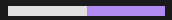


Spritesheet, Palettes & Tilemaps

Sebastian Grygorczuk - STEM Institute at CCNY

Today's Agenda

- **Spritesheets**: Discuss what a spritesheet is and how to process it to extract individual sprites.
- **Tilemaps**: Create tilemaps that allow us to place sprites as if we're painting with colors.
- **Tile Palettes**: Learn about tile palettes, which convert sprites into tiles that the tilemap can read and use in our scene.

Video

Alongside the slide, you can watch the accompanying video, which covers the same topics in a less detailed manner.

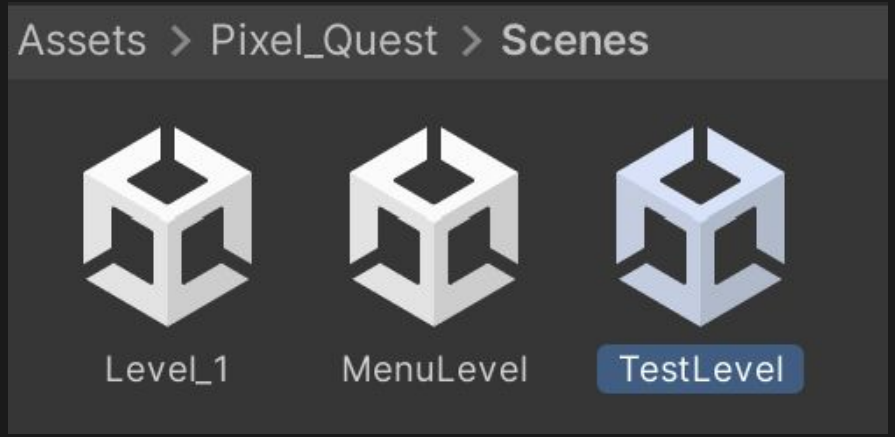
Set Up

Open Up Pixel Quest Level 1

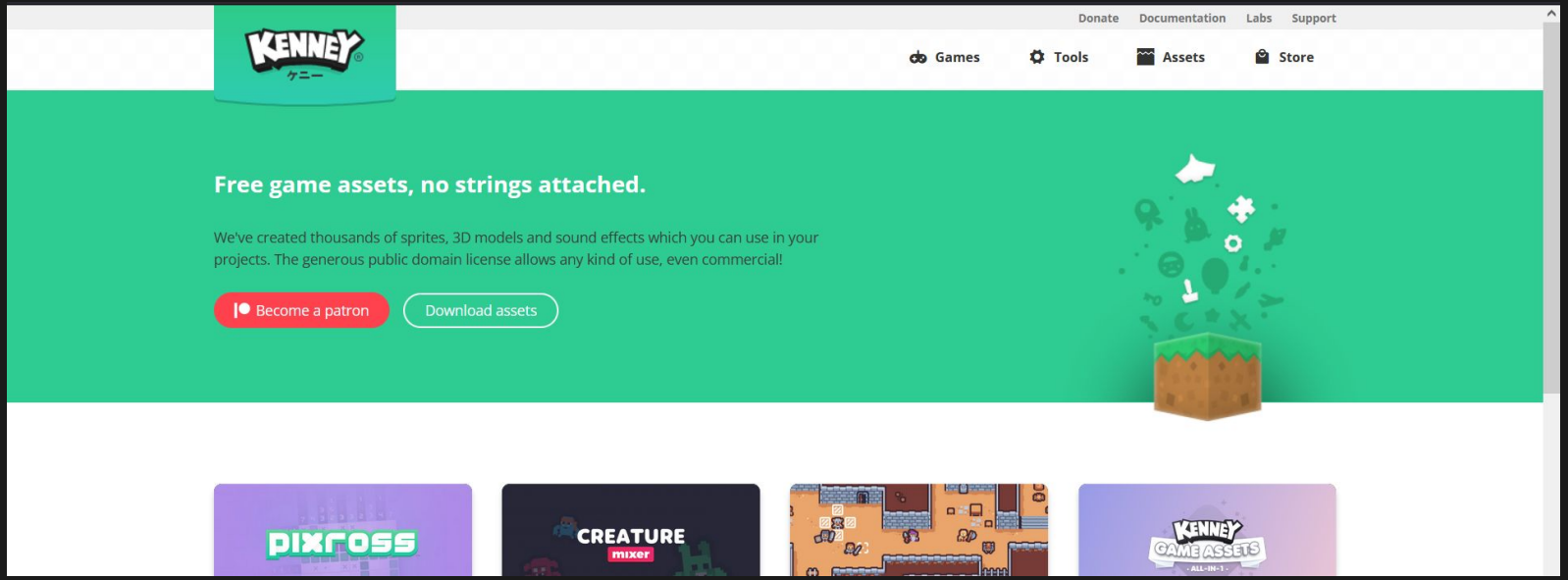
We're going to start fresh in a new scene and begin laying the foundations for **Pixel Quest**.

Navigate to:
Assets > Pixel_Quest > Scenes

Then, open **TestLevel**.



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/

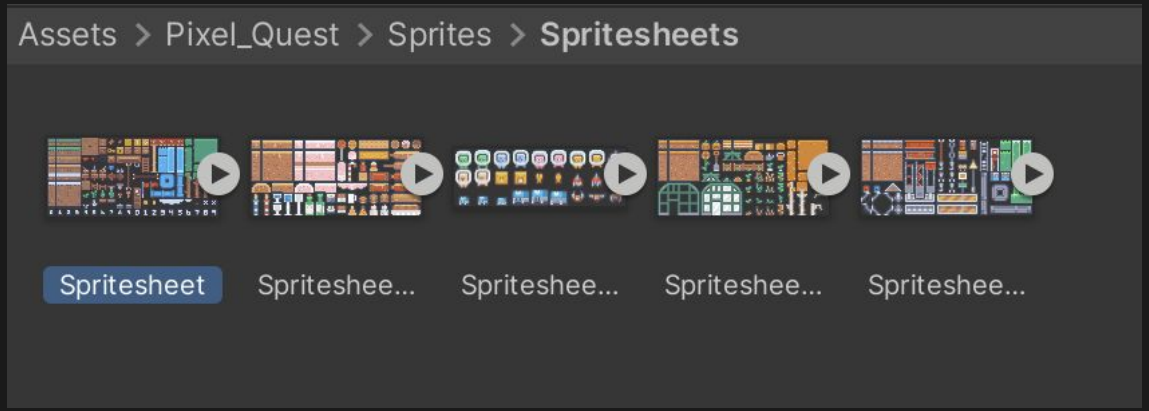
Spritesheet

Getting to Sprites

Geo Quest was made with basic blocks, but most games have visuals that display the theme and character art, we will do the same.

Navigate to: **Assets > Pixel_Quest > Sprites > Spritesheets** Select the first **Spritesheet**, this will be our basis for **Pixel Quest**.

A **spritesheet** is a collection of sprites in one image. It makes it easier to keep track of all the art, but it comes with the issue of being a single piece of artwork. We will need to break it down into individual parts.

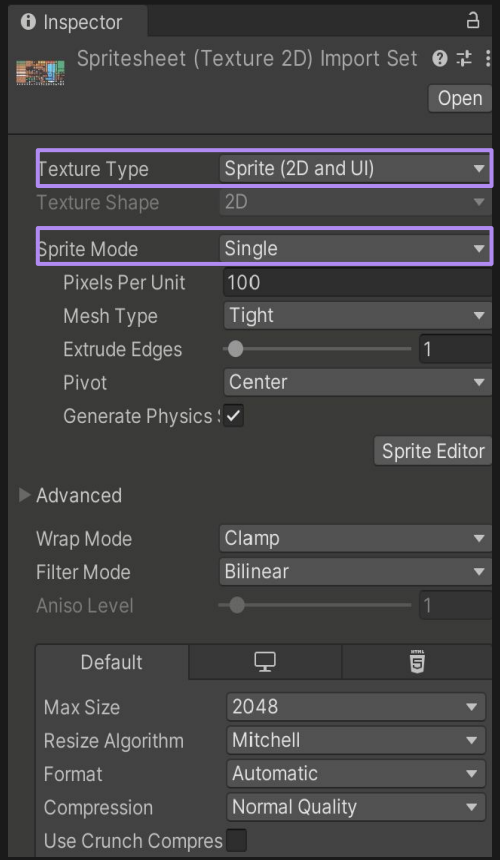


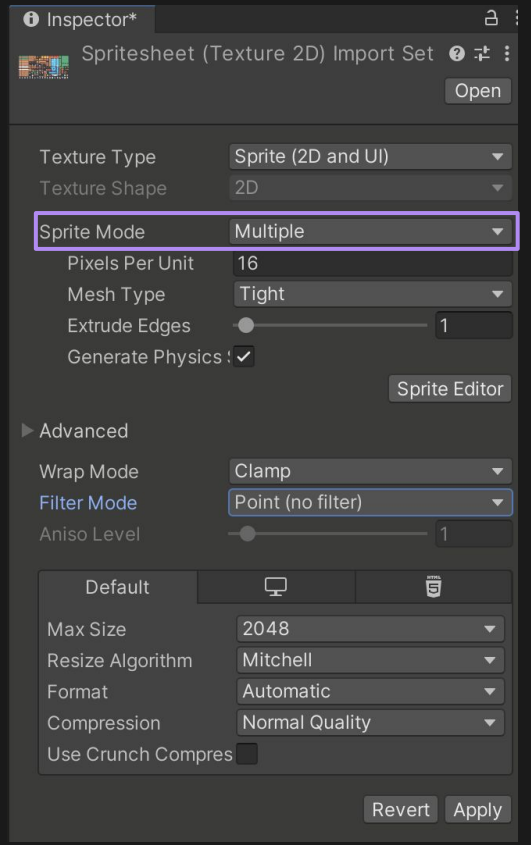
Getting to Sprites

When looking at an art asset, there are a few settings we care about:

Texture Type – Normally, when you have art, it is treated as a **Texture**, meaning it's meant for 3D objects. However, we are working in 2D, so we will always use **Sprite (2D and UI)**.

Sprite Mode – This controls how the image should behave. **Single** means it's one singular image, but we want to swap it to **Multiple** so we can cut it out into many.



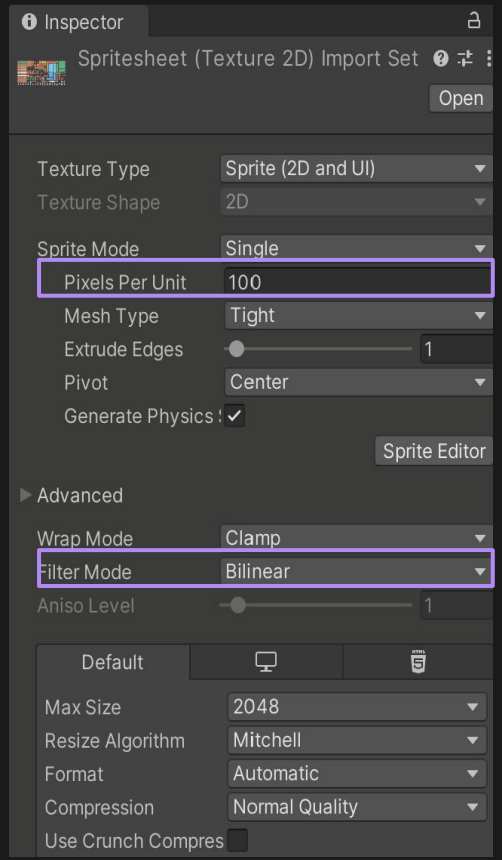


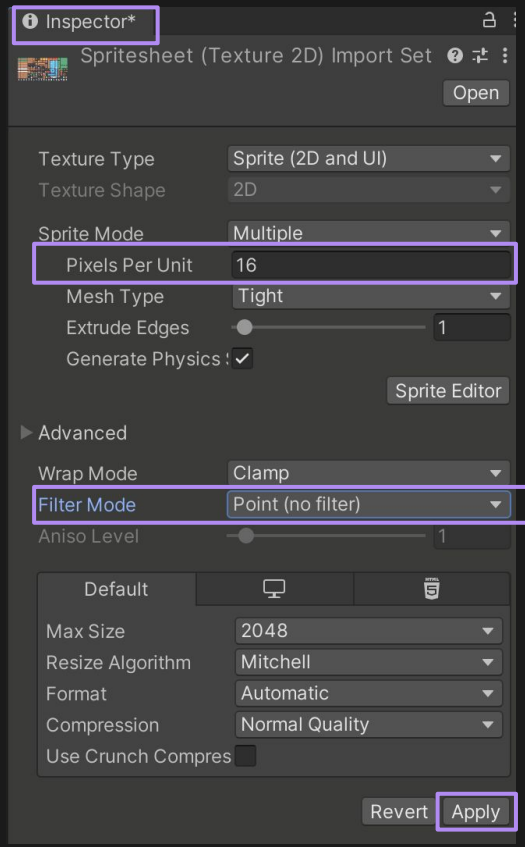
Getting to Sprites

Pixels Per Unit – This controls how much space the image takes per unit of space in Unity. You want to match this to the sprite's actual size. If a sprite is **16x16 pixels**, you should input **16** in this space.

Filter Mode – This controls how the image behaves when shrunk or enlarged. We want to swap it to **Point (No Filter)** to keep the hard edges. By default, it is set to **Bilinear**, which makes it blurry if stretched too much.

When you make changes, you need to **save them**. You will see an **asterisk (*)** in the **Inspector**, and you must click **Apply** to save the changes.



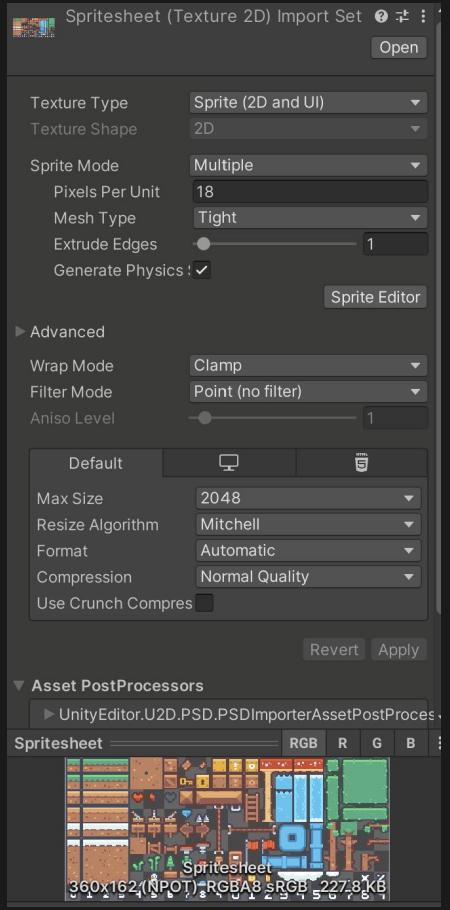


Challenge Sprite Settings

Now, let's put your skills to the test!

Look at the **Spritesheet**, and set the setting values of:

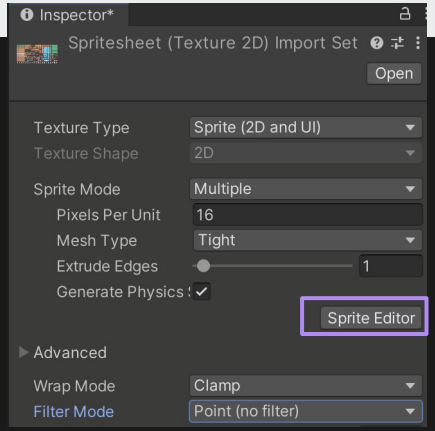
- **Sprite Mode** to **Multiple**
- **Pixels Per Unit** to **18**
- **Filter Mode** to **Point (No Filter)**

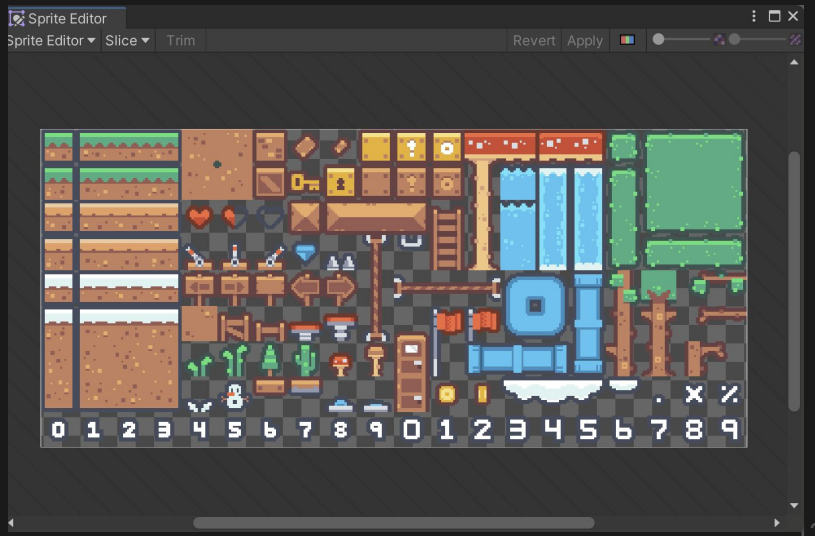


Sprite Editor

Now that we have everything set, we will want to proceed to cut out the images. To do this, open up the **Sprite Editor**. You can access it from the **Inspector** or by going to **Window > 2D > Sprite Editor**.

Either method will open a new window where we can make changes to the image.



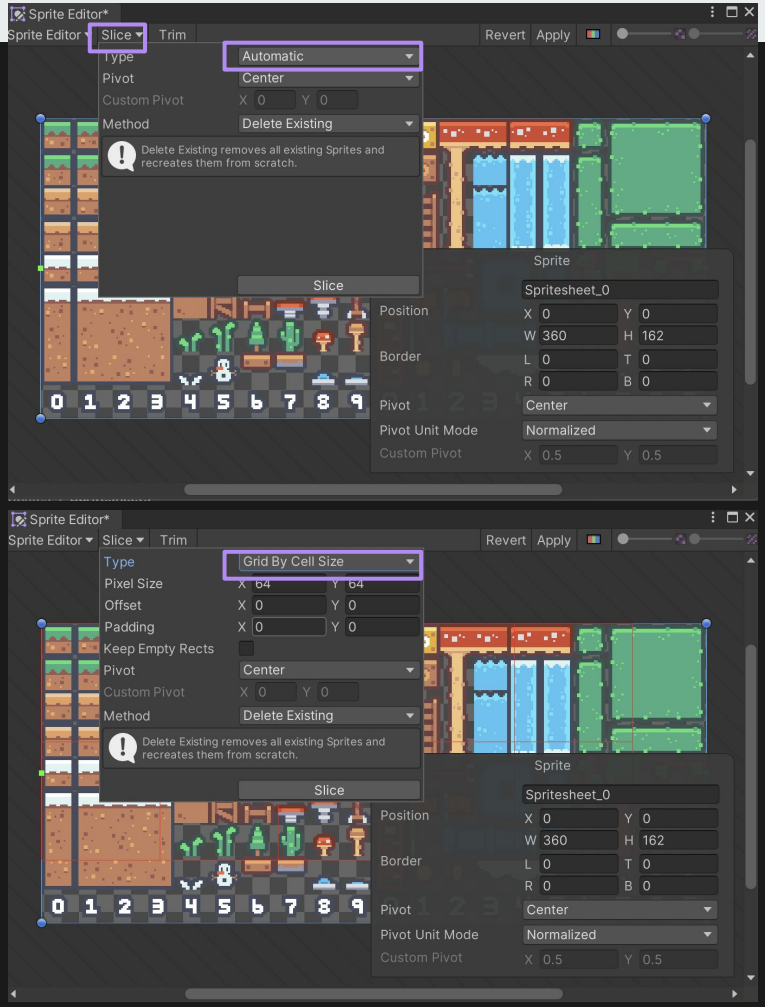


Slicing

To cut the images out, click the **Slice** dropdown, which allows us to use different methods of slicing.

By default, it will be set to **Automatic**, which cuts out images that are not directly touching. However, since we have a spritesheet where blocks are touching, they would be cut out as one big block.

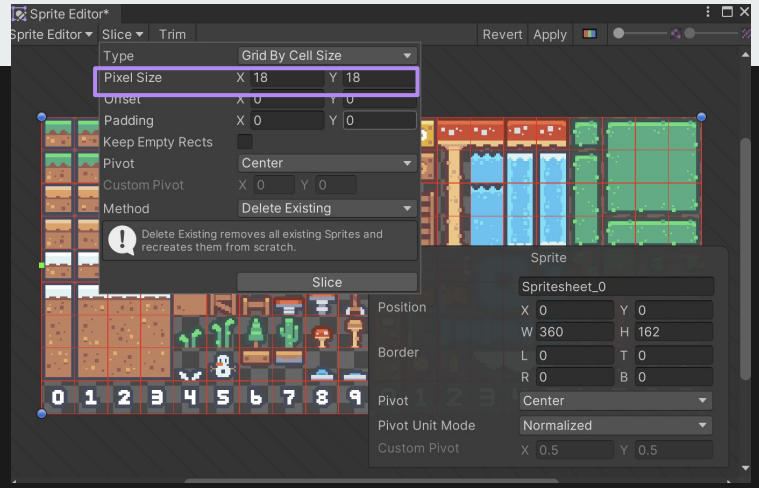
To overcome this, we will change the **Type** to **Grid by Cell Size**, which allows us to cut them all out by pixel dimensions.

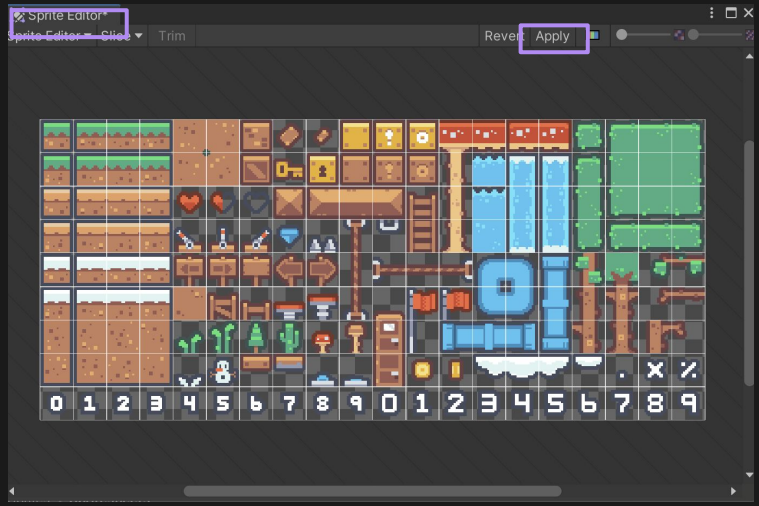


Slicing Grid By Cell Size

We know that this image has sprites that are **18x18**, so we type that in, click **Slice**, and this will cut it out.

Like most things in Unity and C#, this will not save automatically upon slicing, you will see that Sprite Editor has the **Asteris (*)** you need to click the **Apply** button for the changes to be saved.

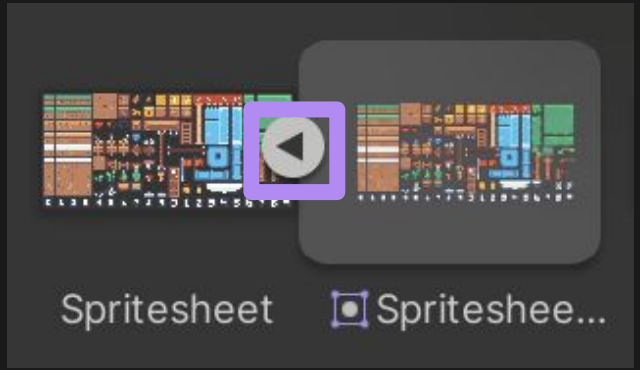




Sprite Asset Breakdown

Once you do all of that, you can look back at the **Project View** and click on the **expand button** (the triangle).

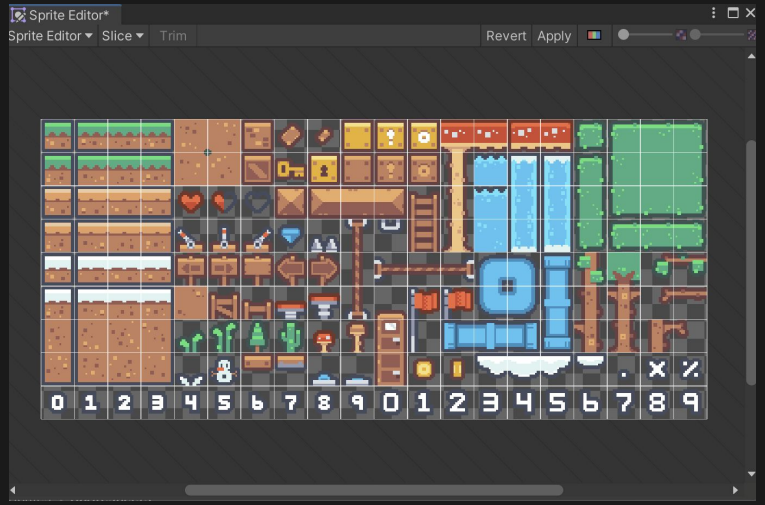
You will see that the image breaks down into each part that was cut out, and you can **drag and drop** these images into your **Scene View**.



Challenge Slice Up

Now, let's put your skills to the test!

Go into Sprite Editor and slice out the first sprite sheet using the 18x18 Pixel Size.



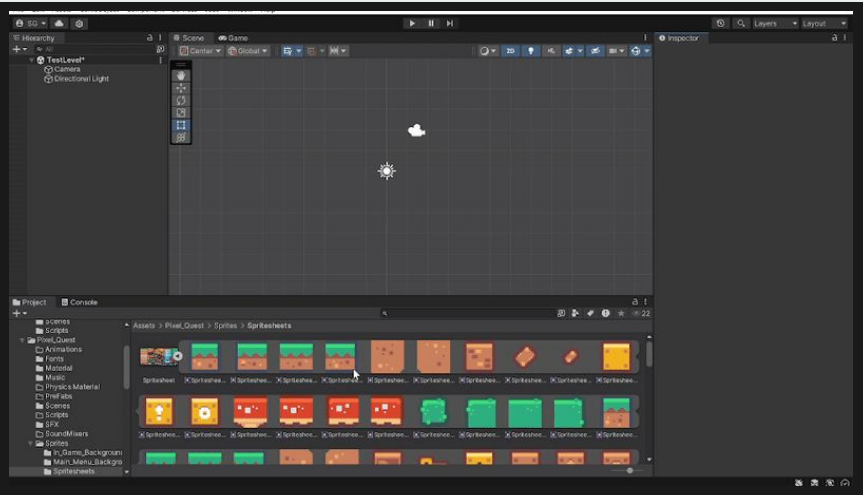
Tilemap & Palette

Drag and Dropping

Now, we could just drag and drop them as we did when making **Geo Quest**, and you can still do that.

However, we will have many more sprites and game objects, and keeping them perfectly aligned by hand will be nearly impossible.

To alleviate that problem, we will use a system called **Tile Mapping**.



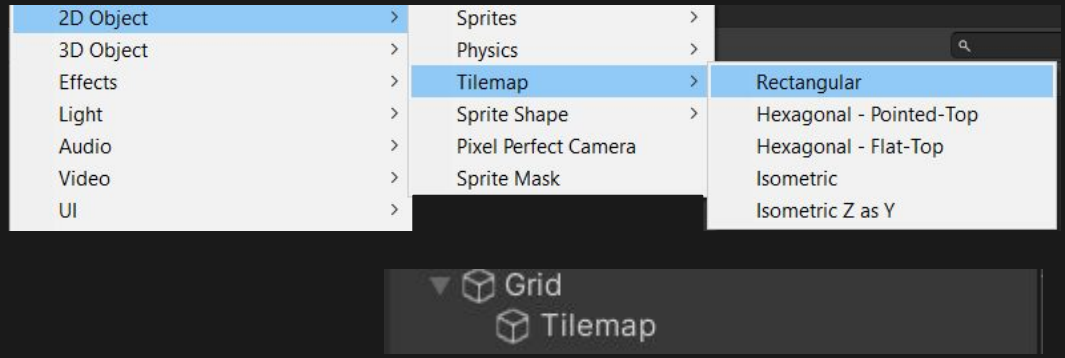
What is a Tilemap

A **tilemap** is a map drawn from tiles.

We take small sprites 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 tilemap becomes a child of.

Everything drawn on the tilemap will be linked to the **Grid**.



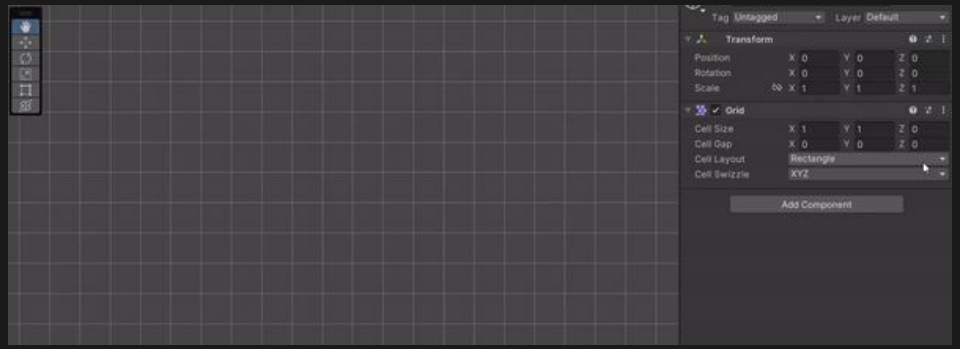
The Gird

The **Grid** can be configured in three ways:

- **Rectangular**
- **Hexagonal**
- **Isometric**

Depending on the style of the game, you should choose the appropriate grid type. We will use Rectangular.

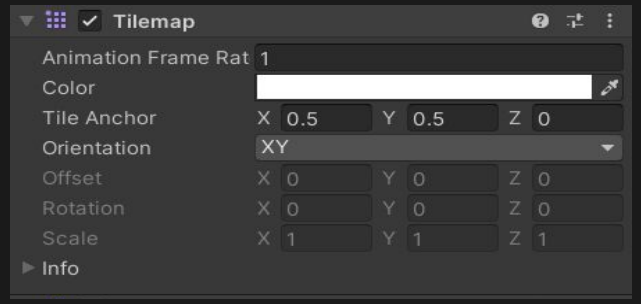
You can also control the size of each cell in the grid and whether there should be any spacing between the tiles.

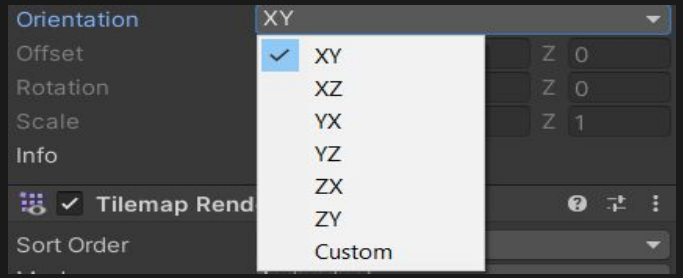


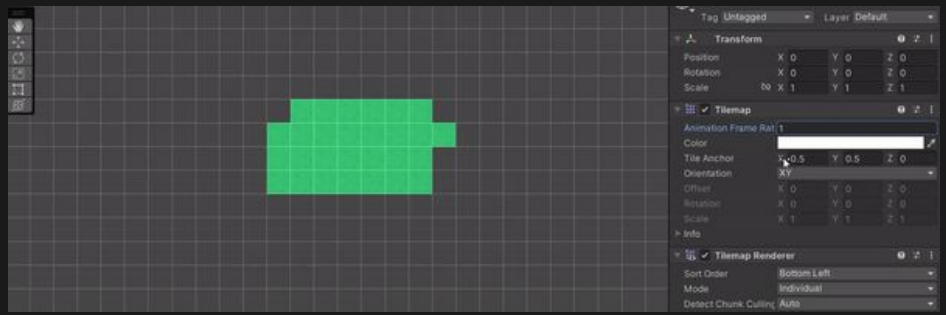
Tilemap Components

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.



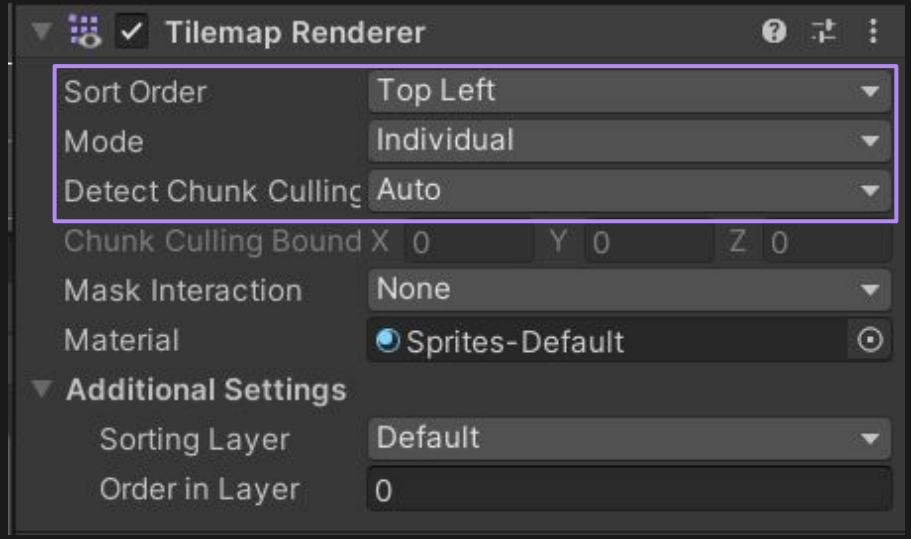




Tilemap Renderer

The **Tilemap Renderer** controls what gets drawn and how.

- **Sort Order** determines which tiles will be drawn first. For example, if **Top Left** is selected, drawing will start from that corner.
- **Mode** defines how tiles should be drawn—either individually or in chunks.
- **Detect Chunk Culling** tells Unity how close the player needs to be for tiles to render, optimizing performance.



Challenge Tilemaps

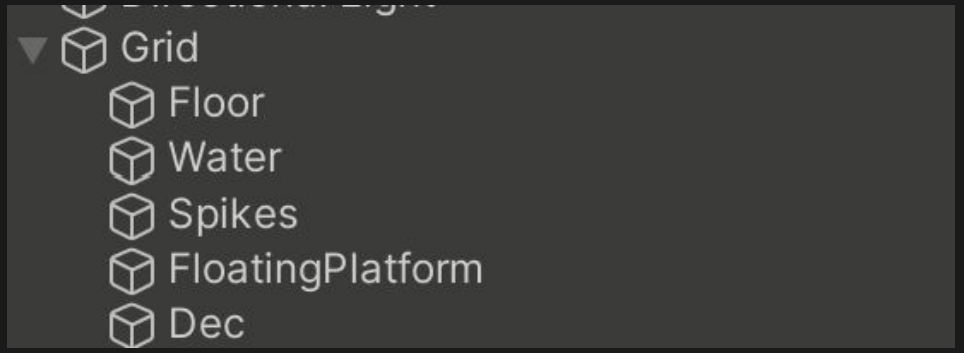
Now, let's put your skills to the test!

Create **five tilemaps** that we will use to build our levels.

For now, they should all be the same, just with different names:

- **Floor**
- **Water**
- **Spikes**
- **Floating Platform**
- **Decorations**

Make sure they are all **parented under the one Grid object** that was spawned with the first tilemap you created.



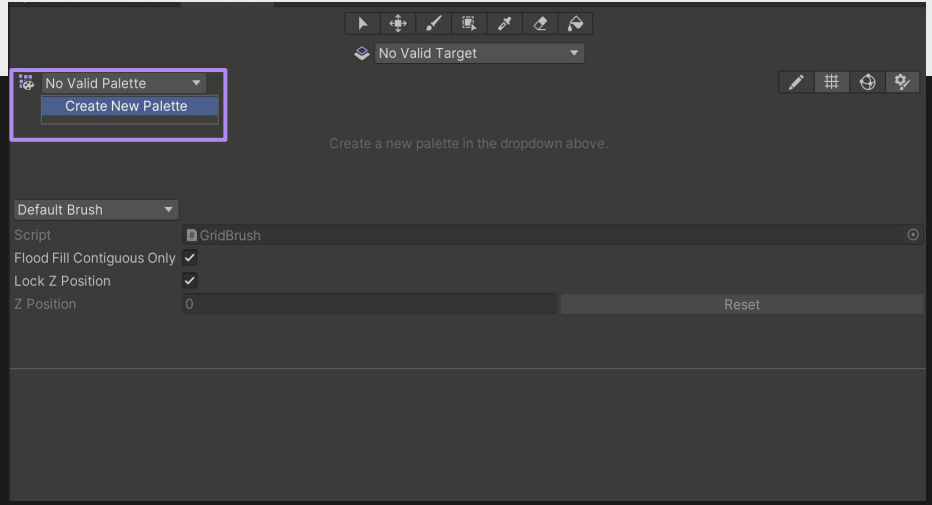
The Palette

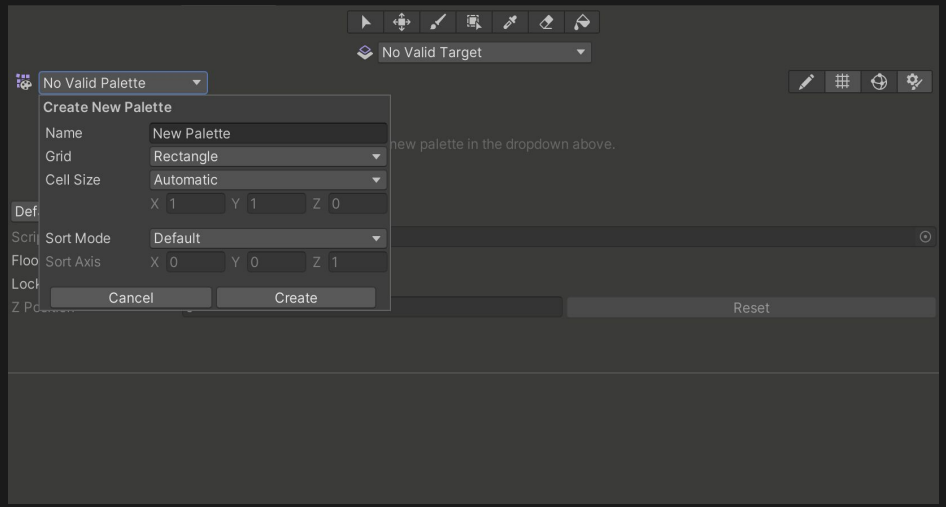
Even though our images are split into individual parts, they are not yet tiles.

We need to create a **Palette**, which will store these images as **Tiles**. These tiles will act as prefabs for the **Tilemap** to draw with.

To create a palette:

1. Navigate to **Window > 2D > Tile Palette**.
2. When it opens, you should see that there is **No Valid Palette**.
3. Click on that message and create a new one.
4. Give it an appropriate name, keep it **Rectangular**, and hit **Create**.





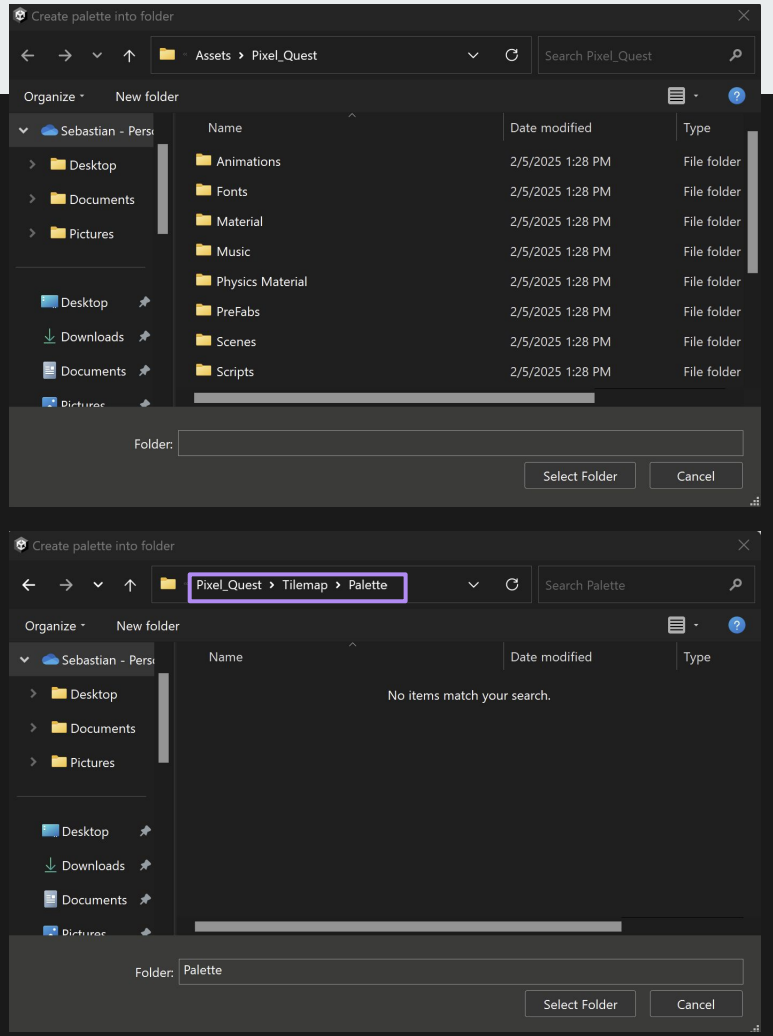
The Palette

Next, you will be prompted to choose where to create this **Palette**.

The file explorer will open in whatever folder you are currently viewing in the **Project View**.

Either open **Pixel_Quest > Tilemap > Palette** and hit **Create**, or navigate to that same folder manually when the file explorer pops up.

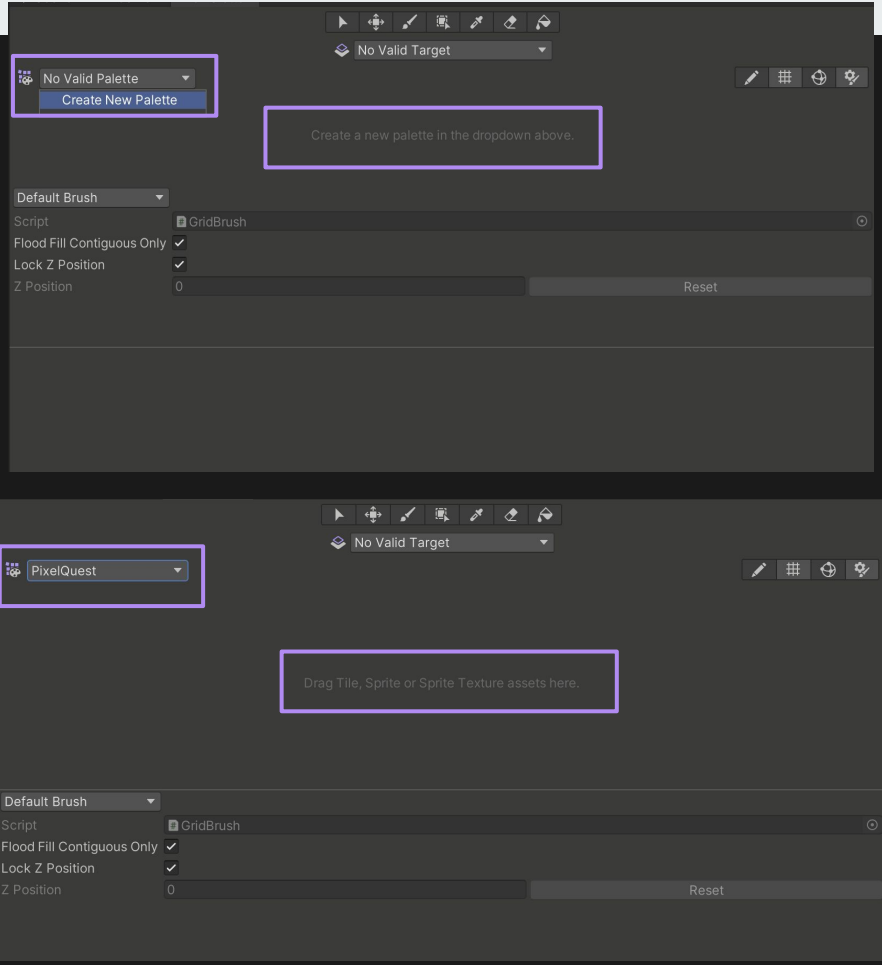
Once you're there, hit **Select Folder**, and the **Tile Palette** asset will be created.



The Palette

Once the palette is created, the menu won't look like much has changed, but the text in the center will update, and the name of your palette will now appear in the dropdown menu.

The text will change from "**Create a new palette in the dropdown above**" to "**Drag Tile, Sprite, or Sprite Texture Asset here**".



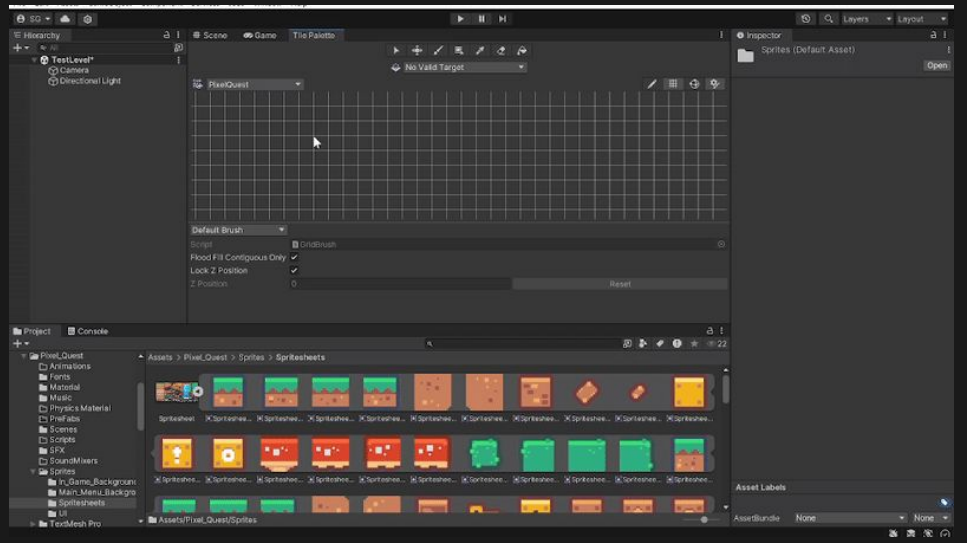
The Palette

Drag the **sliced spritesheet** into the **palette**. Once again, a file explorer window will pop up asking where you want to store the tiles.

Choose to place them in: **Pixel_Quest > Tilemap > Tiles**

This will keep the project organized.

After selecting the folder and clicking **Select Folder**, your **Tile Palette** should now be ready to use.

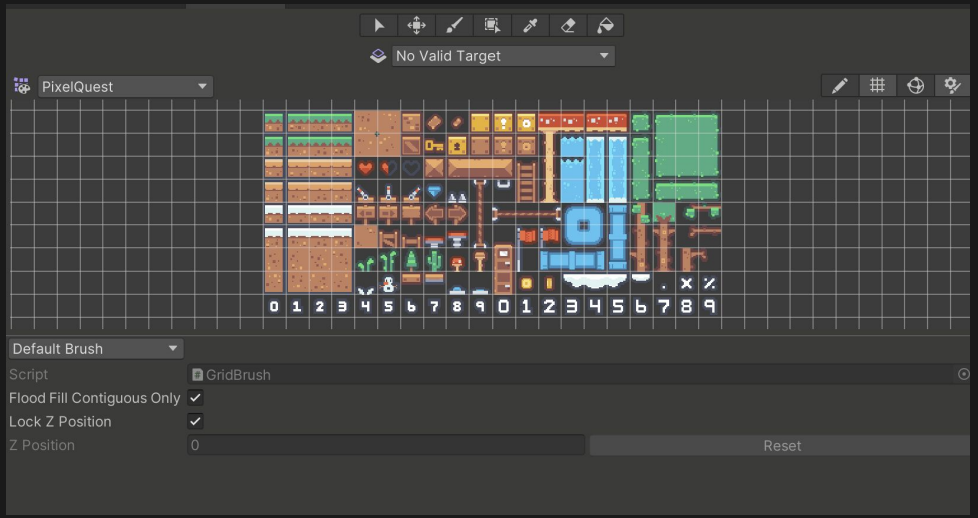


Challenge Create Palette

Now, let's put your skills to the test!

Follow the steps we just covered and create your own palette.

Try to keep your assets organized.



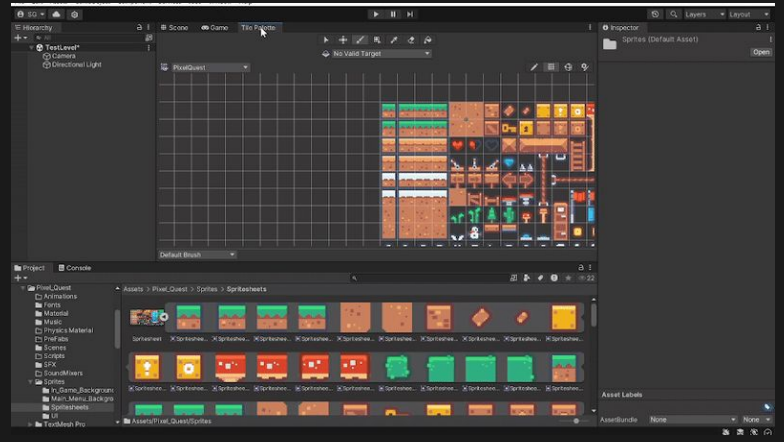
View Set Up

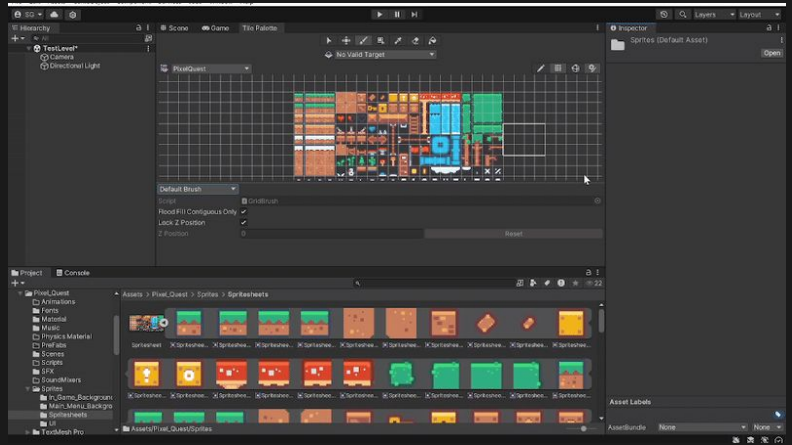
Now that we have the palette, we will want to use it to draw our level.

I suggest organizing your views so that half the screen is the **Scene View** and the other half is the **Palette View**.

Additionally, I recommend shrinking the **Default Brush** settings at the bottom to better see your palette.

You can also navigate the palette just like the **Scene View**, use the **mouse wheel** to zoom in and out, and hold the **right mouse button** to move around the palette.



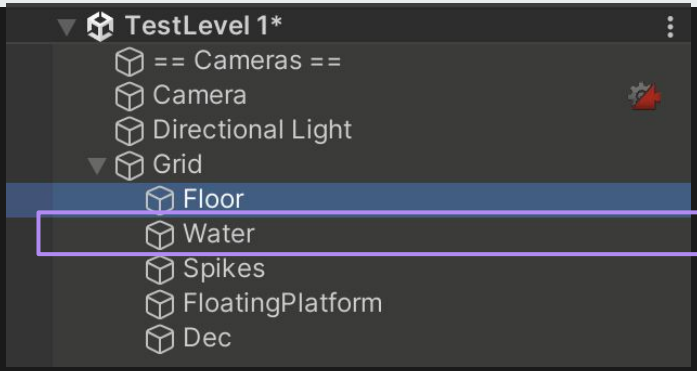


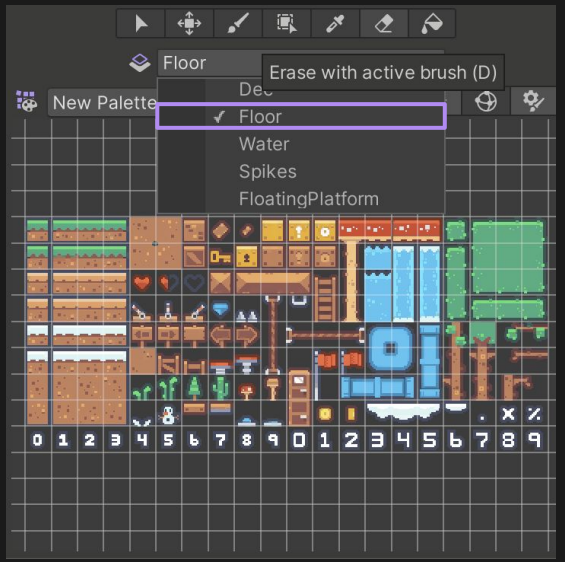
Selecting Layer

Make sure to first select the **GameObject** or **Active Target** before you start drawing.

- If you don't select the correct target, the tiles you draw may have incorrect properties.
- To select the **Tilemap**, either:
  - Click on it in the **Hierarchy**, or
  - Use the drop-down menu to pick the layer you want to draw on.

When selected from the **Hierarchy**, all similar tiles will be outlined in **orange**, while tiles from a different **Tilemap** will not be highlighted.





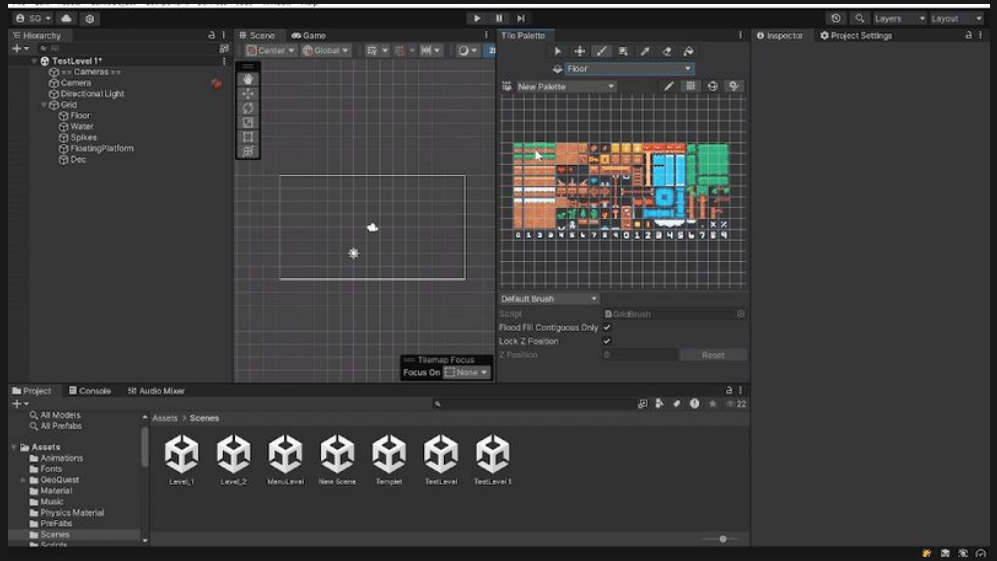
Sebastian Grygorczuk - STEM Institute at CCNY

Painting

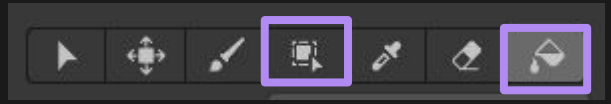
Select a tile and use the Brush Tool to start painting your map.

To erase tiles, either:

- Select the Eraser Tool, or
- Hold Left Shift while painting.

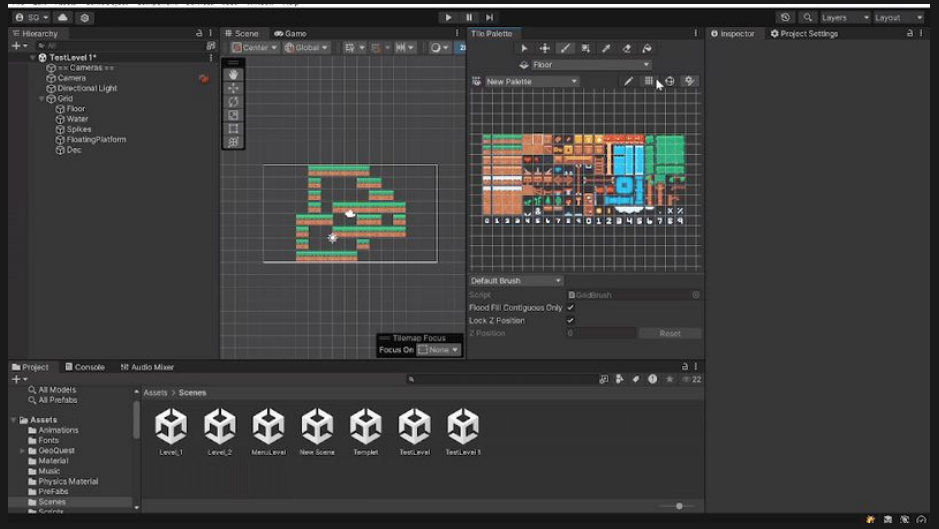


Rectangle and Fill Tool



At your disposal, you also have the **Rectangle Tool**, which allows you to copy and place the same tile or a selected tiled area within the rectangle you create.

Additionally, the **Fill Bucket** can be used to paint all connected tiles of the same type, filling the area as long as there's a continuous space linking them.

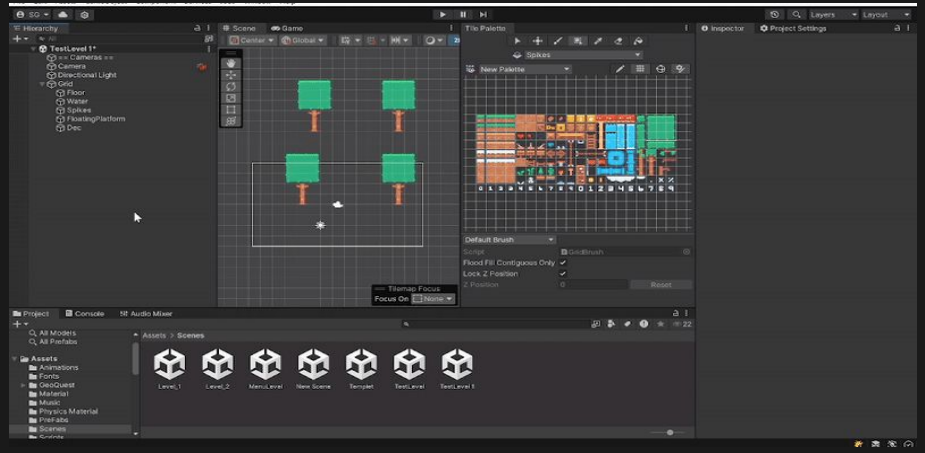


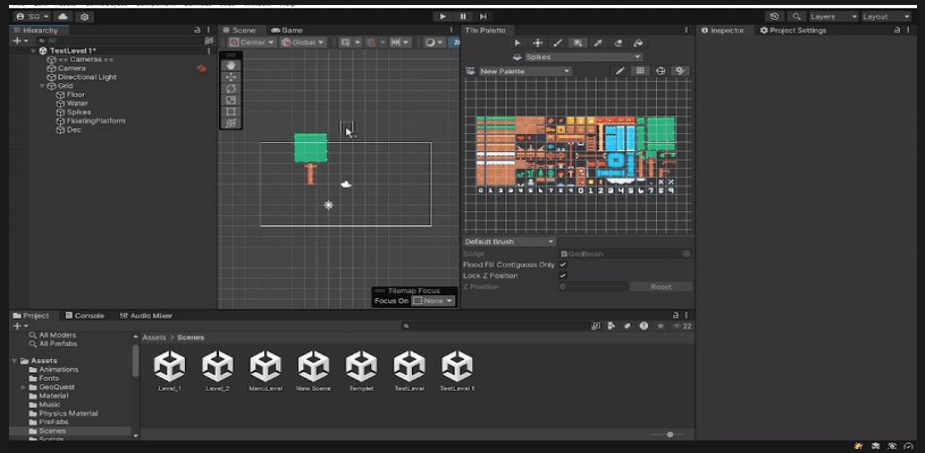
Sebastian Grygorczuk - STEM Institute at CCNY

Rectangle Copy, Paste and Erase

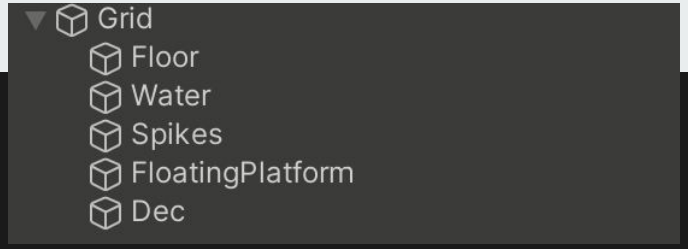
When you have the **Rectangle Tool** selected, you can combine the **rubber band selection** with **Shift** to erase anything on that layer.

Meanwhile, if you **hold Ctrl** and rubber band an area, you will **copy** that selection. Then, using the **Rectangle Tool**, you can **paste** the copied tiles onto your tilemap.



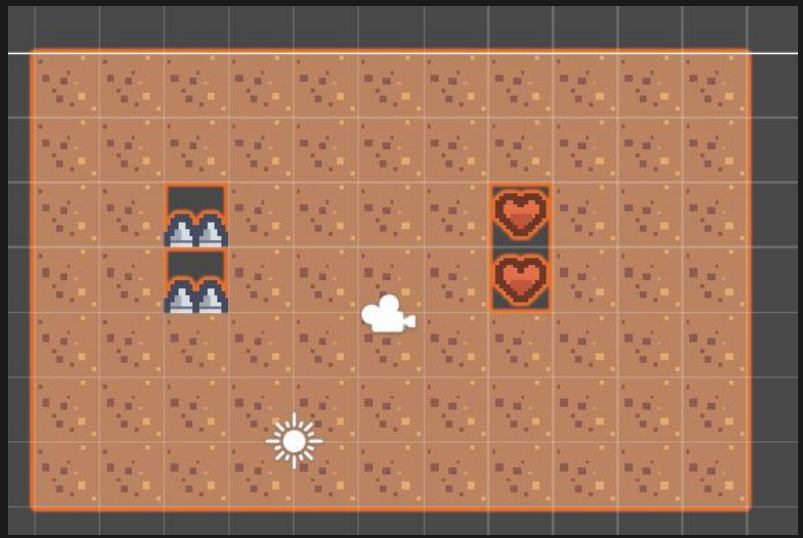


Layering Maps



If you try to place a tile on top of an existing one, it will **replace** the original tile with the new one.

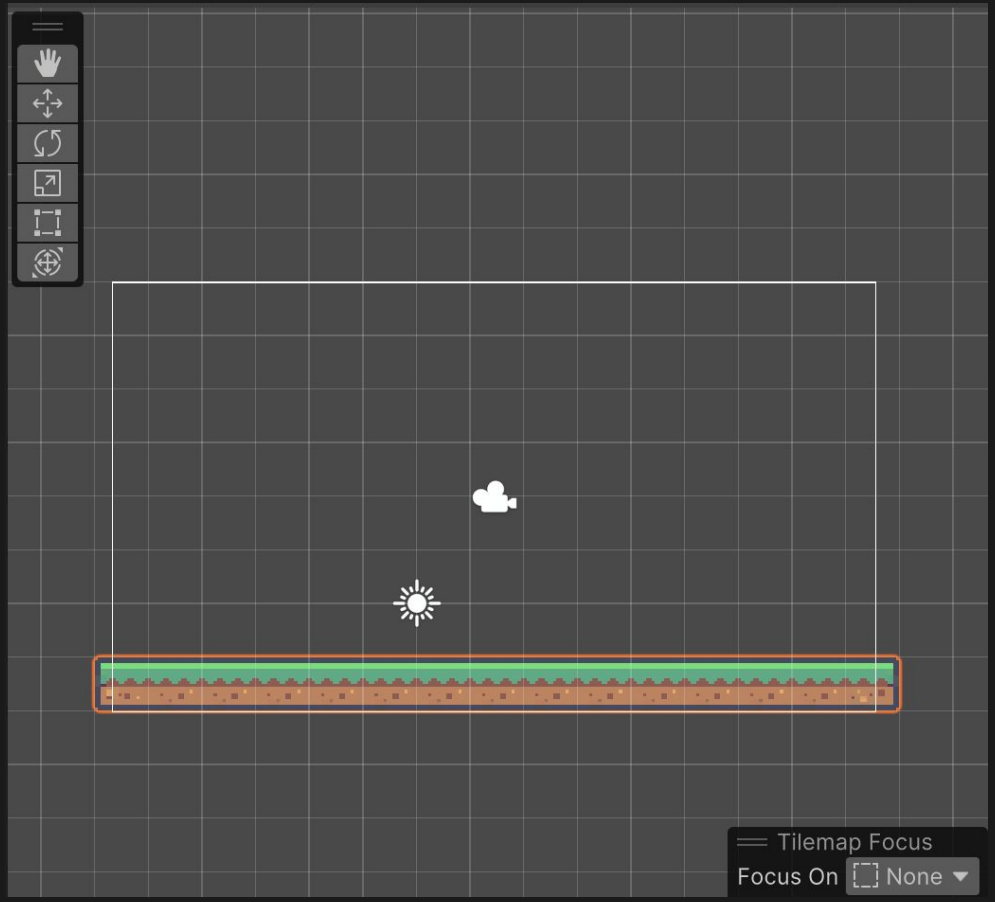
If you want to **layer** tiles on top of each other, you need to use **different tilemaps** and set their **Order in Layer** values correctly so that one appears above another.



Challenge Floor

Now, let's put your skills to the test!

Select the Floor Tilemap you've created and draw a floor for your player to walk on.

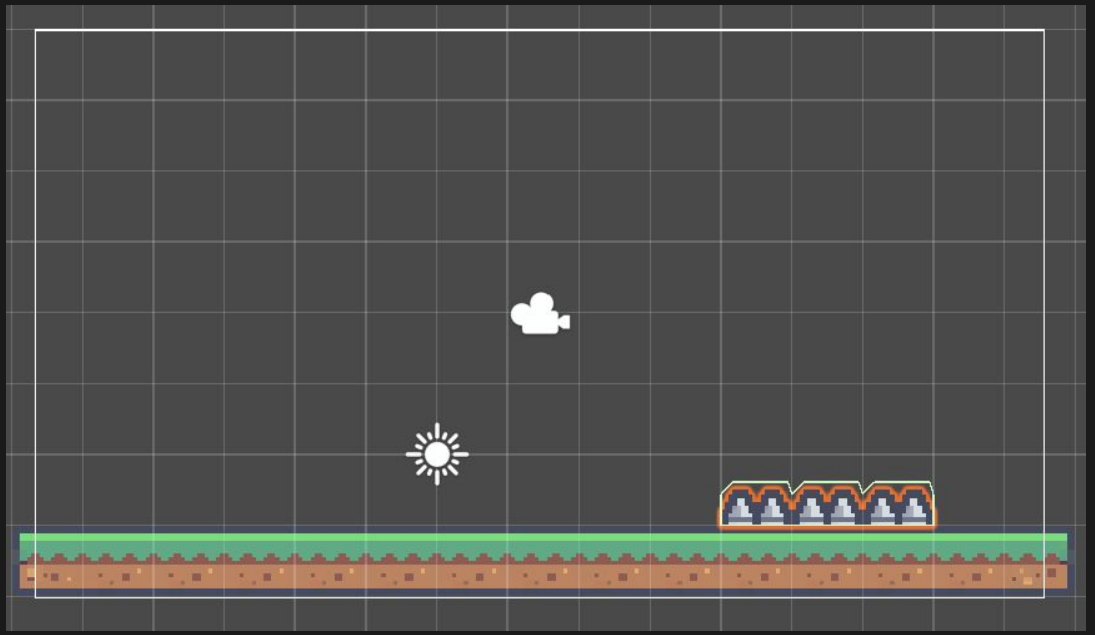


Challenge Spikes

Now, let's put your skills to the test!

Go to the Spikes Tilemap and create a few spikes on the level.

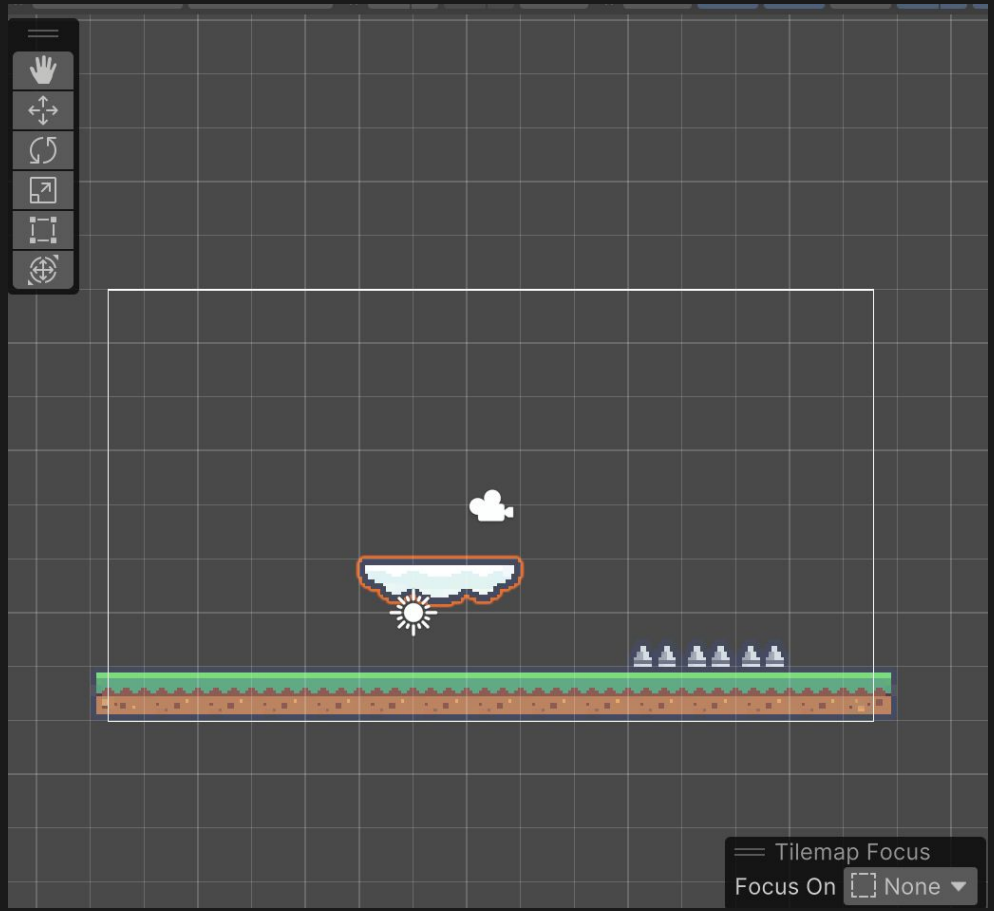
Make sure to mark the Tag as “Death”



Challenge Floating Platform

Now, let's put your skills to the test!

Go to the Floating Platform Tilemap and draw a cloud.

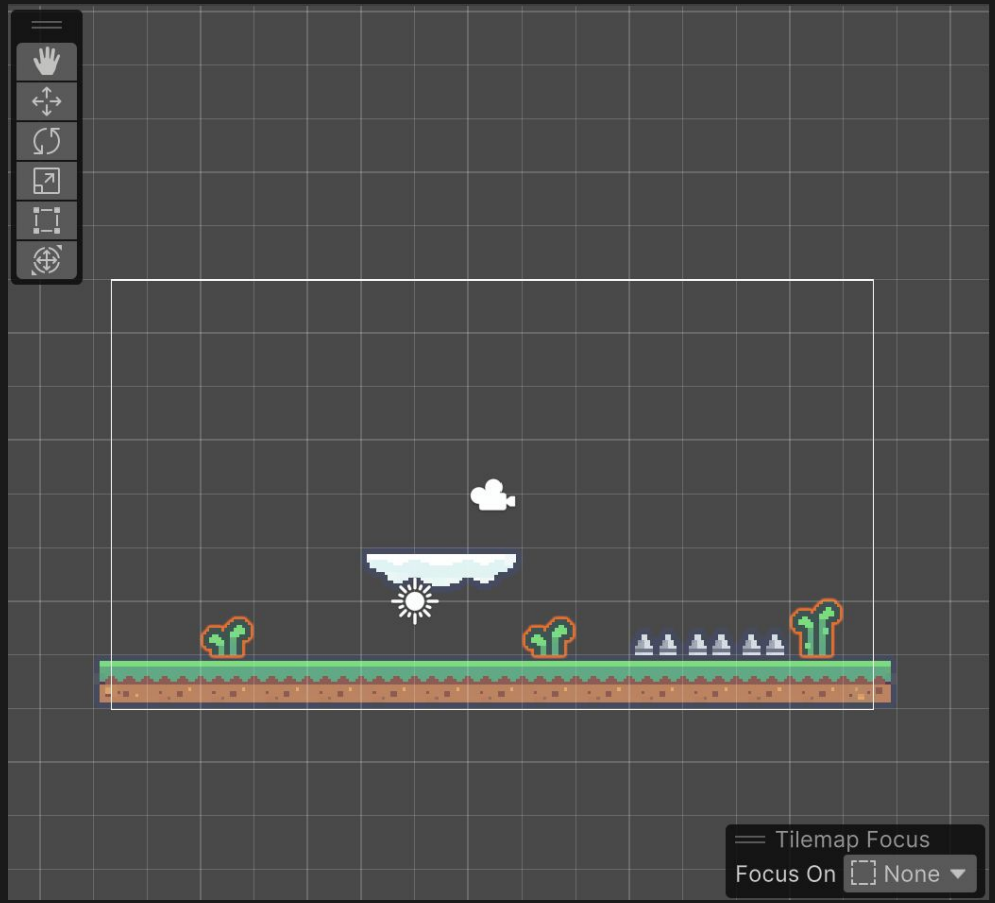


Challenge Decorations

Now, let's put your skills to the test!

Go to the Decorations Tilemap draw few plants.

Make sure to set the Order in Layer to be -1 so they’re behind the player.

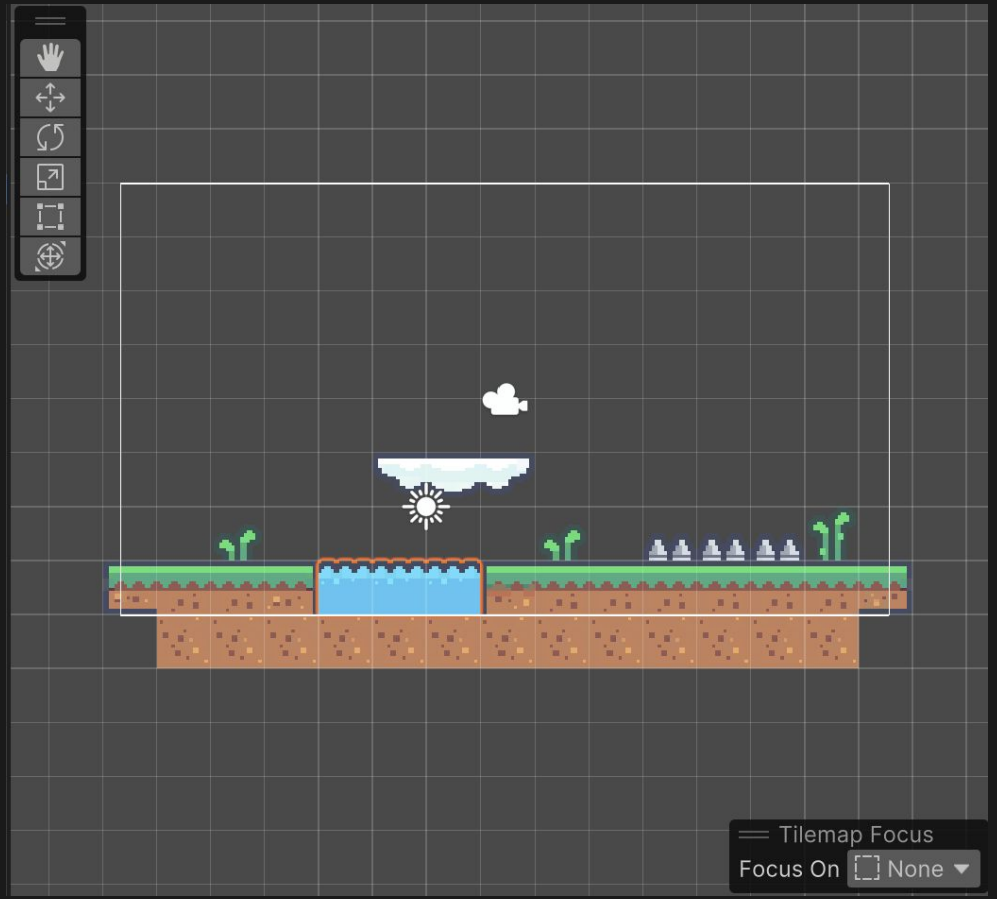


Challenge Water

Now, let's put your skills to the test!

Go to Floor Tilemap , cut out a piece and create a row of dirt below the first one.

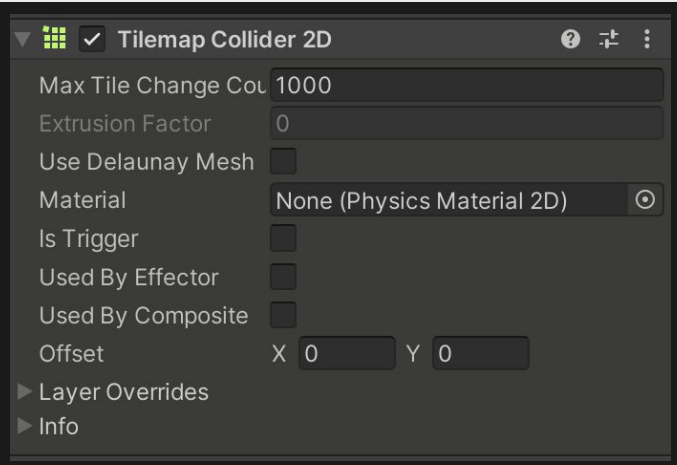
Then go to the Water Tilemap and draw your puddle.

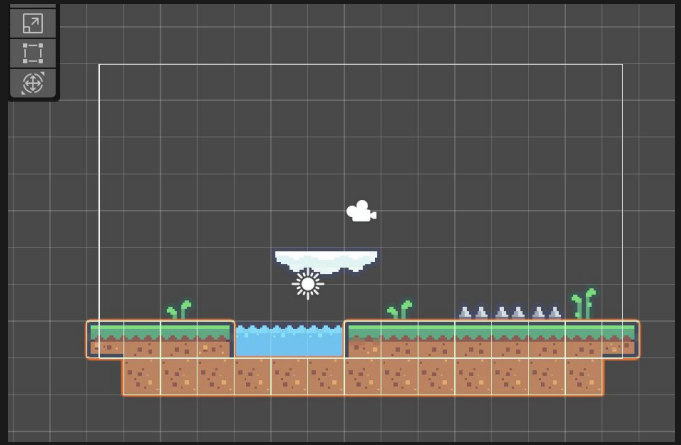


Tilemap Collider 2D

Tilemaps have a special **collider** that you can attach, allowing you to create platforms or pathways for the player to interact with.

Adding the **Tilemap Collider 2D** component will generate a **polygonal collider** around each tile, matching the shape of the tile.



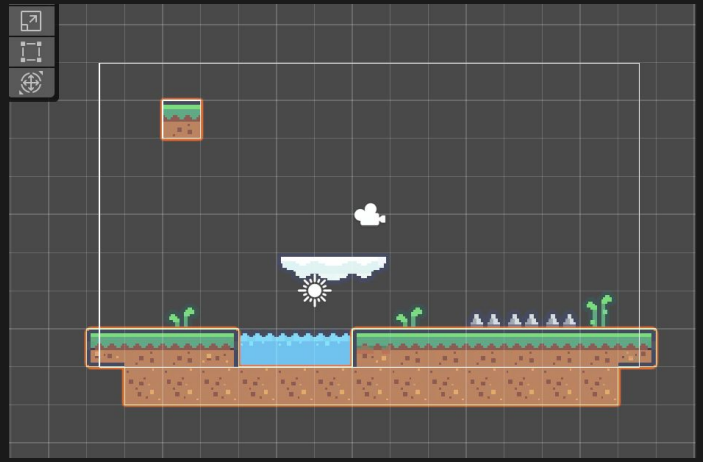


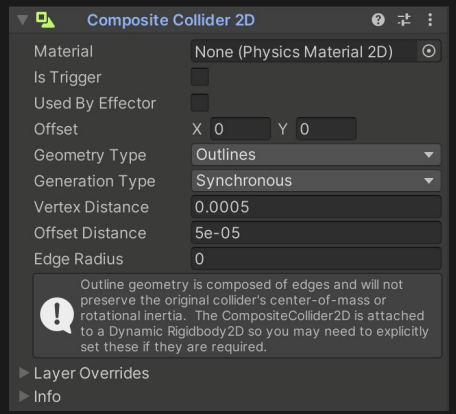
Composite Collider

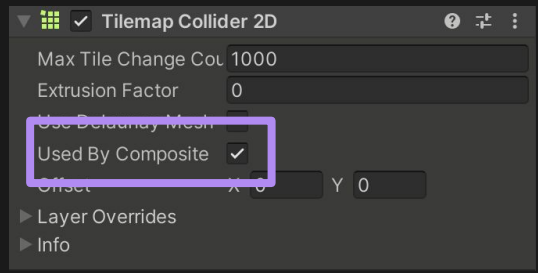
Right now, if the player walks on these individual tile colliders, they'll get caught on the edges, making movement feel rough and inconsistent.

To fix this, we can add a **Composite Collider 2D**. This will merge all the small, adjacent colliders into one large, smooth collider, creating a seamless surface for the player to walk on.

For this to work, go to the **Tilemap Collider 2D** component and check the box for **"Used by Composite."**



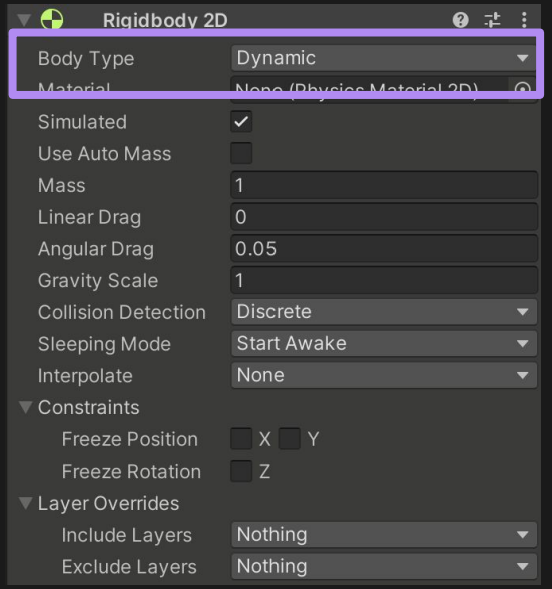


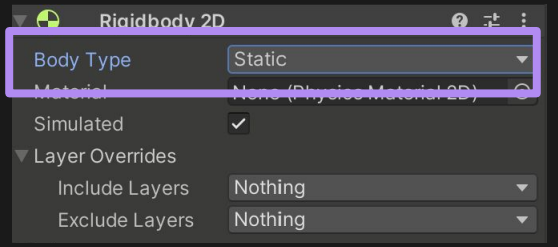


Composite Collider 2D + Rigidbody 2D

When you add a Composite Collider component, a Rigidbody 2D will also be added to your game object.

Now, we don't want this Rigidbody to move around, as it represents our world. Therefore, make sure to change the Rigidbody's type from 'Dynamic' to 'Static.



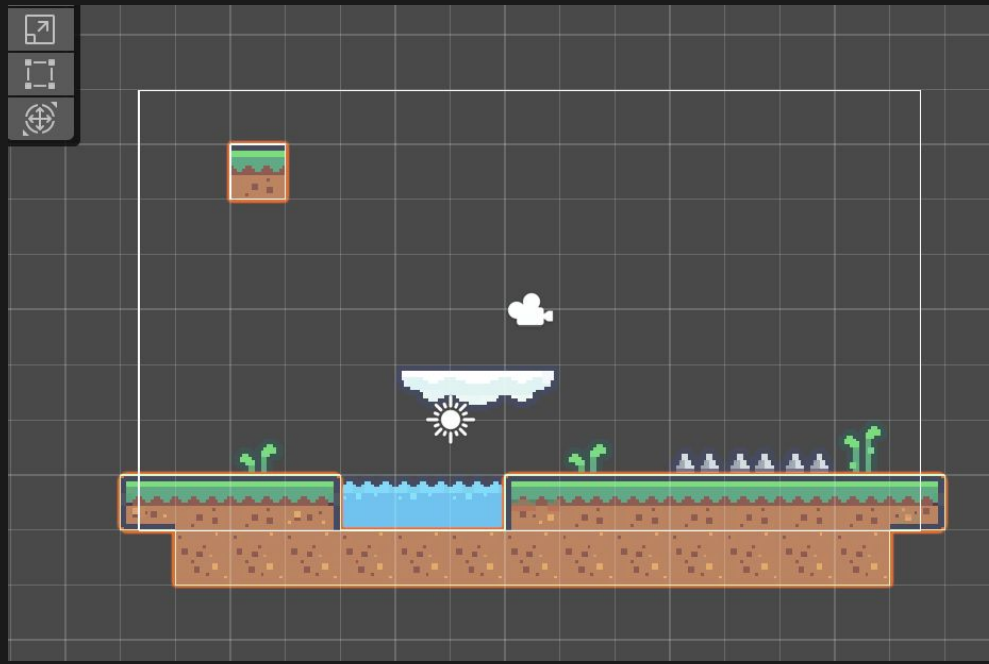


Challenge Floor: Add Collision

Now, let's put your skills to the test!

Add a Tilemap Collider 2D and a Composite Collider 2D. Make sure that the Tilemap Collider is set to use the Composite Collider.

Next, set the Rigidbody 2D to 'Static' to prevent it from moving.

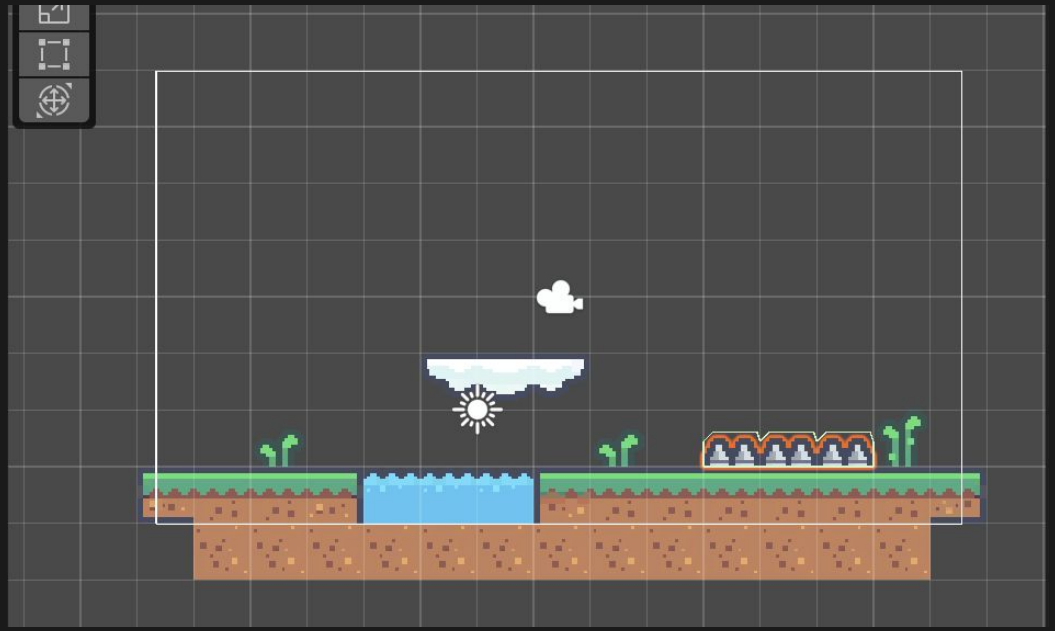


Challenge Spikes: Add Collision

Now, let's put your skills to the test!

Add a Tilemap Collider 2D and a Composite Collider 2D. Make sure that the Tilemap Collider is set to use the Composite Collider.

Next, set the Rigidbody 2D to 'Static' to prevent it from moving.

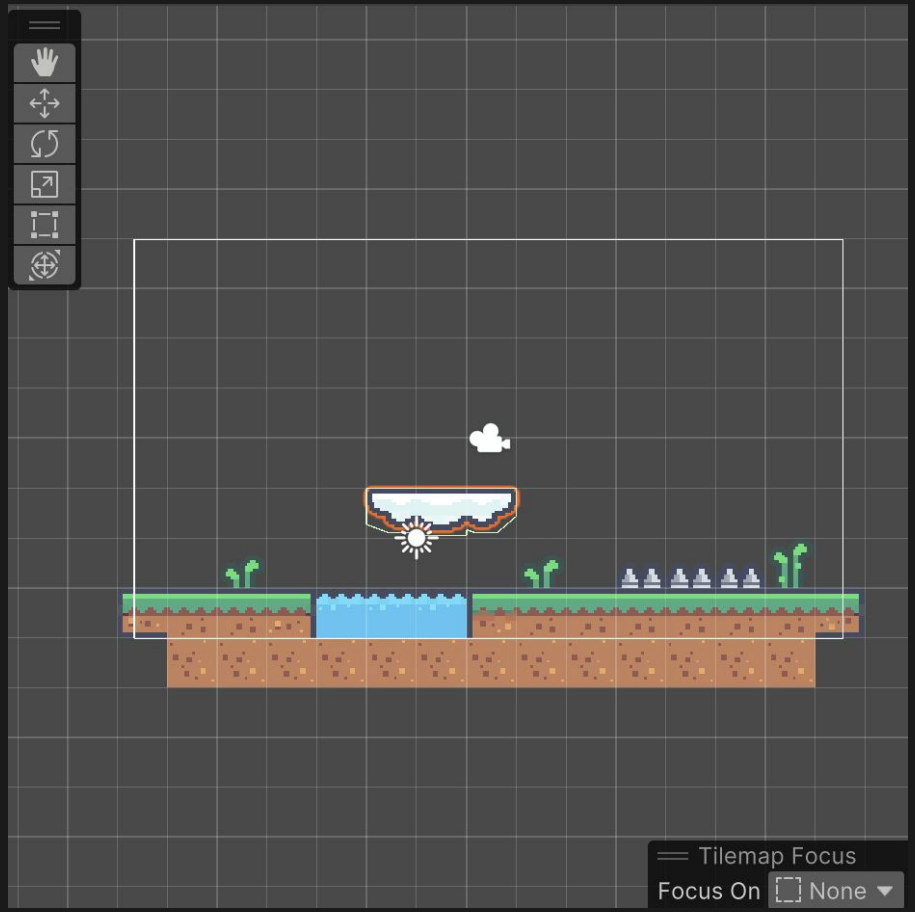


Challenge Floating Platform: Add Collision

Now, let's put your skills to the test!

Add a Tilemap Collider 2D and a Composite Collider 2D. Make sure that the Tilemap Collider is set to use the Composite Collider.

Next, set the Rigidbody 2D to 'Static' to prevent it from moving.

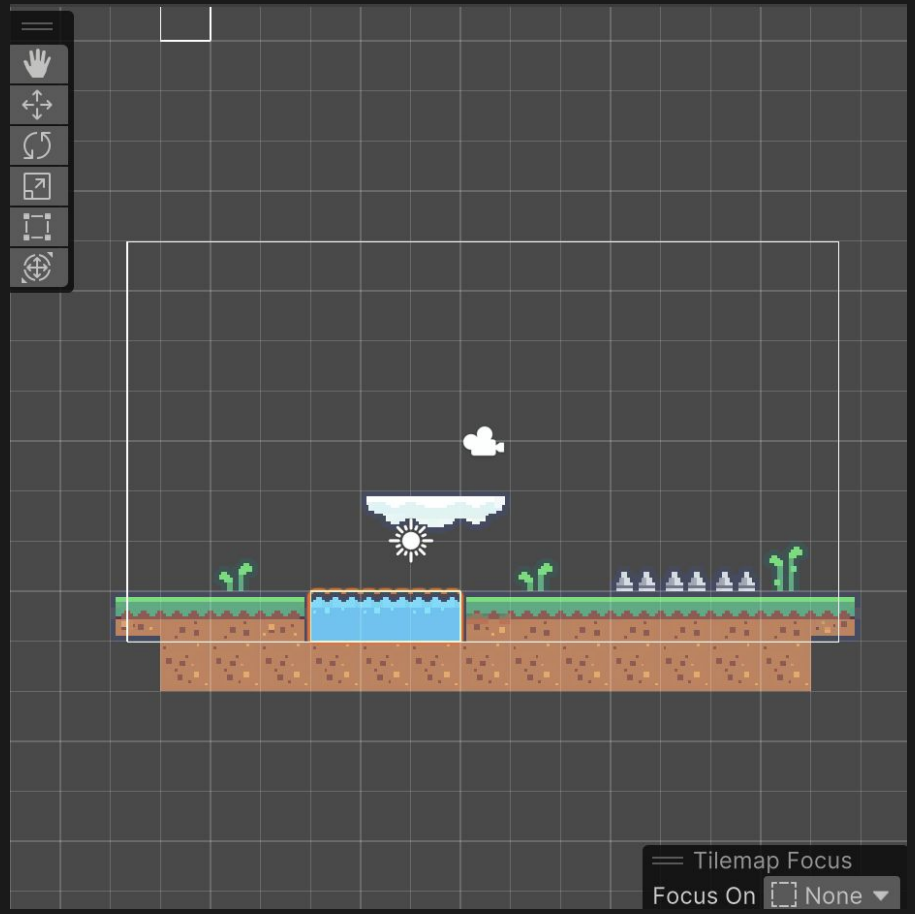


Challenge Water: Add Collision

Now, let's put your skills to the test!

Add a Tilemap Collider 2D and a Composite Collider 2D. Make sure that the Tilemap Collider is set to use the Composite Collider.

Next, set the Rigidbody 2D to 'Static' to prevent it from moving.

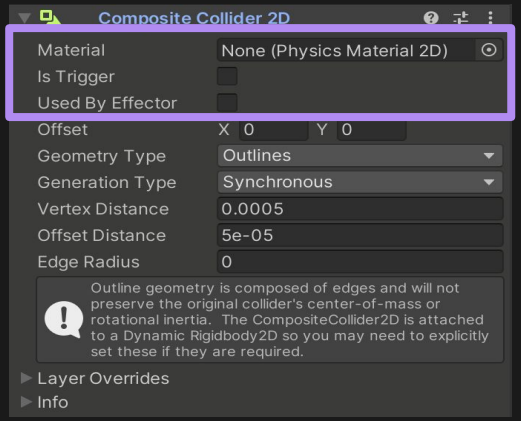


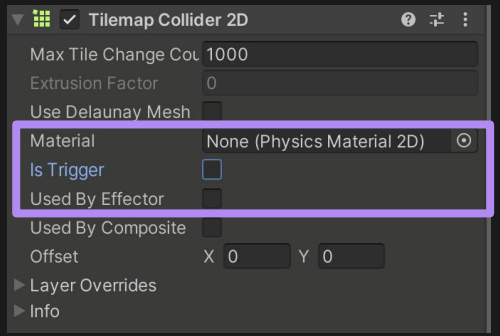
Composite Collider 2D

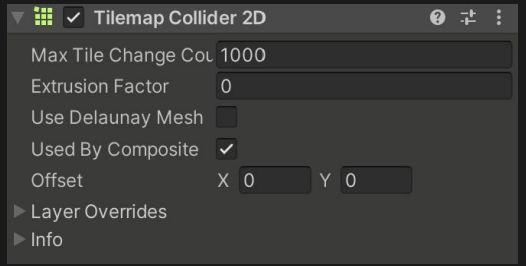
An important thing to mention is that if you want something to be triggerable or to use an effector, you now need to check the settings within the Composite Collider to make that action occur.

When you click on 'Used by Collider' in the Tilemap Collider, you'll notice that the settings for Material, Is Trigger, and Used by Effector disappear. However, the values you previously set do not simply vanish. You must be careful, as switching to the Composite Collider will clear out these values.

If you don't reapply them correctly, your game may behave strangely, as the two colliders will conflict over the actions of Material, Is Trigger, and Used by Effector.





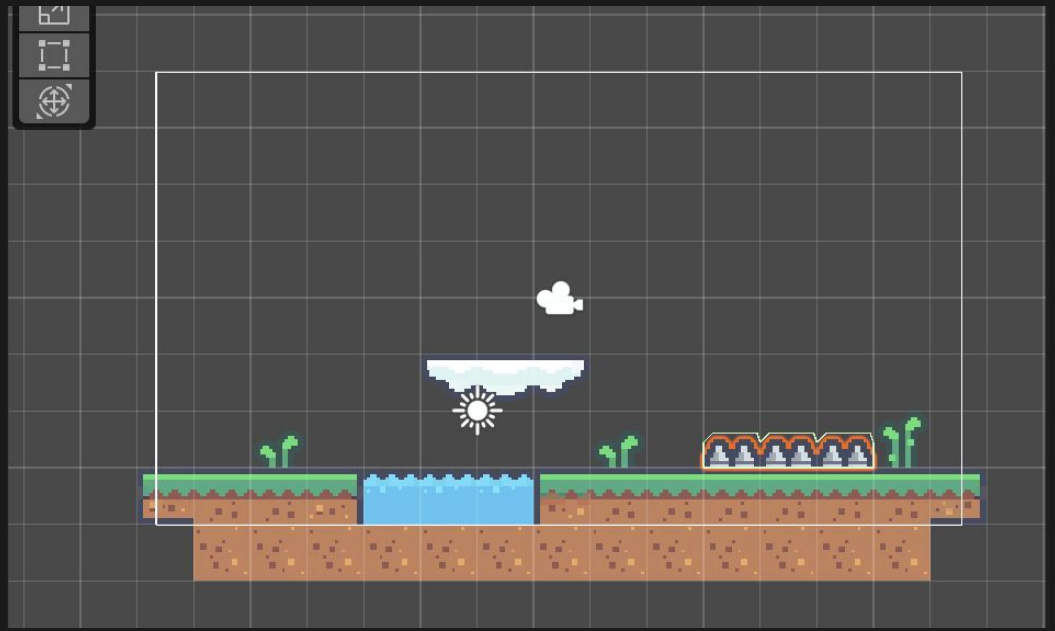


Sebastian Grygorczuk - STEM Institute at CCNY

Challenge Spikes: Is Triggable

Now, let's put your skills to the test!

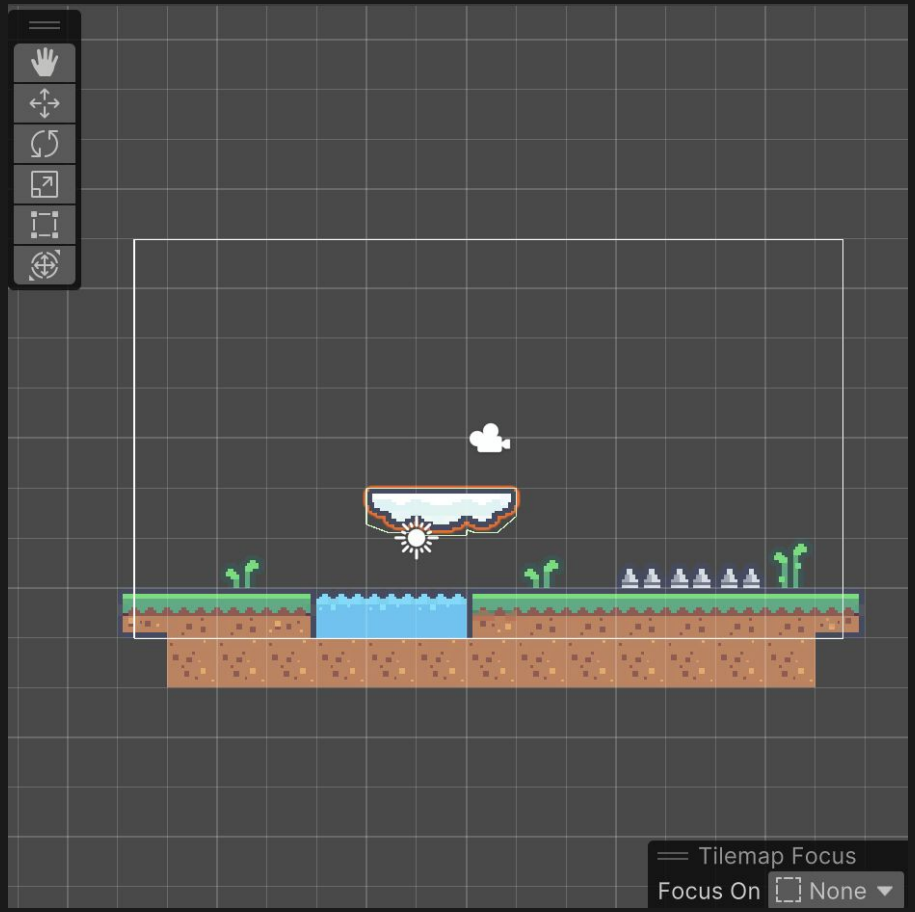
Set the Spike Composite Collider to be triggerable by enabling the 'Is Trigger' option in the Composite Collider settings



Challenge Floating Platform: Platform Effector

Now, let's put your skills to the test!

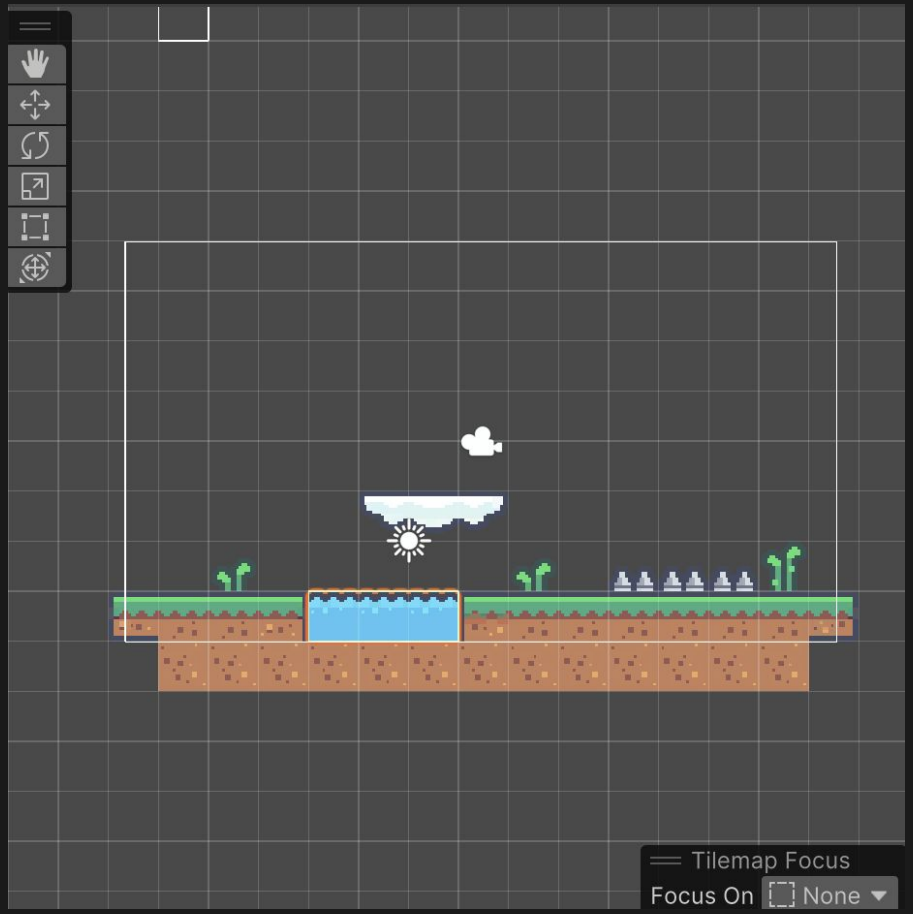
Add a Platform Effector to the Floating Platform Tilemap and connect it to the Composite Collider.



Challenge Water: Bouncy Effector

Now, let's put your skills to the test!

Add a Bouncy Effector to the Water Tilemap and connect it to the Composite Collider.



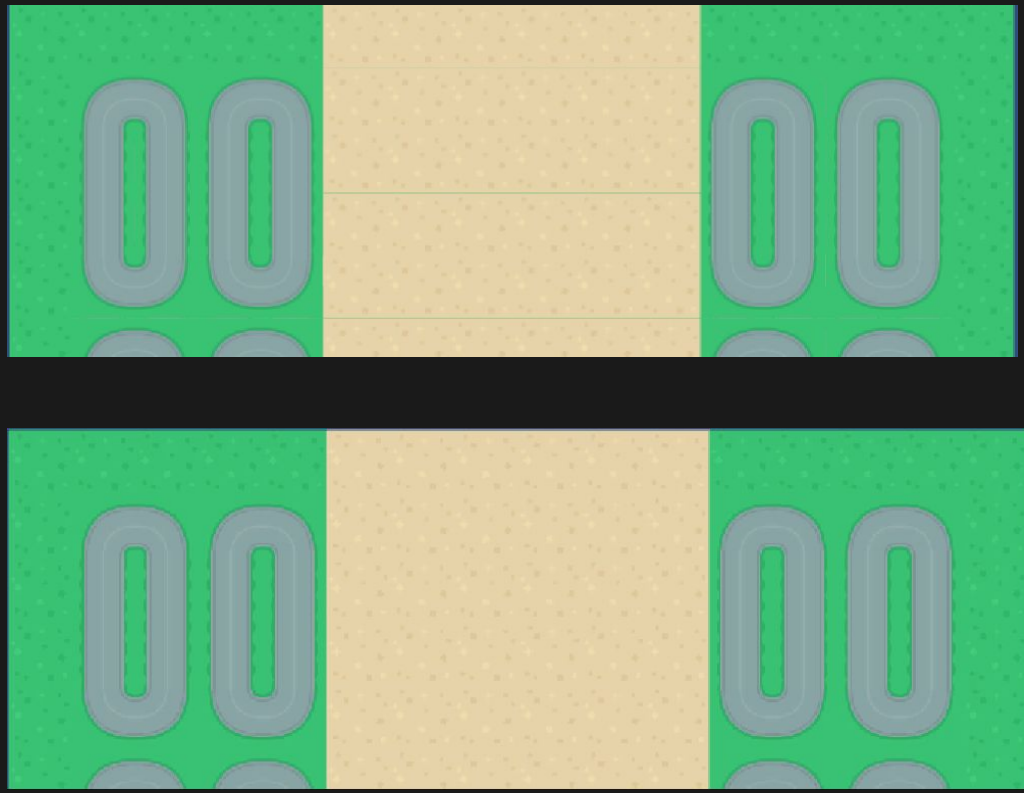
Tearing Between Tiles

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

The reason for this is that each sprite is pulled individually from the Asset folder.

Although they might have originally been part of a larger sprite sheet, when split, they become separate sprites. The process of the computer fetching these individual images and drawing them on screen can cause small gaps or tears.

A Sprite Atlas solves this issue by combining all the images into one large texture, which is then referenced whenever any sprite is drawn. This reduces the workload and helps eliminate those gaps.

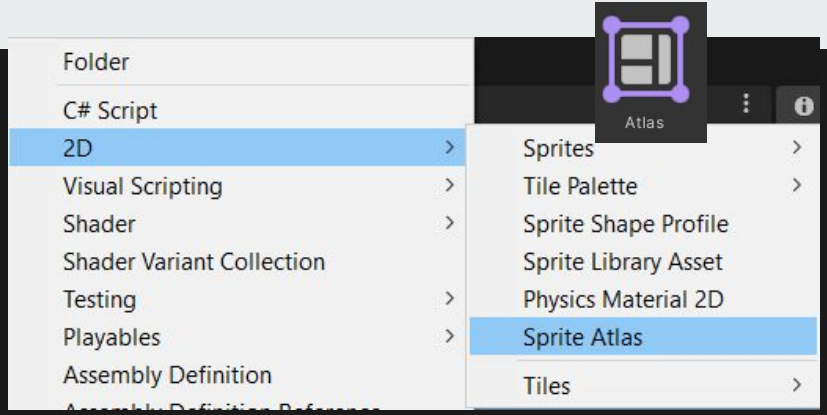


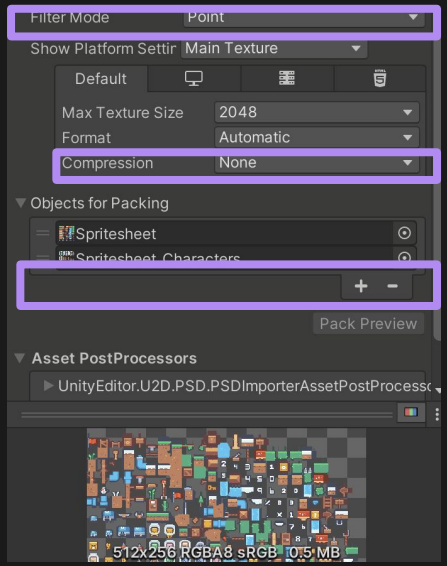
Sprite Atlas

A Sprite Atlas is a game asset, which you can create by right-clicking in the Project view and selecting 'Create' > '2D' > 'Sprite Atlas.'

In the Atlas settings, make sure to set the Filter Mode to 'Point' so the textures retain their pixel-perfect appearance when scaled. Also, set the Compression to 'None' to prevent any blurriness in the sprites.

Afterward, you can add the sprites you want to include in the Atlas. Unity will automatically generate a new texture that can be used directly in your game



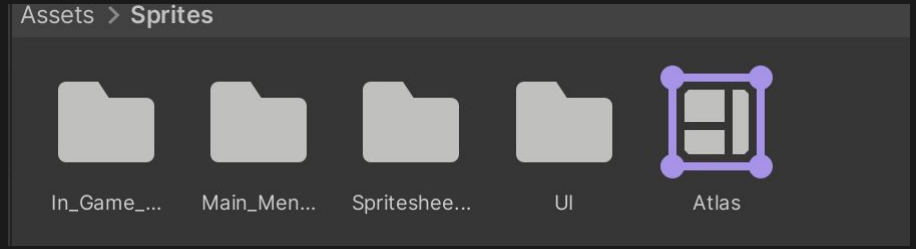


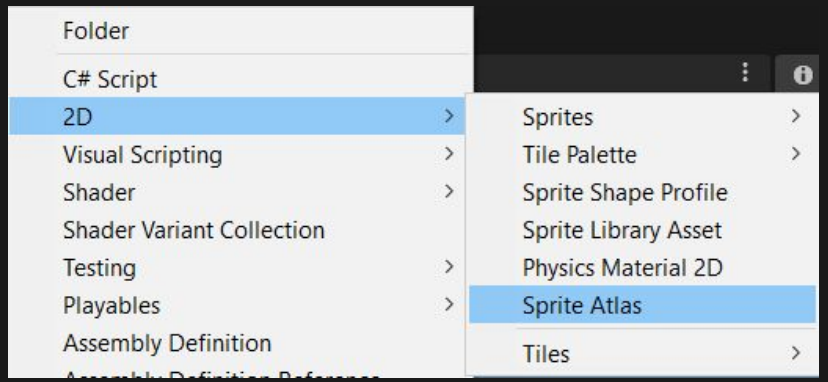
Challenge Sprite Atlas

Now, let's put your skills to the test!

Create a Sprite Atlas inside the 'Assets > Pixel_Quest > Sprites' folder.

Add the sprite sheet that you cut out to the Atlas. Then, set the Filter Mode to 'Point' and the Compression to 'None'





Challenge Background

Now, let's put your skills to the test!

Create a Sprite Atlas inside the 'Assets > Pixel_Quest > Sprites > In_Game_Backgrounds' folder.

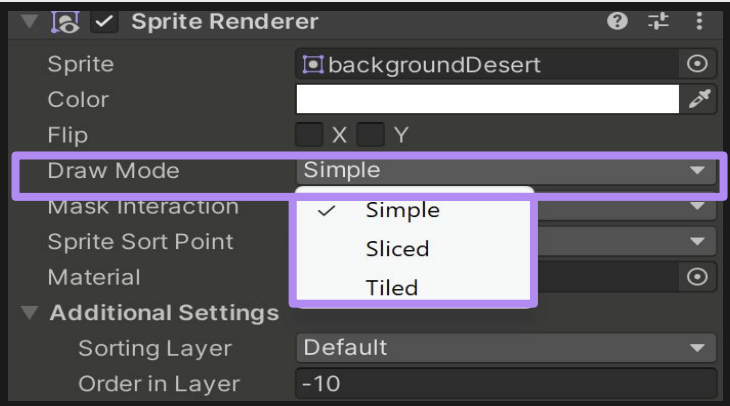
Choose a background and add it to your game. Then, scale it to the appropriate size to fit your scene.

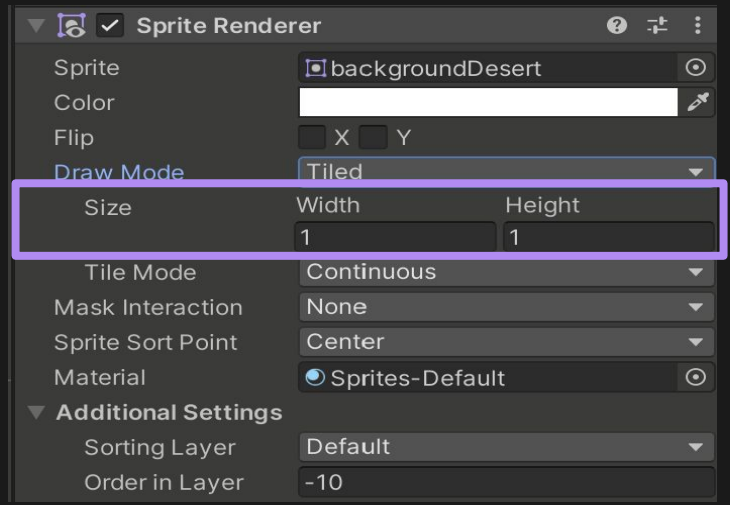
Sprite Renderer Draw Mode

Sprite Renderers offer many options that we haven't covered yet, one of which is the Draw Mode. This controls how the image is displayed, stretched, or altered.

- **Simple** will keep the image exactly as it is, without any modifications.
- **Sliced** allows you to expand or shrink specific parts of the image, which is mainly used for UI image processing.
- **Tiled** enables you to create copies of the image, treating it like a tile, so rather than stretching, the image will repeat seamlessly next to each other.

When switching to Tiled, you'll be able to adjust the width and height. We will use the width to expand the image and repeat it, making it longer.

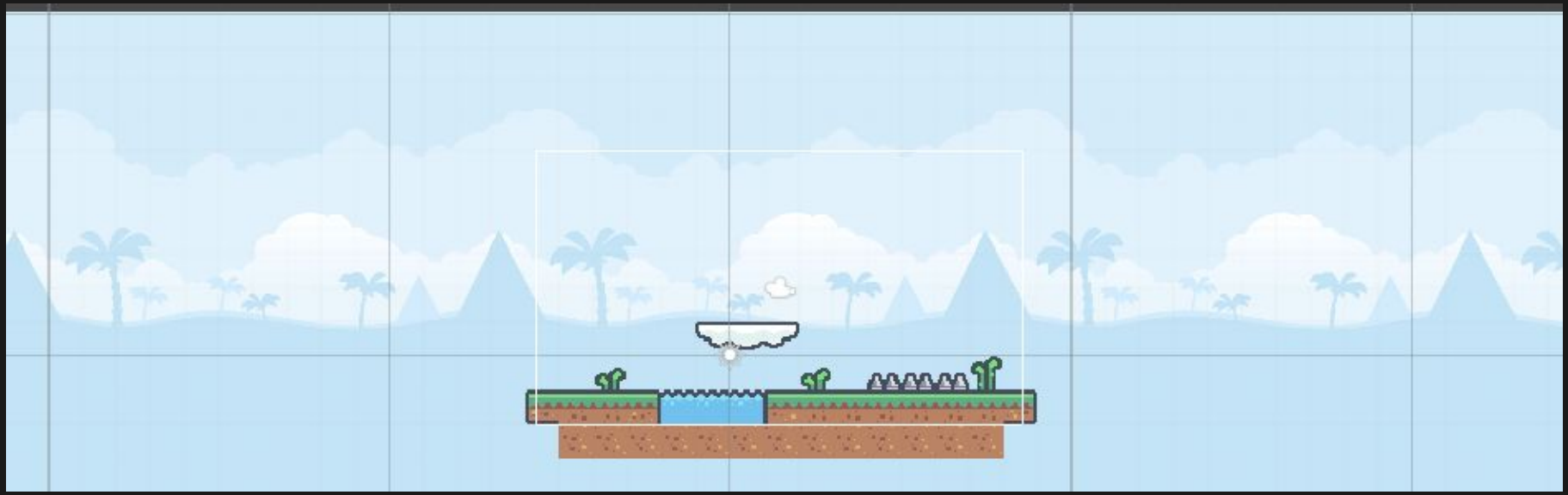




Challenge Background Extension

Now, let's put your skills to the test!

Change the Sprite Renderer's Draw Mode to **Tiled** and increase the width to extend the image.



Challenge Add Old Prefabs

Now, let's put your skills to the test!

To test out our new world, add the player and the bounce pad, then playtest your game.

In the next class, we'll create a new player character that fits the aesthetic of the game better.