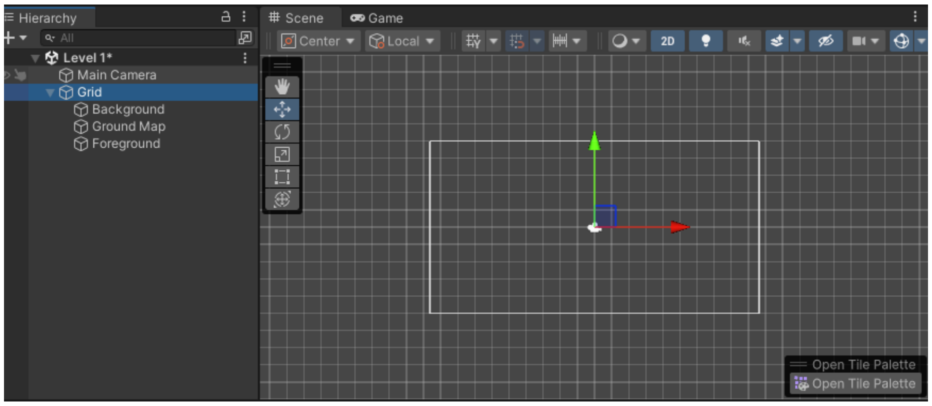
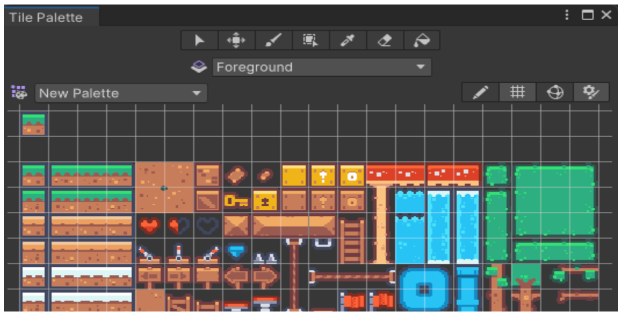
Setting up the Level:

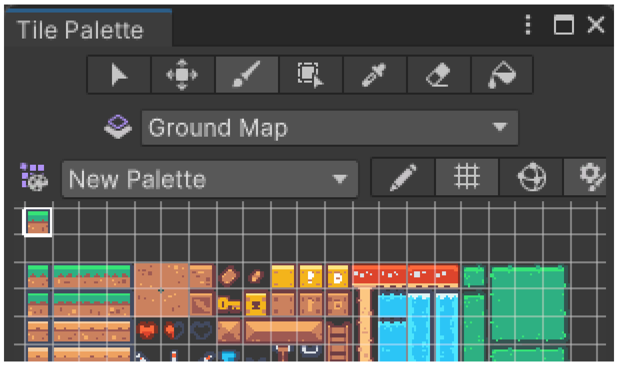
1. Right-Click on the Hierarchy and create a new Tile Map by selecting **2D Object > Tilemap > Rectangular**
   Do this 2 more times, and rename each as: **Background, Ground Map, Foreground**



2. Open the **Tile Palette** by either clicking on it in the bottom right corner or selecting **Window Tab > 2D > Tile Palette**

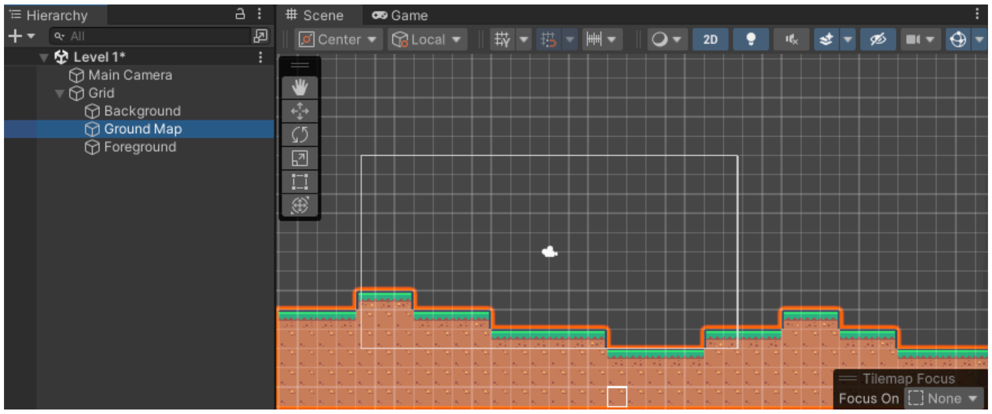


3. First, ensure that the **Ground Map** Tilemap is selected inside the **Tile Palette**, then use the **Paint Brush Tool** to select and draw the Tile you would like onto the **Scene.**

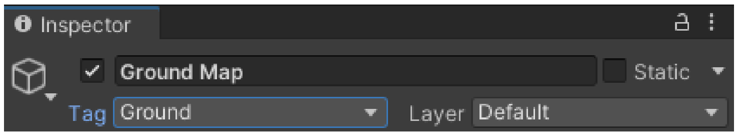


(You can use the **Ground Tile Rule** you created to automatically format the tiles)

Your Game should look like this now!



4. Your game now needs to recognize it as **"Ground"** by **tagging** it and adding a **Collider 2D.**
    - After selecting the **Ground Map**, create and add a **"Ground" Tag** to it

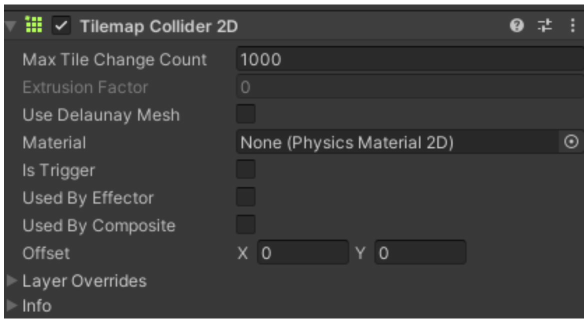


Inspector

Ground Map | Static

Tag: Ground | Layer: Default

    - Then, use the **Add Component button** inside the **Inspector** to add the **Tilemap Collider 2D**

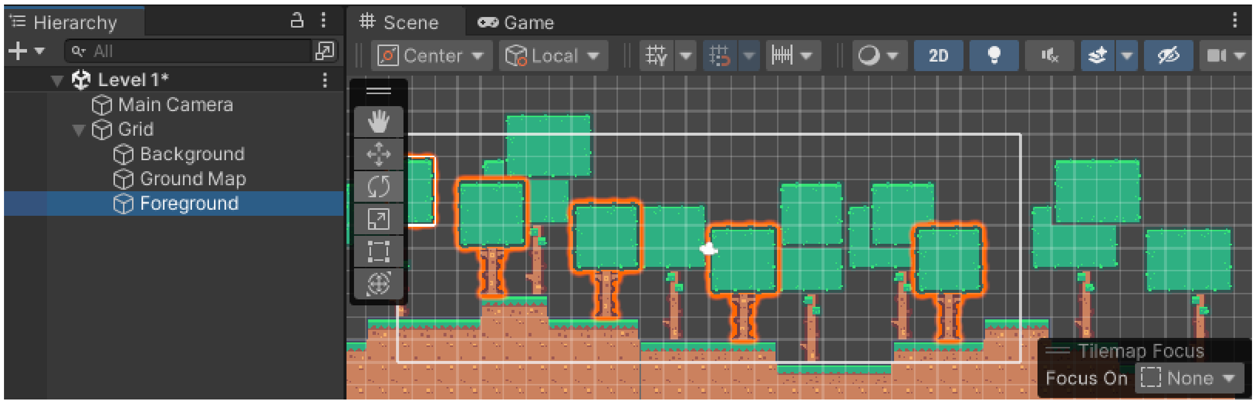


Tilemap Collider 2D

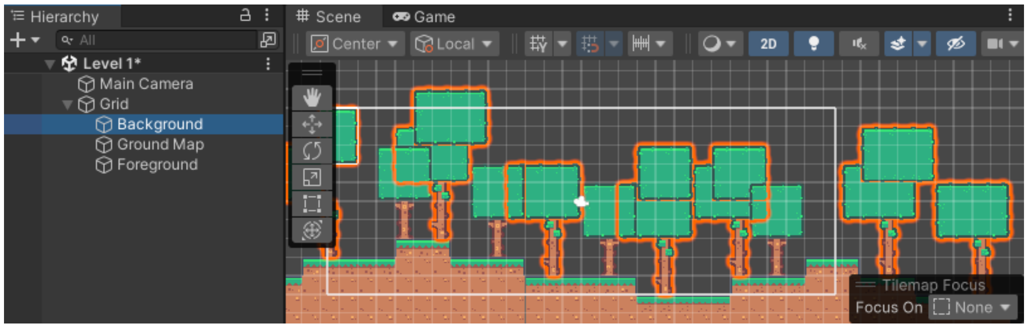
Max Tile Change Count	1000
Extrusion Factor	0
Use Delaunay Mesh	☐
Material	None (Physics Material 2D)
Is Trigger	☐
Used By Effector	☐
Used By Composite	☐
Offset	X 0 Y 0
Layer Overrides	
Info	

IMPORTANT! - When designing the **Background** and **Foreground,** make sure that the **correct tile map is selected** inside the **Tile Palette** <u>**before**</u> you start painting!

5. Create a few objects in the **Foreground**, and set the **Z Position to -1**



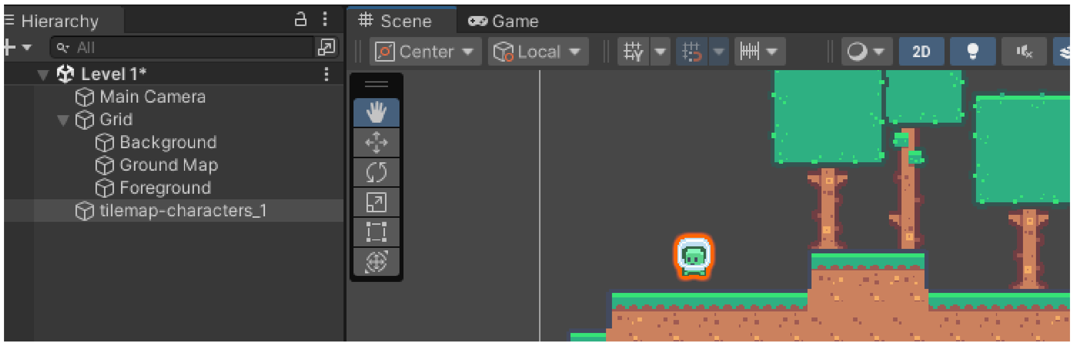
6. Create a few objects in the **Background**, and set the **Z Position to 1**



Since these Tilemaps lack a Tilemap Collider 2D, the player should walk right through them!

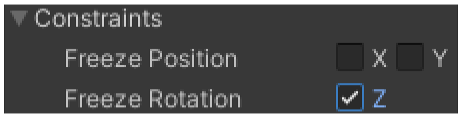
<u>Setting up the Player:</u>

1. Find a **Sprite** for the **Player** Character in the Assets **OR Import a new Sprite** from your computer, and drag them into the **Scene**



2. **Rename** the Game Object to **"Player"**
3. **Tag** the Game Object with **"Player"**
4. We need to add **PHYSICS** to the **Player** so they respond to "gravity"
    - Add the **Rigidbody 2D** component to the **Player** Object

    - In the the **Constraints** dropdown in the component, make sure to **Freeze Rotation**


Constraints
Freeze Position X Y
Freeze Rotation ✓ Z

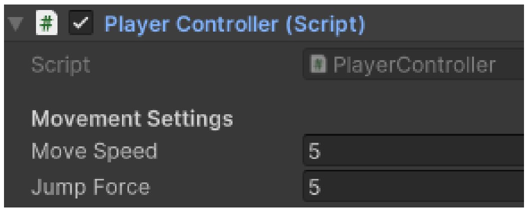
5. We need to add **COLLISION DETECTION** to the **Player** so they hit other objects.
    - Add the **Capsule Collider 2D** component to the **Player** Object

6. To control the player, we need to add the **Player Controller Script** to the **Player** so the user can use the keyboard to move the player.

7. To make the camera follow the player,
    - Move the **Main Camera** inside of the **Player Object** so that the camera becomes a **Child Object** of the **Player.**
    - Reset the **Main Camera's Transform** component
    - Set the **Z Position to -2** (so that it is in front of all other object)
    - Make any final adjustments to the Camera Position as you like

Play Testing your Player Controller:

1. **Play** the game and **observe** the way the player moves.
2. Select the **Player Object** in the **Hierarchy** and look for the **Player Controller Script** inside the Inspector



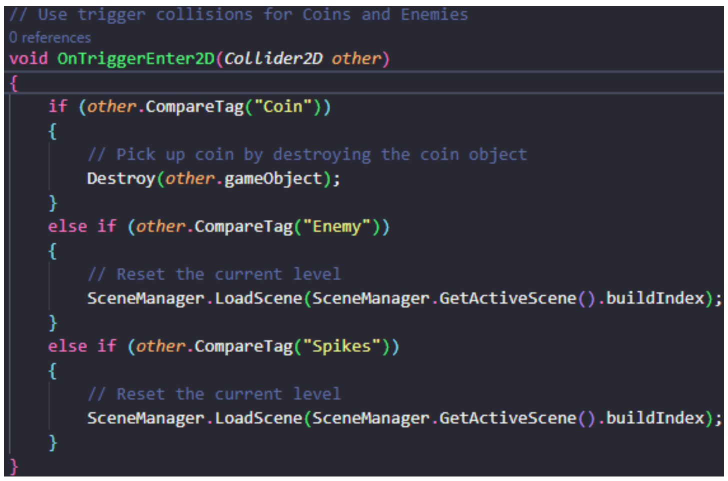
Player Controller (Script)

Script	PlayerController
Movement Settings	
Move Speed	5
Jump Force	5

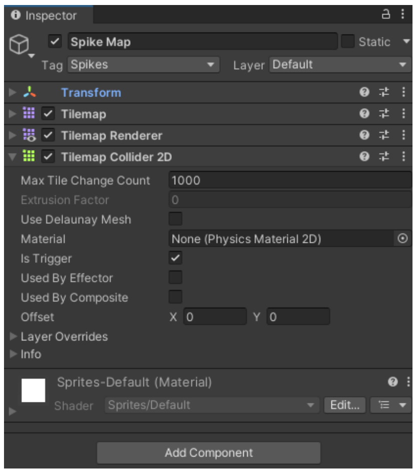
- You can adjust these variables while you play the game to test values fast

YOUR TURN: Create a new Tilemap

- ☐ Create a **SPIKE MAP**! (Hint: **2D Object > Tilemap > Rectangular)**
    - ☐ Paint Spikes onto the Spike Map Tilemap
    - ☐ Add a "Spikes" **Tag** onto it
    - ☐ Give it a **Collider**
    - ☐ Toggle the **Is Trigger** variable to **True**
    - ☐ Adjust the **Z position** so it appears in front of the background
- ☐ We need to adjust the **Player Controller Script** now so it resets the level when coming into contact with the Spikes. Add this to the **OnTriggerEnter2D function**



```csharp
// Use trigger collisions for Coins and Enemies
0 references
void OnTriggerEnter2D(Collider2D other)
{
    if (other.CompareTag("Coin"))
    {
        // Pick up coin by destroying the coin object
        Destroy(other.gameObject);
    }
    else if (other.CompareTag("Enemy"))
    {
        // Reset the current level
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
    else if (other.CompareTag("Spikes"))
    {
        // Reset the current level
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
}
```



Inspector

Spike Map ✓ — Static
Tag: Spikes — Layer: Default

Transform
Tilemap ✓
Tilemap Renderer ✓
Tilemap Collider 2D ✓

Max Tile Change Count	1000
Extrusion Factor	0
Use Delaunay Mesh	☐
Material	None (Physics Material 2D)
Is Trigger	✓
Used By Effector	☐
Used By Composite	☐
Offset	X 0 Y 0

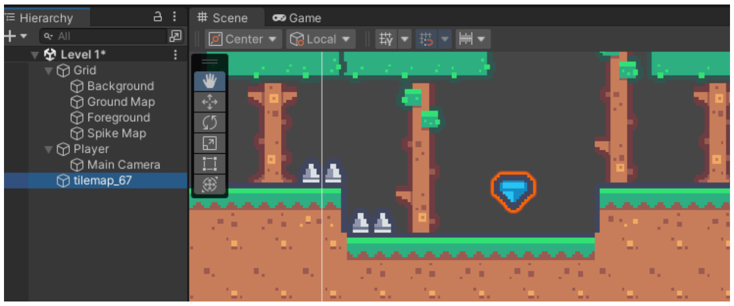
Layer Overrides
Info

Sprites-Default (Material)
Shader: Sprites/Default — Edit...

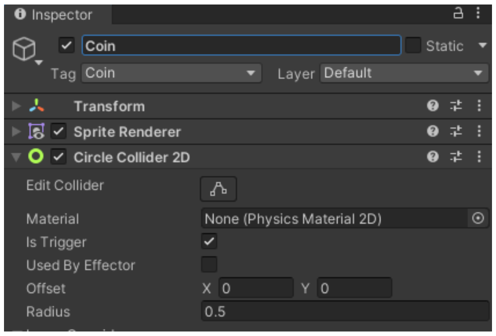
Add Component

<u>Creating Collectible Items:</u> (This will show you how to create a coin)
This is VERY SIMILAR to the way we created the Player Object

1. Find a **Sprite** for the **Player** Character in the Assets **OR Import a new Sprite** from your computer, and drag them into the **Scene**

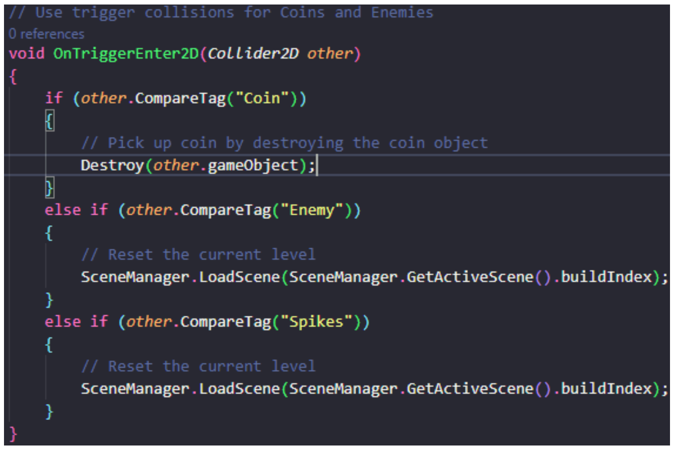


2. **Rename** the Game Object to **"Coin"**
3. **Tag** the Game Object with **"Coin"**
4. We need to add **COLLISION DETECTION** to the **Coin** so they hit other objects.
   - Add the **Circle Collider 2D** component to the **Coin** Object
   - Toggle the **Is Trigger** variable to **True**



Inspector
Coin — Static
Tag: Coin — Layer: Default
Transform
Sprite Renderer
Circle Collider 2D
Edit Collider
Material: None (Physics Material 2D)
Is Trigger: ✓
Used By Effector:
Offset: X 0 Y 0
Radius: 0.5

5. The Player Controller Script already takes care of what happens if the player hits the Coin.

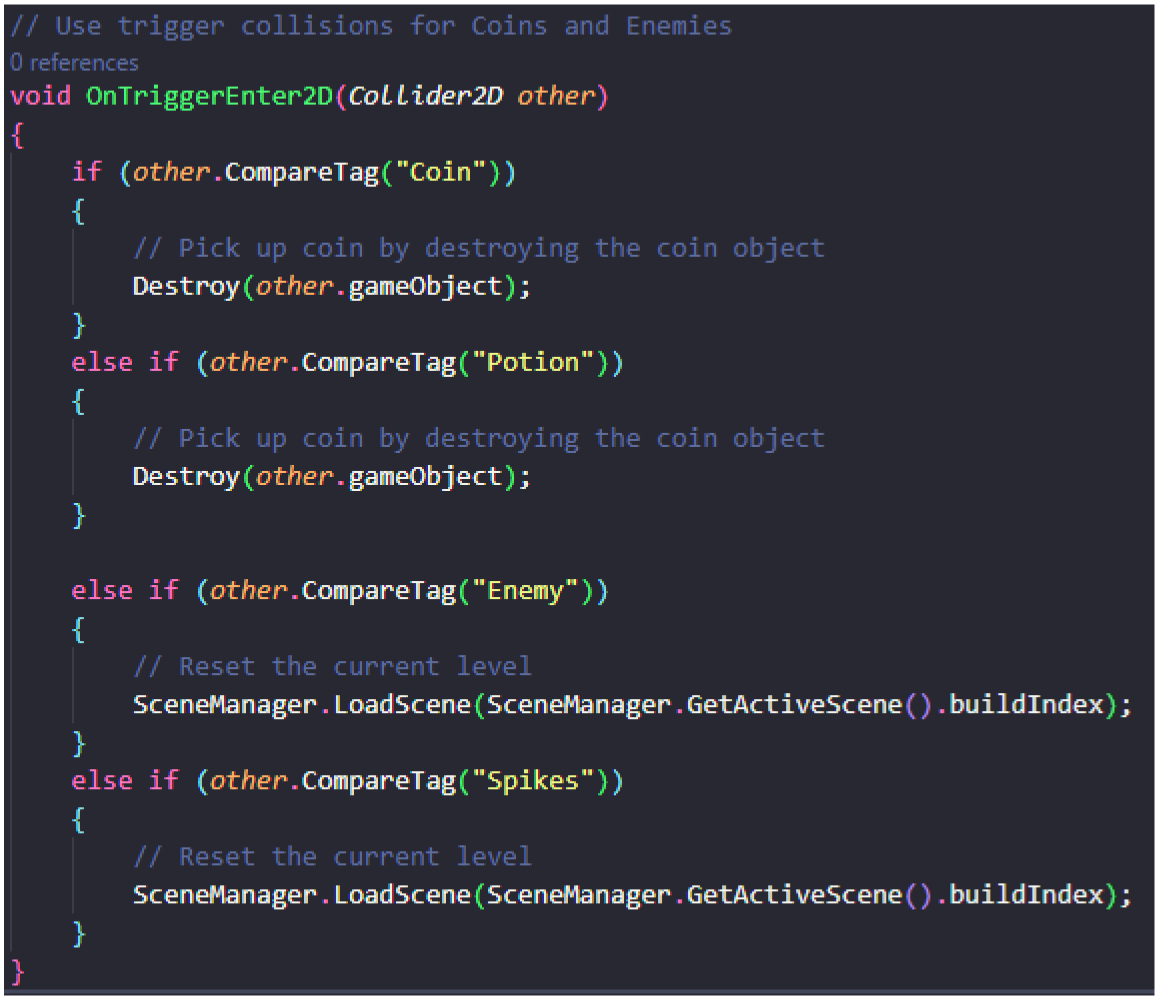


```
// Use trigger collisions for Coins and Enemies
0 references
void OnTriggerEnter2D(Collider2D other)
{
    if (other.CompareTag("Coin"))
    {
        // Pick up coin by destroying the coin object
        Destroy(other.gameObject);
    }
    else if (other.CompareTag("Enemy"))
    {
        // Reset the current level
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
    else if (other.CompareTag("Spikes"))
    {
        // Reset the current level
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
}
```

<u>YOUR TURN: Create a new Collectible</u>

- ☐ Create another Collectible Item which the player can pick up

Here's how you can code the **Player Controller** to allow the player to pick up **Potions:**



```
// Use trigger collisions for Coins and Enemies
0 references
void OnTriggerEnter2D(Collider2D other)
{
    if (other.CompareTag("Coin"))
    {
        // Pick up coin by destroying the coin object
        Destroy(other.gameObject);
    }
    else if (other.CompareTag("Potion"))
    {
        // Pick up coin by destroying the coin object
        Destroy(other.gameObject);
    }

    else if (other.CompareTag("Enemy"))
    {
        // Reset the current level
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
    else if (other.CompareTag("Spikes"))
    {
        // Reset the current level
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
}
```