Art Bible

Mood Board

Original Art

This is the art that was given by the client that should be integrated in some capacity. Because this is already used in their therapy session with the children.

Setting

The game takes place in an alternative present where all are replaced by cats. To create a playful and safe environment.

The building style is inspired by buildings in the Mediterranean/ alpine regions of Europa. Because it is an ecstatically pleasing style. Adding some modern elements like solar panels and electricity wires, etz.

Rules

Shapes

Gematrical Shapes

We want to create a **save environments** so the shapes should represent this feeling. Make use of **rounded shapes** for the safe feeling, **rectangular** for stability and strength or a combination of both.

Distorting Shapes

Don’t use straight edges in your shapes, instead make **use of crooked** and **distorted** effect.

Scale

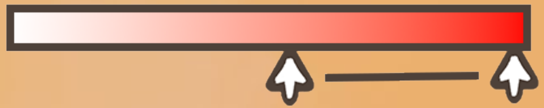
The proportions of the characters/ objects should also be a bit stylized in the form that important parts get bigger in comparison to less important parts: like making the head of the character almost as big as the body.

The scale of the objects is also distorted, the characters/hero props would be larger then normal compared to the buildings, trees, bushes, etz. The Adults should also be larger then the player character.

Environment Colors

Overal

Saturation: Overall, a High saturation value.



Value: Used as contrast between building accessories and the wall.

Background

- Analogue color pallet
- Red range for buildings
- Yellow for plants

Interactable

- Between blue and green for the network buildings.

Character Colors

Child

The color used for the play character is changeable in the **customisation** at the beginning. Chosen out of a range of **skin colors/ hair colors**, so they can make the character represent themselfs.

Skin

Hair

Cats

The colour of the cat is representative of its **breed** in the real world. This is created by a proses of masking it with a pattern (with colors get replaced) see later under workflow.

So the value and saturation has no fixed range.

Clothing

This part has to do with giving the player the ability of self expression. Very important because we want them to create a connection between the cats and their contacts in the real world and gives the player the abilities of the character looking like them.

Parts

The clothing consist of **3 pieces:**

- **Accessory/hat**
- **Shirt or top**
- **A pair of pants/ skirt**

Should be compatible with all the others.

Shape

The most important factor of the clothing elements is **recognisability**. The shape of the outfit needs to be readable from a further angle, so **important elements** of the outfit are allowed to be **scaled up,** but be carful **not to overdo** it. It might look **weird** in **close up.**

Color

The colours used should be **monochromatic**. So the details can be highlighted with **saturation and value changes**. The **base** of the outfit should have a **high saturation and value.**

Camera

Main angle

Most of the game is played with a **top down** camera that is under a little bit **under an angle** to give the more view to see further in the distance and thus move easier. Because this is the main angle it is imported that everything looks good from this angle.

Talk angle

There is one other angle that the camera takes, for the **conversations** with the other individual in the world. In this angle it is important that elements like the **character** and there **clothing** look good because it **provides a close up**.

UI

Shapes

Shapes used in the UI use like the base shapes **rounded shapes** like circles and rounded rectangles. But unlike the base shapes we **don't want wonk** in these shapes.

They need to **look drawn**, so they have a bit of a **fussy effect** on the **edges**.

Use

The UI must be as **language neutral** as possible, but if **not able** to do so **text can be used.** In general, the game should have **as little UI as possible** to make it easier for children, particularly with the HUD.

Colors

The UI Consists of **2 colors**. The base **brown color #4e4139** that is also used for the lines in the world. And **white on top if needed**.

Texture

Character

The characters is textured with **solid colors and lines + optional normal map**.

Texture map: a mask map:

- Giving **4 color** channels and **1 line** in engine.
- More info in Character Shader.

Outfit

One Albedo map:

- The colors used should be **analogues in the red range.**
- Make use of **gradients**
- Add **lines** on edges of colors and creases.

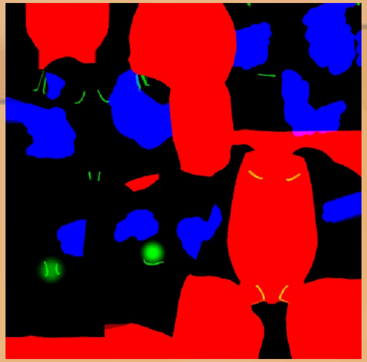
One Optional **Normal Map**

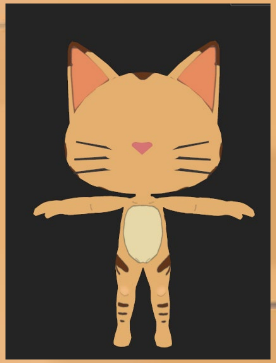
Environment

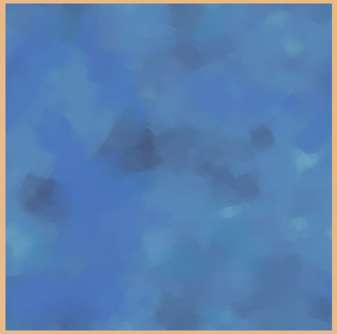
For the **Albedo map** we want to created a **water painted** looking effect for the environment. There are 2 ways to create this:

- **Provided generator** (See Environment texturing)
- **Hand painted highlights**

Normal map.





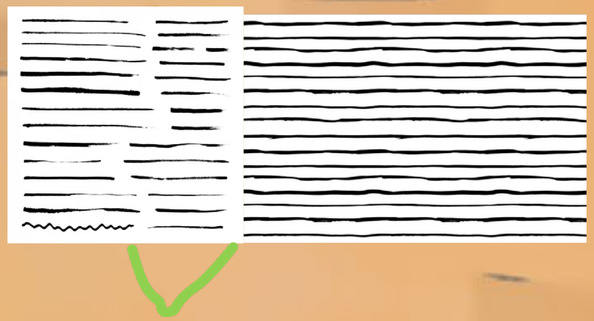


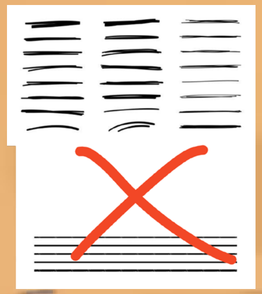
Lines

Construction

The lines are an imported aspect of the style we are going for. The lines should look a bit sketched but not in the way that it is messy ,more in the form of **semi strait lines**. This should also stay away from being to bold, they should be **thin but still visible** from the cameras.

The lines have a specific **color** show below, and get a transparency of 80 in most cases (UI exception).





Use

The lines are used in both a contour as for edge highlight sense.

- In the case of the **buildings** all the line creation is done by the shader

- With the **characters** only the contour is done by shader, the edge highlight are painted in see texture.

Pipeline

Character Texturing

Base

If said model should have a **different lighting effect**, then the low poly can **provide a normal map (baking is allowed)**.

The body texturing uses a **mask system**. So in substance you **don't paint with colors** but **4 user channels** (representing **4 colors in engine** colors). **Low values** for these channels **are** also **allowed** to create a **blending effect**. The **higher priority channels override** the lower.

Priority

- **Non:** if all channels are 0 the **base color** is shown.
- **Red – blue**: highlight colors (channels for the colors **different** from the **main color**).
- **alpha:** stores the lines.

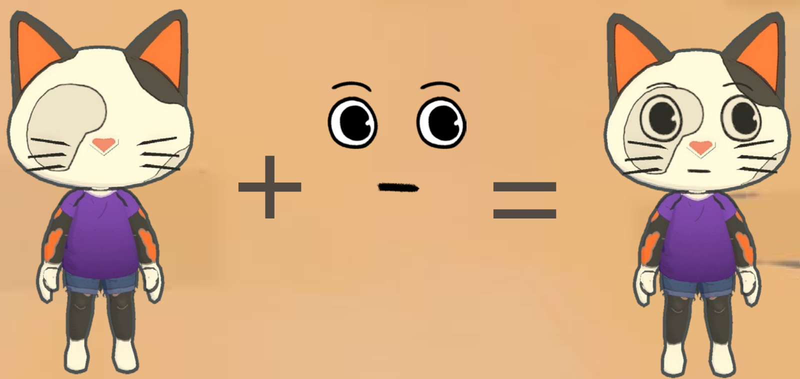
Face

In the **engine** the **face is placed** on the **body separately**. And from the different face elements texture **sprite sheets** need to be made. (See later under character shader)

The drawn faces then use a **custom LUT** to convert them to the **correct color system** for the in-engine shader.

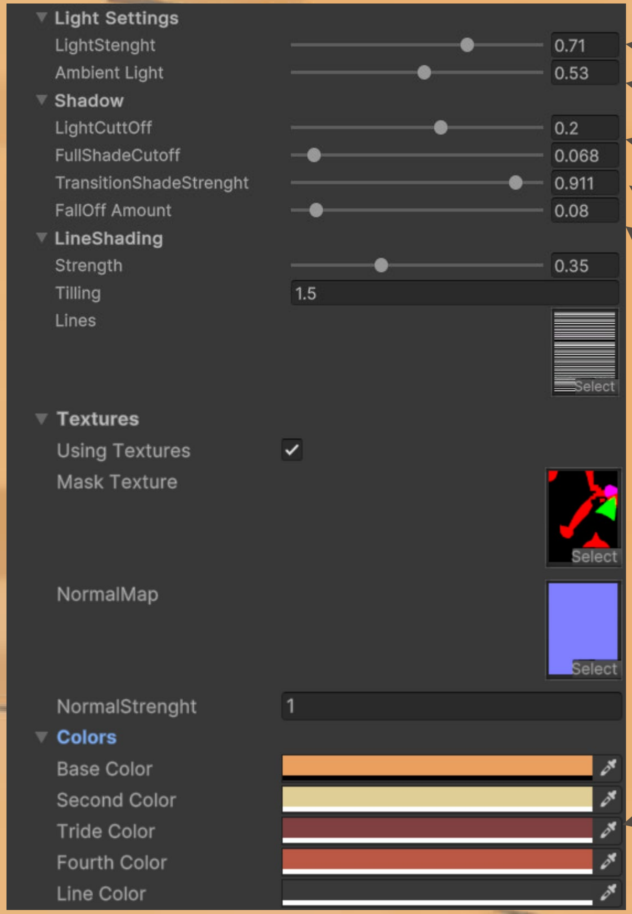
- **Size : 50px**
- **Kyle Charcoal Pencil form adobe**

- Outline (0,0,0,1)
- Fill (1,1,1,1)
- Details(1,0,0,1)
- Nothing (0,0,0,0)



Character Shader

The **character body** uses **2 shaders**: one for the face **SH_Face** and another for the body **SH_Body**. They have most in common except that the **face** has some **more functionality** (descripted on the next slide).



The strength of the light

How dark are the shadows: 0 black -> 1 no shadow

The **cutoff** point of the (1^{st}) **full shadow and** (2^{nd}) **transition shadow** (lines) **respectively**

Darkness of the **transition shadow** is.

How **sharp** is the **shadow layer transitions** the lower the sharper.

The **dankness**, **tilling** and **texture** for the **shading with lines**.

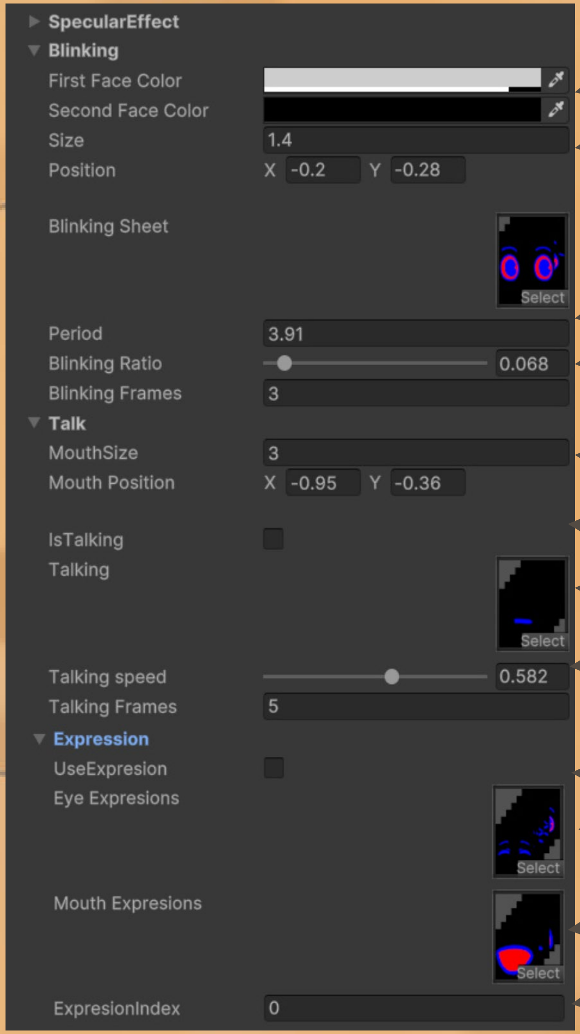
The **mask texture** showing **where** what **color** should be used

Normal map

Colors corresponding to **the channels** of the mask map, see **texturing part** above.

Character Shader

The **SH_Face specific elements.**



The **face specific colors**: first **fill** color, second **highlights** like blush.

The **size** and **position** of the **eyes**.

Texture 2D array of the **blinking eyes**.

How **fast** the blinking **repeats.**

How **long** the **blink** is compared to the **hole period.**

Number of frames in the texture.

The **size** and **position** of the **mouth** on the head.

Activation talking.

Texture 2D array of the **mouth** talking

The speed of the **mouth movement.**

Amount of **frames** in the array.

Activation Facial expressions.

Texture 2D array of the **eye expressions**

Texture 2D array of the **mouth expressions.**

The current **selected expression index.**

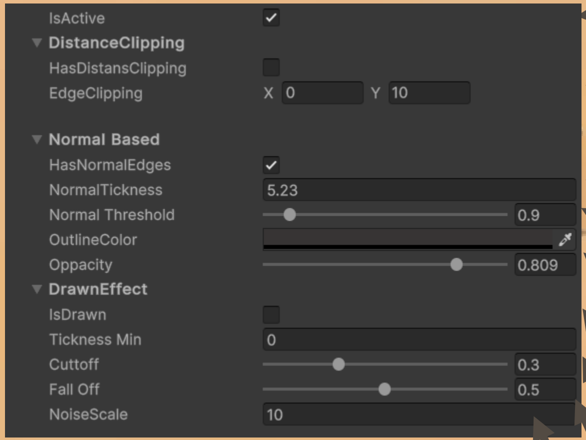
Line Shader

There are 2 outline shaders:

- Screen space used for environmental element, advantages of having outline on every shape
- Object based used in characters and collectable items, advantages of this is only contour, less Noice for more dense shape language.

Screen Space

Only one material in hole scene.



Activating effect

Near and far distance with determine the thickness

The thickness of the edge

The tolerance of what is a edge.

The color and opacity

Has line thickness variation

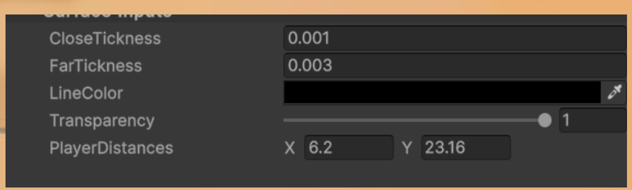
The mine thickness

The cutoff and transition of the noise used

Scale of the noise

Object based

This shaders material should be applied on the mesh render 2nd material.



The thickness of the lines far and close

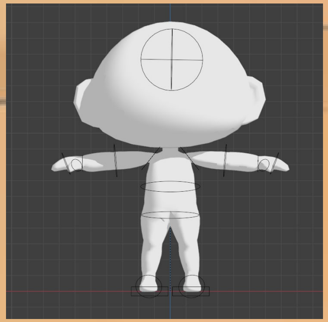
Line color and transparency

Near and far distances with determine the thickness.

Outfit Creation

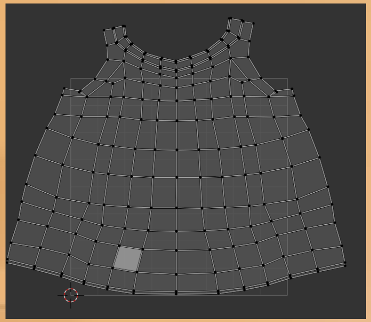
Modelling

Design the outfit for the **child base model**. **After** designing it and unwrapping, it should be **modified** a bit for the **cat body**. Normally not much work because their bodies are similar.



Unwrapping

During the unwrap make a **2nd uv** channel. This channel will be used to place a decal on the front. It is important that the **0-1 space reprecent** the space where the **decal** is wanted.

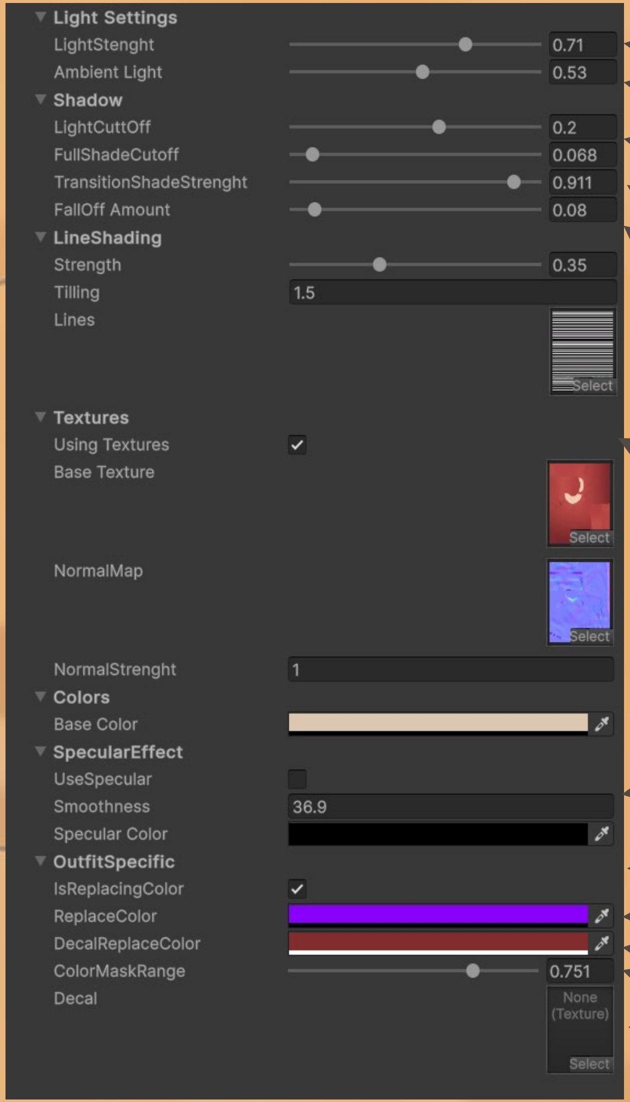


Texturing

The texturing procedure is **fairly simple**. Everything in done on the **base color and normal map**. While coloring it is important that you **only use red tints** (**exception** for **lines** and **wite**). The reason for this is makes **color shifting** possible in shader (for this reason **check** if with the **hsl filter** in substance painter, it it still looks good In **other colors**).

Outfit shader

The outfits use the **SH_Outfit** descripted here.



The strength of the light

How dark are the shadows: 0 black -> 1 no shadow

The **cutoff** point of the (1^{st}) **full shadow and** (2^{nd}) **transition shadow** (lines) **respectively**

Darkness of the **transition shadow** is.

How **sharp** is the **shadow layer transitions** the lower the sharper.

The **dankness**, **tilling** and **texture** for the **shading with lines**.

Use textures in case not active it uses the base color.

The **outfit specific textures**: base map and normal map with strength.

The **color** in case of **no texture**

Use of spacular higlite, its color and size.

Use color replacement

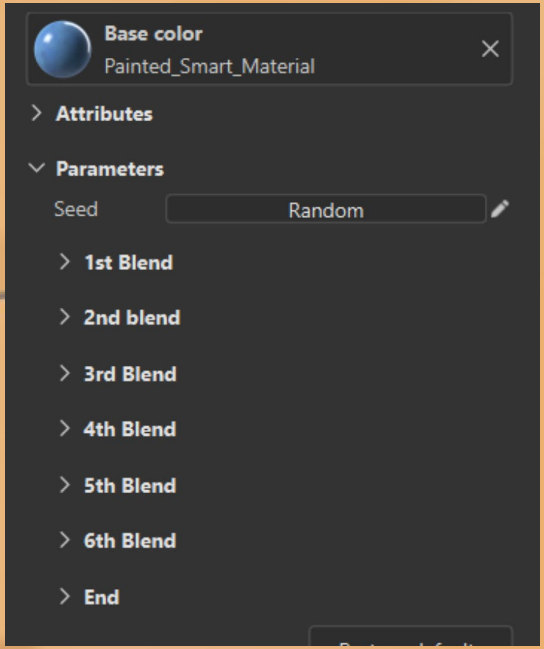
The **replacing color** of the outfit base

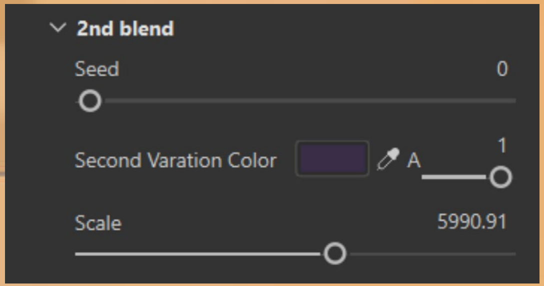
The **decal replacing color** where it is **wite.**

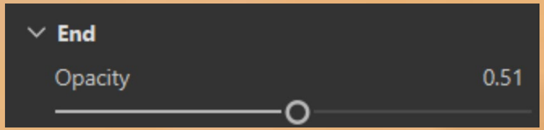
The sharpness of the decal masking

The **decal** uses the **2^{nd} UV** space.

Environment Texturing







In substance the only two outputs we need are the **base and the normal**. For the base there is a **generator provided** to help with the creation of the painted effect explained in this chapter.

Elements

The generator consists **of 7 collections** which correspond to a different noise layer within the texture. The **lower** in the order the **more** it is **on top**.

Each blend consists of a **seed** (noise seed), a **color** that the noise has, an **alpha** how visible it is and a **scale** of the noise .

Rule of thumb

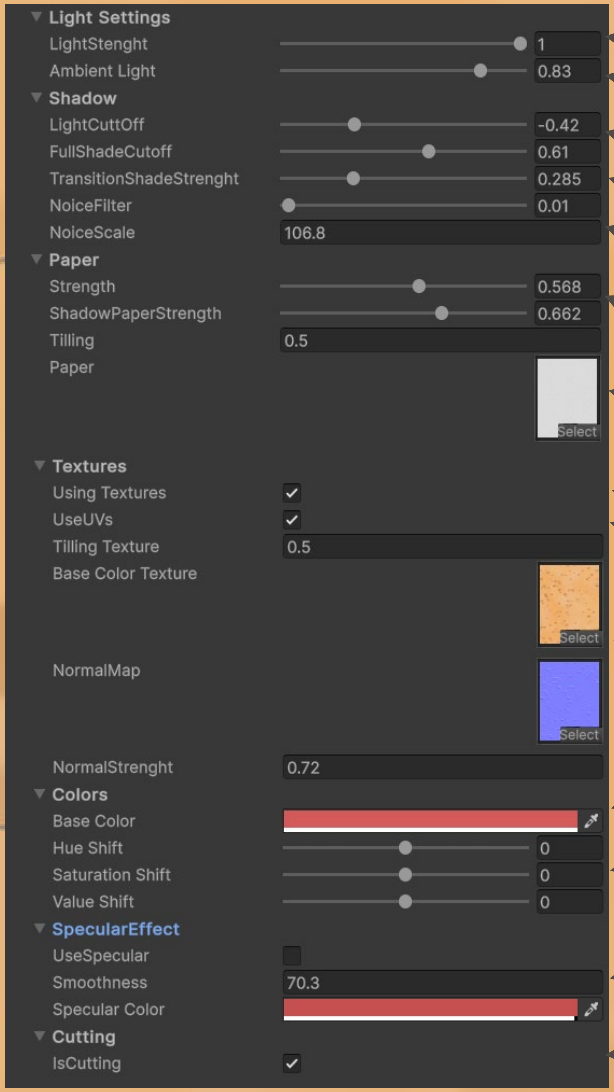
In the **end section** there is **a opacity** for the **fine tuning** of the blend:

- The **higher** the opacity the **more** of the **noises** the
- **Lower** the more it becomes the **base** from blend 1
- Make **full during** the creation prosses **only** mend **for the end**.

Not all the blends are needed they can be **dropped** by setting the **alpha to 0**. While **creating drop them all** to 0 so you can **build up by adding noises**.

Environment Shader

Each environment material makes use of the **SH_Envirenment** descripted here.



The strength of the light

How dark are the shadows: 0 black -> 1 no shadow

The **cutoff** point of the (1^{st}) **full shadow and** (2^{nd}) **transition shadow respecitifly**

Darkness of the **transition shadow** is.

The **amount** of **edge noice and scale** of it.

The **base strength** of the **paper** and how much **stronger** in the **shadows.**

Tilling and the texture itself

If we use a **texture or base color**

If it uses **uv's or a topdown world projection** (ex. Roads and pavement).

The tiling and the **base and normal maps** with its strength.

The color in case of no texture use.

Some **shifting** of the color **values** for the creation of variation. (**subtle** use, **not for texture adjustment**)

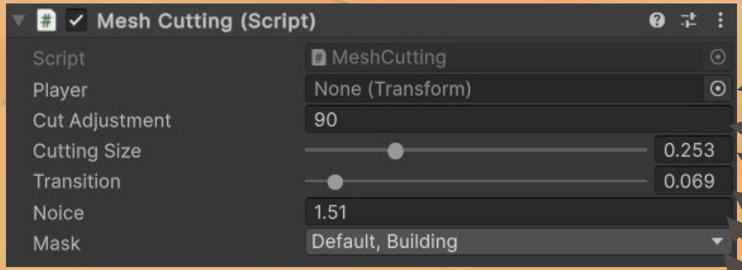
Adding a **specular highlight** and adjust its size and color.

Activates the cutting of the mesh when **player is behind** it if wanted.

Global Settings

For these scripts to give visual **feedback** to dev's you're **draw gismos** must be on. The script **control** these parameters for **all the materials** in the scene.

Mesh Cutting



The mesh cutting needs a reference to the player position.

How much shifting is needed on the y axis (in pixels) from the player pivit.

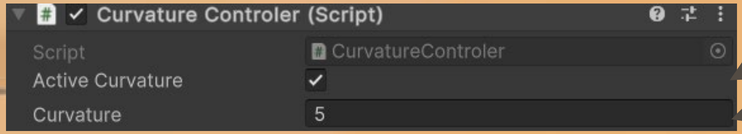
Size of the cut

The range of the noise edge

The scale of the noise

The layer mask of the elements the raycast that determines if a building is Infront can hit.

Curving world



Activates the curvature effect, for ease of creating the environment.

How strong the curvature is.