



# Animation

# Today's Agenda



We're going to be going to be continuing with Chapter 17: Animation, Chapter 18:

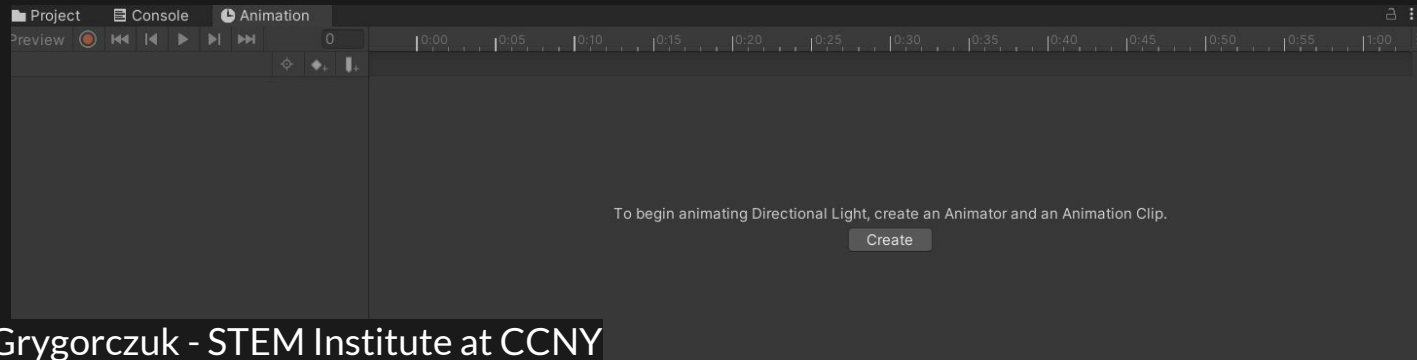
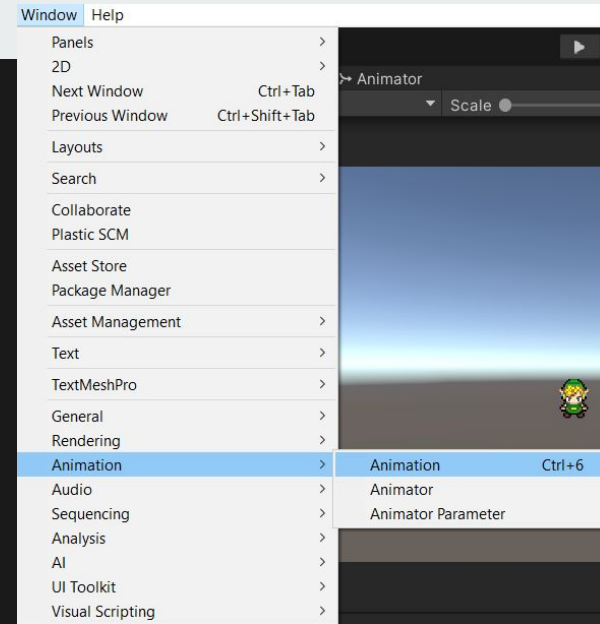
## Animators

- How to animate any game object
- How to use Sprite Sheets and Rigging to animate characters
- How to use The Animator to switch between animations on the fly

# Animation

To Start animating we will open up the Animation View which can be found Windows -> Animation -> Animation

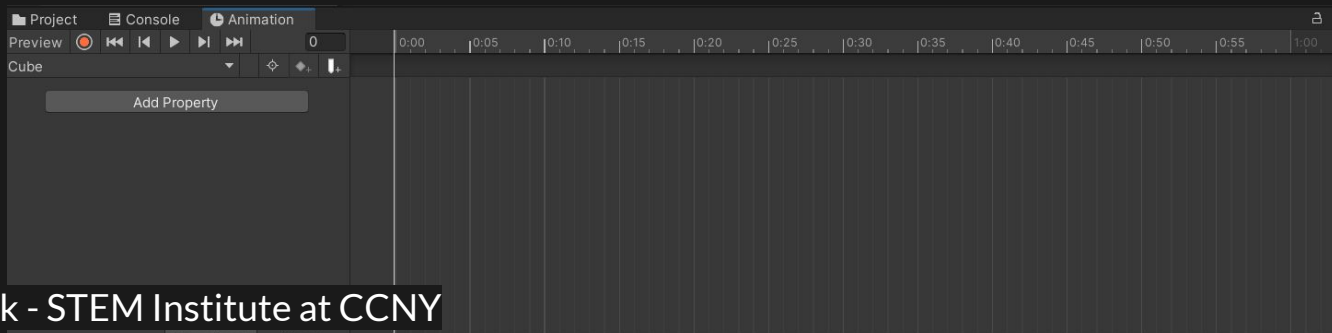
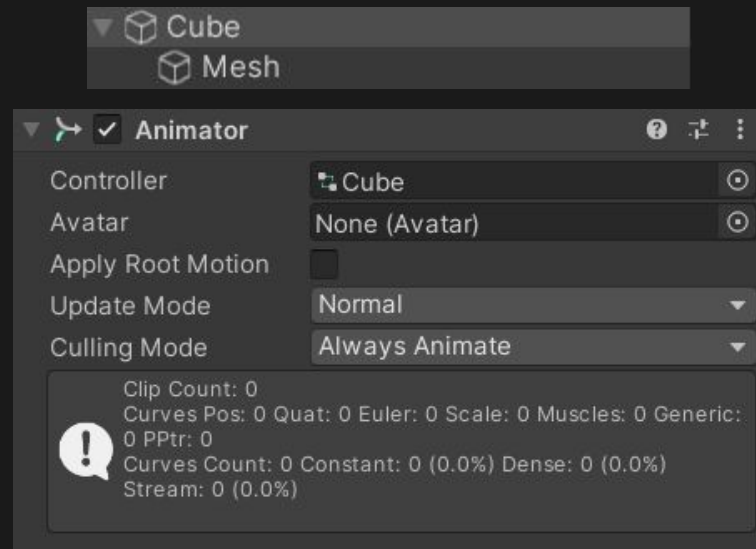
I like to put this window on the same area as project and console but it's up to you where you place it.



# Animator Component

When you select a game object such as Cube you want to have a 3D Mesh or 2D model as a child so that editing the transform won't affect the control of the game object.

Once you've selected the parent you can click create this will create two assets. A controller and an animation clip.

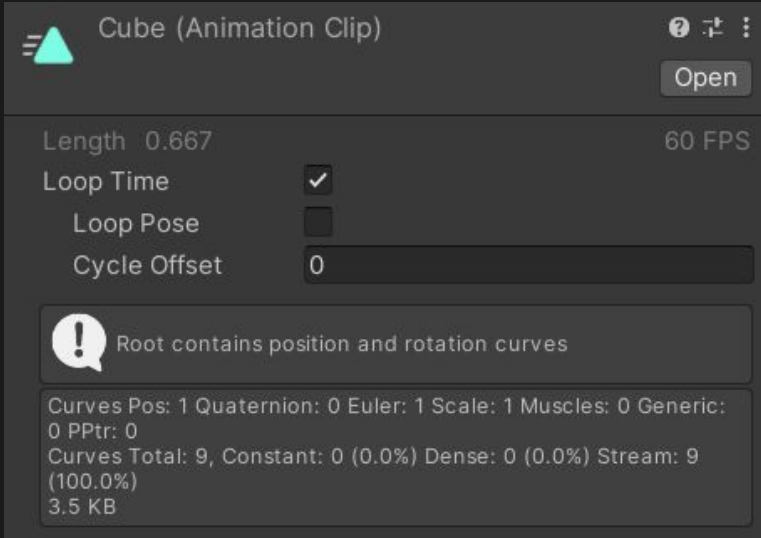
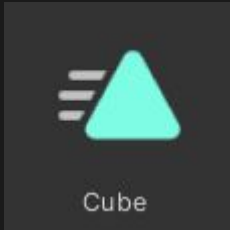
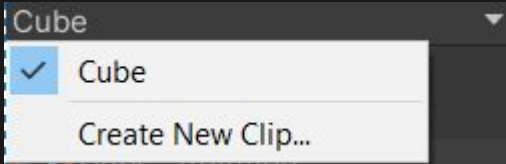


# Animation Assets



When you create your assets you will have two things, the three squares is the controller, it holds all of the animations and later we'll see how it allows us to switch between animations.

The Triangle is the animation, if you click on it you will see that you can control so of the data of it such as if it loops or if it starts few seconds into the animation.

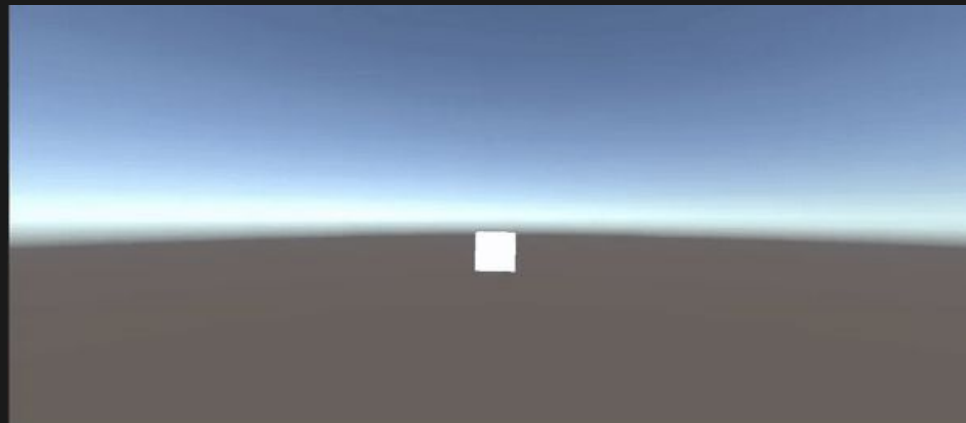


# Dope Sheet



The Section below is called a Dope Sheet and it lets you make nodes at different times.

You can only modify the nodes while you are in record mode, which will make your time red. You can affect anything in here, as long as it's the game object or it's children.



Preview 37

Cube

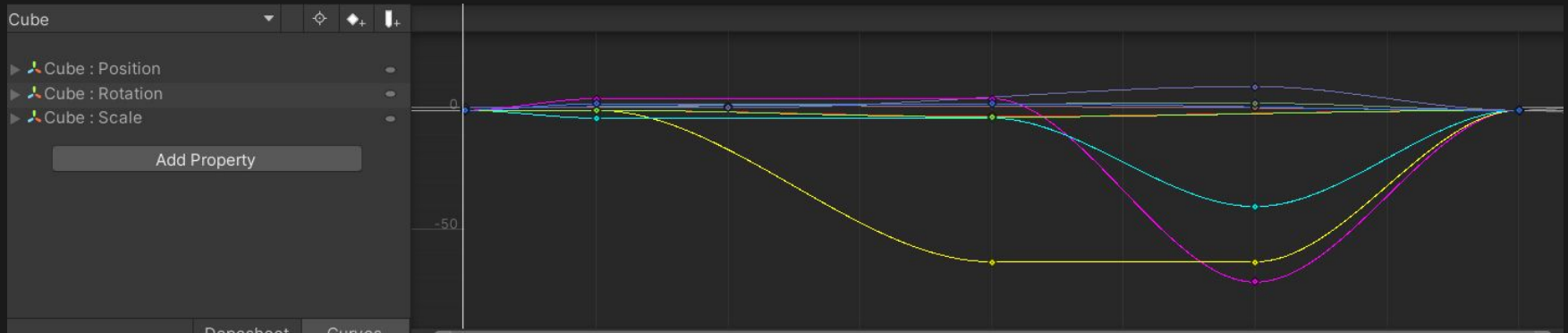
- ▶ Cube : Position
- ▶ Cube : Rotation
- ▶ Cube : Scale

	0:00	0:05	0:10	0:15	0:20	0:25	0:30	0:35	0:40	0:45	0:50	0:55	1:00

# Curves



When you are controlling something's transitions, such as transforms Unity calculator the gradual steps between where you start and where you end. The Curves Menu allows you to see and modify that transition using the curves. You can make something happened instantly, or make it very slow and speed up at the end or vice versa.

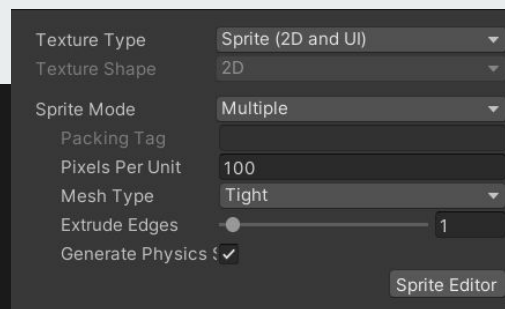


# Sprite Sheet



Sprite Sheets are very useful tool for animation. They hold all of the frames of one action and you can just switch between them.

Make sure your Texture Type is set to Sprite (2D and UI), Sprite Mode is set to Multiple and go into Sprite Editor to slice up the sheet automatically or by cell.



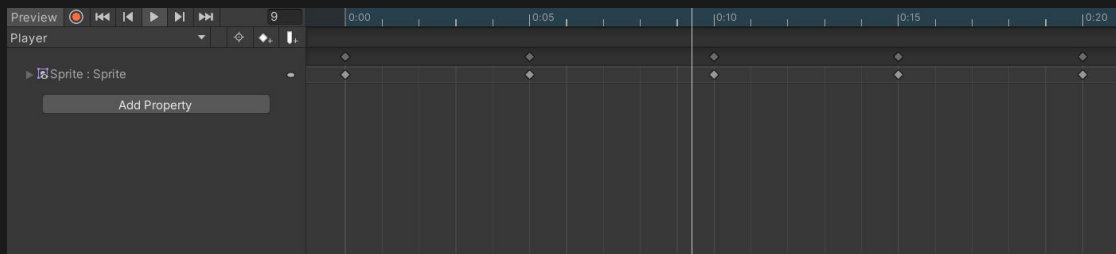
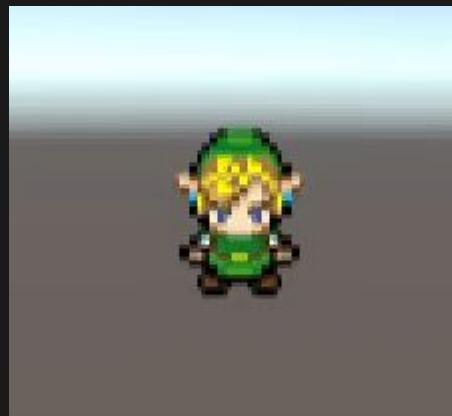


# Sprite Animation



To Create a Sprite based animation you just replace the sprite render at x time intervals.

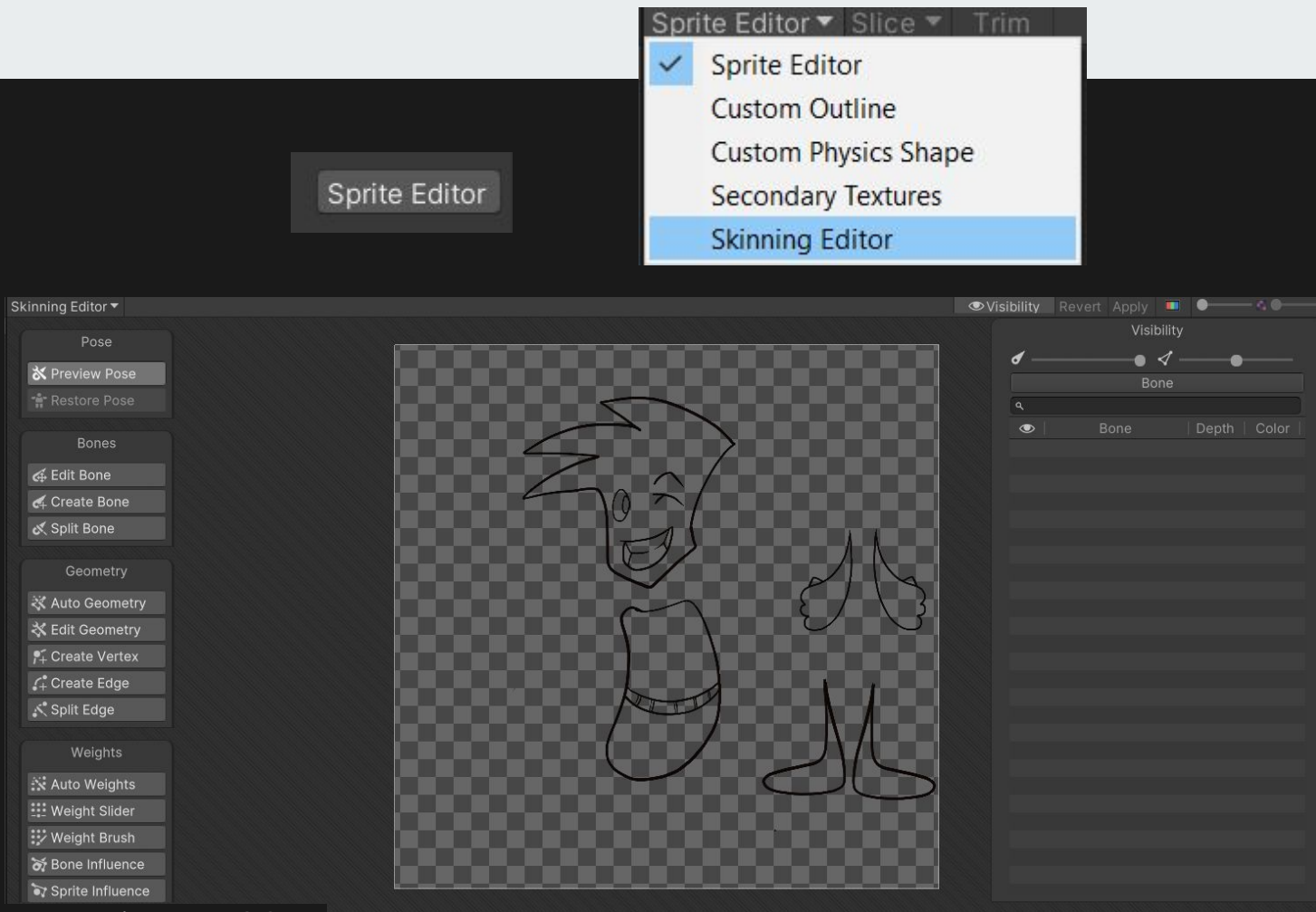
Just make sure once you reached your final frame you put the start frame at the end so you get that last frames time



# Rigging

Rigging is the process of giving a skeleton to a image or 3D model that can be used to animate it.

To create bones we will go into an image Import settings, the go to Sprite Editor and switch the editor to Skinning Editor



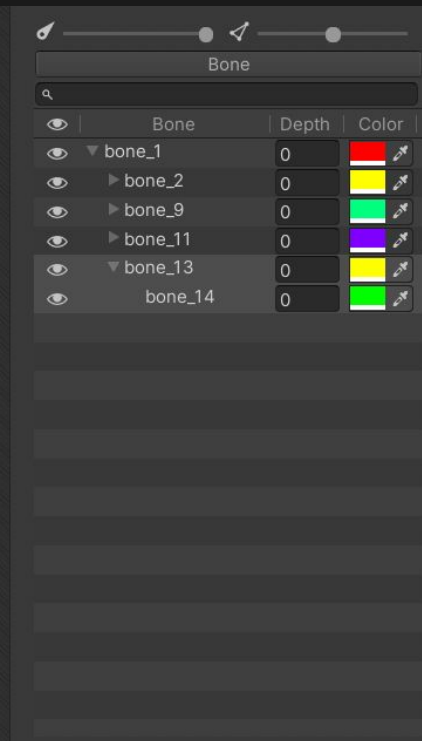
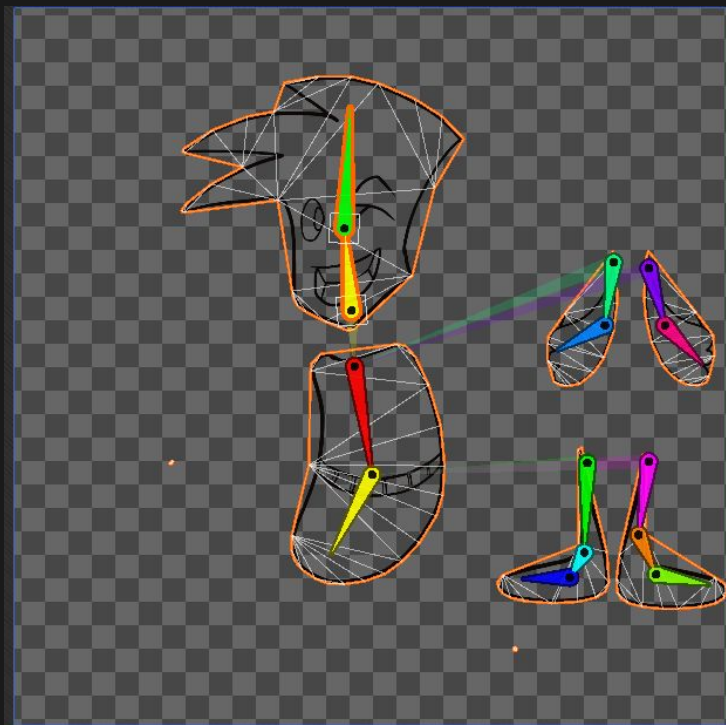
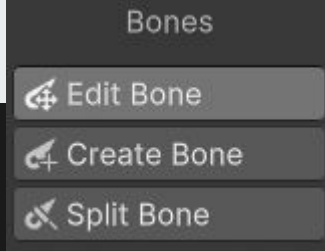
# Adding Bones



Doubleclick and the select all of the objects. It will glow orange when it's selected.

Go to the bones and click Create Bone, then head to the body. Create a few in the body, once you're done right click to exit.

If you want it to connect to other part click on the circle in the and move it to another orange area, it will be transparent meaning it's connected.



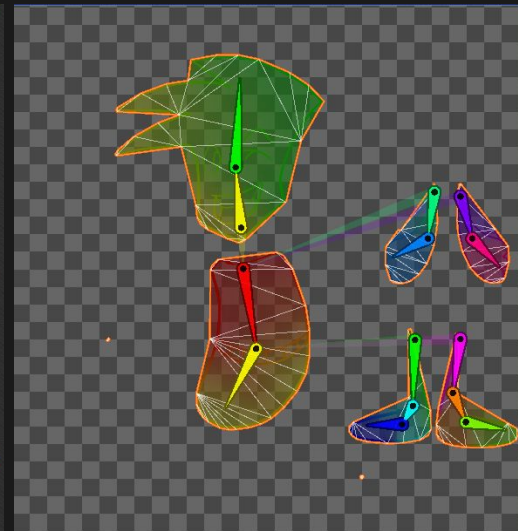
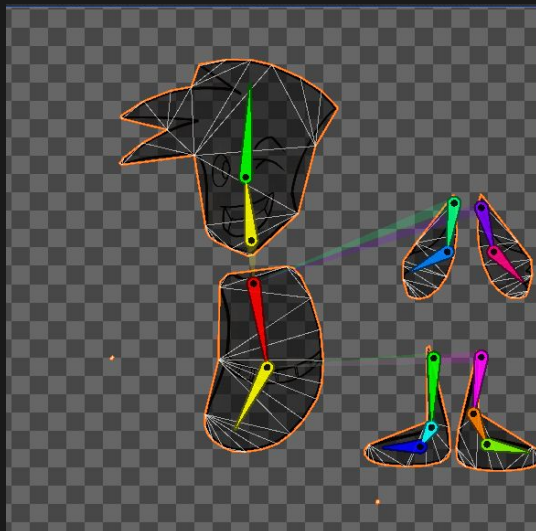
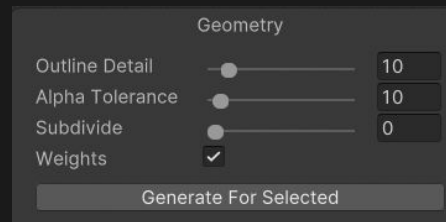
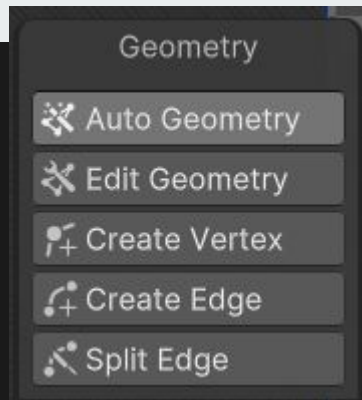
# Geometry



White Lines is the geometry of the character, it's how the image is divided up internally.

The color is the weight it shows how much the area in the geometry is affected by the bone near it.

When you click on Auto Geometry the Geometry field will show up in the visibility. Click Generate for Selected and it will auto fill all of it.



# Bone Depth



Bone Depth is the order which will bones will show up above others.

It works like Sprite Layer Order. Depth being larger makes it show closer to the camera and lower further back.

In visibility you can also rename your bones. This will help you animate.

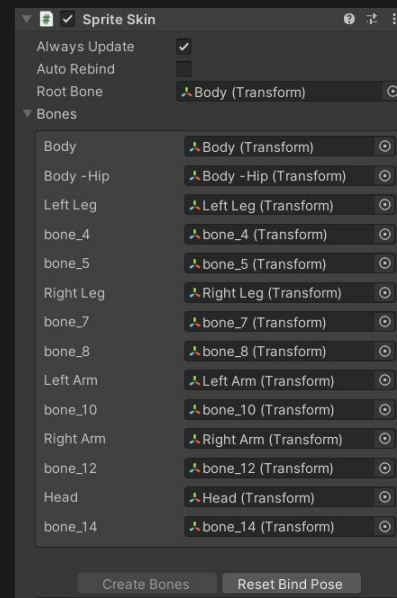
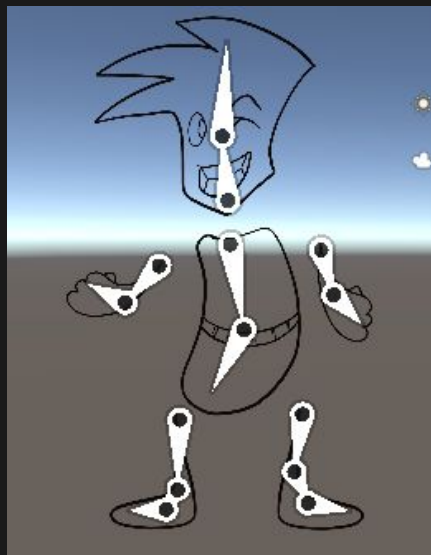
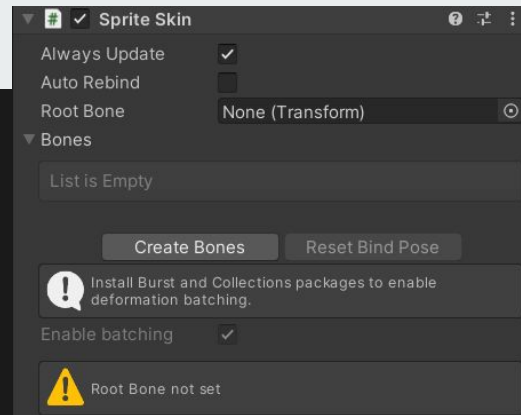
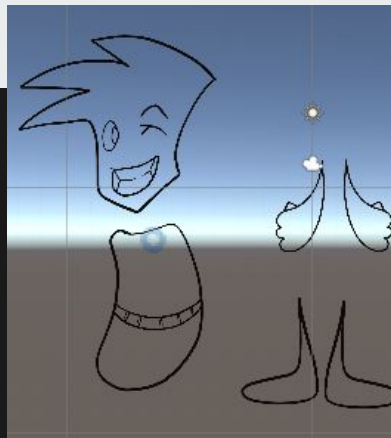


# Sprite Skin

Now all of that is saved we can bring our sprite into the Scene.

You will notice that it's still broken apart. To remedy that we will add a Sprite Skin and click create bones which will create the bones and children game objects that correspond them.

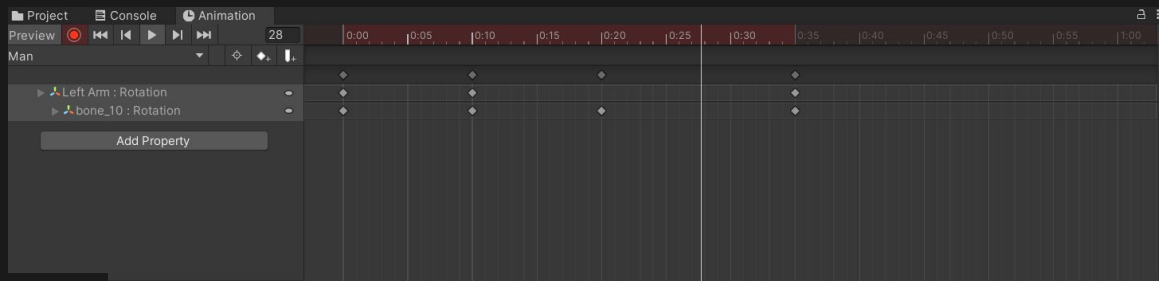
Once that happens you will be able to move the limbs in their correct place.



# Animate



Now that we have that we  
can create an animation  
controller and use the bones  
to animate our character the  
same way we did with  
everything before.

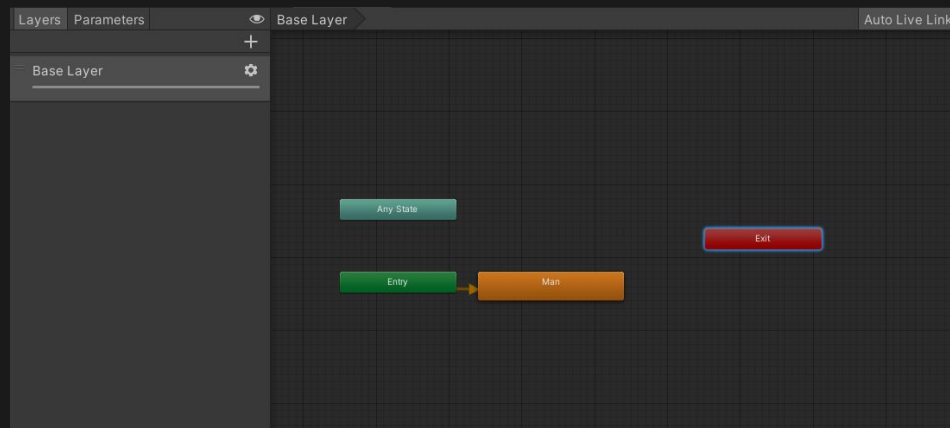
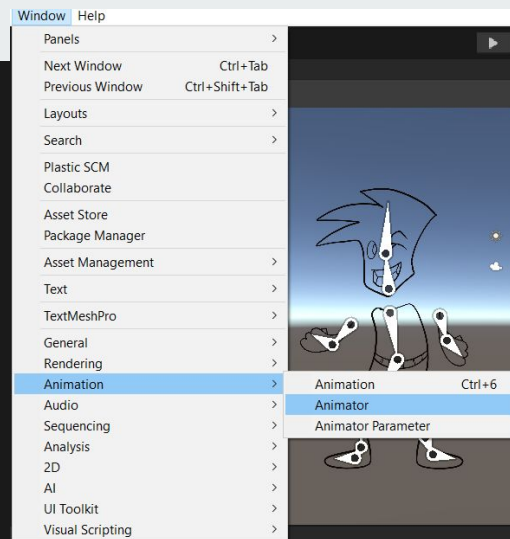


# Animator View

Now that we have a couple animation set up we can use the Animator View to link them and set Transition between them.

Click Window -> Animation -> Animator

In here we will create a State Machine we'll have the animation alternate between several differentiate state based on player's actions





# Base States



Every animator will have these three connections.

Entry is where the animation starts, it will have an arrow pointing to a Animation clips, that will be the first animation to play when the game object is in playmode.

Any State means that no matter where in the machine you State Machine you are you if the parameters permit will move to the specified animation.

Exit lets you end an animation and go back to Entry.



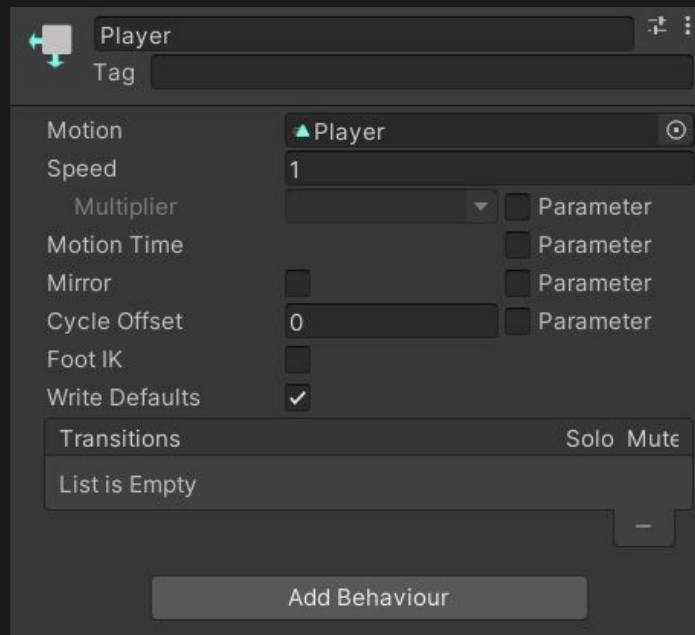
# Animation Clips



Animation clips will be shown in either Orange or Gray.

Orange Animation Clip is the initial clip that will play when the game object begins to animate while the gray ones will be states that you can enter.

Currently this only has the two states we created in the Animation View but we can add additional clips by dragging them from Project View.



# Transitions

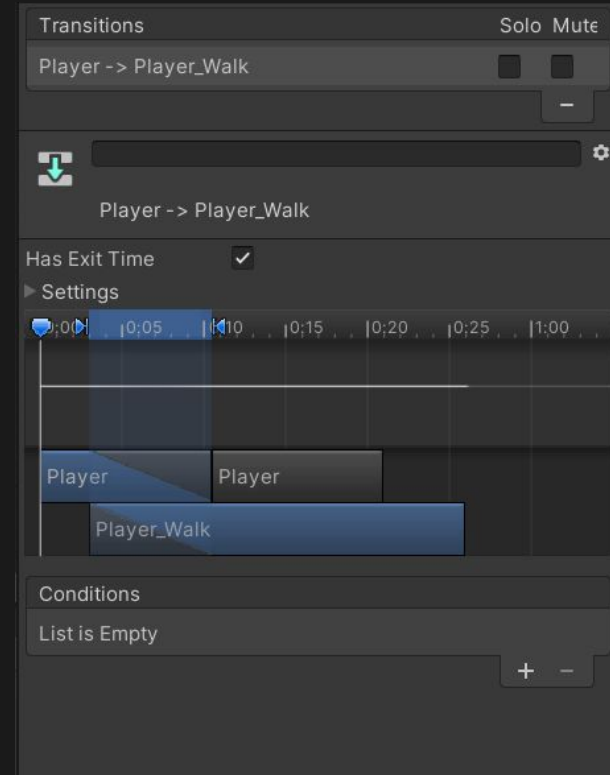
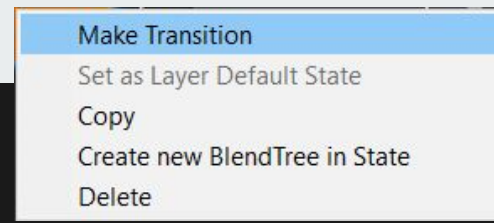


The arrows going between each animation are the transitions, when you click on the their settings.

The blue area is how long it will take to move between one animation and another.

Usually you will want it to be almost zero for things that the player controls such as movement, otherwise they feel input lag.

It can vary for other animations.



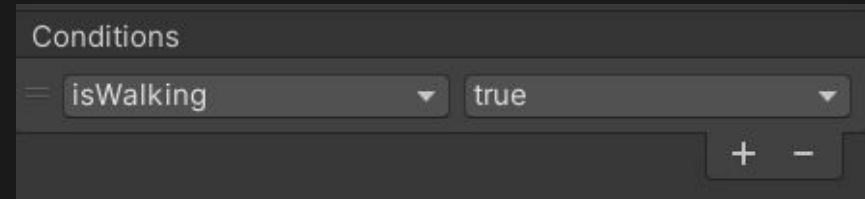
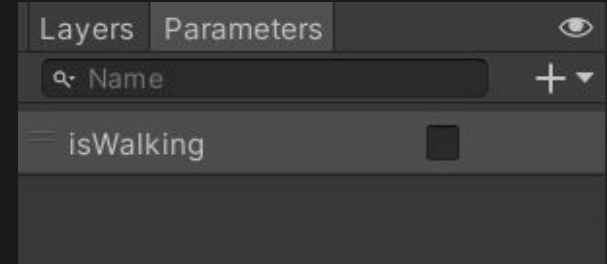
# Parameters & Conditions



Now to have something transition between different states you will need to create parameters. In this case we will create a bool call is walking.

Then we'll click on the two transition, add a condition and set one to be true and the other false.

But to actually have them move between these animation we'll have to use code.



# Script - Call A Single Animation

Player  
Sprite

Sprite : Position  
Sprite : Rotation

Before we update the parameters make sure you have an Animator class, and connect that class to the component that's on the body.

You can call individual Animation by using `Play(name of the animation);`

Warning: Make sure that the game objects that you are animating are not the same you will use to program, animation locks up the variables used in the component and you won't be able to change them using code.

```
private Animator animator;
```

```
animator = GetComponent<Animator>();
```

```
//Listen for the play to click to the  
if (Input.GetKeyDown(KeyCode.Space))  
{  
    animator.Play("Cube");  
}
```

# Updating Parameters



Here we have two snips of code that let us animate our character properly.

We take in input from the Horizontal keys and check if the player is moving, if they are we set the isWalkingParamter to true otherwise false.

We can also update the scale, by setting the x axis to be negative when the speed is 0 we will have animation look like it's moving in the opposite direct saving us the trouble of animating several walk cycles.

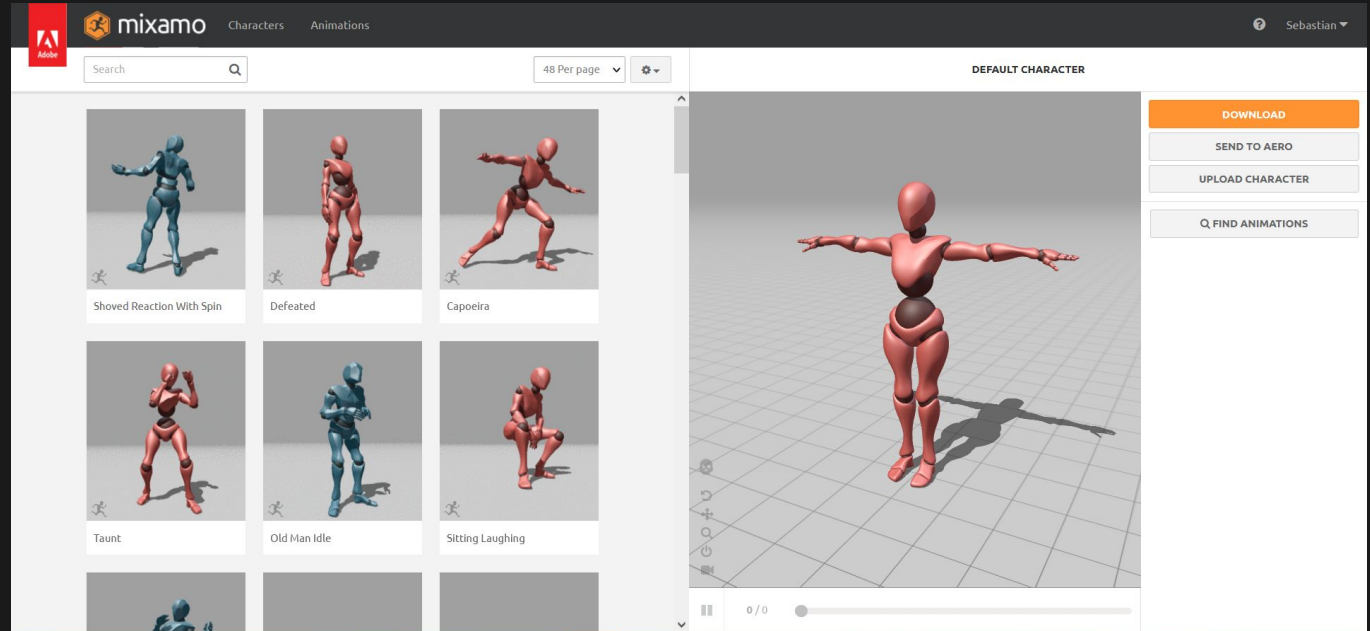
```
//Gets the player input  
xSpeed = Input.GetAxis("Horizontal");
```

```
//Updates the Animator States based on player input  
if(xSpeed != 0)  
{  
    animator.SetBool("isWalking", true);  
}  
else  
{  
    animator.SetBool("isWalking", false);  
}
```

```
//Uses scale to flip the player left or right so we can u  
if(xSpeed < 0)  
{  
    spriteTransfrom.localScale = new Vector3(1, 1, 1);  
}  
else if(xSpeed > 0)  
{  
    spriteTransfrom.localScale = new Vector3(-1, 1, 1);  
}
```

# Mixamo

Mixamo is a website owned by Adobe that allows you to download free rigged animation that can be attached to 3D humanoid meshes.



<https://www.mixamo.com>

# Downloading

Before you click download make sure you select in place, otherwise the animation will affect the transform of the object which we want to control through scripts.

Format defines what data will be included with the animations

Skin lets you choose if you want to keep the mesh or not

Frames Per Second is how many frames, the more frames the smoother the animation but larger the file.

Keyframe reduction allows you to downsize the animation at the cost of key frames which define the actions taken.

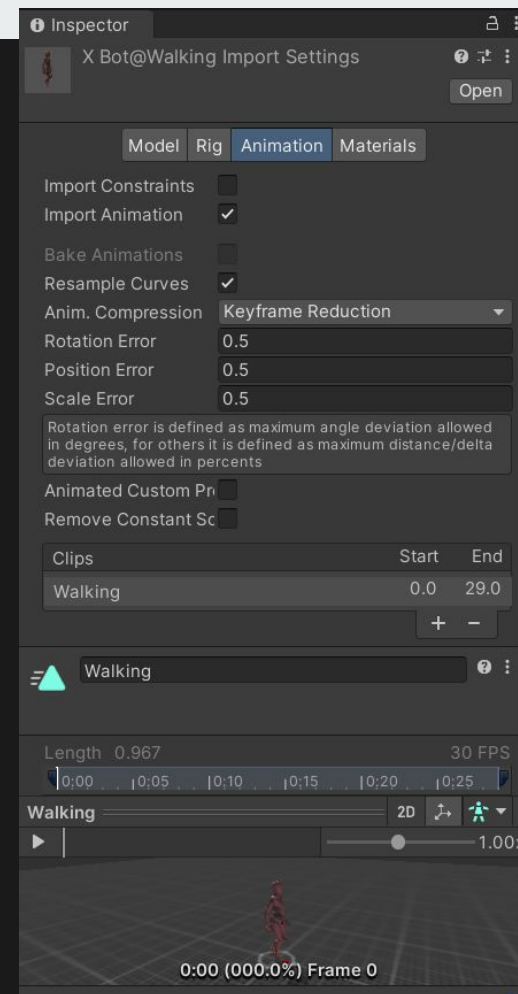
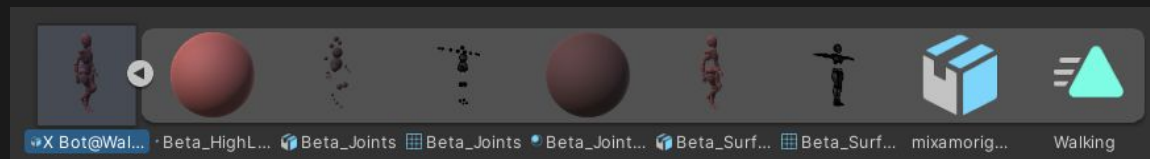


# Importing The Files

Once you've download an animation you can drag it into your project view, you will get the mesh, materials, riggs and the animations downloaded with the Unity format of the object.

If you click on the Game Asset you you will see the inspector breaks down into a detailed aspect of each part.

If you go to the animation tab you can preview the animation that were download with the model.



# 3D Animation

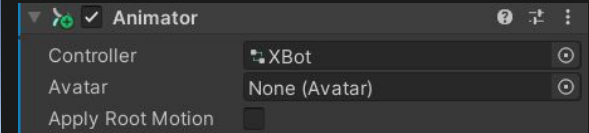
▶ X Bot@Walking

To make this work you will first drag the game asset into the scene.

Once the object in the scene we will have to go to the inspector and add the Animator Component.

The object comes pre imported with animations but we still need to create a controller.

We will create an animation controller and call it XBot.

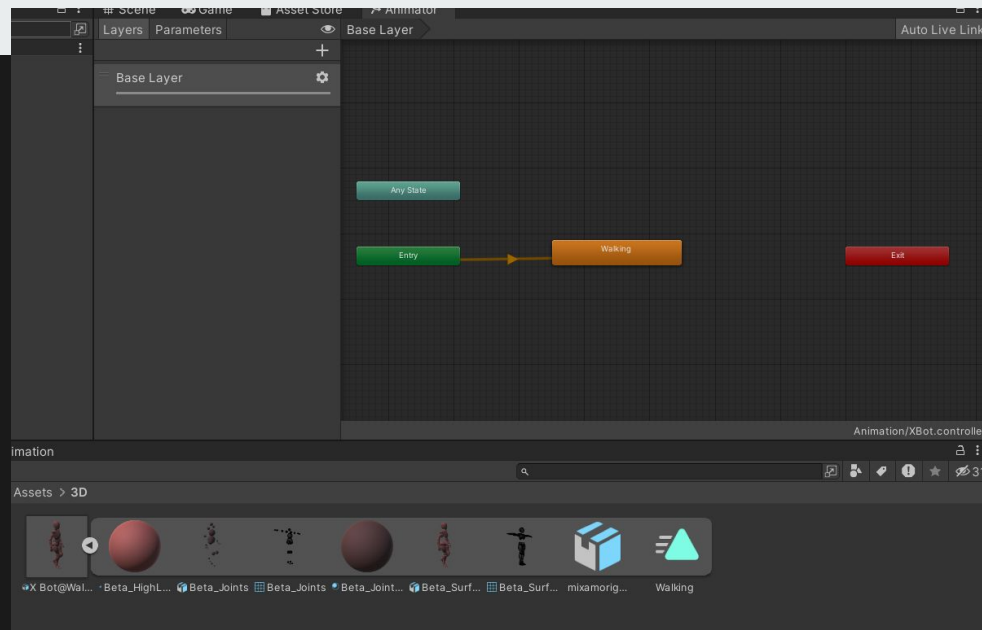


# Adding the Animation



The animation controller will be empty. What you have to do is drag the animation asset into the Animation Controller View and it will be auto added.

Since the asset comes with a rigger the animation will know what part goes where and how to move. Turning on the level will make the object be animated.





# Audio, Sound Effects, and Lighting

# Today's Agenda



We're going to be going to be continuing with Chapter 21: Audio, Chapter 16:

## Particle Effects

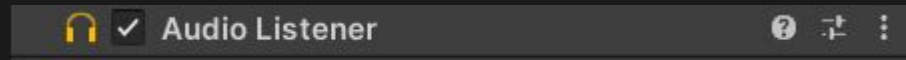
- How to add audio to our game
- What is 2D vs 3D audio
- How to use sound Mixers to control different audio in the game
- How to use Particle Effects to enhance the game

# Audio Listener



Audio Listener is a component that is attached to the camera whenever it's created.

It is the component that will listen to the Audio Sources and play them in the speaker based on how distant it is from that source.



# Audio



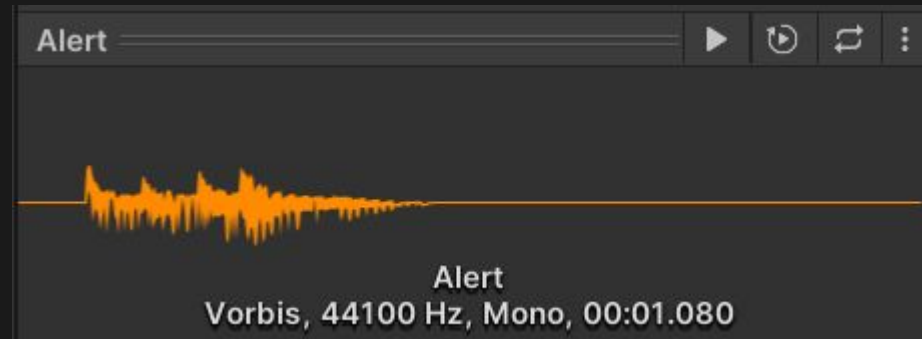
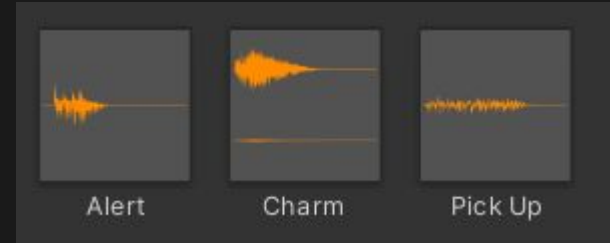
<https://freesound.org/>

What are some audio you can use in your game

You can use Music, Sound Effect to give player feedback and Voice Acting.

All of these will be shown to you as their Waveforms that show the amplitude of sound.

You can test out all of the sounds inside of the Inspector in the preview on the bottom.



# Credit



Crediting is very important when working on a game. Imagine you put together your project and some nabs it shows it off and never credits you.

We haven't talked about it yet but now you will have tons of effects and you need to keep track of the information but the same idea applies to any art, mesh, animation, font or anything else you haven't made on your own. Credit people's work.

It's good keep an notepad, or a script full of comments that saves the name of the piece you used and who it was from and what website you found it from.

```
From FreeSound.com
```

```
Music
```

```
"PianoKeys_Meral.wav" by Meral
```

```
SFX
```

```
"Charm by Scrampunk
```

```
"Alert" Video Game Sound by EVRetro
```

```
"Retro Bonus Pickup SFX" by suntemple|
```



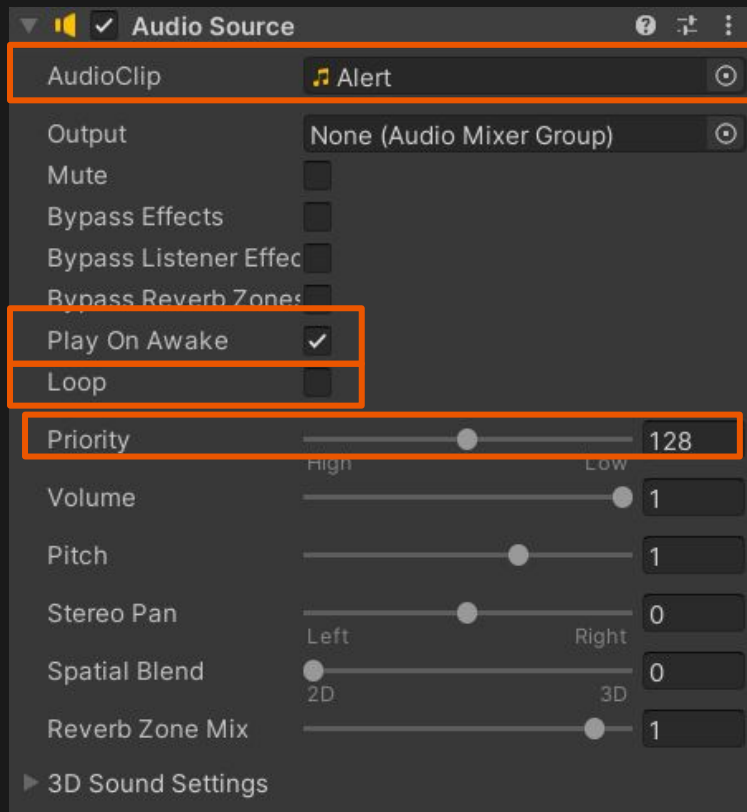
# Audio Source

The AudioClip sets what Audio you want to use in this component.

Play On Awake means that whenever the audio source is turned on, we will use that to our advantage in animations but for the most part you will want to turn it off otherwise all the sound effect will play at the same time when you start the game.

Loop will make the audio source restart playing from start when it ends, very useful for implementing music.

Priority tells us which audio should be played more clearly when several are playing at once, if you have both music and sfx what should be louder. Lower the number to more clear it will sound over sfx playing at the same time.



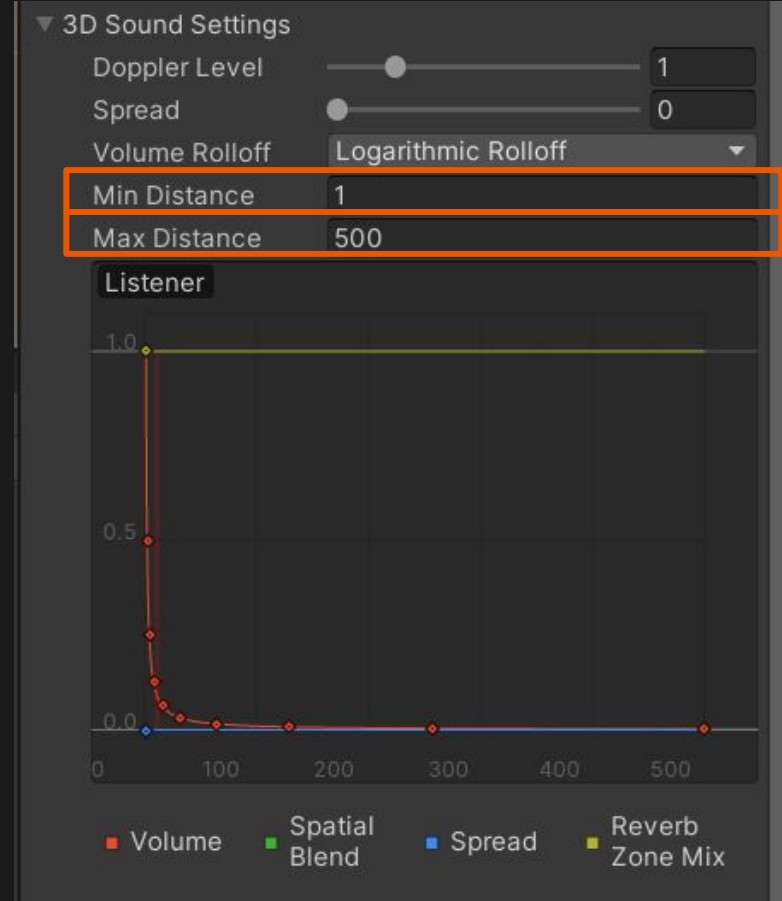
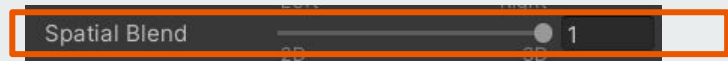
# 3D Audio



Spatial Blend in the regular settings allows you to control how the sound will be perceived. When it's set to 2D that means you will hear it the same regardless of meanwhile 3D will care about how far or close you are to the object.

You can control these when you open up the 3D Sound Setting.

Main thing we care about is the Min Distance and Max distance, you want to make them small so that the player has to be near the object to hear it play.



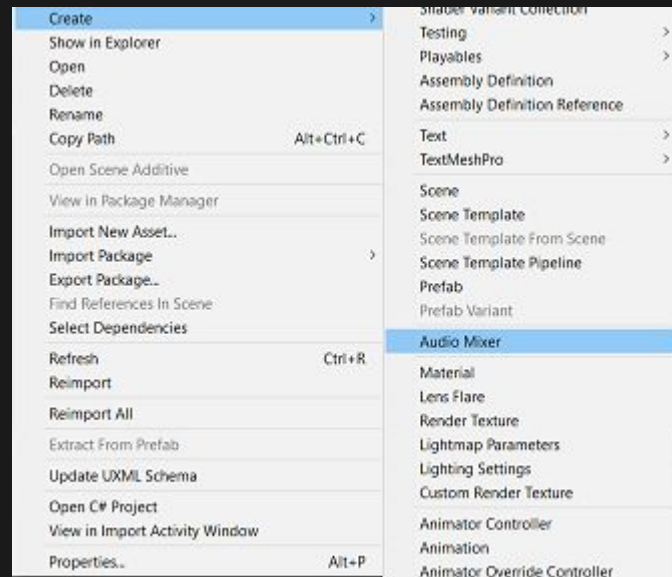
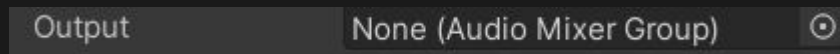
# Audio Mixer



We will now create an Audio Mixer.

Audio Mixers are a way for us to be able to control all of the sound effects and music in the scene with one slider.

To create an Audio Mixer add a Game Asset in your project view and add Audio Mixer.



# Audio Mixers

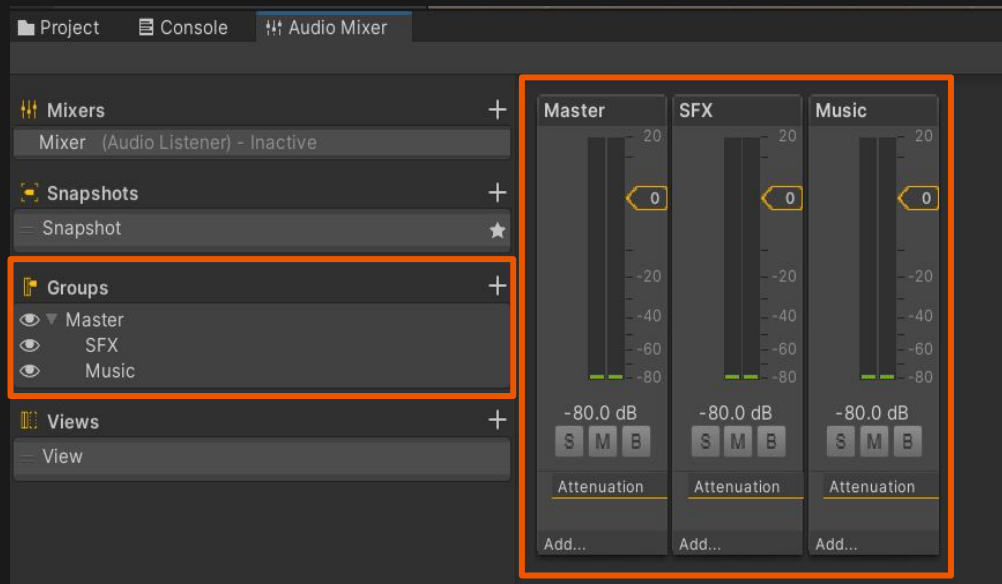
Once we created the Mixer you can double click on it and it will open the Audio Mixer View.

Here we can create and manage the different Mixers using the Mixer Tab.

Create Snapshots, which have preset values for all of your groups.

Create Groups so you can contain different SFX together or separated. Groups can also be nested such like here Master is Parent of SFX and Music.

And View allow you to see only specific groupings.



# Groups

On the Right Hand Side you see our three groups. The main part of them is the sound slider that controls how loud audio sources connect to that group can be.

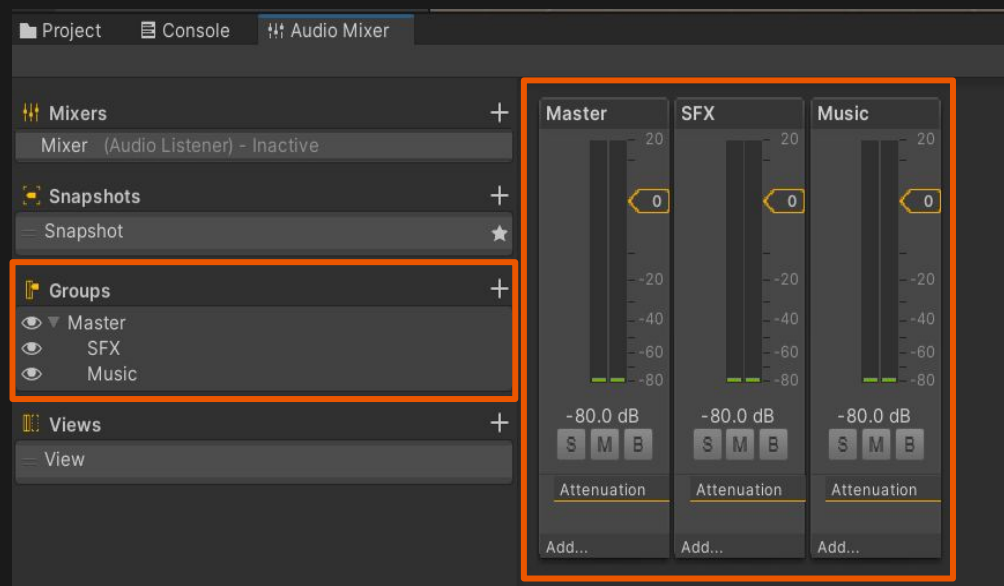
Next you have the three button that allow you to

S - SoloGroup, only play that group when selected

M - Mute

B - ByPass/ Ignores effects

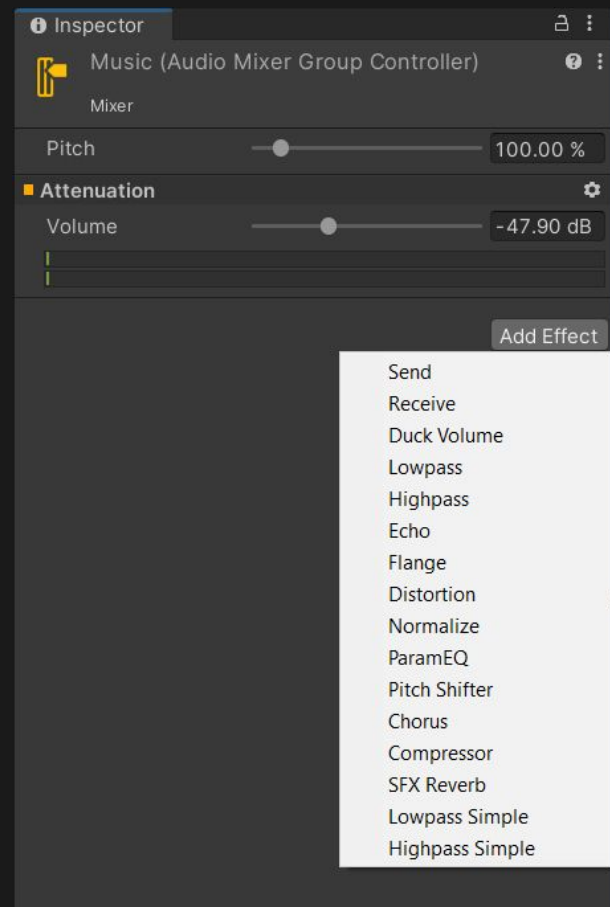
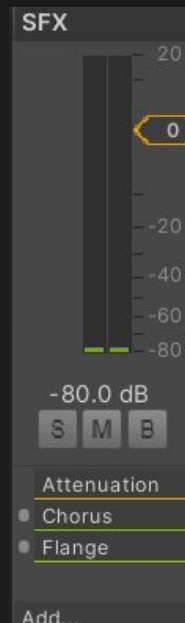
Below the buttons you see the Effects placed on the group currently there are none but once we add them it will list them,



# Effects

Audio Effects allow us to apply the action to all of the Audio Sources that are connected to the group.

They can absolute change the sound from what it was to something completely new.



# Audio In Script

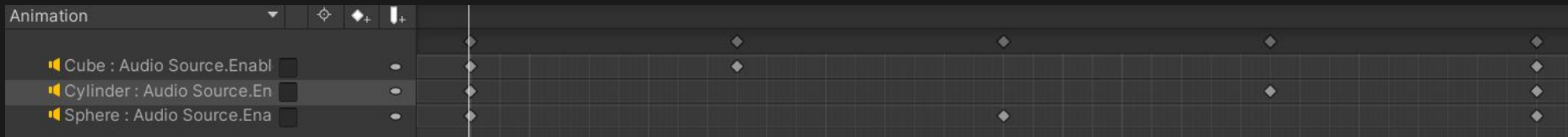
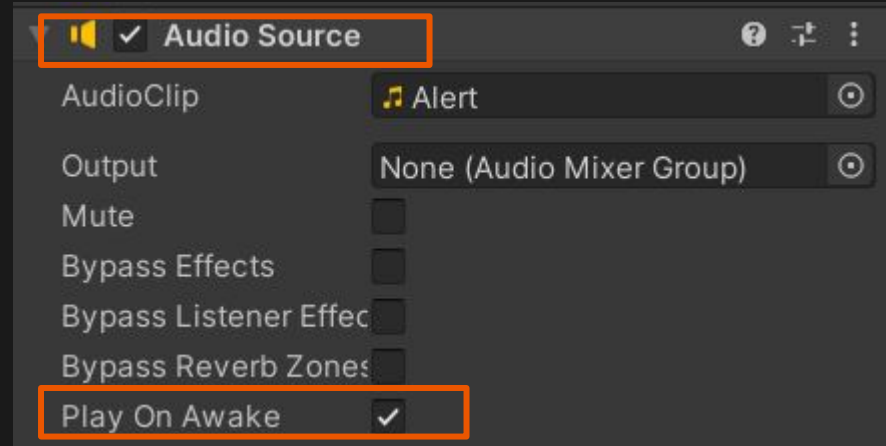


Making an audio play using script  
is as simple as import and  
calling the Play() function

```
AudioSource audioSource;  
  
// Start is called before the first frame update  
📦 Unity Message | 0 references  
void Start()  
{  
    audioSource = GetComponent<AudioSource>();  
    audioSource.Play();  
}
```

# Audio In Animation

As mentioned before if we set the Play On Awake to be on and keep the audio source off, we can use the animation tool to make the sound effect play.





# Particle Effects System

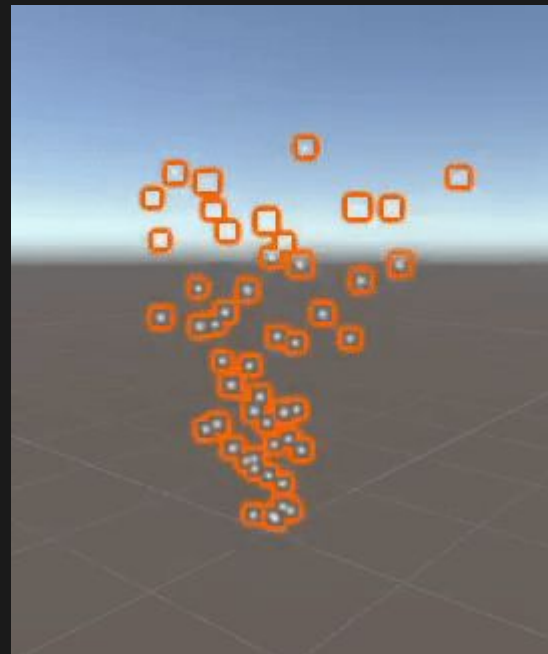


Particle effects are very important as they provide feedback just like sound but rather visually than audio wise.

Whenever you hit something you want there to be some kind of flash, or when you walk a dust cloud get left behind. Particle effects allow you make the illusion that things are touching as if they were in the real world.

To make a particle effect go to Hierarchy and create effect Particle System.

Effects	>	Particle System
Light	>	Particle System Force Field
Audio	>	Trail
Video	>	Line

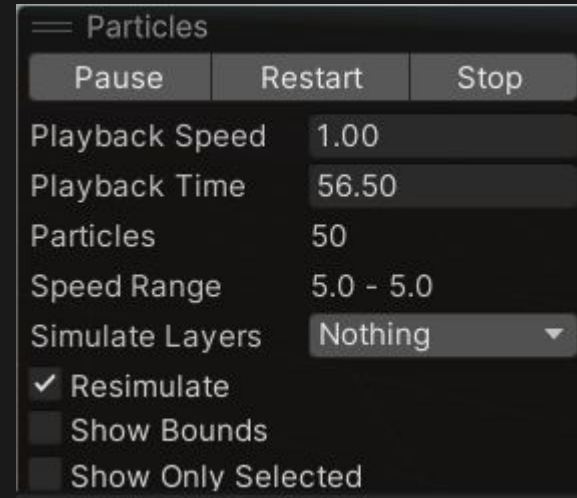


# Controller



When you select your particle system you will have the Controller for the particle effect will pop up in you scene view.

This lets you preview how you particle effect will act when played



# Component



You will notice that the Particle Effect Component is massive.

It's to allow you to create as detailed effect as possible.

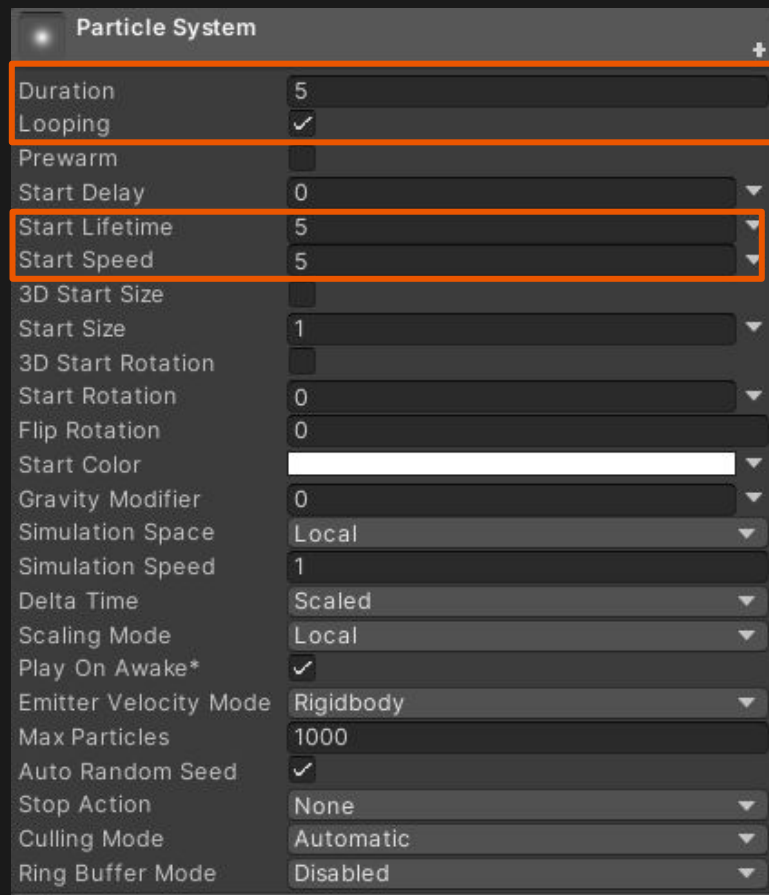
In the general area of the comment we will case about four variables.

Duration, it's how long the whole effect goes on for.

Looping, if you want it to repeat upon completion

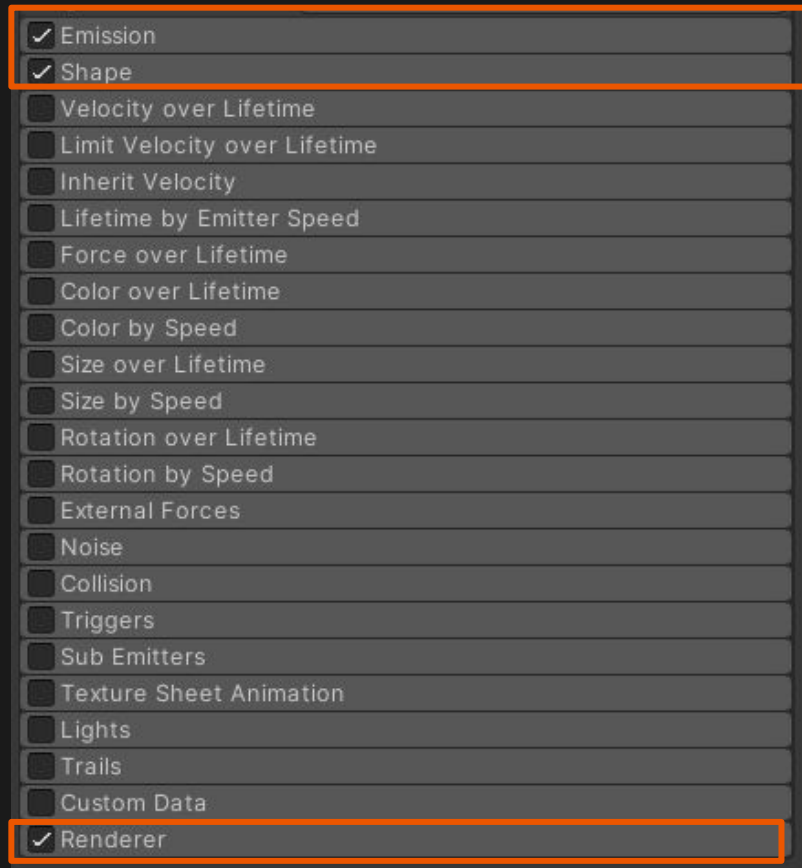
Start LifeTime, how long the individual particles will live for from start

How fast should they be going.



# Effects

You will notice you have a lot of optional effects that you may want to play around with the main three will go over are  
Emission, Shape and Renderer



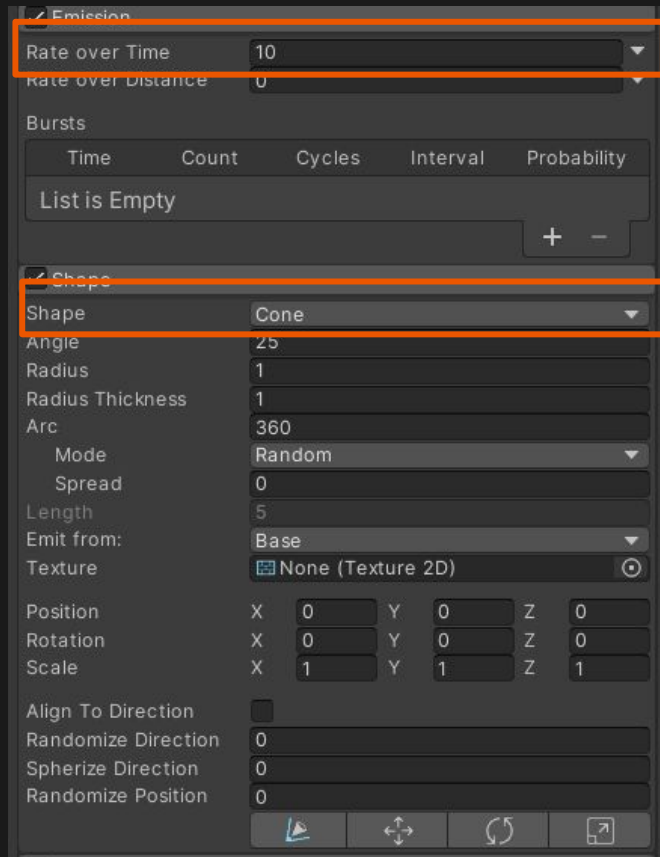
# Emission and Shape



Emission control how many particle effects are created over time,

You can also allow it to create a burst of particles at once so they all show up together.

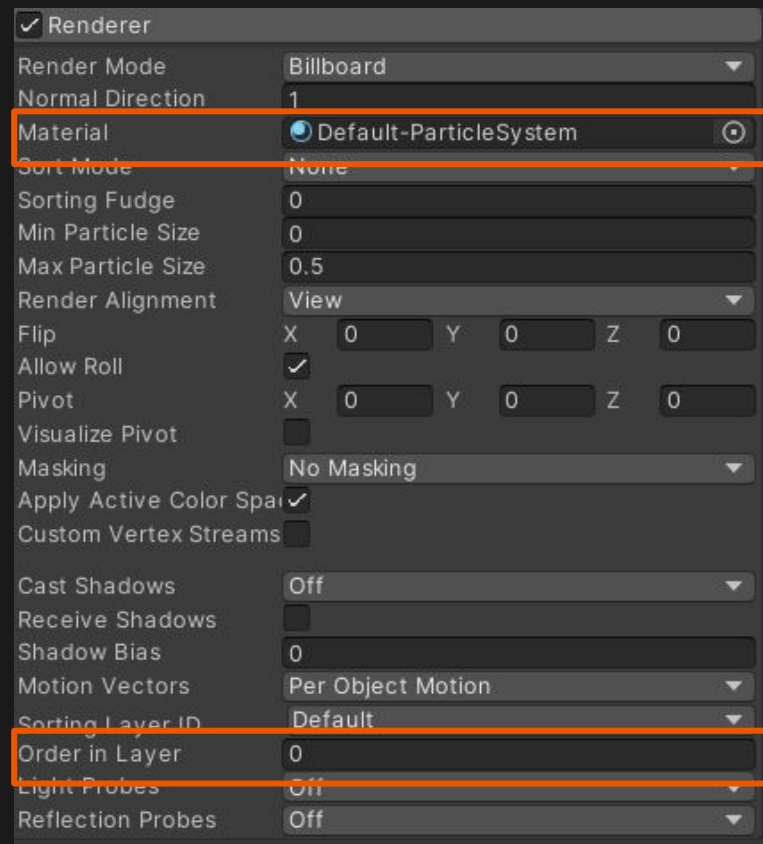
Shape on the other hand controls in what way the particle will be sent out. It starts out with cone but there are tons of ways you can have your particles spawn.



# Renderer



Like a Sprite Render the the Render component controls what the particles will look like. We mainly care about two thing Material which is the image or texture that's going to be attached to the particles. And the Order Layer which will controls it's visibility.

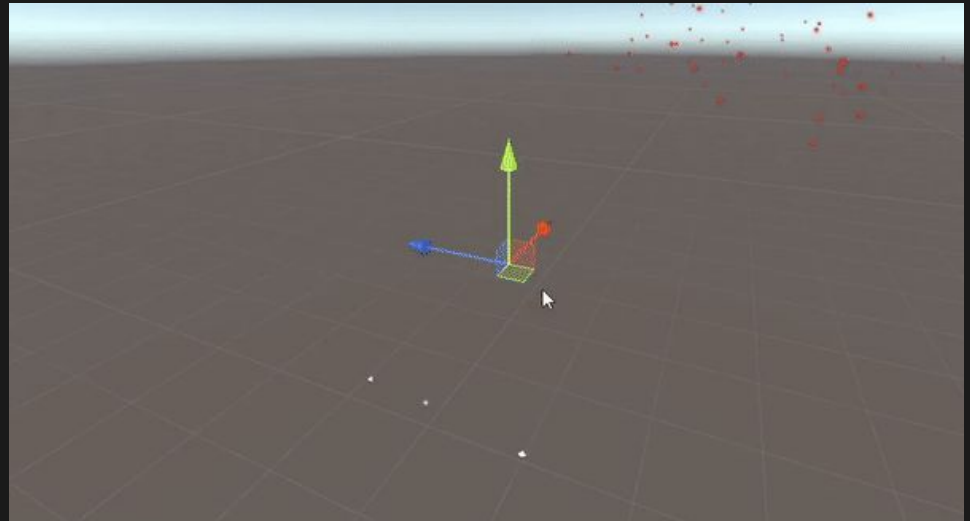


# Trail



Effects	>	Particle System
Light	>	Particle System Force Field
Audio	>	Trail
Video	>	Line

Trails are another effect system that can be very useful to you, they can help you show that something is moving very fast, or you can use it to show the players action with the mouse.

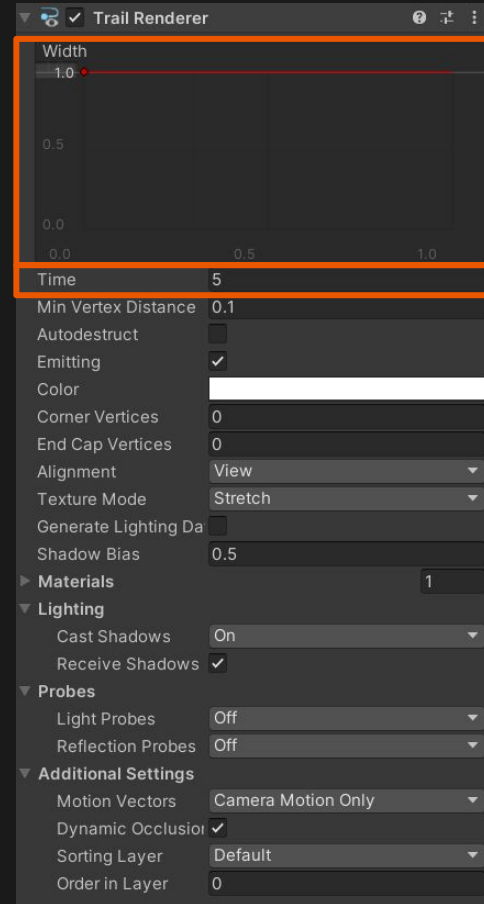


# Trail



The trail has this big graph that controls how thick the trail will be, you can make many nodes on it and move it about.

You can control how long it will stay on screen for, what color it is and what it's made of.





# Lighting

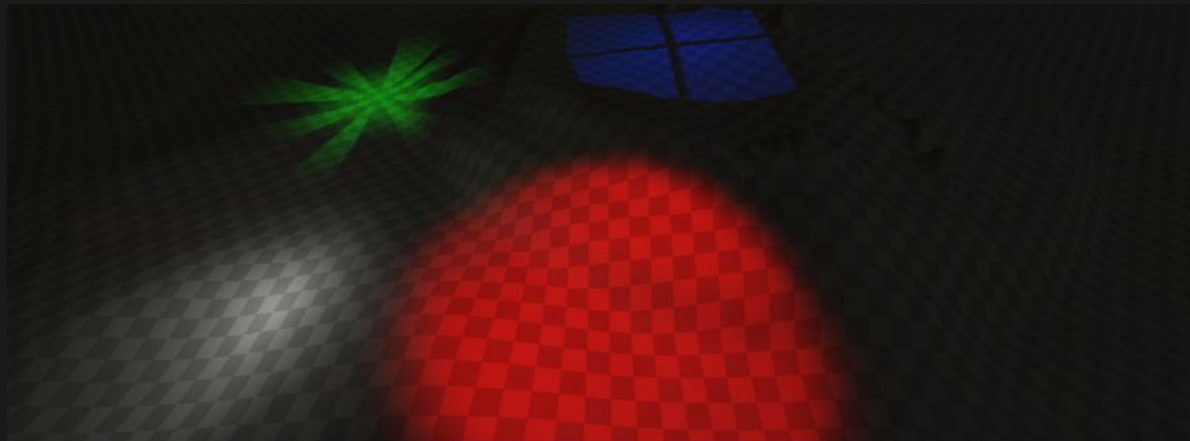


Lighting can affect the games mood and even set the tone for what game you are making.

Making a game with dark lighting is great for setting up a Horror exploration game.

We will look at three lighting sources. Directional, Point and Spotlight.

Light	>	Directional Light
Audio	>	Point Light
Video	>	Spotlight
UI	>	Area Light
UI Toolkit	>	Reflection Probe
Camera	>	Light Probe Group
Visual Scripting	>	



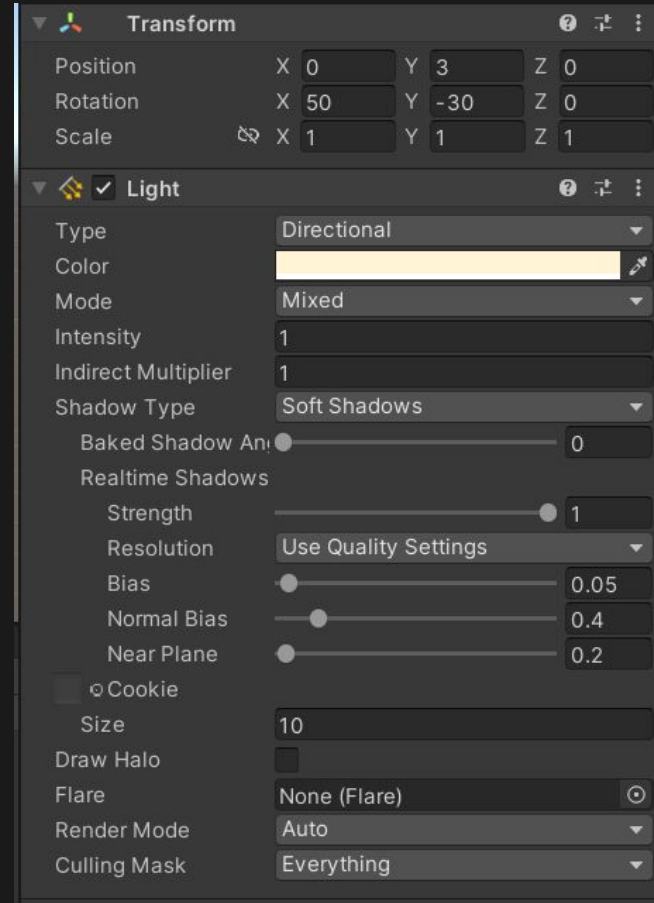
# Directional Light



Directional light is created in any 3D game, it the Sun of the scene.

Position and Scale don't really matter however the rotation is crucial to it. Which way the light is point determines how much light is being shown.

You can also control the color of the light and how you want the shadows to be rendered.

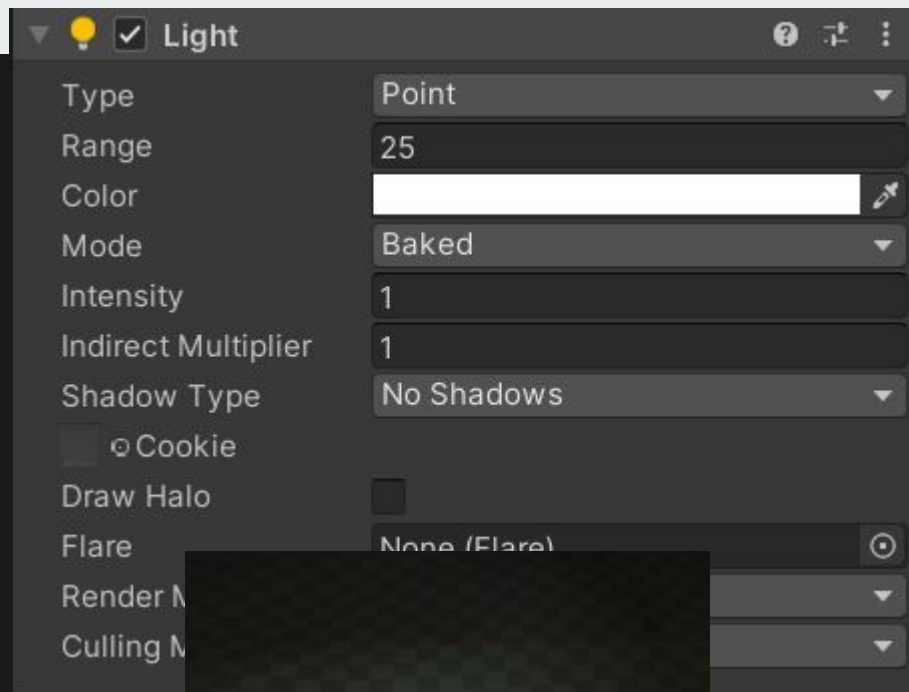


# Point Light



Point light are area of effect lights. You give it a sphere that is controlled by the range and it will have a glow tha slow loses power from center to the end of the sphere.

You can also use Intensity to how strong the light coming from the source is.

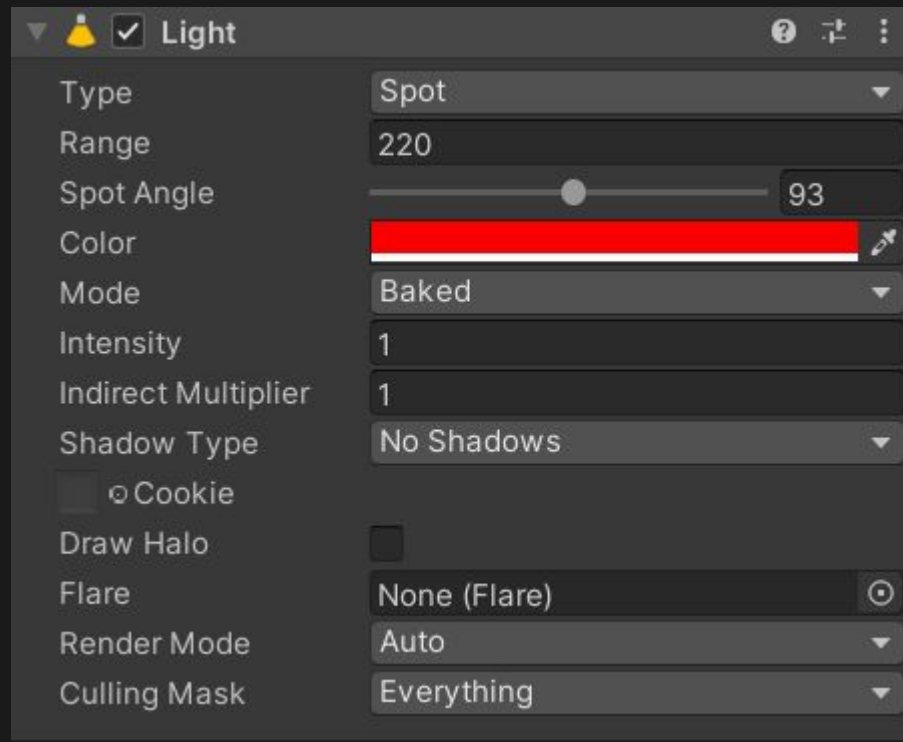


# Spot Light



Spotlight is what it sound like a light that shoots in on direction rather than radiates from the center.

You can control the angle of the and the range of the cone that it uses to show the light.



# Cookie

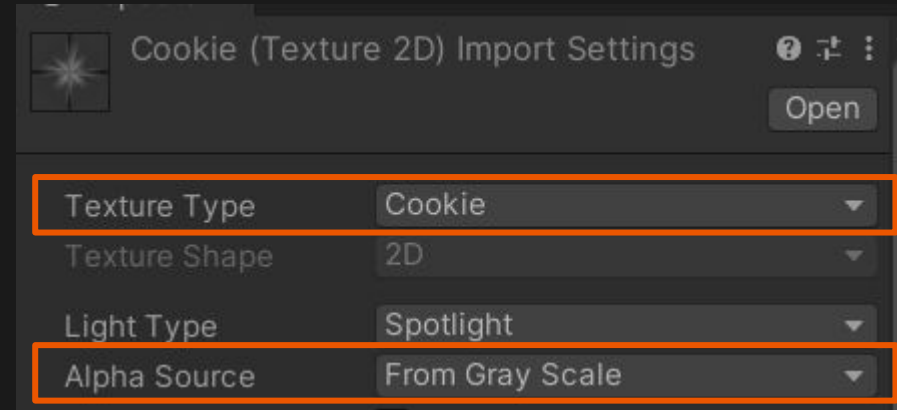


Cookie is a way of taking a image and having the light system display it.

You first want to make sure that the image you are using is black and white.

Next you will set the texture Type to be Cookie and the Alpha Source to be From Gray Scale so we only care about the black and white values .

After all that apply the changes and you can drag your image into the Cookie slot on the lights.



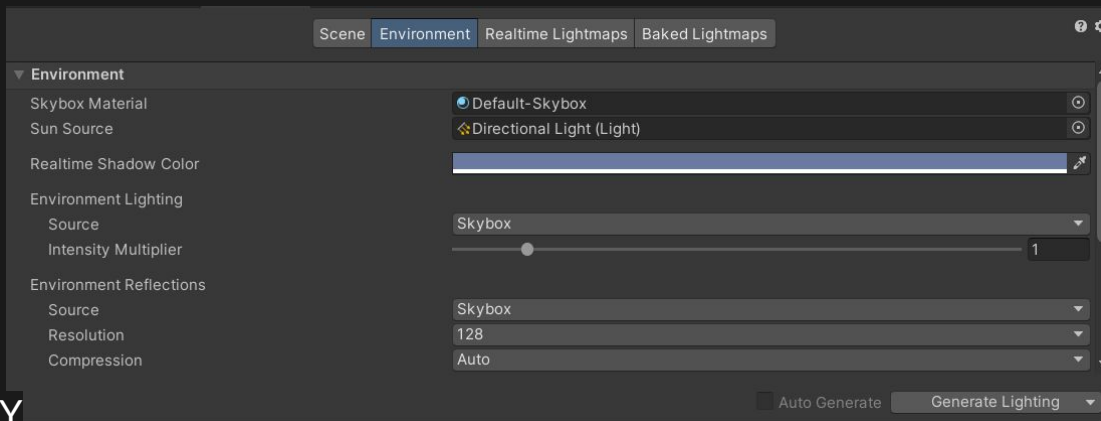
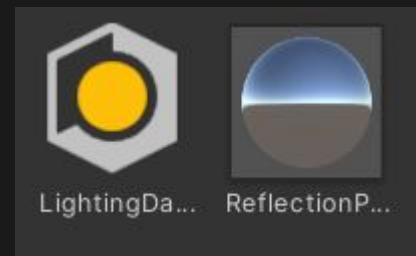
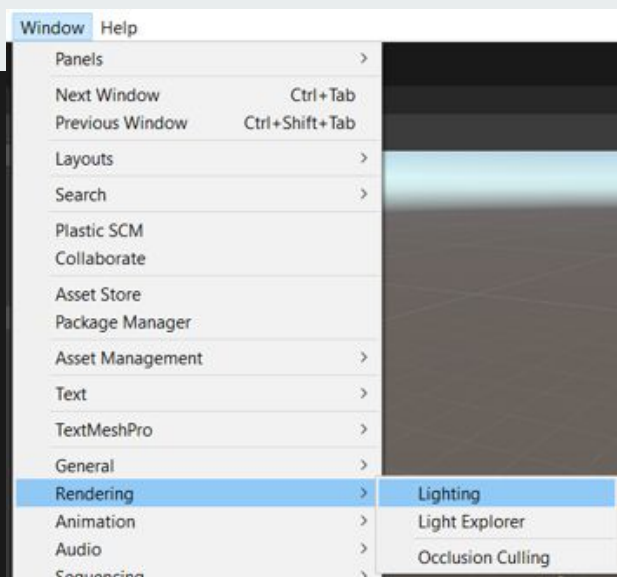
# Baked vs Real Time

Currently as you are editing your scene the Lighting is Real Time but the game isn't running so it's not hard for unity to do that.

Once the game operates and all of the game element swork it's much harder to keep up with demanded calculations.

That's why we Bake the lighting into the scene. Meaning once you're happy with the way it looks you can save the setting on all of the object and bake them into the scene.

To do that we will open the Rending Lighting View and go to the Environment and click Generate Lighting which will create two object that will save what the scene will look like.



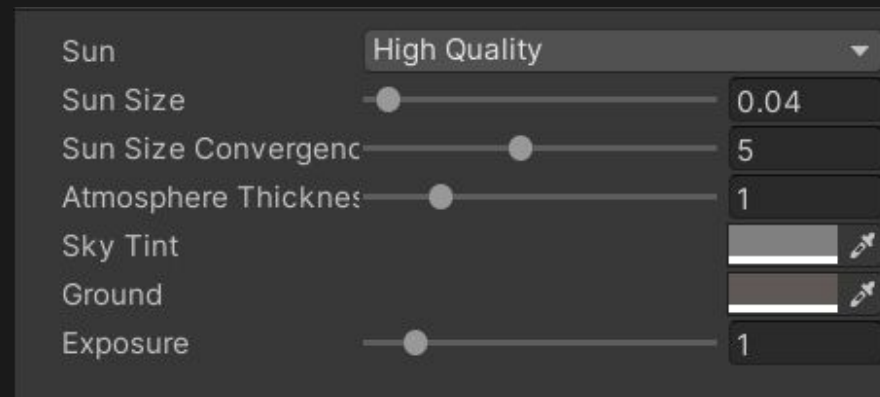
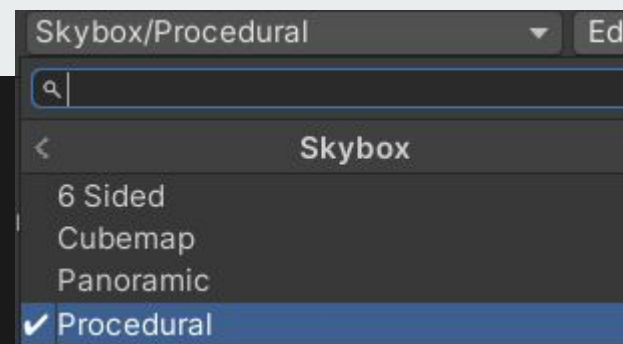
# Skybox Procedural



First skybox we will look at is procedural.

To Create a skybox we will create a material and inside the Material Inspector we can change it from Material to Skybox Procedural.

This will allow us to create a gradient sky with a ground.



# Skybox CubeMap

Now a different way of creating a sky box is by getting a skyboxed cube image.

Inside the Import Settings you will set the Texture Shape to Cube which will make it look like a material.

Then you can Create a new Skybox Material and set it to Skybox/Cubemap pass in the image and now you will have to vista in your game.

