Giovanni Ferrari
Al MU
Daniel Hinckley
Denys Davydov
Fat Cows
Alessandro pacini
Jake Brown
Zoran Petrovic
cassie baer
Jack Ibbotson
Conor Haining
Hannah reslan
Gav Harvey
Ioana Oprisan
Soviet Matthew
Ashley Martin
Sergii Zlobin
Pedro Yapor
Sam
Giovanna Neiva Cardoso dos Santos
Antonio Bonet
E Arthur
Thomas Skinner
Scot Daniel Burns
tendercrisp
alan field
Cameron Bragg
James Williams
Abid Sultan
Nikita
Cristian Bercea
Sanjay Kumar
Joseth Segura
TheArtofGee
Cordell Cohoon
Giacomo Bonanno
Edward Osei Amoah
Jim Düwel
Rhi Thomas
jan
HEEHO
wyrine
Dawid Wrona
Leo Ier
Adam Anastasia
Thiago Carneiro Matthew Wilshaw
Laird Wt
Sr Saito
Mohsen Ghafari
matt
Leonardo Menezes
Tom Calcoen
Tropical Moisture
DenA
Damien Prevost
William Andrews
BananaBread
Leo Bushkin
Michael Fairchild
Red Charton
TJ Post
Markus Esch
MrDrey
Derek Schumacher
Brocktoon
Michael Daw
Luis Mesquita
Lack of oesophagus Male
Aniket Aggarwal
Maria Raducanu
Kang Jia
Will
Ze Olive
Hamish Ames
Isaac Johnston
Raj Joshi
Chelsea Bradford
Ahmed Matar
Sanekum
Fabiano Oliveira
Scott Rafferty
ofek abo
Alan
Tobias Rintoul
Jake Parrott
Walnussrücken
Saif Alhmeyrat
Fateh Shehryar
Aria Baskakova
Jono Sutton
Erfan Beiraghdar
Thomas Ray
DK
AdaBee
Kelly Seekings
Mora Kaitu
Andrew S Lytle
Sady Fofana
KérioX Jenkins
Andrew Whiteman
Paula Maria Röhmer
Morrissey Alexander
M Orrostieta
Sasho Lazeski
Shivam Annapure
Shawn Uttley
Robert Allen
GuiH