2D Game Design

Sebastian Grygorczuk - STEM Institute at CCNY

Today's Agenda

We're going to be going over Camera portion of Chapter 5: Lights and Camera

Chapter 12: 2D Game tools, and Chapter 13: Tilemaps

- Using Orthographic Cameras in a 2D Scene
- Learn about 2D Assets and Tilemaps
- Create Game Objects with 2D Collision and Physics
- Create and Share Your 2D levels

What is a Prefabs

A Prefab is a complex asset that has been bundled and can be reacted with little extra work. It also gives us the power of Inheritance.

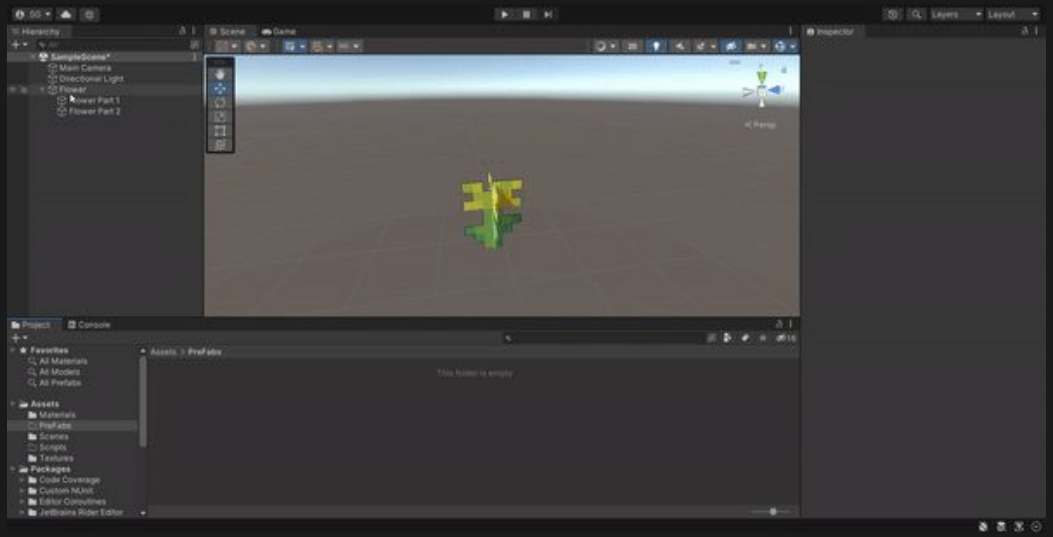
Inheritance is when all of the Game Object are linked to the Prefab, giving us the power to edit the PreFab and passing on those changes to all of its children Game Objects.

For example, let's take the Flower we had in previous Lesson. Let’s say we wanted to make a lot of them and after we made all of them we wanted to change their size, without a Prefab you’d have to go to individual game object and edit those characteristics.

Additional Resources: Pages 191-192 of Textbook

Creating A Prefab

Creating is as simple as dragging a Game Object from the Hierarchy View into the Project View.



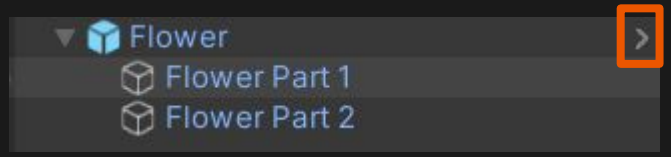
Visual Difference of PreFabs in Views

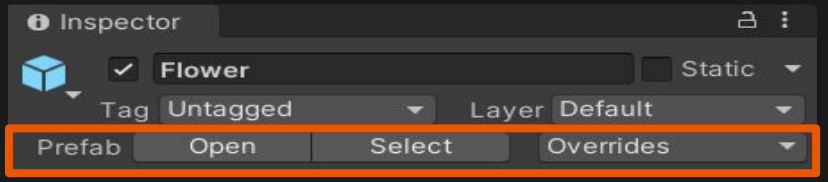
Once you've turned a Game Object into a Prefab few things will change/became available to you.

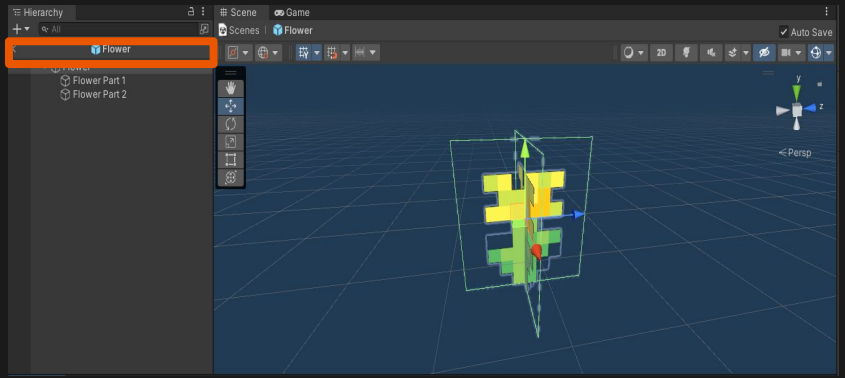
First, the Game Object and all it's copies will appear with Blue Text in the Hierarchy view to indicate that it is an Instance of a Prefab.

Along side of that the Inspector View Header will gain another row giving you [2] Prefab options, [1] The Arrow in Hierarchy View and the Open Button will send you to the Prefab Mode where only the Prefab and all of its children will be for you to edit directly.

You can exit the Prefab mode by clicking the back [3] Tab in the Hierarchy View.

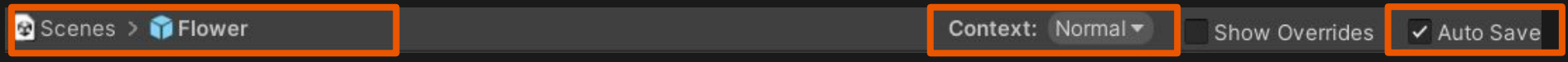






Sebastian Grygorczuk - STEM Institute at CCNY

Prefab Mode



1

2

3

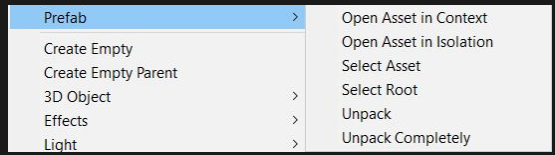
Scene View will receive a new row of options as well. [1] Showing you the Breadcrumb Trail of how you entered the Prefab Mode. [You can nest Prefabs in prefabs]

Next you can change the way the background of the Scene View looks with [2] Context DropDown

And lastly you can [3] Auto Save, if you check it, it will save and copy all of your edits on the Prefab asset. You can also uncheck it and a manual Save button will appear next to it but I recommend keeping the Auto Save on.

Additional Resources: Prefab Mode

You can get into PreFab mode in two ways, Context which is still in the Scene or Isolated where it's just the Prefab. . Just right click on the Game Object in the Hierarchy View and select Prefab for more options.

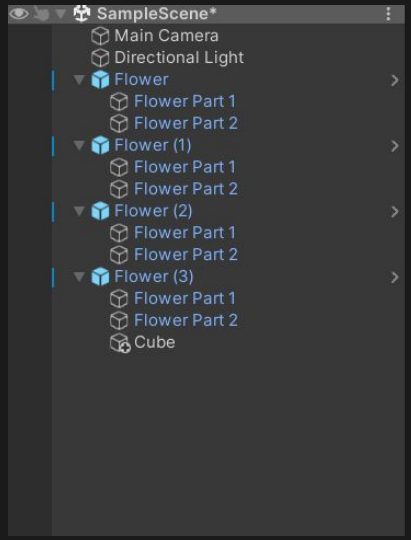


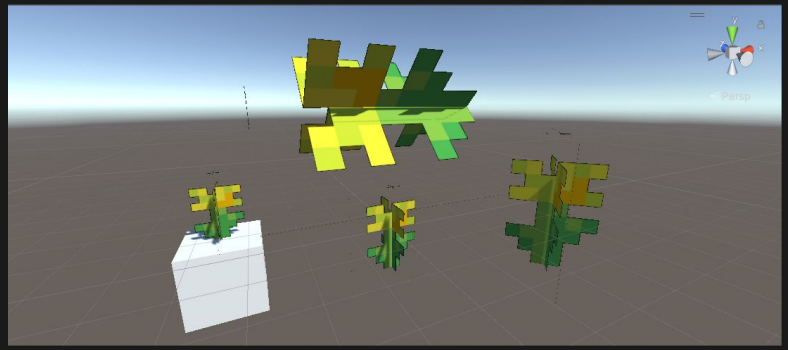
Sebastian Grygorczuk - STEM Institute at CCNY

Modifying Prefabs in Scene

While in the regular Scene View you can modify the Prefabs and all of its components as long as it does not affect the structure of the Prefab. Meaning you can change the Components settings like Position, Rotation, Scale, if the Collision is on or off, you can even add additional components or meshes to the Game Object.

The only thing you can't do is Remove the Game Objects or Components that are already attached to the Prefab.

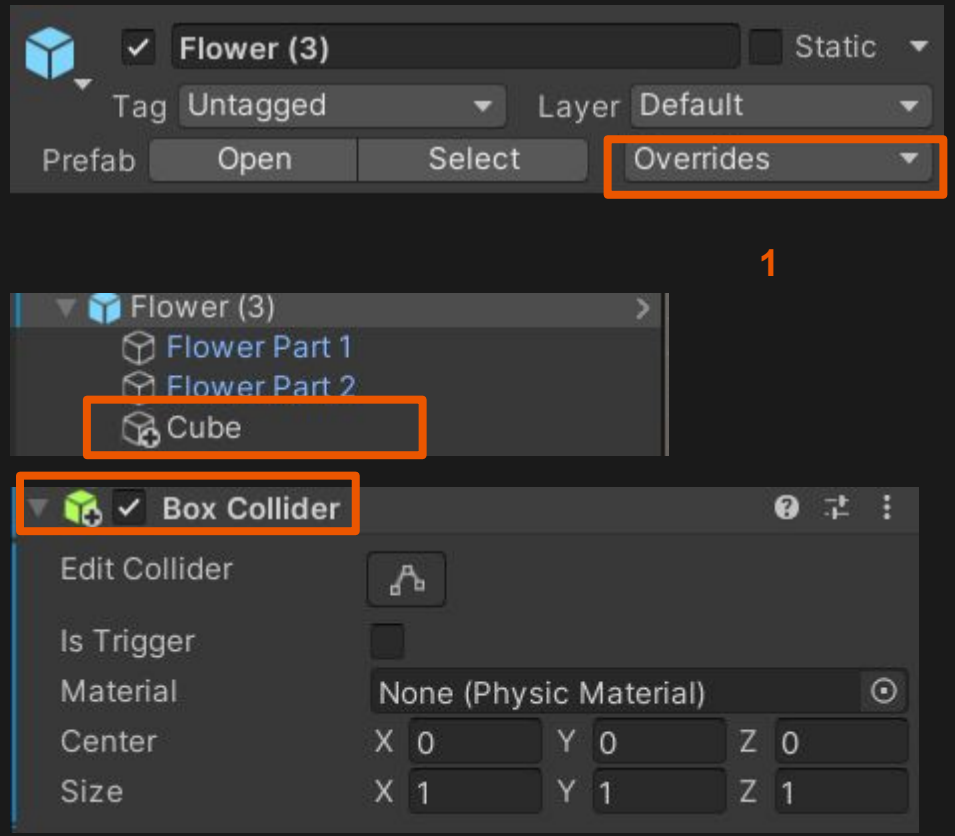




Modifying Prefabs in Scene

While Modifying Prefabs by adding Game Objects or Components to the Instance of a Prefab you will notice the newly added. This is because you can actually apply those changes to the Prefab and pass them onto all the other instance.

By going to the Inspector View Header and cling the Overwrite DropDown you can see all of the changes you've made to the Instance in reference to

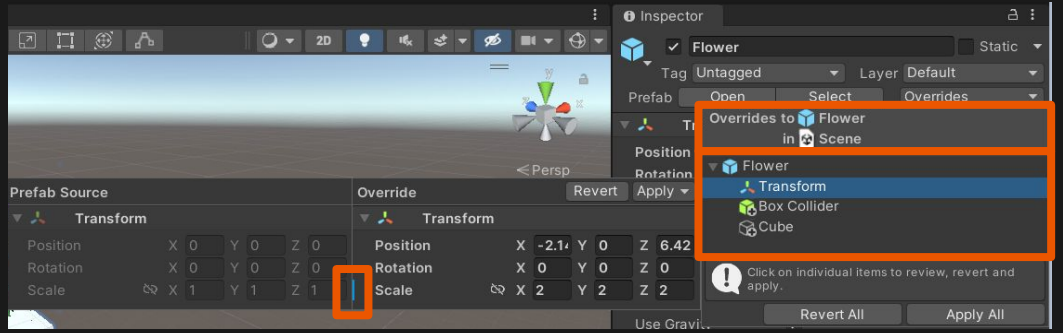


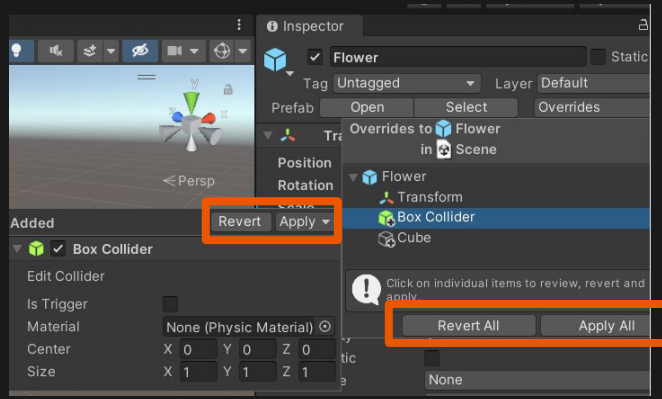
Inspecting Overwrite Changes

When going through the overwrite you will have the Header that shows you where the changes occurred. Then you will have a break down of all of the Game Objects or Components that were affected.

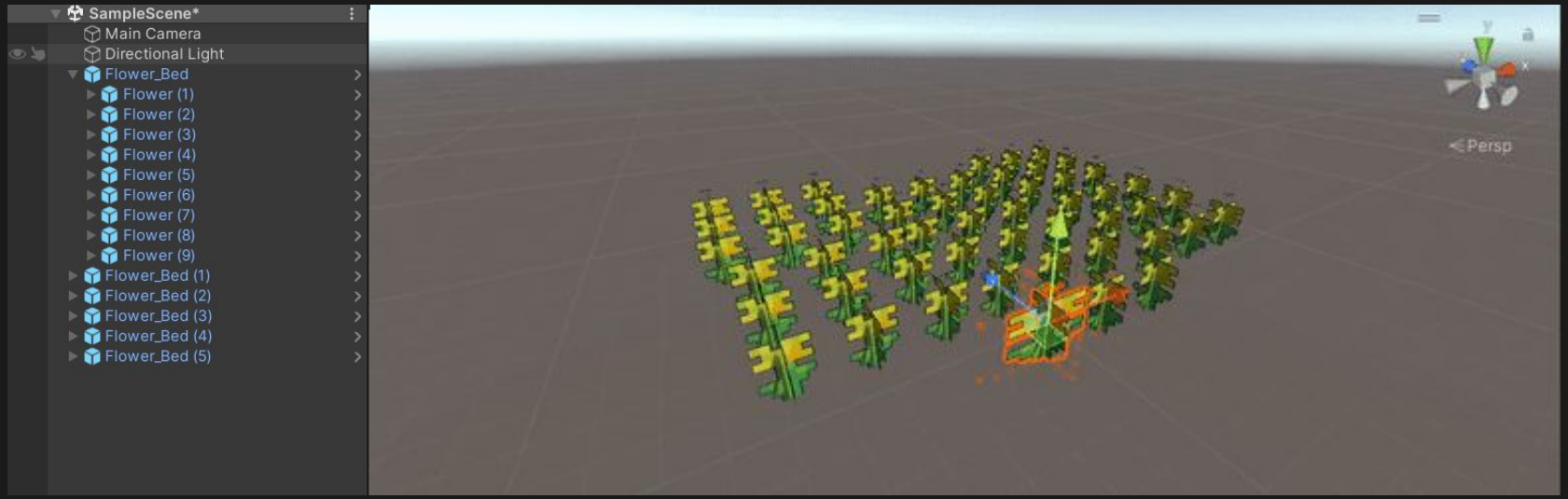
In case of Components that pre existed in the Prefab and were changed you'll get to see what the values are in the reference to original. In this case the Scale was changed indicated by the Blue tab.

On Game Objects or Components that were Added you'll just have them pop up. You can Revert/Apply the changes individually or Revert/Apply All at once.





Nesting PreFabs



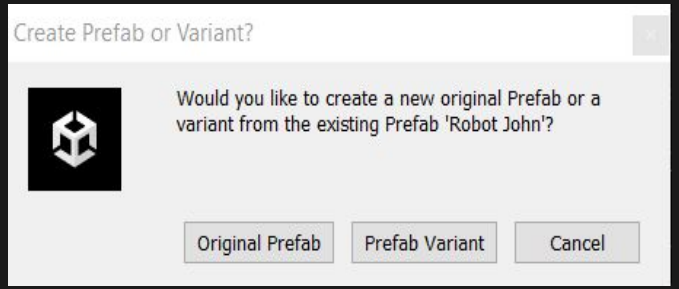
Say you want to make a field of flower out of the one we made, it would be much easier to fill the Scene with flower is you made a Flower_Bed Prefab that contained ten or more of our Flower Prefabs and then used the Flower_Bed to decorate the level.

And now if you think that the flower is too small or the wrong color you can change the original Flower Prefab and all of the Flower_Beds will be updated with the new flower as well.

Making a Variant or Original

You may have done enough changes to a Prefab in Scene that you may want to make it an entirely new prefab.

If you drag and drop the changed Prefab into the Project View you will get this PopUp, allow you to create and Original or a Variant.



Original will create a fully separate Prefab with no connection to the one used to make it.

Variant will create a new Prefab that is connected to the one used to make it, this is great for creating enemies with small variants. Any changes to the original used to make the varraint will also be added to the Variants.

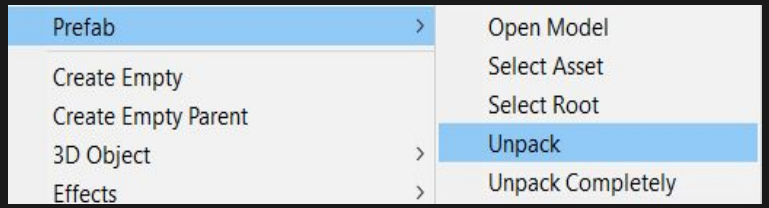
Additional Resources: Prefab Variant

Disconnecting Prefabs

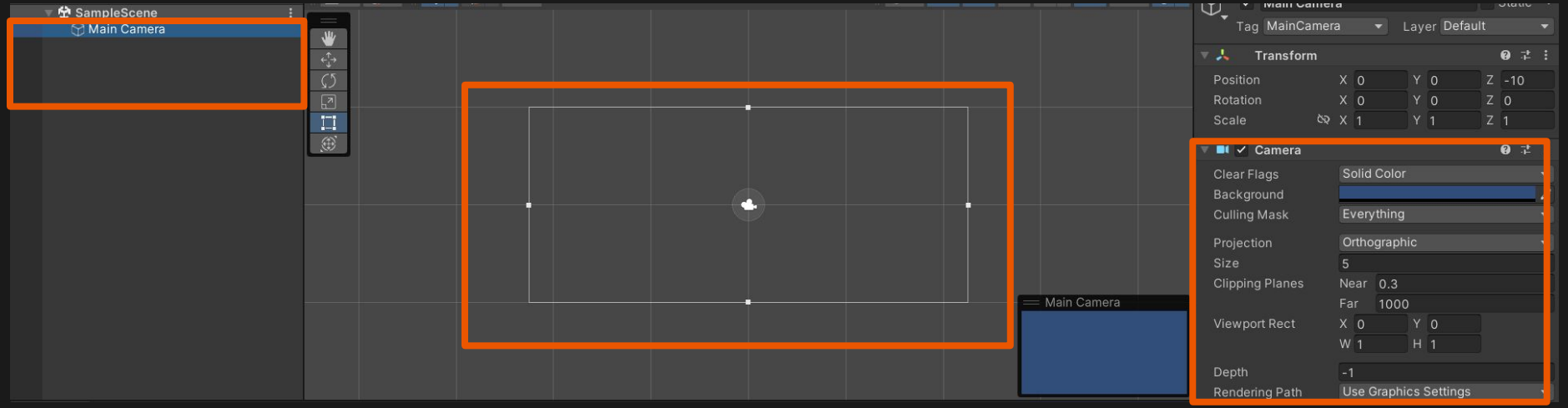
Last thing to mention is sometime you may need to change a core component of the Prefab in the scene but you don't want to make the Game Object from start.

To do this you can right click on the Prefab Object and choose to Unpack it. This disconnects it from the Prefab and allows you to modify or delete any Game Object or Component that was part of it.

Unpack Completely will break down any Nested Prefabs inside the game object too.



2D Game Project

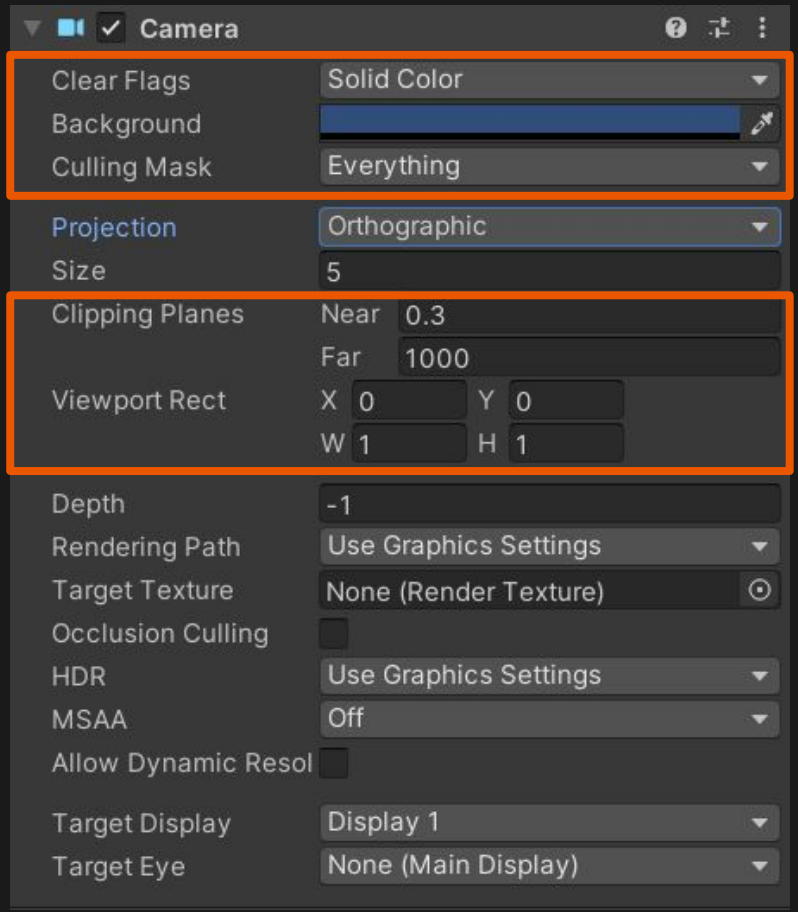


When we start a 2D Game Project there will be no light source and Camera set into Orthographic Projection. We don't need light because we're not loading any meshes and Orthographic Project is better for 2D Games.

Camera

There are two general areas we are [1] controls what's visible, Clear Flags describes the background type we have, normally we're used to using Sky Box but now we're going to use Solid Color defined by the Background. While Culling Mask defines what the camera should render. Currently is rendering everything.

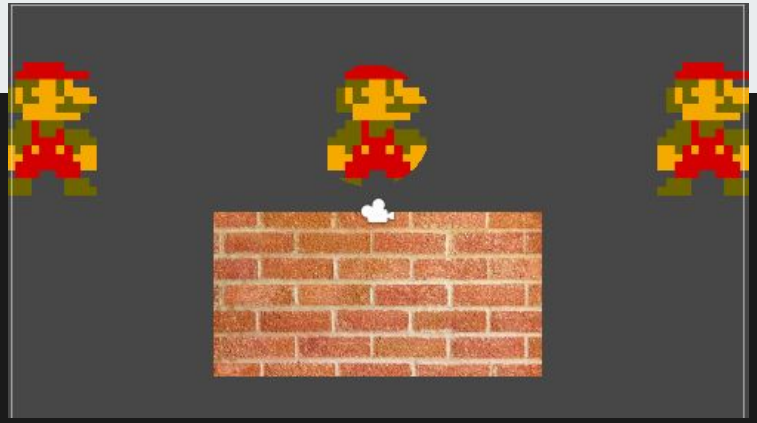
[2] Controls the size of the camera and it's behaviors. Clipping Planes tells us how close or far we need to be for the camera to clip into it, Viewport defines how much of the screen this camera should take up, you could have two cameras taking up half the screen with it or so.

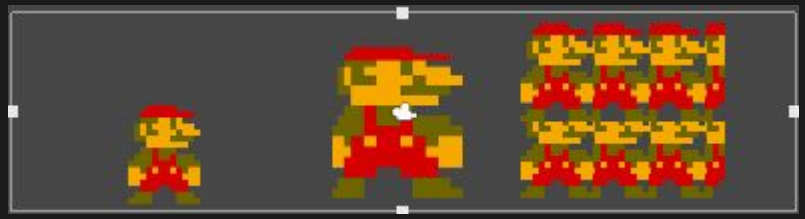


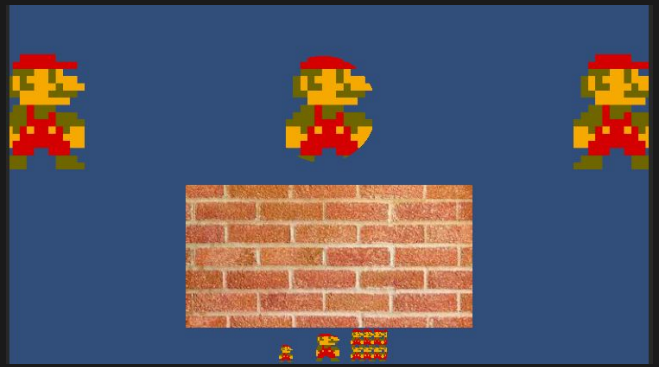
Multiple Cameras

Having multiple Cameras with different Viewports will allow you to display two different areas of the scene at once.

You could show the goals that the player is looking for or split the screen so that multiple players can play at once.





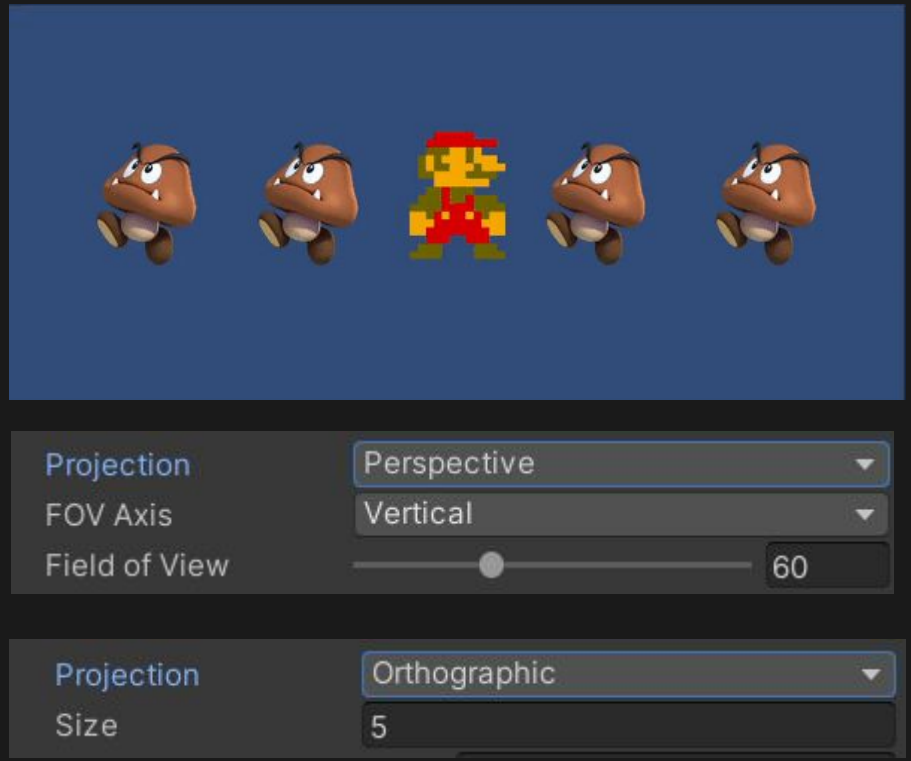


Sebastian Grygorczuk - STEM Institute at CCNY

Perspective vs Orthographic

Perspective Projection is what we've been working with until now, it allows for depth on the 3D axis.

On the other hand Orthographic Projection removes the 3D axis and makes everything on the same 2D plane.



Sprites

2D Sprite are images that are flat on the screen, they can be anything from pixel art, to hand painted, to picture of real people.

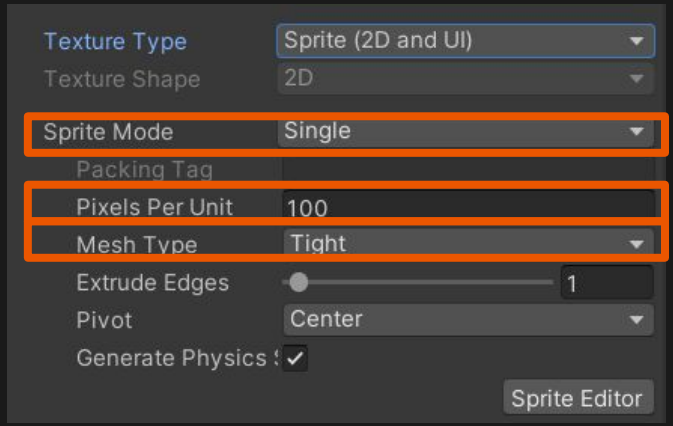
So example of how you can use them, 2D Platformer - Shantae, RogueLite - Binding of Isaac, FPS - Doom Classic, Top Down Farm - Stardew Valley, 2D Fighting -Mortal Kombat, and Point and Click - Paradigm.

Sebastian Grygorczuk - STEM Institute at CCNY

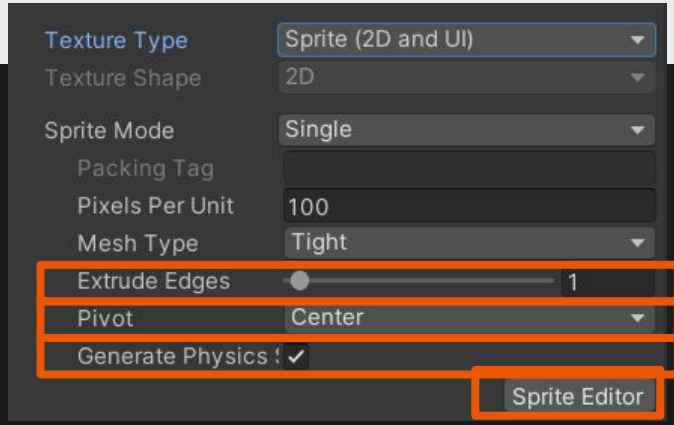
Sprites Import Settings

When working with sprites the import setting on the image has to be set to Sprite, unlike when we were working with Textures and it was set to Default.

With this we can edit how the image will be used in the Scene. Sprite Mode allows us to cut up or keep the image as one, Pixels Per Unit tells us how much space should the image take in respect to the world space. Mesh Type tells us how the Scene should view it as a rectangle the size image width and height or the shape of the Sprite.



Sprites Import Settings



Extruded Edges connect to Mesh type of figure out how much space off of the image should be considered the image.

Pivot is where you want to rotation to occur from and

Generate Physics will build physics based on the Sprites look.

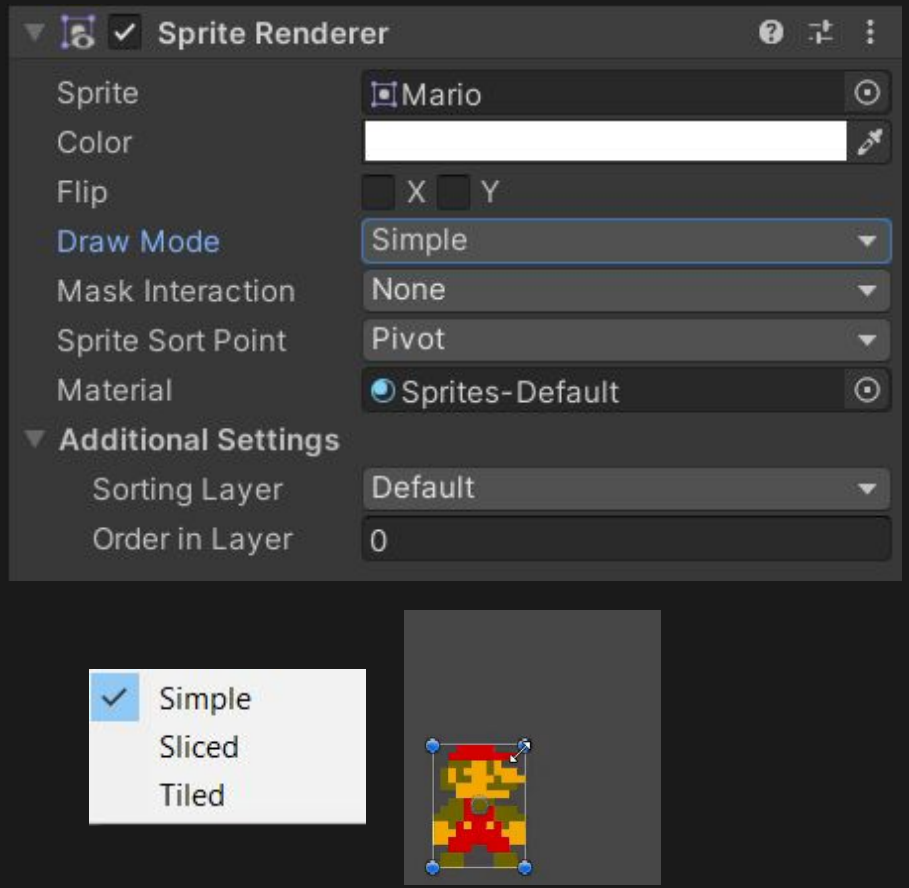
The Sprite Editor Button will bring you to the Sprite Editor View from which you'll be able to edit the 9-Slicing Box that we'll use in a moment.

Sprite Renderer

Sprite Renderer is the component through which we can draw the sprite.

The Sprite controls what Image is draw, like the mesh choosing the shape of the 3D object.

And Draw Mode dictates how it will be drawn and how scaling is dealt with. Simple will stretch and compress the image equally.



Sliced And Tiled

For Sliced and Tiled it's important to know that they are controlled by the 9 Slices box in the Sprite Editor while the Sprite is set for Single Sprite Mode.

Initially the box is filled out so that the center covers all of the image, however you can edit how it's framed such as making the center Marios body and that being are being affected by the Slice and Tiled Effects.

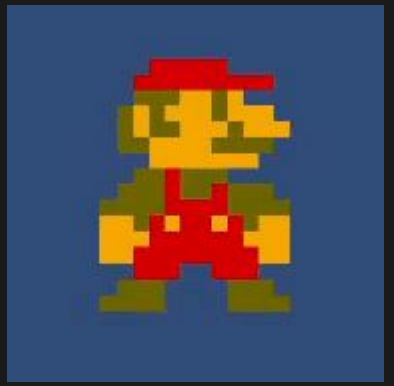
Sebastian Grygorczuk - STEM Institute at CCNY

Materials - Shaders

As you can see the Sprite Render uses a Material to draw you could swap in one of the Materials we've worked on but you will notice that the Sprite is distorted.

That's because the Material is meant for meshes not sprites. To give our sprite different ways of display you need to create a Shader.

Shaders are Scripts that calculator how the image should be drawn, in there you can specify that the material should cling to the Sprite while adding many material effects.

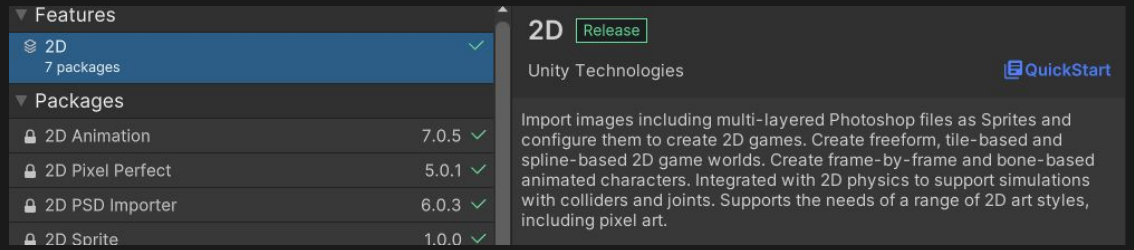


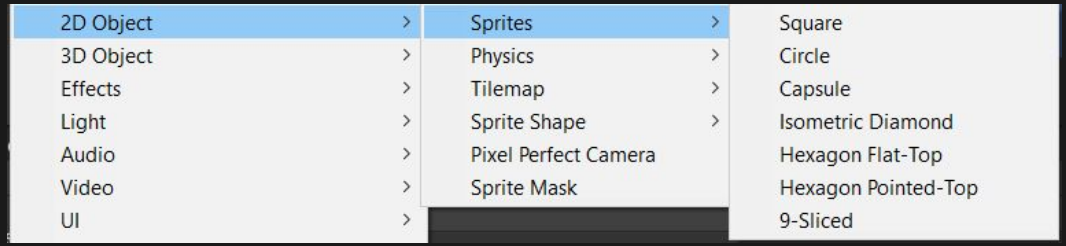
[Get started with 2D Shader Graph in Unity - Dissolve Tutorial]

2D Sprites Game Object

When Unity is created in 2D set up it also allows us to create premade 2D Game Objects, these weren't available to us in 3D beauce the 2D Package Manager isn't include in the set up.

You can always add this to the 3D projects but searching 2D in the Package Manager.





Sprite Sheet

Having manage multiple of the same image can be headache. Many animation take several images to give the appearance of a character moving. For that we have Sprites Sheets. They're one image that has all of the desired sprites that we can divide up and use as it they we all individual image files.

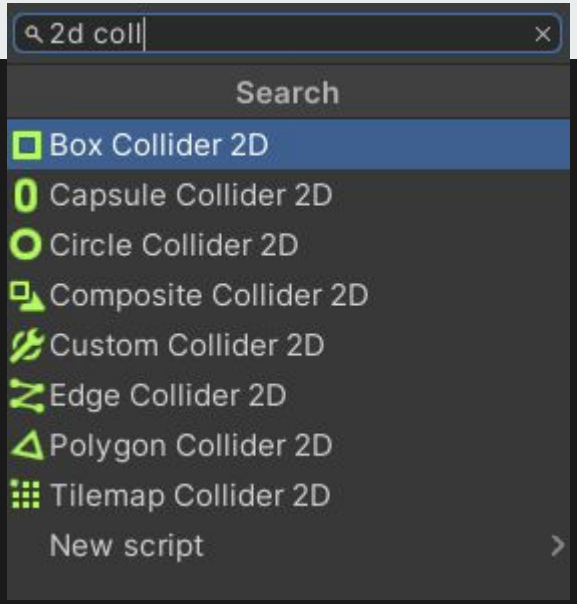
Sprite Editor

When you set your Image Import Setting of Sprite mode to Multiple and endter the Sprite Editor you will be able to slice the image into individual images.

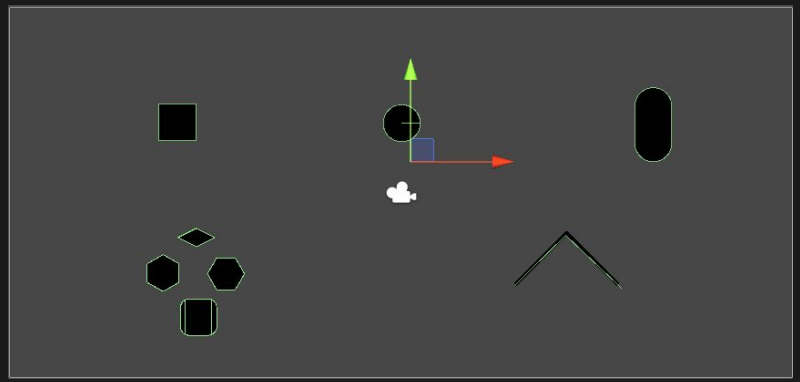
You can do this by either Rubber Banding around the image or you can select Slice which will give you several ways to cut up the image. Automatic, by how big each cell should be or by how many cells you want in total.

Once you've separated all the images you will see that now you can see all of the individual images in the Project View.

2D Collision



2D Colliders work very similarly to their 3D counterparts. With three basic one Box, Capsule and Circle with addition to Edge and Polygon collider that have exclusive properties.



Additional Resources: **Collider 2D - Official Unity**

Sebastian Grygorczuk - STEM Institute at CCNY

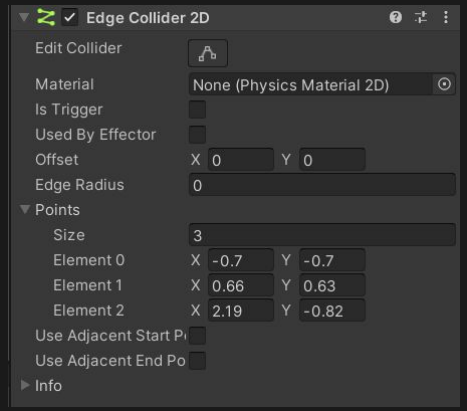
2D Edge and Polygon Colliders

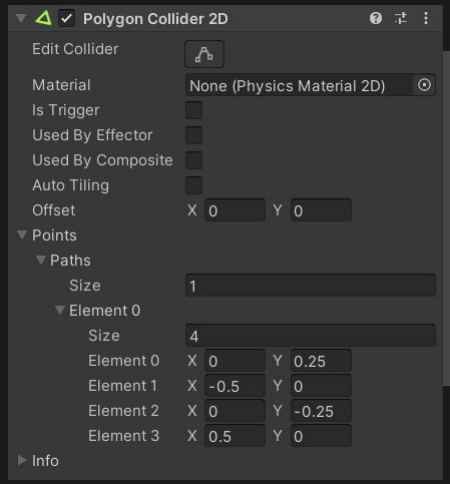
The two new interesting colliders that we have at our disposal are the Edge and Polygon Colliders.

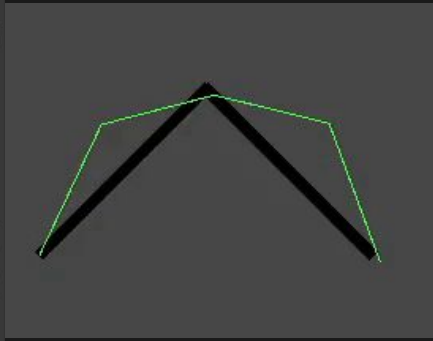
They share the same trait that you can just draw desired shape with main difference being that Edge Colliders can be left open while Polygon have to create a loop where the last point and first point connect.

Adding a Polygon Collider to a given sprite will auto generate a composite that stimulates its shape.

You can edit or add the vertices when Edit Collider Mode is toggles, and if you hold Left Ctrl you can delete the edges.

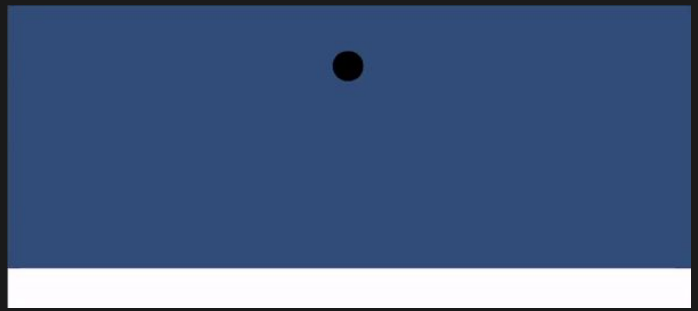






2D Rigidbody

2D Rigidbody work very similarly to the 3D one. You may have notice that unlike 3D Rigidbody the component doesn't have a toggle next to the name, this one can be controlled using the Sleeping Mode.



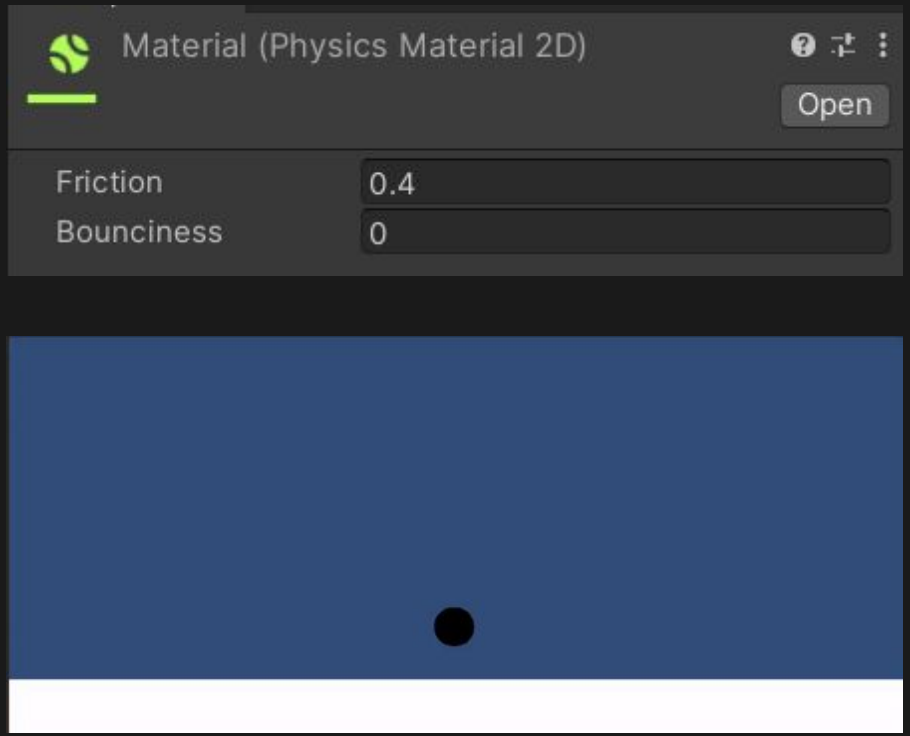
Additional Resources: [Rigidbody 2D - Official Unity Tutorial]

Sebastian Grygorczuk - STEM Institute at CCNY

2D Physics Material

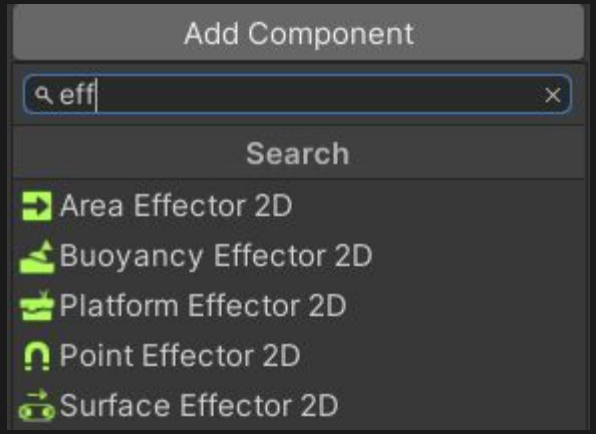
2D Object just like 3D have Physics Materials you can attach to them, work the same way with less options.

Just friction and bounciness you can attach to the object.



2D Effectors

2D Colliders have a unique connection called Effectors, effectors are additional Physics components that continuously exert a force. There five of them, Surface, Area, Buoyancy, Platform and Point.



Additional Resources: **2D EFFECTORS in Unity - Tutorial**

Sebastian Grygorczuk - STEM Institute at CCNY

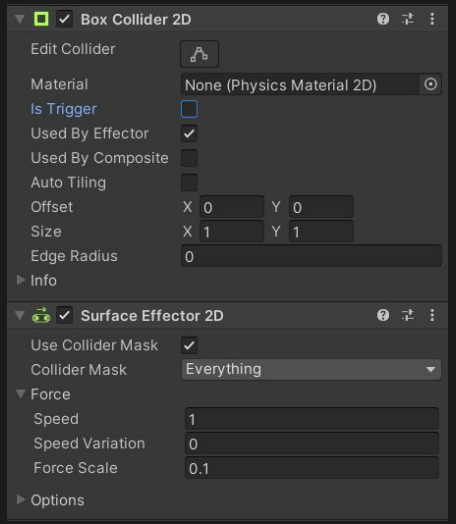
2D Effectors - Surface Effector

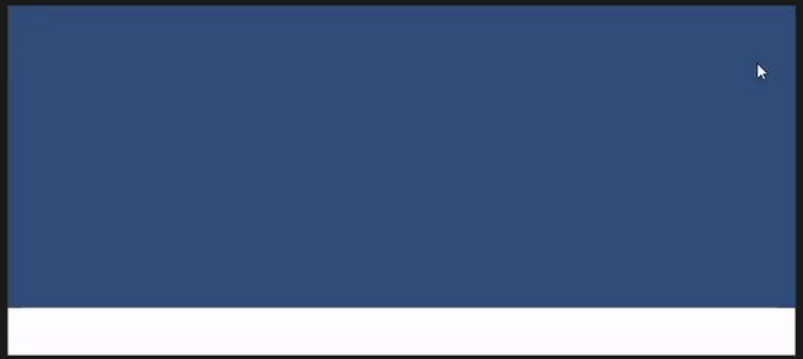
The Surface Effector gives a constant speed in the X axis, this is connected to the white platform and requires only the effect to be toggled on the box collider.

The Speed Variable will send it left or right depending on if the value is negative or positive.

The Force Scale is how fast the object accelerates towards the specivided speed

The Speed Variation is any sudden upticks that occur while accelerating toward given speed.





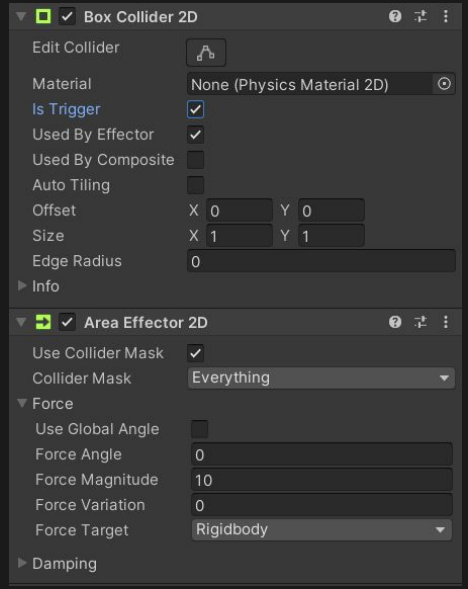
2D Effectors - Area Effector

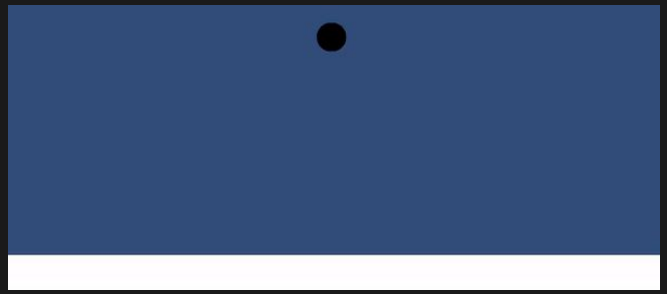
The Area Collider will apply a force force in a give direction to any object that enters it. This requires the Box Collider to have both the Trigger and Effector Toggled.

The Force Angle is which way the object will be sent, with 0 being X axis.

The Force Mangicture is how much force is put into the object.

And Force Variation is how much can be randomly added to the force. if negative how much randomly will be taken away.



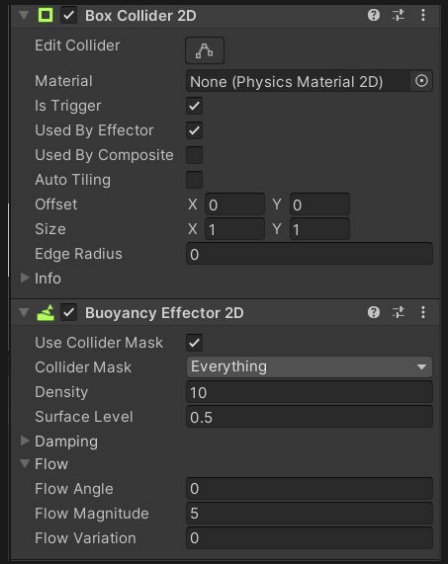


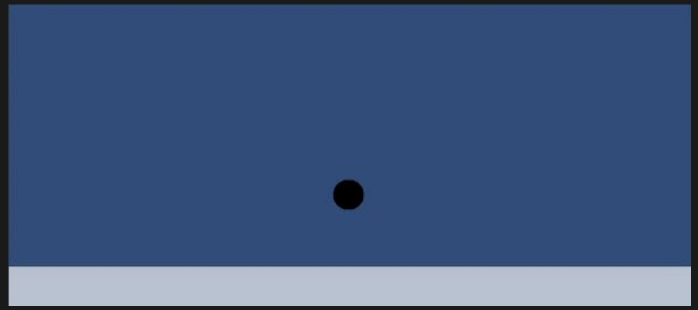
2D Effectors - Buoyancy Effector

The Buoyancy Effector is great at simulating a bodies of water. It requires the Box Collider to have Trigger and Effector on.

Density determines how if the object that lands in it will sink or float. While Surface Level will determine how much of the object will stay above water when floating.

It also has Flow menu that use the same ideas as the Surface Effector adding force to left or right of the collider to push the object.





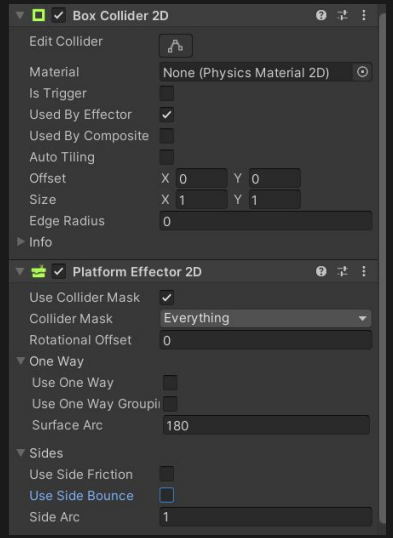
Sebastian Grygorczuk - STEM Institute at CCNY

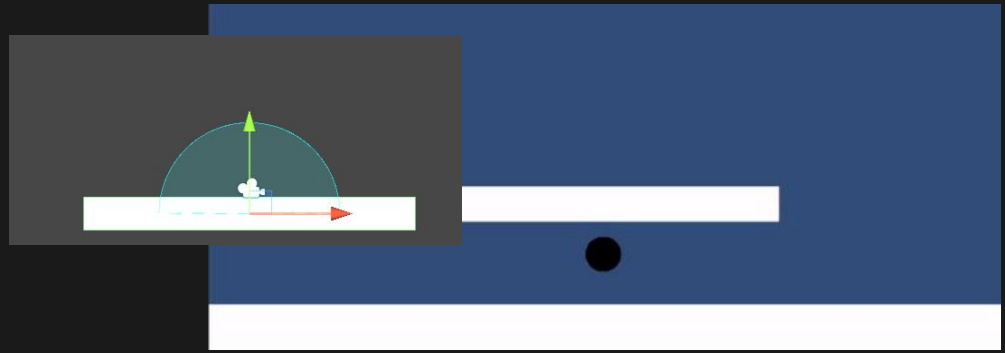
2D Effectors - Platform Effector

Platform Effector or One Way Platform allows us to create the effect where the play can jump through one side of the platform while keeping the other one solid.

Rotation Offset set which direction it should be facing. Surface Arc determines which area should count as solid surface.

Sides allow you to control what the behavior should be when something collides with the platforms side.





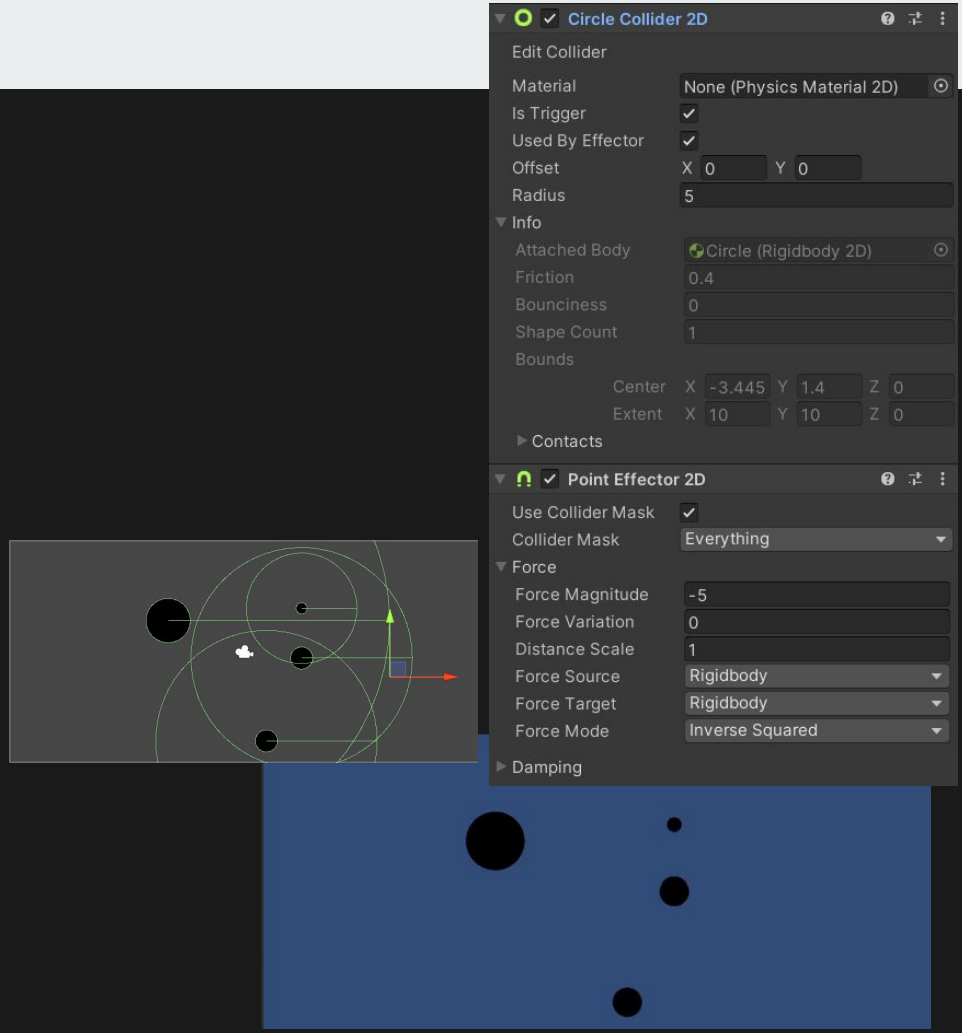
2D Effectors - Point Effector

Point Effector is very different from others, it create attraction or repulsion force between objects.

Force Magnitude tell us how strong the pull/push is.

And Force Mode allows us to determine at what rate should the force be applied.

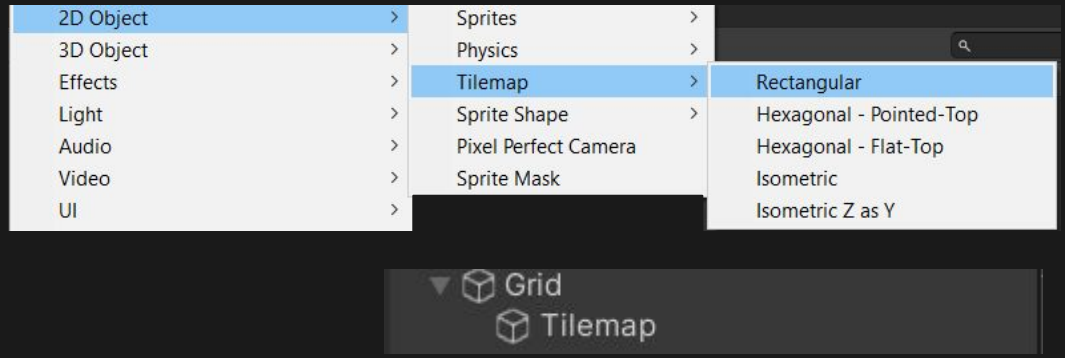
As you see the circles have two colliders, on for actually colluding and a larger one for the effector to give range for the force to act upon.



What is a Tilemap

Tilemap is a map draw from tiles. We take small sprite and line them up together to create the world of the game.

When creating a tilemap you will also instantly create a Grid that the tile map will be a child to. Everything that's draw on the tilemap will link to the Grid.

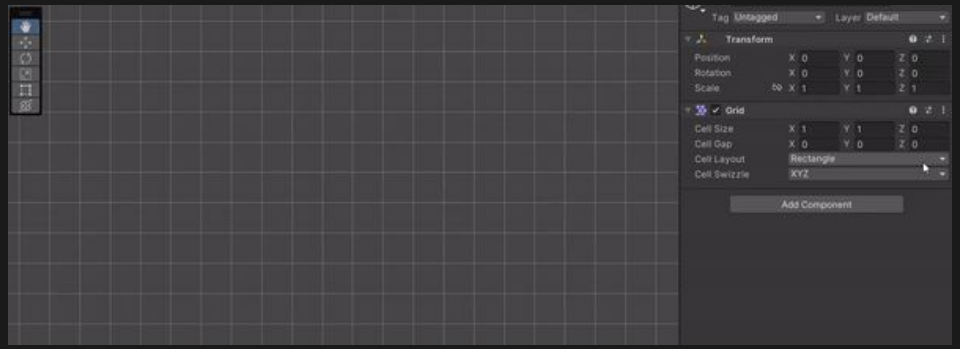


The Gird

The Grid can be configured in three ways, Rectuagl, Hexagonal and Isometric.

Depending on the style of game you should choose appropriate grid type. The main three being Rectangular, Hexagonal, and Isometric.

You also can control the size of each cell in the grid and if there should be any spacing between the tiles.



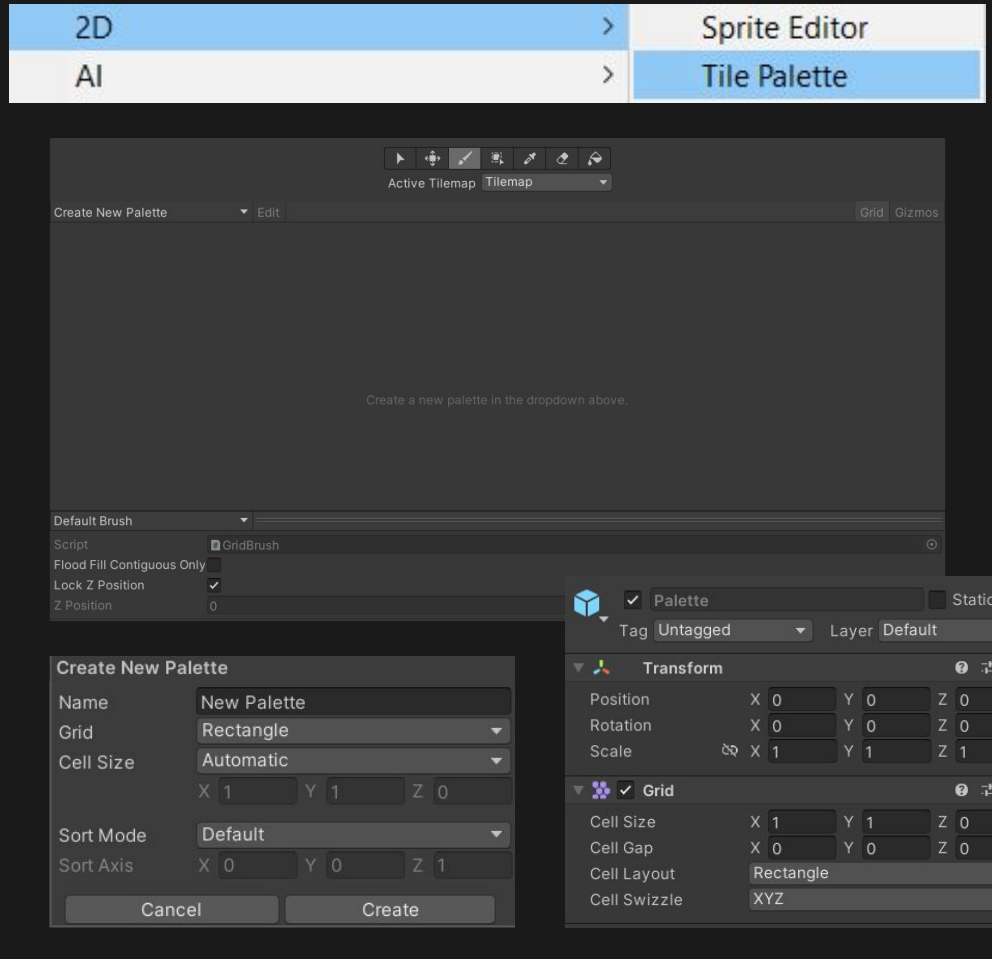
The Palette

To draw on the tilemap you will first have to open up the Tile Palette View. Go to Window -> 2D -> Tile Pallet.

This give you the control screen that will allow you to crate a pallet.

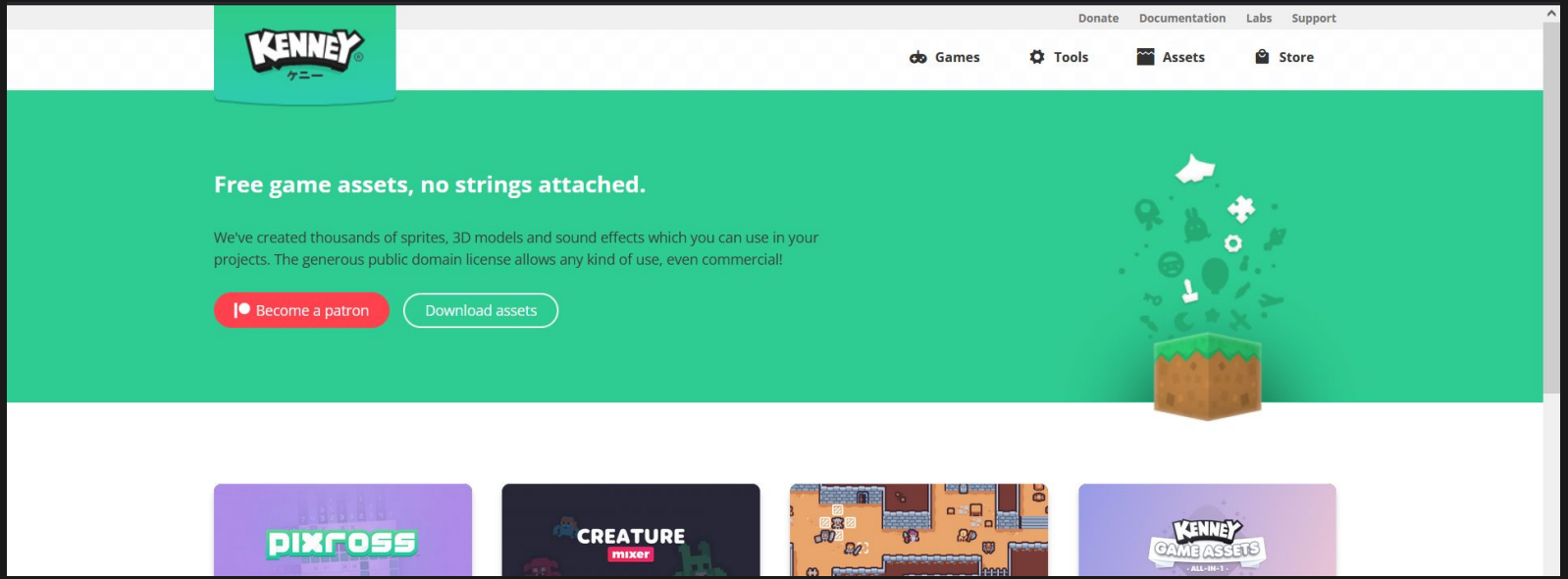
Choose appropriate dimensions for the pallet that work with the gird you created.

This will create a Palette Prefab that will have same components as the grid.



Sebastian Grygorczuk - STEM Institute at CCNY

Kenney

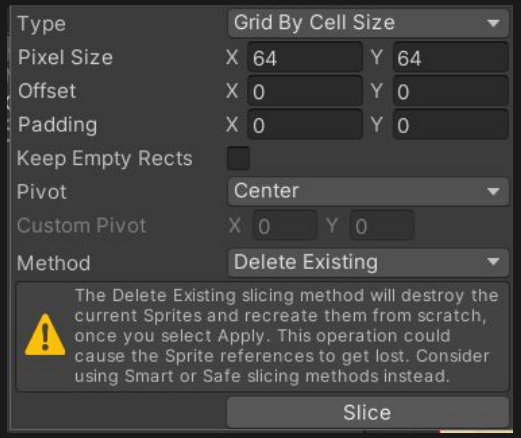


Kenney is a game asset creation group from which you can get assets for your 2D and 3D games.

Feel free to explore the website: https://www.kenney.nl/

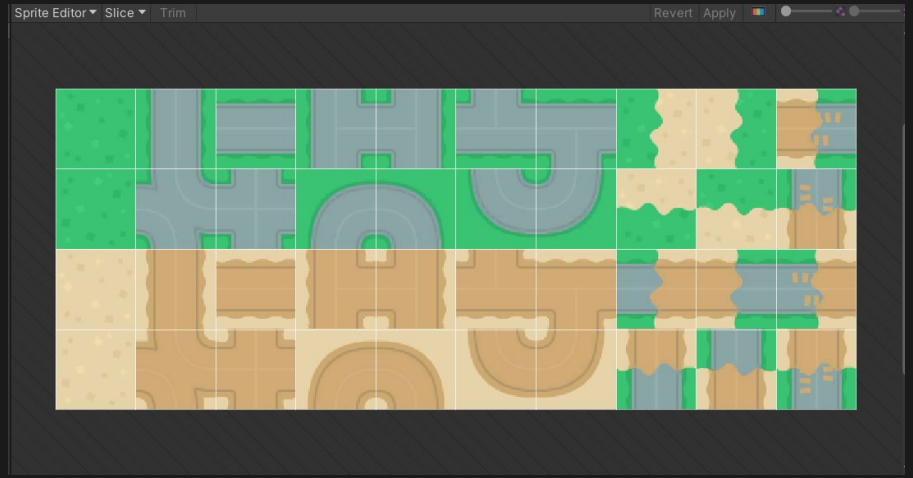
We're going to be using Top-down Tanks Redux Asset Set for tiling

Tile Sheet



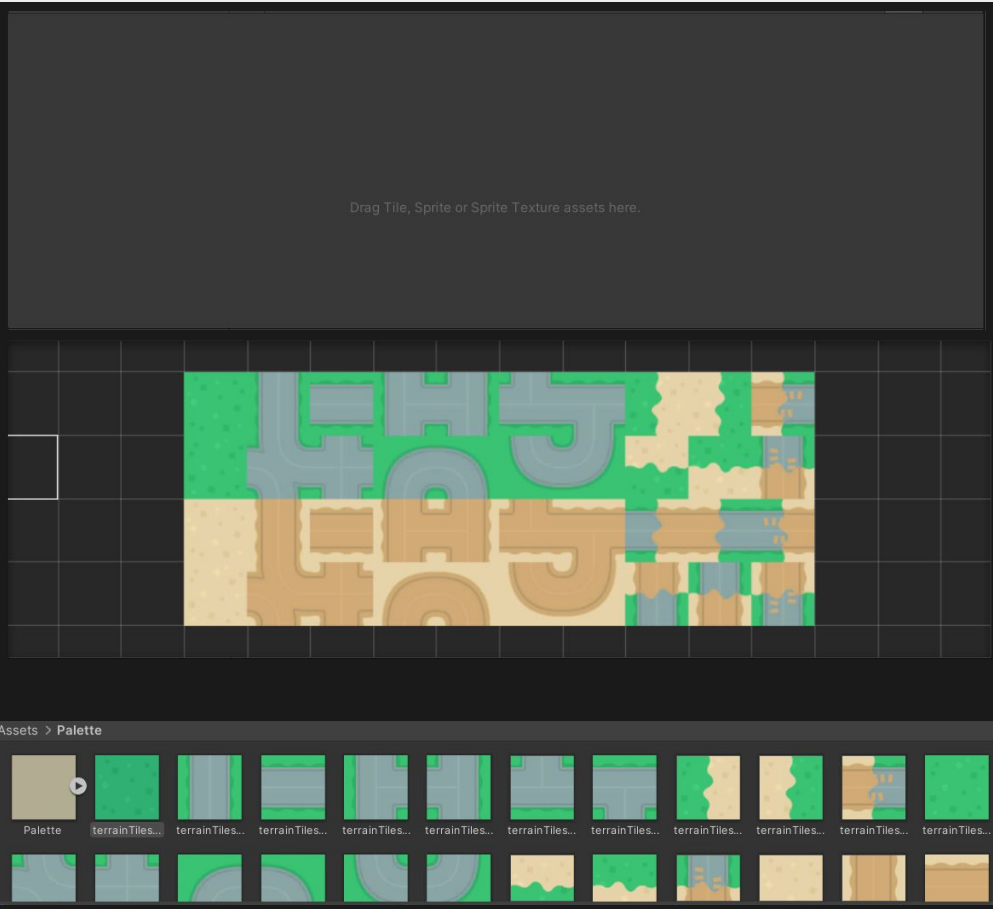
Once you have the palette set up you will need the Tile Sheet, the sprite that you will input into the Palette.

To make sure your image works correctly first make sure that the Pixel Per Unit is equal to the size of each cell, in this case the cells are 64x64.



Tile Sheet

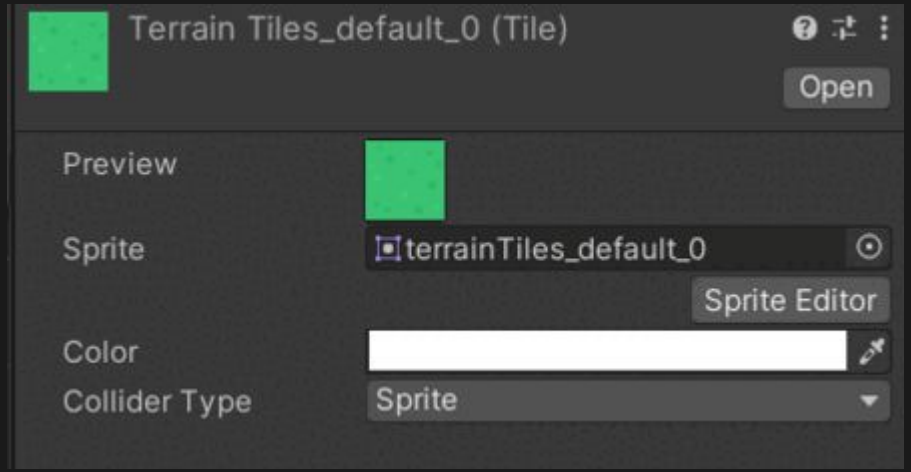
Once your Sprite is correctly configured you can drag the parent sprite to the Pallet window and it will create Tile Assets. Have a folder ready as each cut out will have its own asset created.

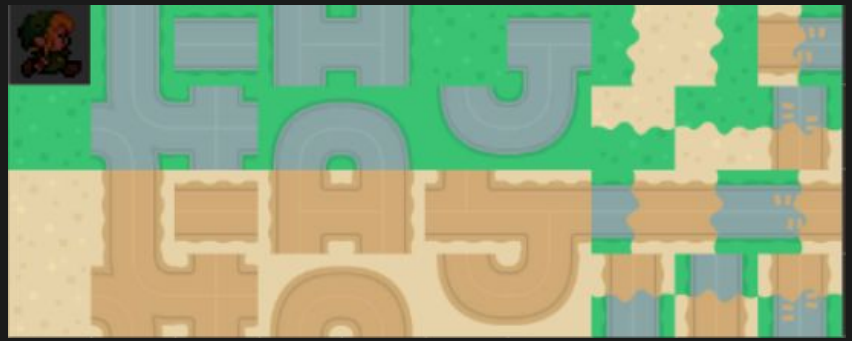


Tile Asset

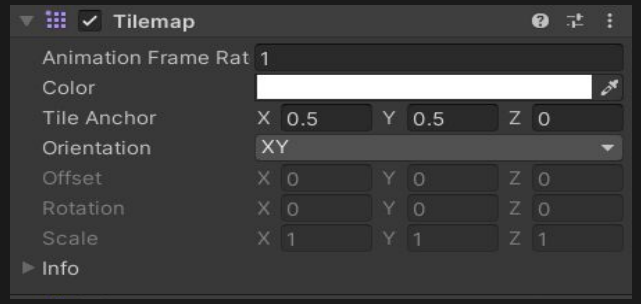
The Tile Asset gives you control over changing what sprite is used and in what color it should be rendered.

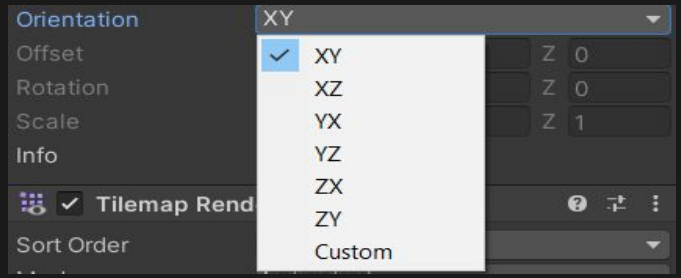
Any changes you make will be immediately reflected in the pallet.





Tilemap Components

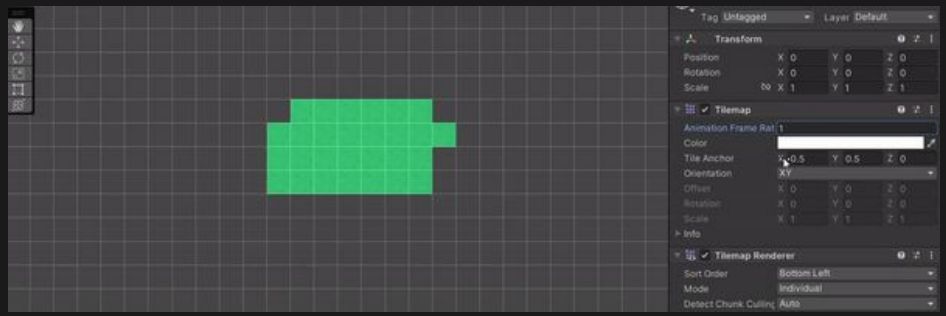




A Tilemap always comes with two components, Tilemap and Tilemap Render.

Tilemap acts as an internal Transform for the tiles draw on, with few rendering actions such as color and Animation Frame Rate, which we won't bother with at the moment.

You also have the power to change the plane the tiles are drawn on using Orientation.



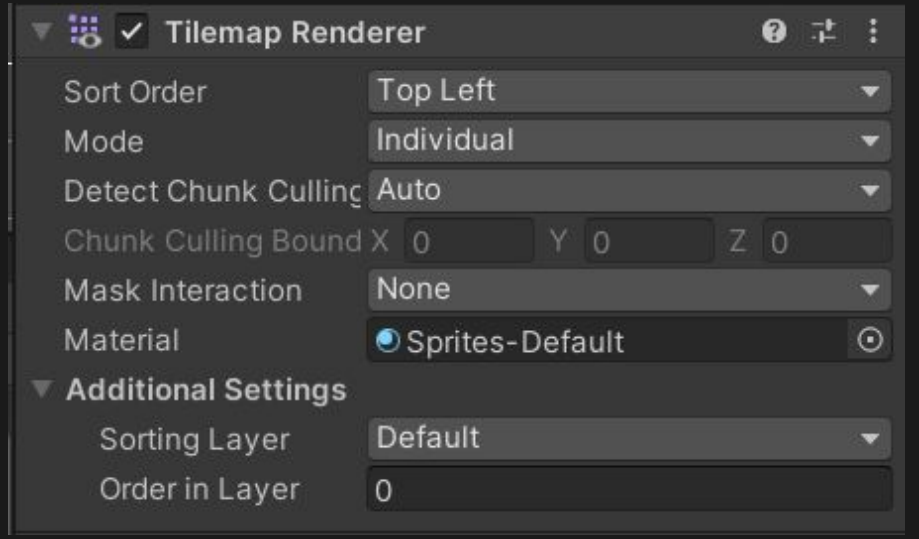
Additional Resources: **Tilemap**

Sebastian Grygorczuk - STEM Institute at CCNY

Tilemap Renderer

Tile Renderer controls what get drawn and how. Sort Order tells us which Tiles will be drawn first, so if Top Left is Sort order that's where the drawing starts. Mode tells us how should they be draw, each tile on its own or in chunks. And Detect Chunk Culling will tell Unity how close the player has to be for it to be drawn. Think of it as Billboarding in 3D.

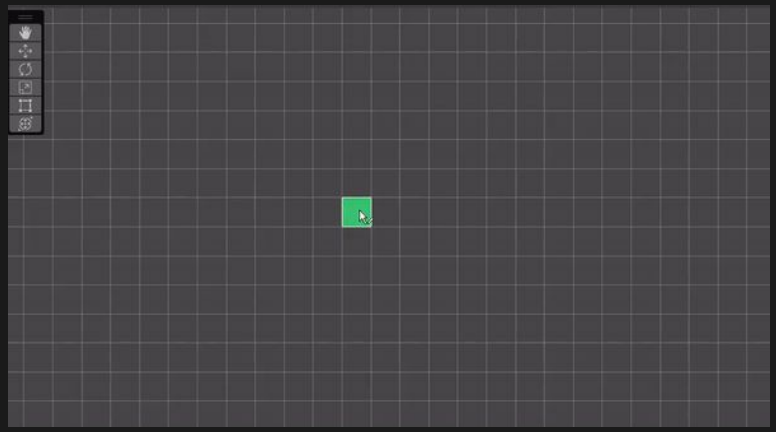
In addition to that it has our familiar Sprite controls of material, masking and layer sorting.

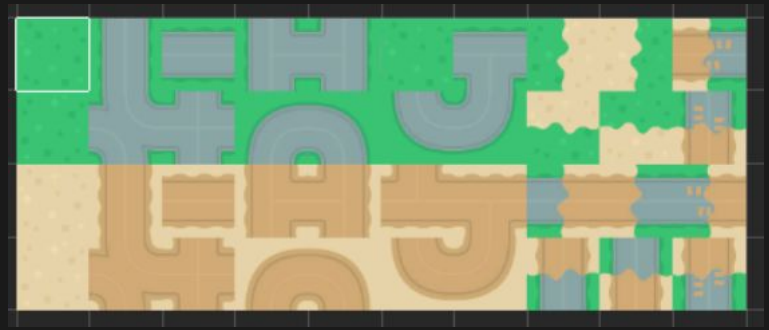


Painting

Now that we're all set up select a tile and use the Brush Tool to start painting your map.

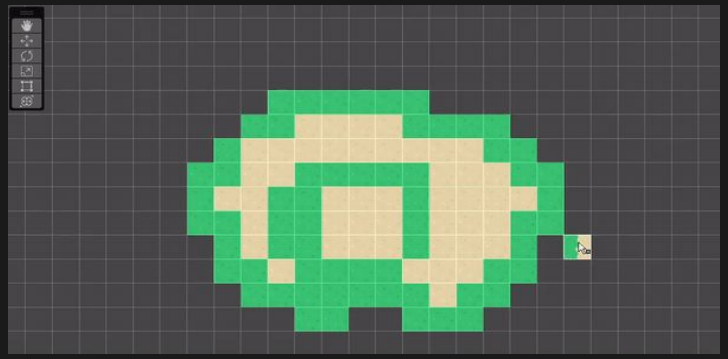
You can also erase any tiles drawn by either selecting the Eraser Tool or holding Left Shift while painting.



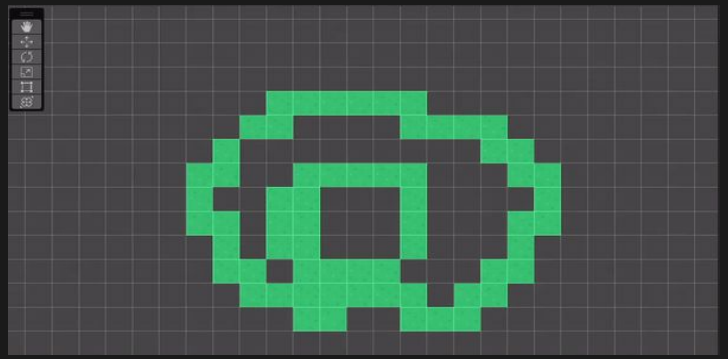


Rectangle and Fill Tool

At your disposal you also have the Rectangle Tool which will copy the same Tile or Tiled Area of the Rectangle Spot you create.



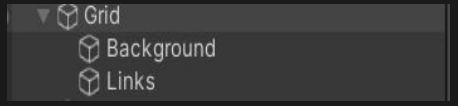
And the Fill Bucket which will paint all of the tiles that color as long as there's a space connecting them.

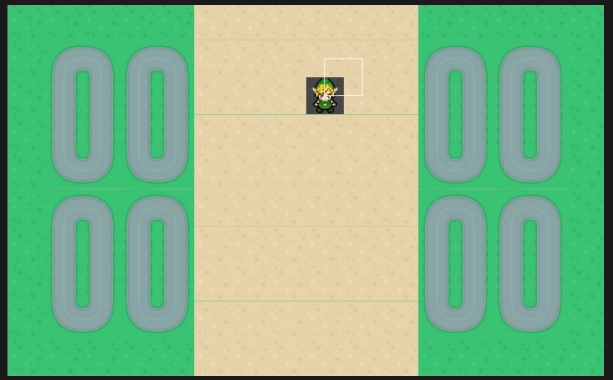


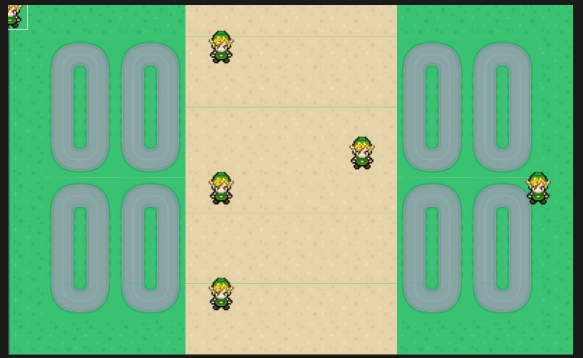
Layering Maps

If you try to place a tile on top of something you've already draw on and that tile has nothing draw on it the emptiness will be part of the tile.

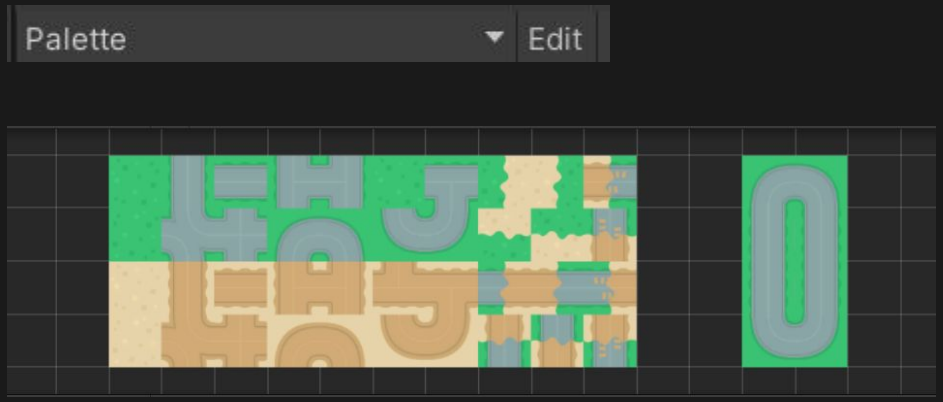
To fix this you will have to create a new tilemap who's layer will be rendered on top.





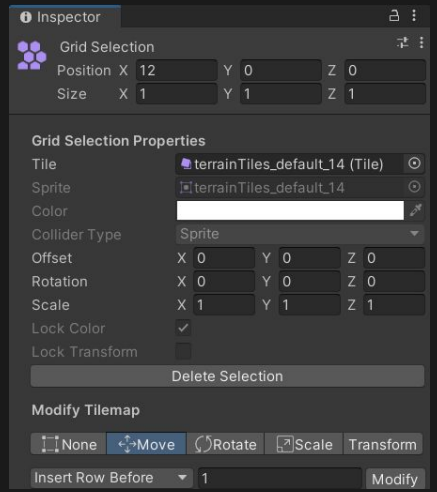


Editing Tile Sheet



Your pallet might give you a lot of option on the grid but it can take time to create complicated areas by placing the tiles one by one.

Thankfully you can create template for areas by going into the Edit Mode and painting in the pallet.

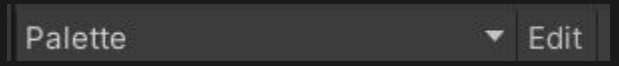


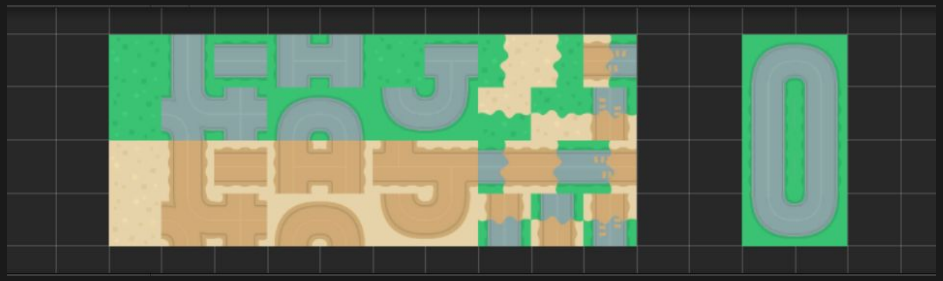
Sebastian Grygorczuk - STEM Institute at CCNY

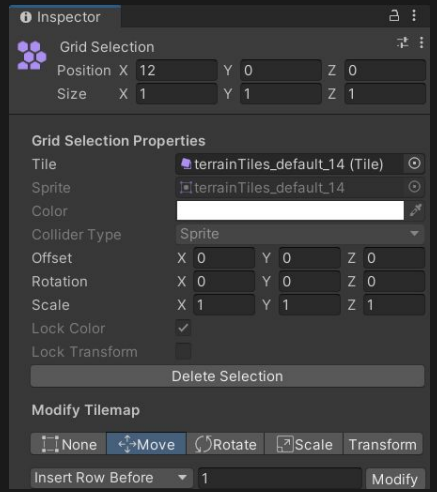
Editing Tile Sheet

Once you're in Edit mode you can use the Eyedropper Tool to copy or modify the selected tile. Once you have one Selected you can use the Brush Tool to paint in the palette.

And to edit the variables of the tile you can use the Select Tool for the tile giving the Inspector view controls to rotate, translate and scale the tiles.







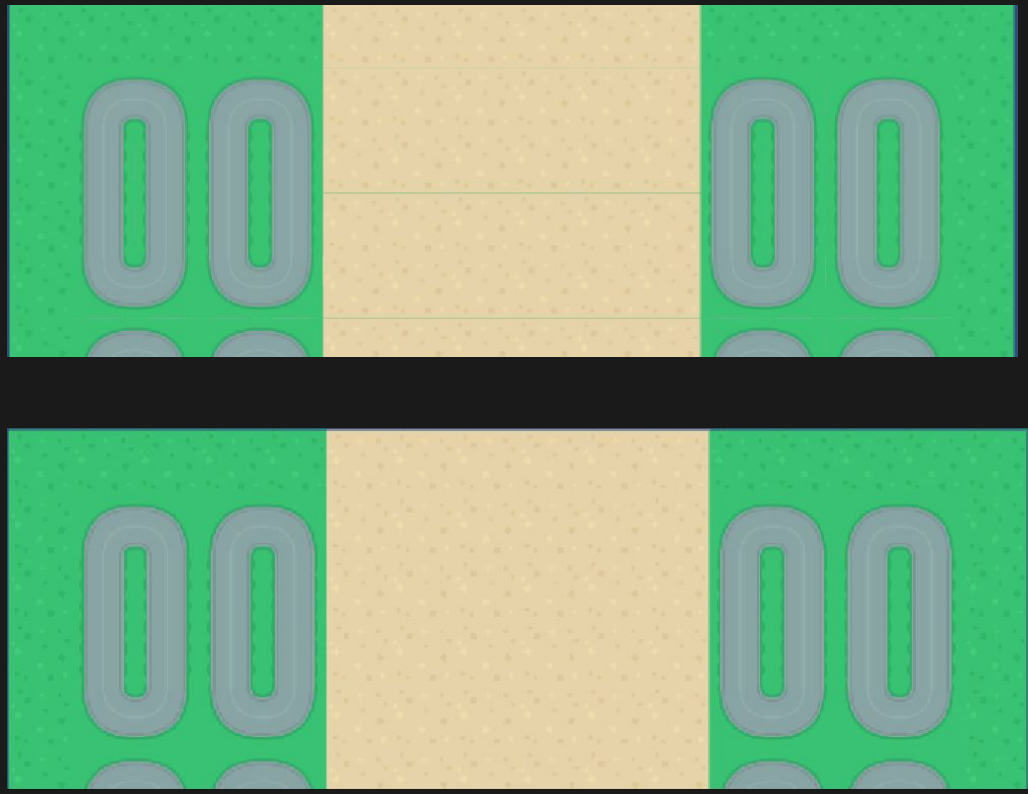
Sebastian Grygorczuk - STEM Institute at CCNY

Tearing Between Tiles

You may have noticed that some of the tiles have gaps between them.

The reason behind this is that each sprite is called from the Asset Folder individual, [they may all have been connected to the big Sprite but when we split them they became individual Sprites] and the amount of work it takes for the Computer to grab the individual images and draw them on screen it creates small tears.

Sprite Atlas fixes this by connecting all of the images into one big image that will be referenced when drawing any sprite.

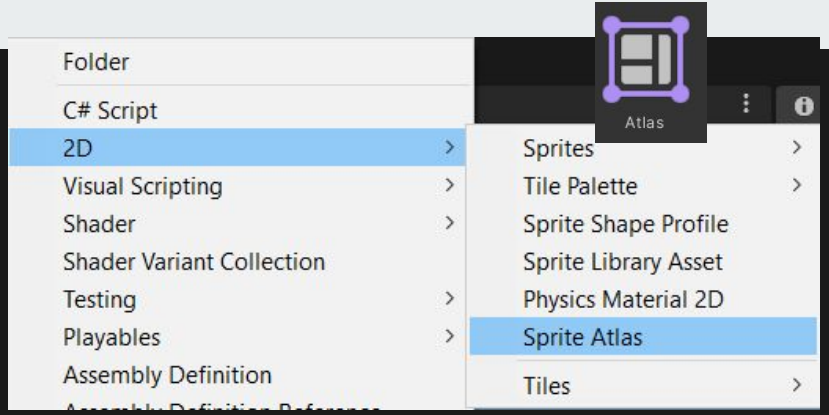


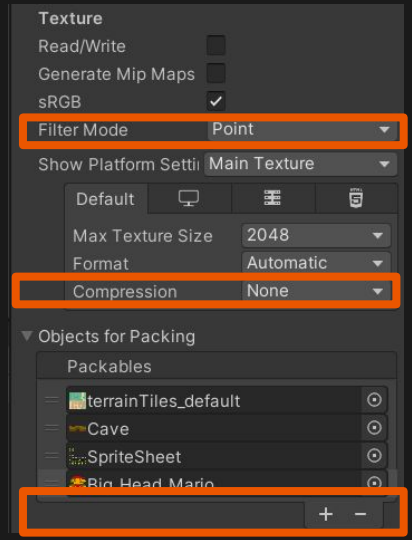
Sprite Atlas

Sprite Atlas is Game Asset, so you can create it by creating a new Asset in the Project View and going to the 2D section.

In the Atlas Settings you want to make sure the Filter Mode is set to Point that way the textures keep to their pixels when scaled and you want to set Compression to None so nothing be becomes blurry.

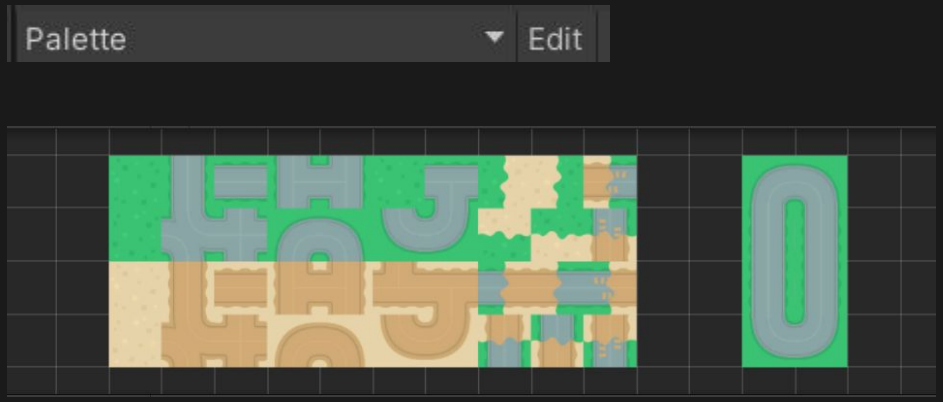
After that you can add the Sprites you want to be part of the Atlas and it will automatically generate a new Texture that Unity will be able to directly use.





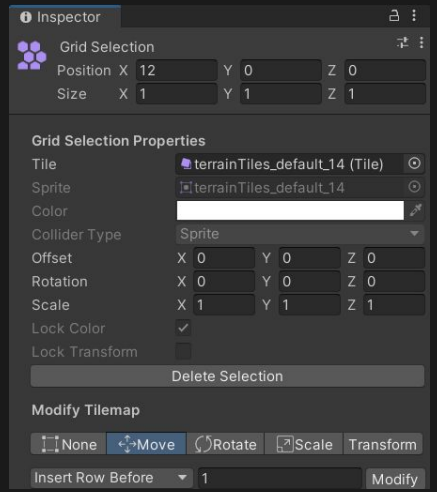
Sebastian Grygorczuk - STEM Institute at CCNY

Editing Tile Sheet



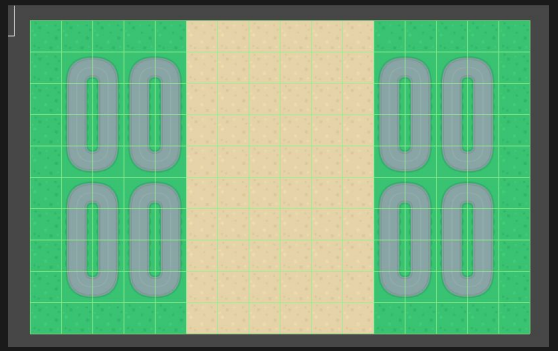
Your pallet might give you a lot of option on the grid but it can take time to create complicated areas by placing the tiles one by one.

Thankfully you can create template for areas by going into the Edit Mode and painting in the pallet.



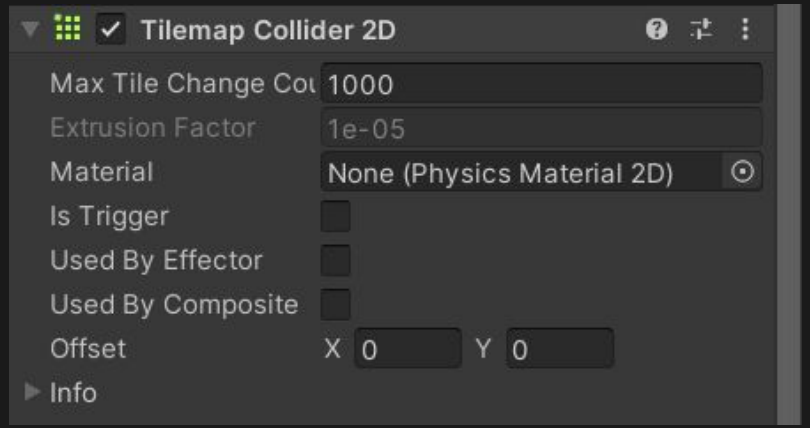
Sebastian Grygorczuk - STEM Institute at CCNY

Tilemap Collision



Tilemap have a special Collider that you can attach to it allowing you to create platform or pathways for the player to interact with.

Add the Tilemap Collider 2D component is a start but for it to fully work it does require a Rigidbody 2D and since we don't want it to move in any way leaving it as Static is enough.



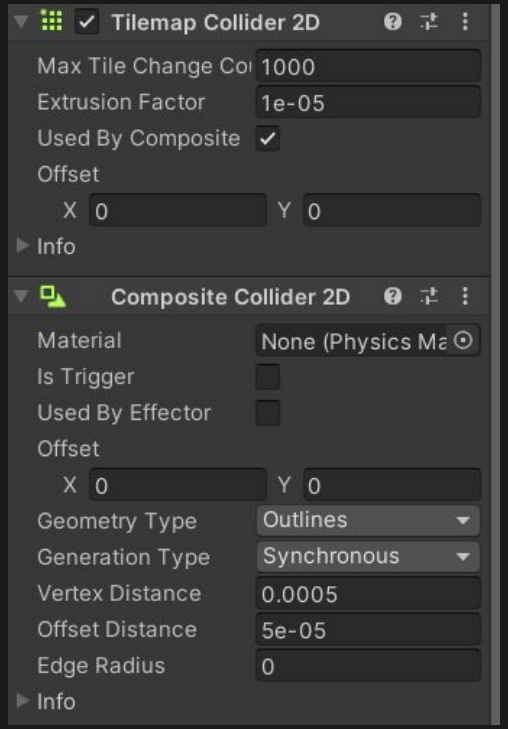
Composite Collider

When you create a Tilemap collider you'll notice that each tile gets its own collision box.

Similar to how drawing individual tiles isn't the best simulating all of the collision boxes isn't either.

So you are making one large area it would be best to use a Composite Collider that will combine all of the nearby colliders into one.

This can be done to a collation of 2D Box Colliders and Polygon Colliders as well.

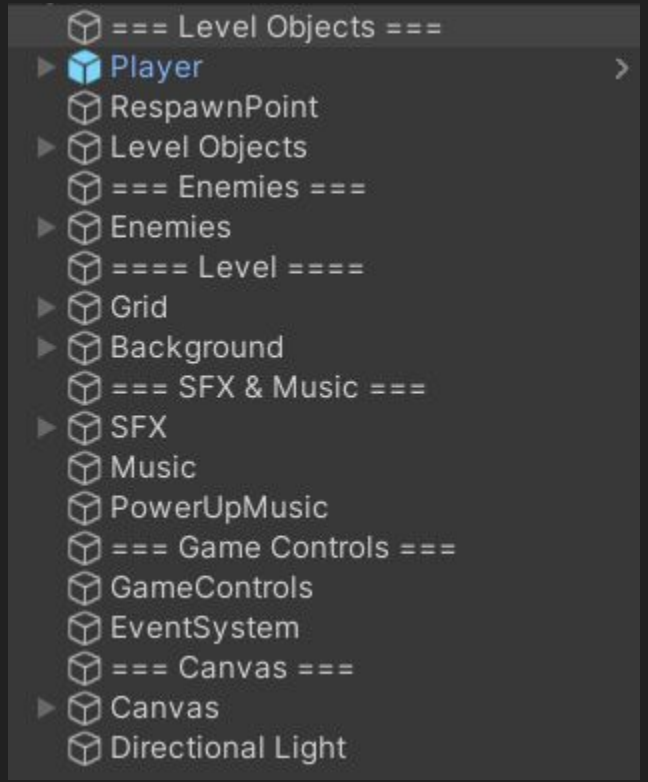


Level

Making the level

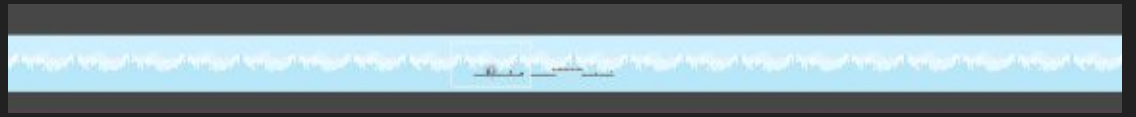
The level is much more complicated but we will break this down into five sections.

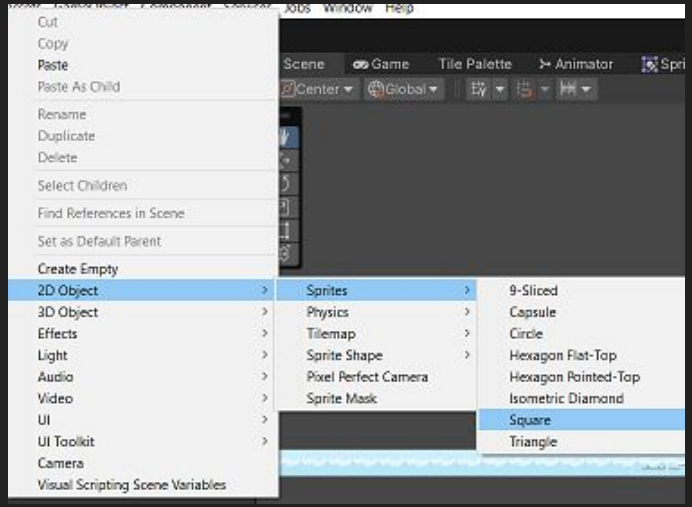
1) Level
2) Player
3) UI
4) Level Objects
5) Music and SFX

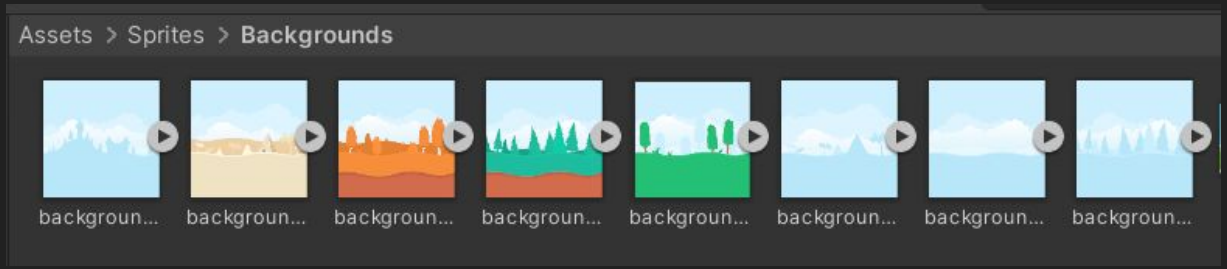


Level Background

We are going to take one of the background images, preferably the faded ones, and tile them across the scene so we have a never-ending background. Start by creating a 2D Sprite Square.

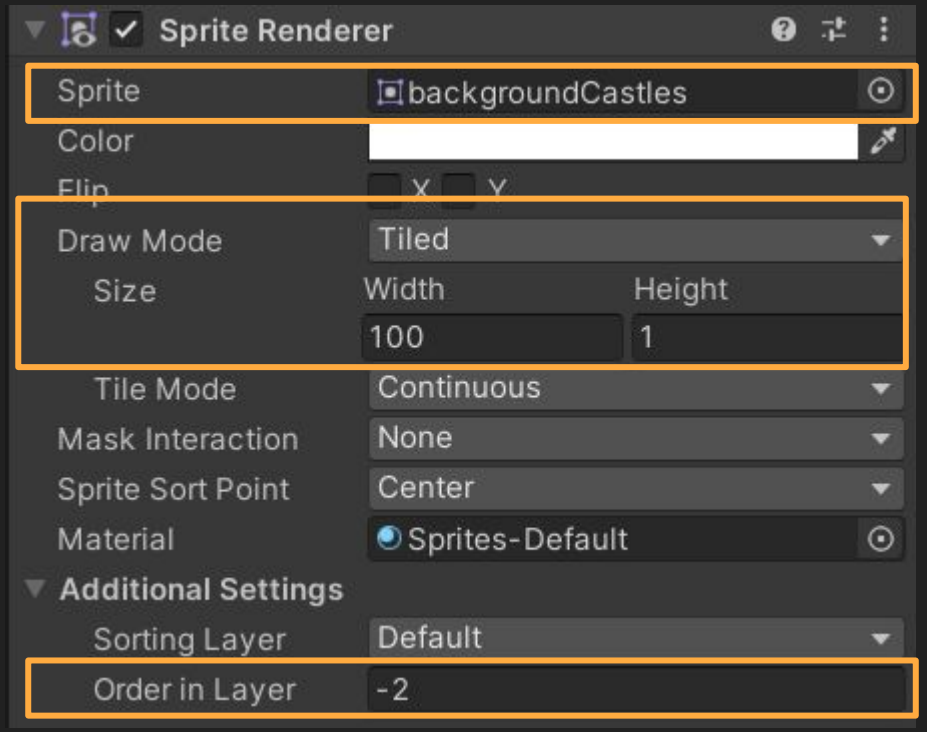






Level Background

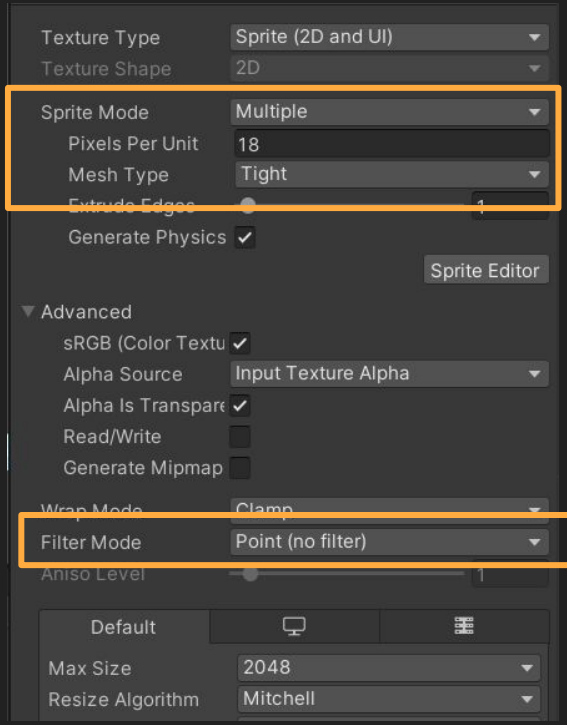
Set the Draw Mode to be tiled, and then make the Width large, like 100. This will tile the image so it's a long level. Also, make sure that the order in layers is low so that it's behind everything else, such as -2.

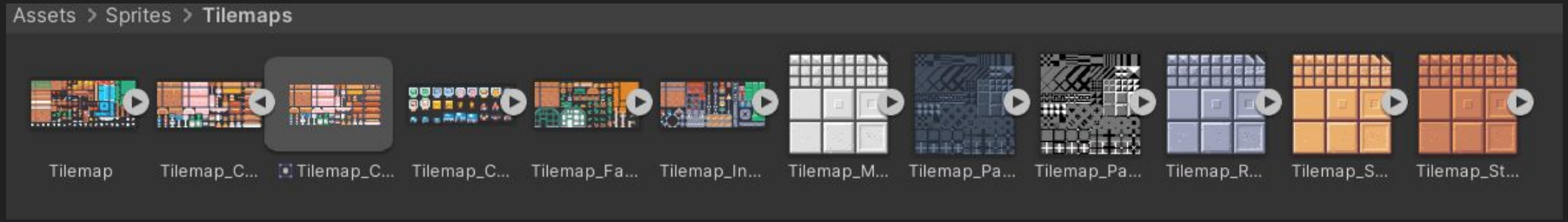


Level Tilemaps

Before we can start making parts of the level, we will have to format the image. In the Tilemaps folder, look for the Tilemap sprite. You will want to set the Sprite Mode to Multiple, the Pixels Per Unit to be 18, and the Filter Mode to be Point.

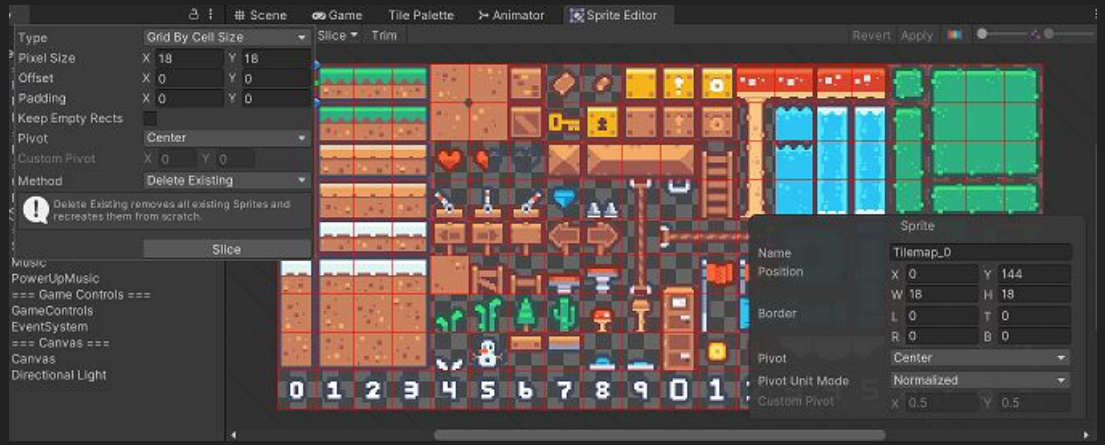
Once all of that is ready, go into the Sprite Editor.





Sprite Editor

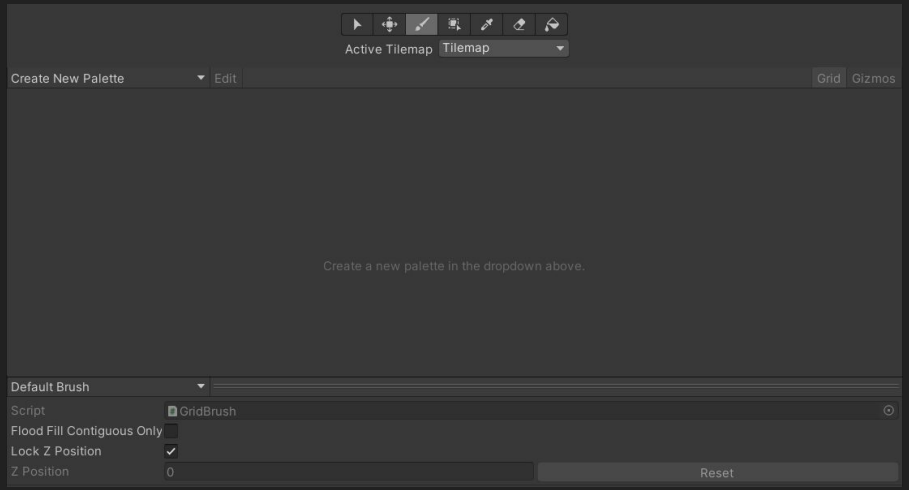
Inside the Sprite Editor, click on "Slice," set it to be Grid By Cell Size, and set the cell size to 18x18. Slice and make sure to apply the changes, and you should have your image cut up and ready to be made into a palette.

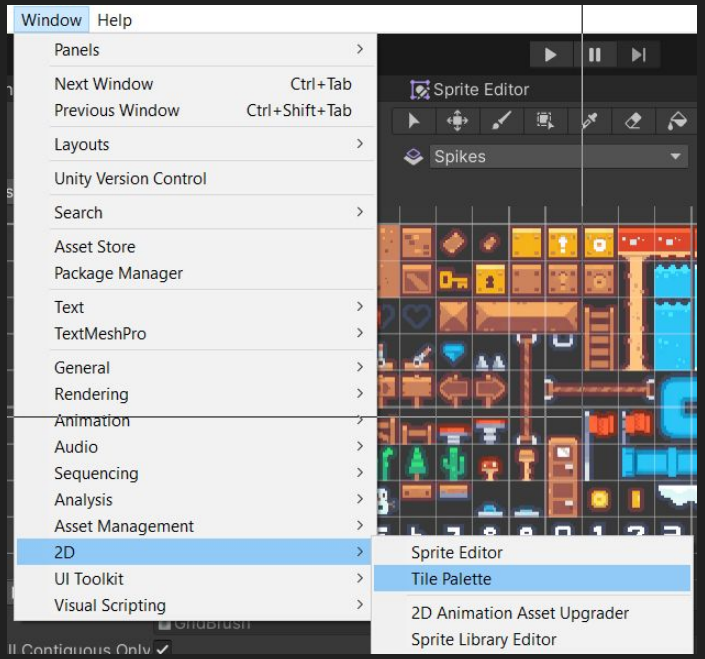


Tile Palette

Go to Windows and open up the Tile Palette.

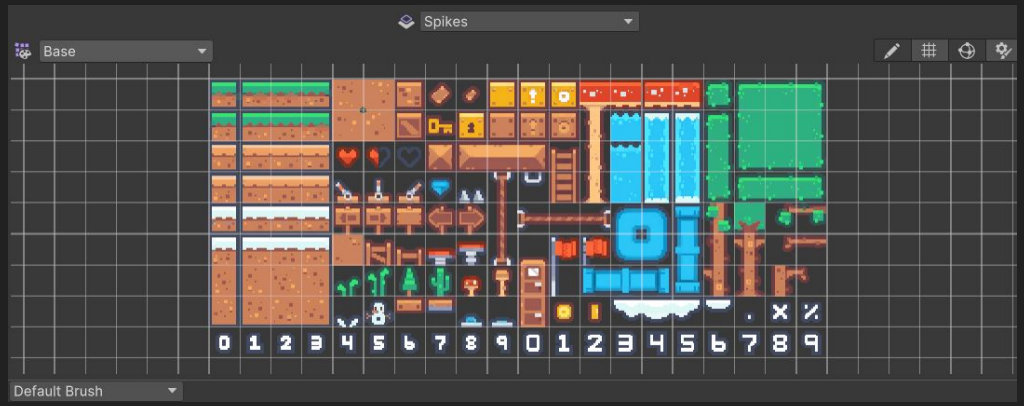
Once you've got it open, create a new Palette and save it to the Palette folder.





Tile Palette

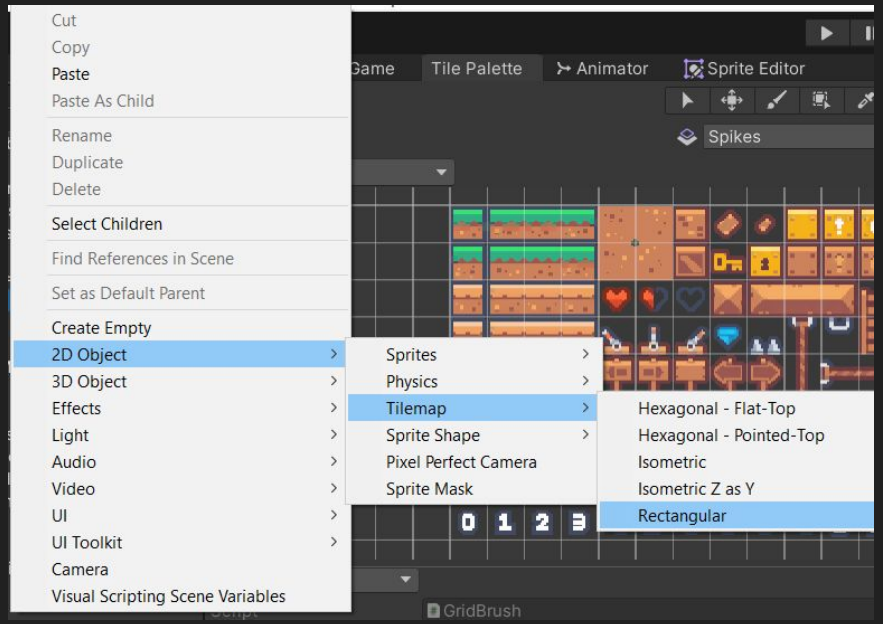
Once you've created the tile palette, drag the tilemap image into the palette, and save the tiles to the Tiles folder.

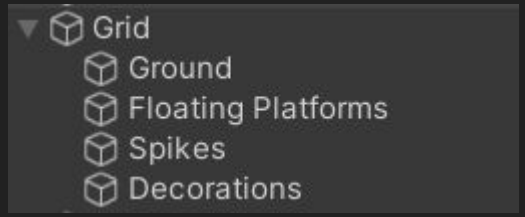


Grids & Tilemaps

Go back to the scene and create four Rectangle Tilemaps.

Call them Ground, Floating Platforms, Spikes, and Decorations. We will go through each one and discuss how they work.

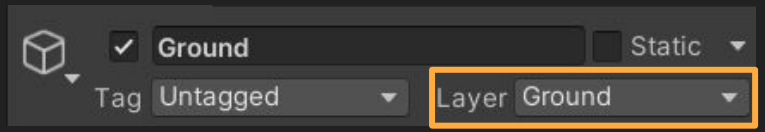


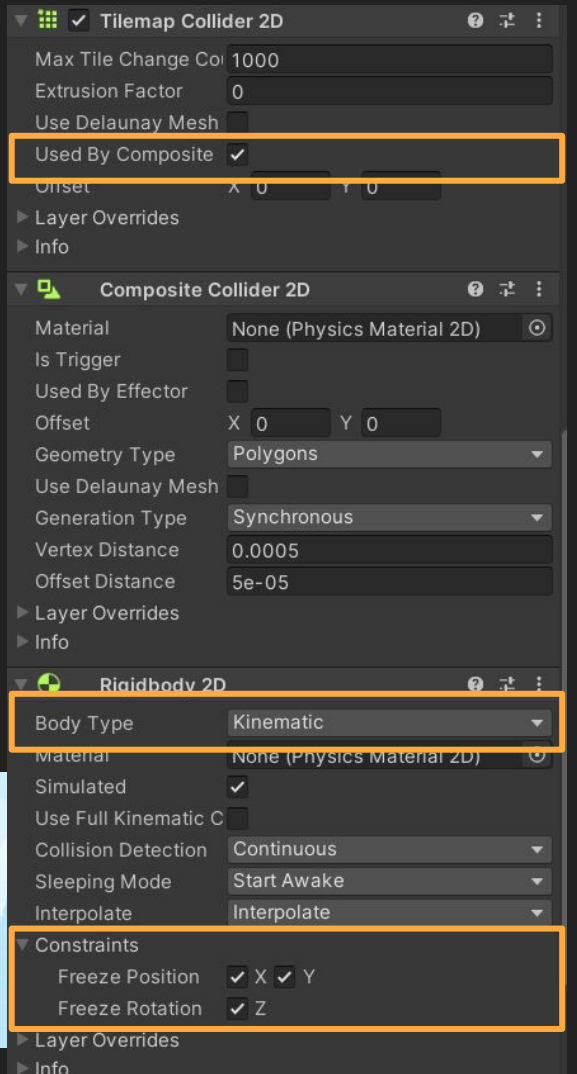


Ground TileMap

The "Ground" will be any surface the player walks on. Therefore, you will need a Tilemap Collider.

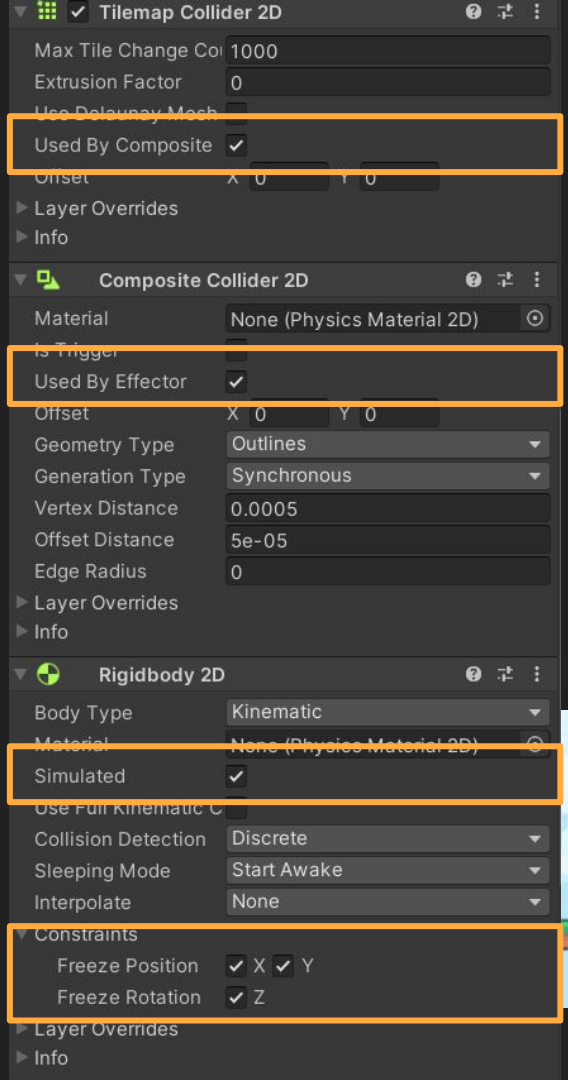
However, the Tilemap Collider creates tiny colliders around each tile, which the player can get stuck on. To address this, add a Composite Collider that combines them into bigger colliders. To make this work, you also need a Rigidbody 2D set to Kinematic Body Type. Lastly, set the Layer to Ground. This will allow the jumping code in the player to work correctly.

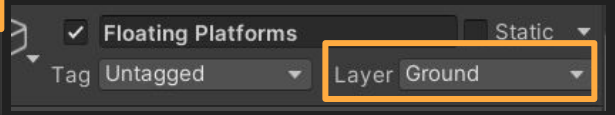


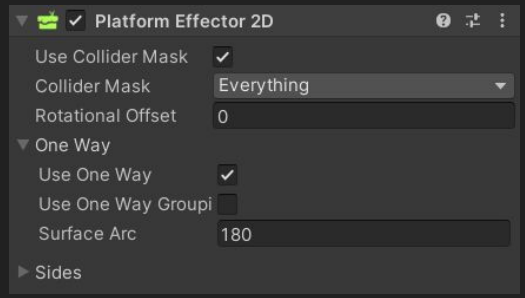


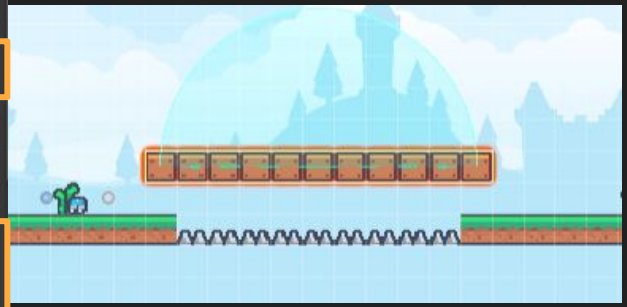
Floating Platform

The "Floating Platform" will have many of the same concepts as the Ground, but with the addition of a Platform Effector. This component will allow the player to jump through the platform from below.





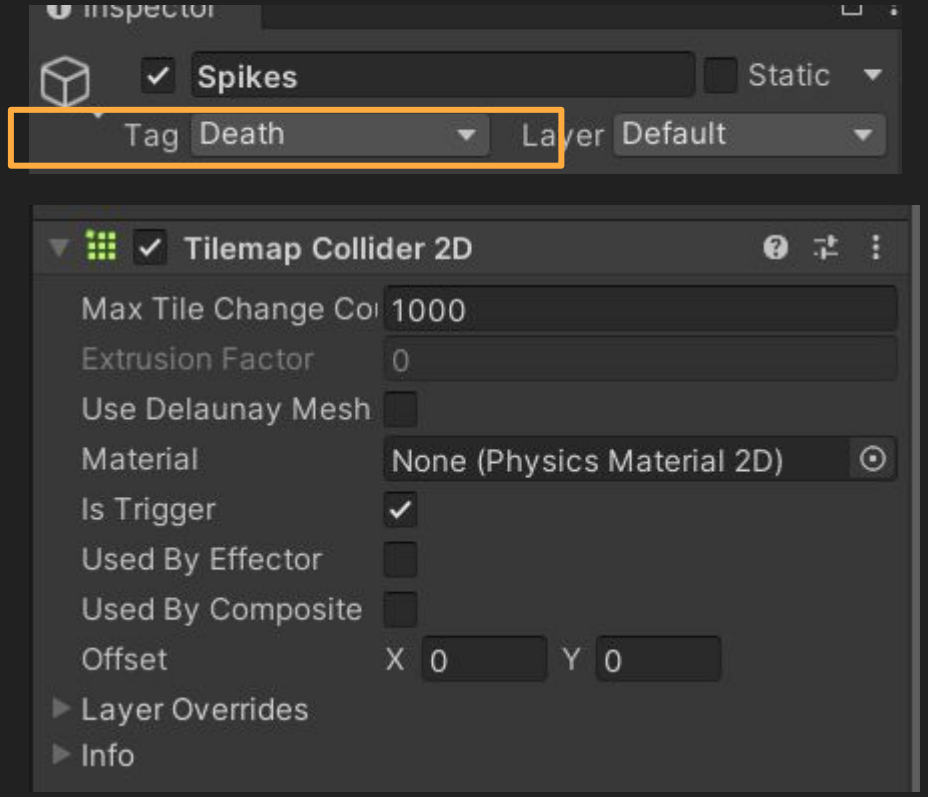


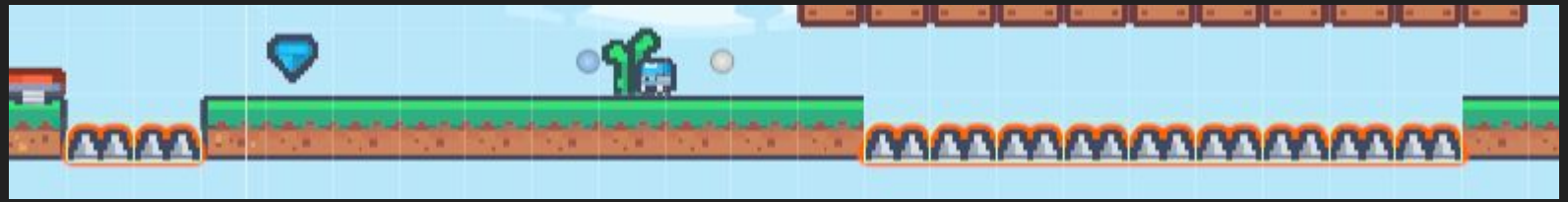


Spikes

For "Spikes," it's much simpler.

Add a Tilemap Collider and set it to trigger so that if the player touches it, they die. Also, make sure the Tag is set to "Death" for the code to work.





Decorations

The "Decorations" tilemap is the simplest, used to make the level prettier. Make sure the order in the layer is set to -1 in the Tilemap Renderer.