

Games Programming Taster

Hide and Seek

An Introduction to Unity Development



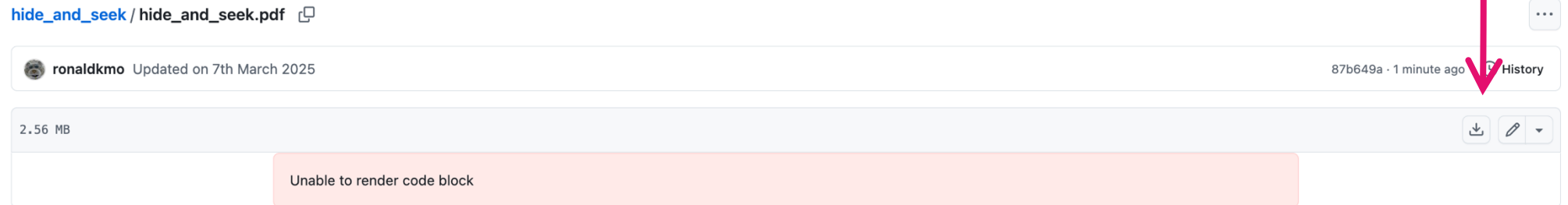
**University of
Sunderland**

Dr. Ronald Mo

ronald.mo@sunderland.ac.uk

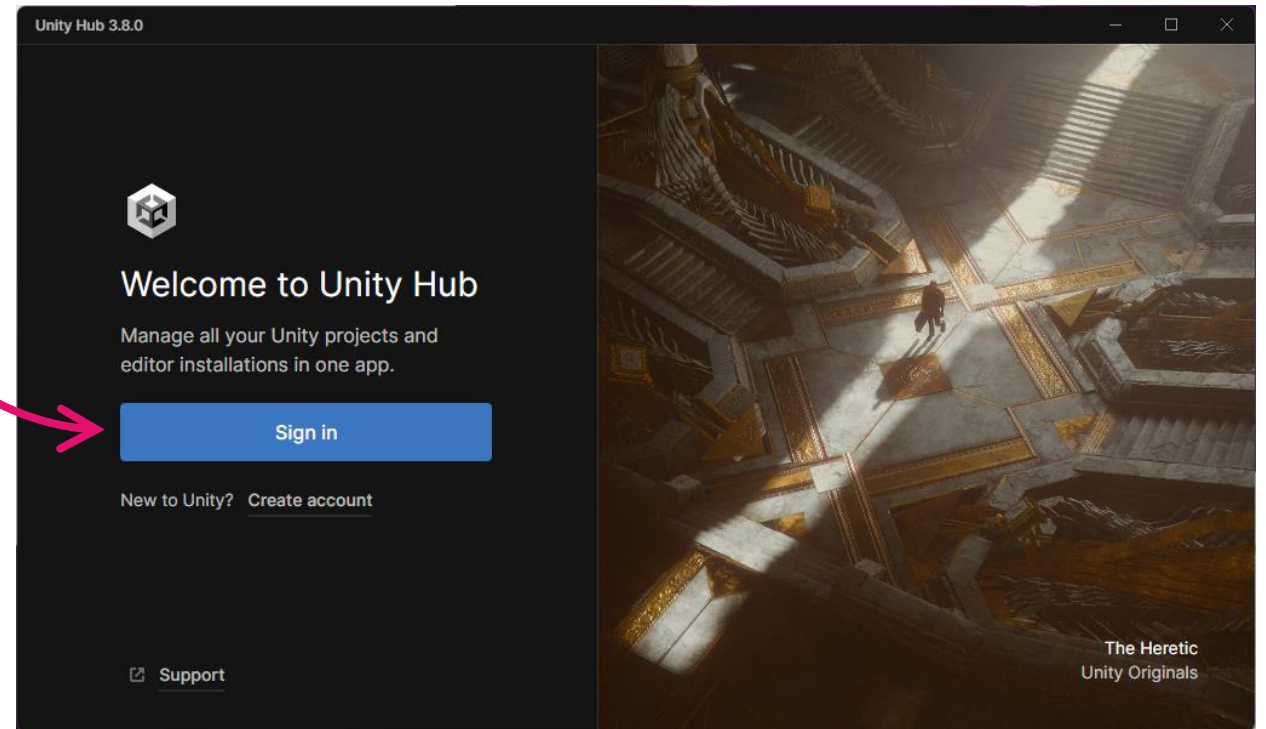
Download this PowerPoint

- Go to
 - https://github.com/ronaldkmo/hide_and_seek/blob/main/hide_and_seek.pdf
- Click this



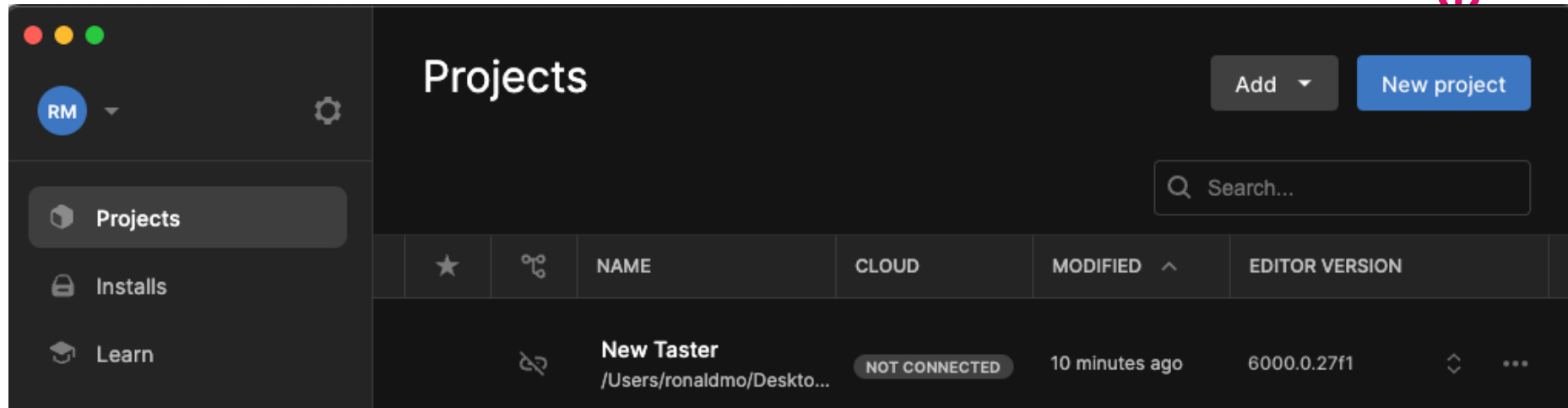
Create a Unity Account

- Open Unity Hub
- Create an account...
- ...and Sign in 😊



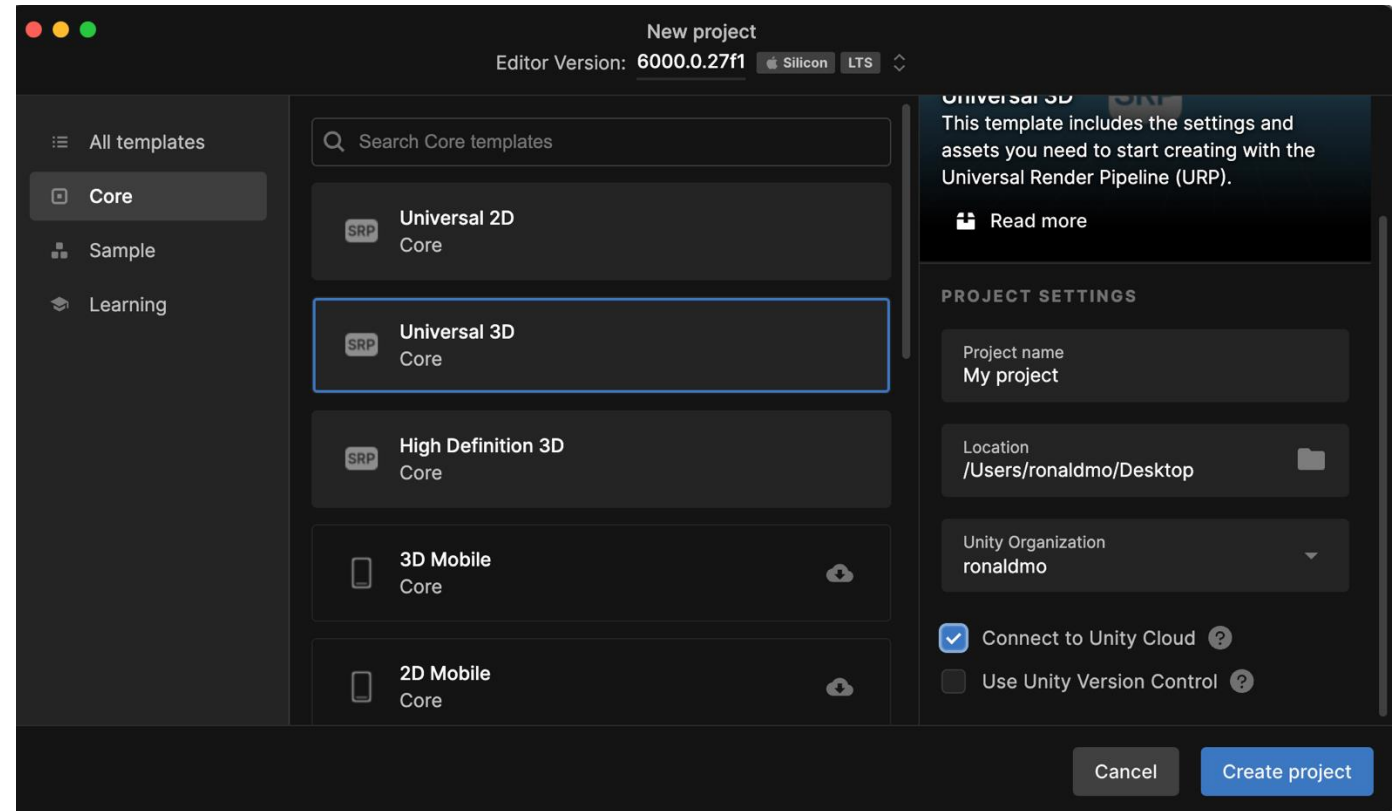
Create a New Project

- Create a new project by clicking this button

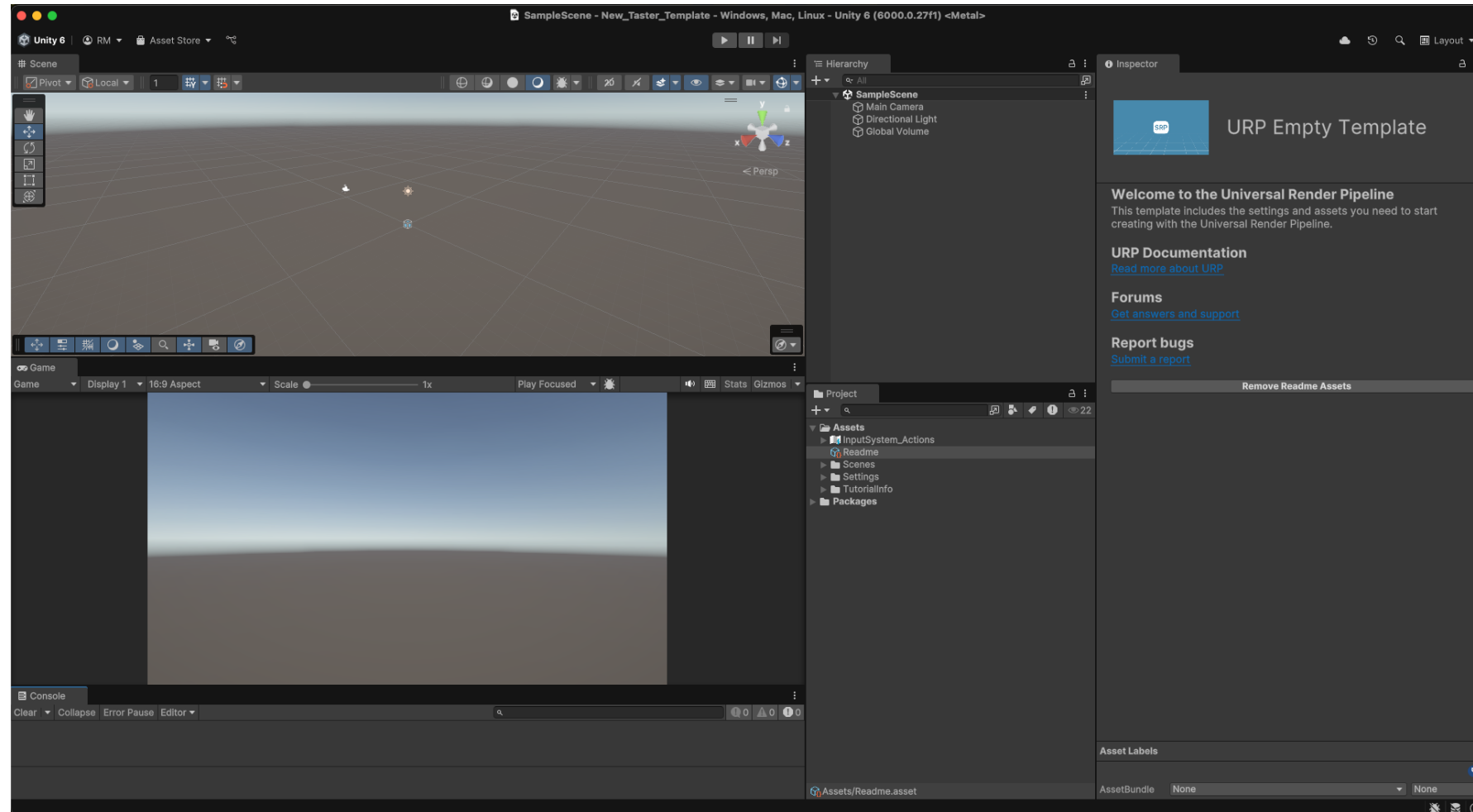


Create a New Project

- Select **Universal 3D**
- Give your project a name
- Click *Create project*

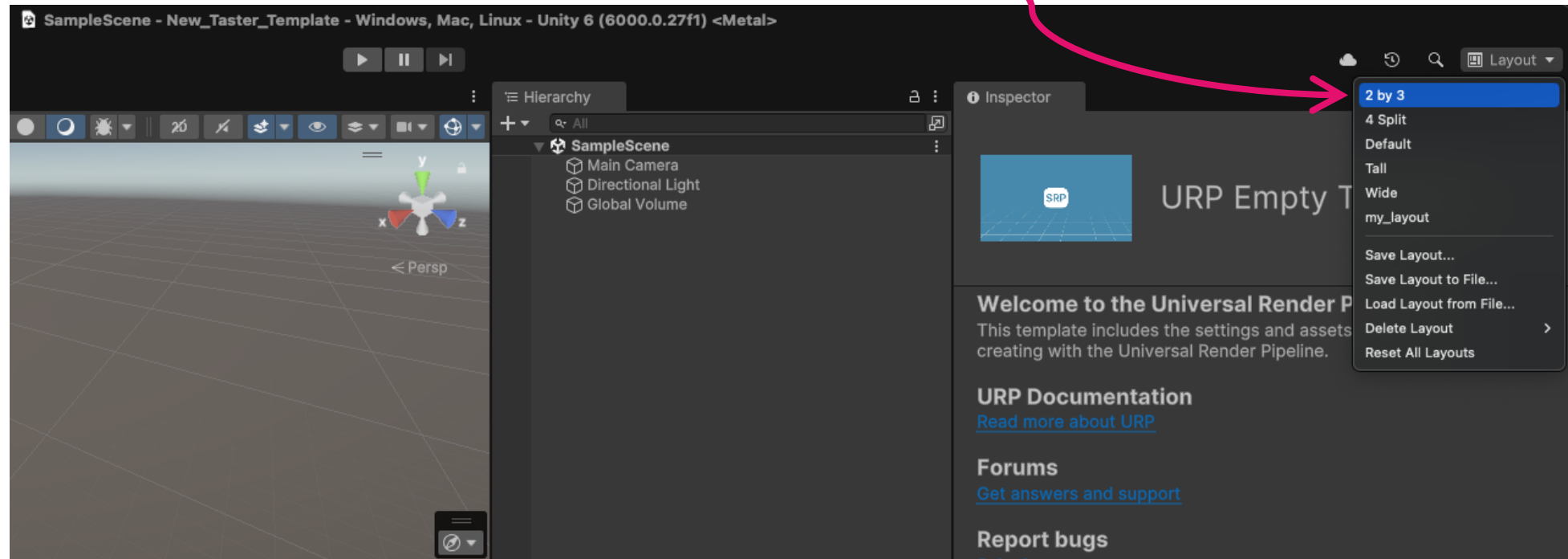


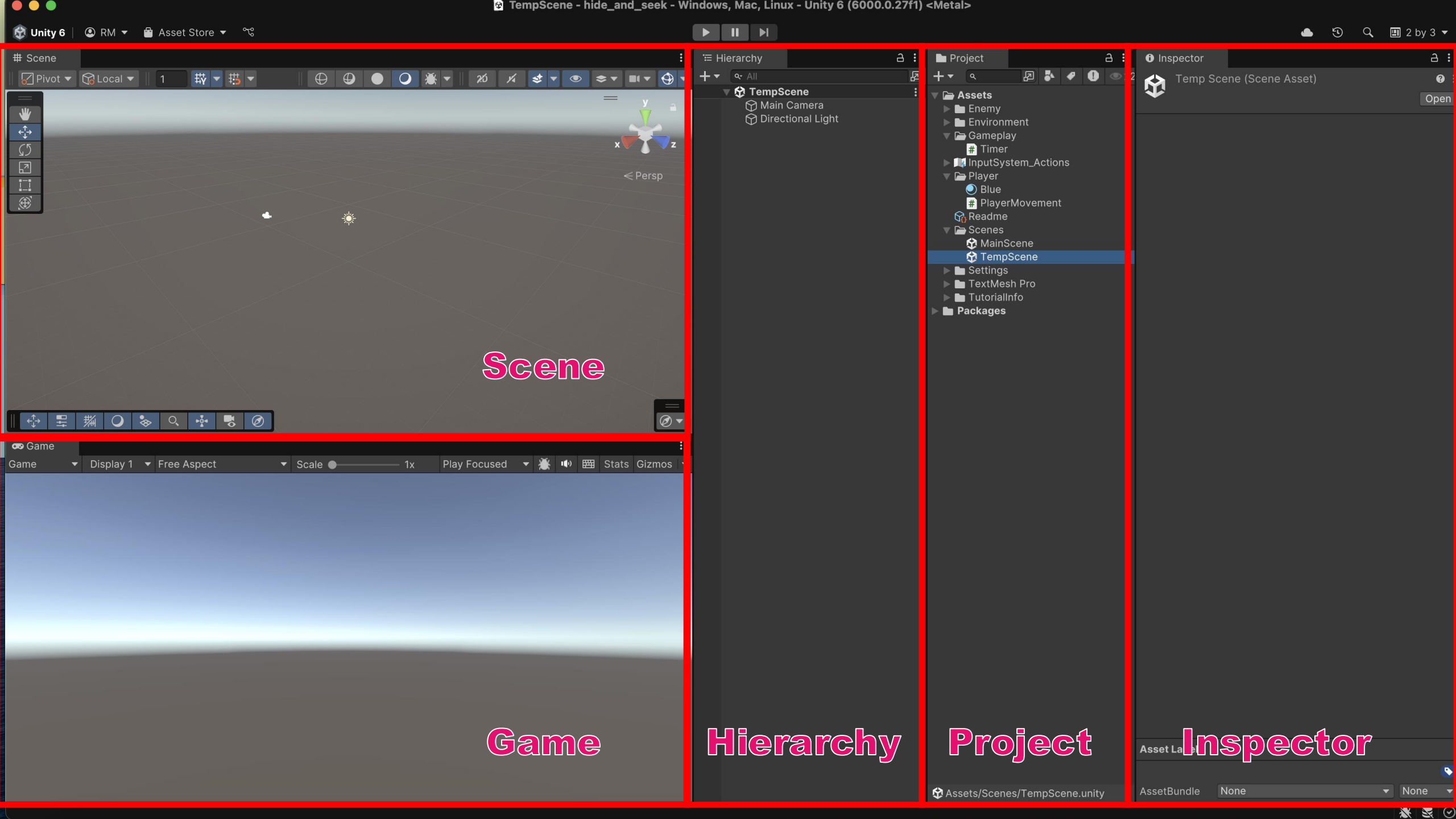
Unity 6



Change the Layout

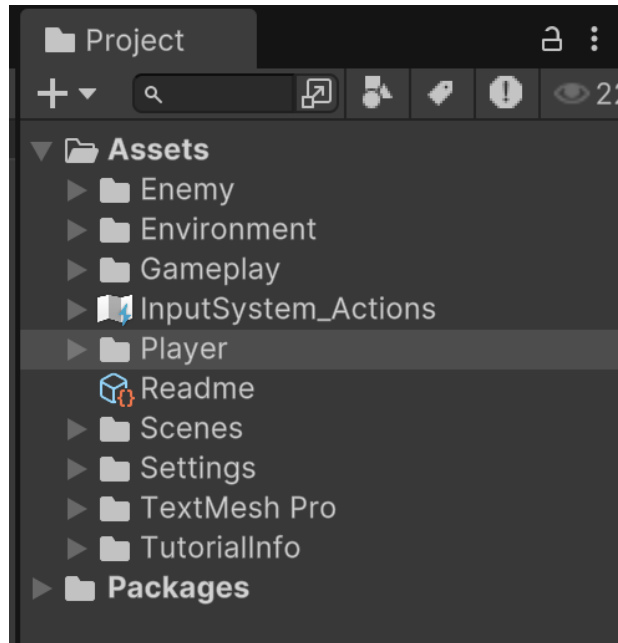
- Let's change the layout to 2 by 3



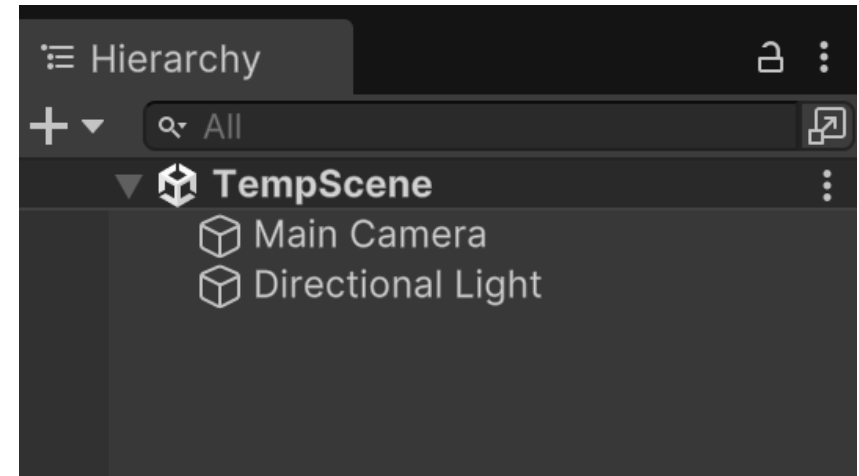


Where is Everything?

- **Project** pane contains all the elements we can use to develop our game

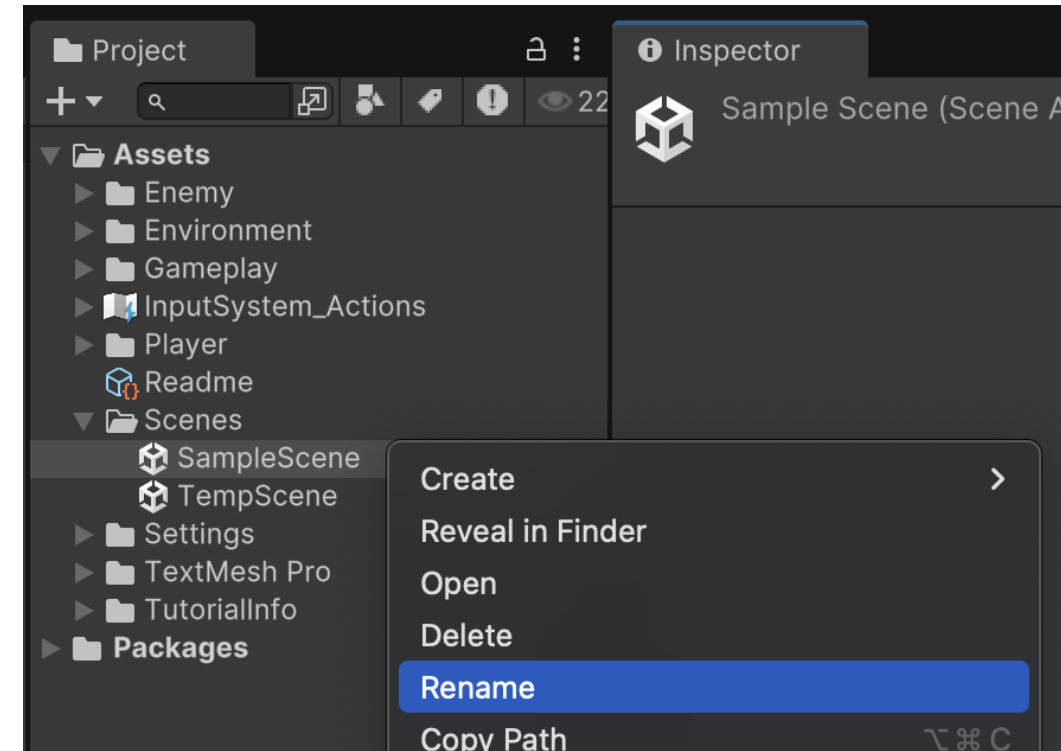


- **Hierarchy** pane contains all the elements in the game



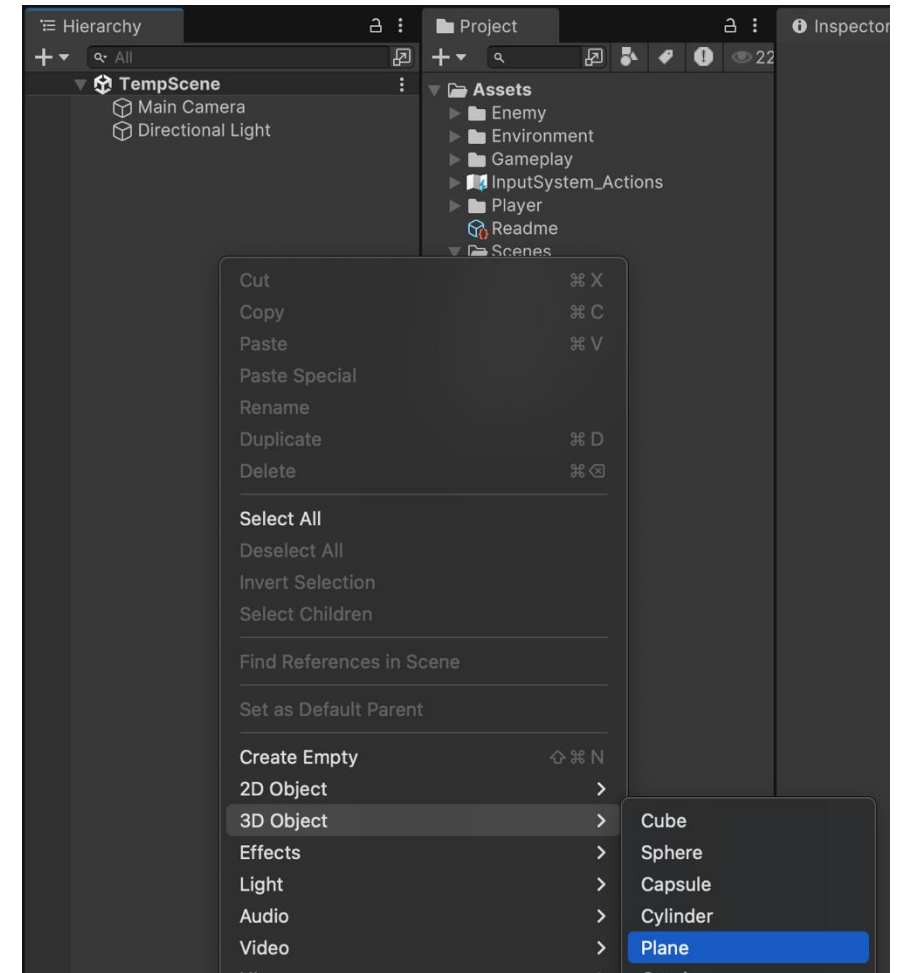
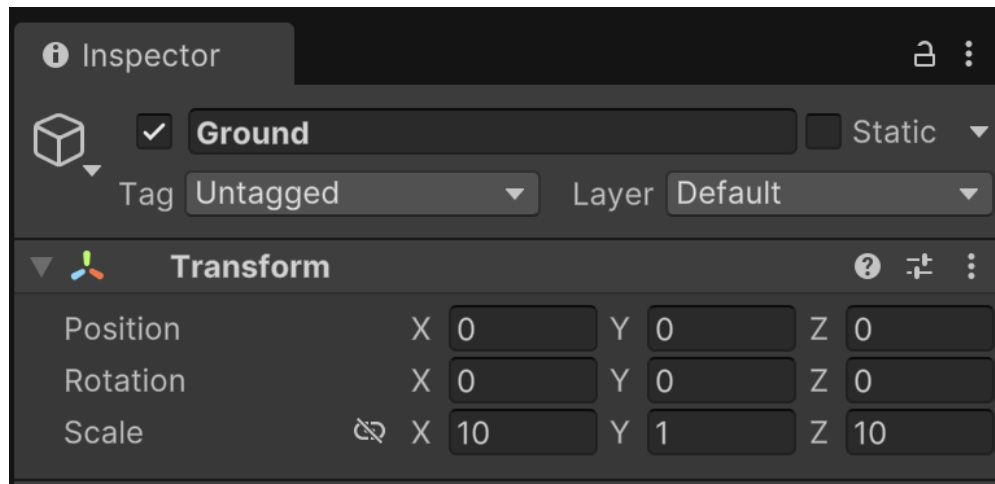
Set Up the Game Environment

- Rename the current scene to something more meaningful (e.g., MainScene)
 - Assets/Scenes
 - Right-click on SampleScene and choose *Rename*



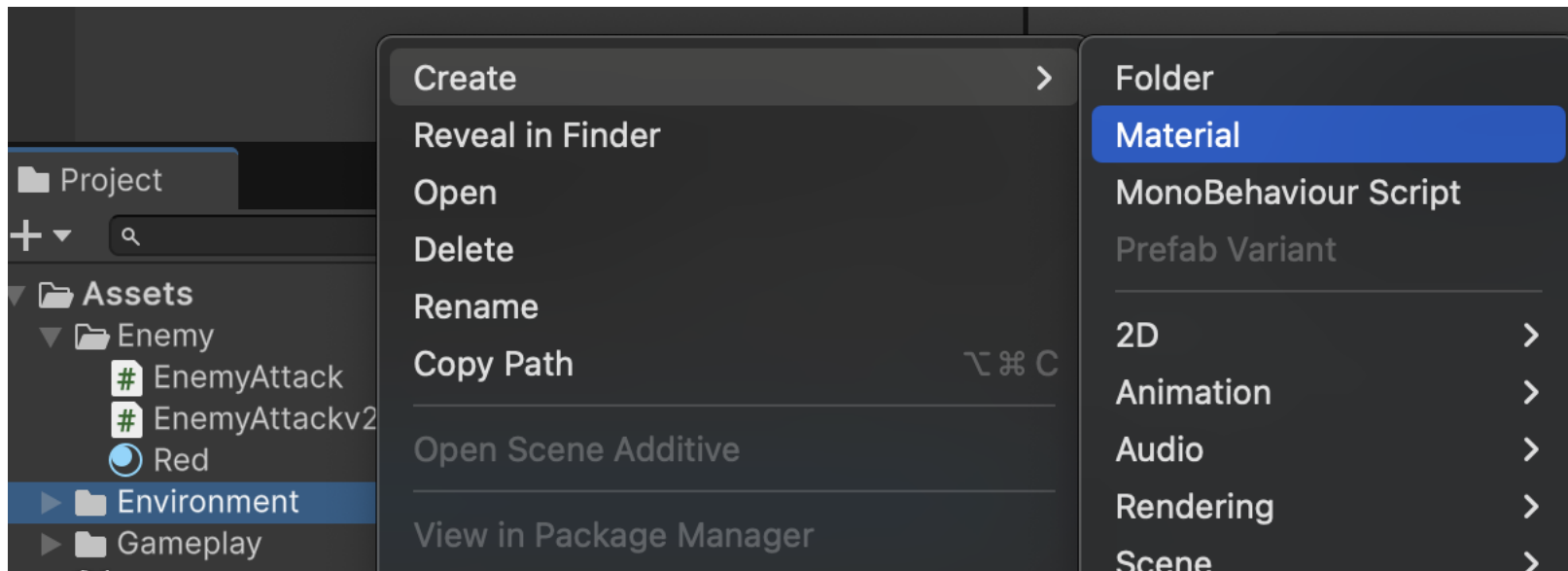
Set Up the Game Environment

- In the **Hierarchy** pane, add a new plane, and name it *Ground*
 - Right-click -> 3D Object -> Plane
- Go to the **Inspector** pane and follow the below setting



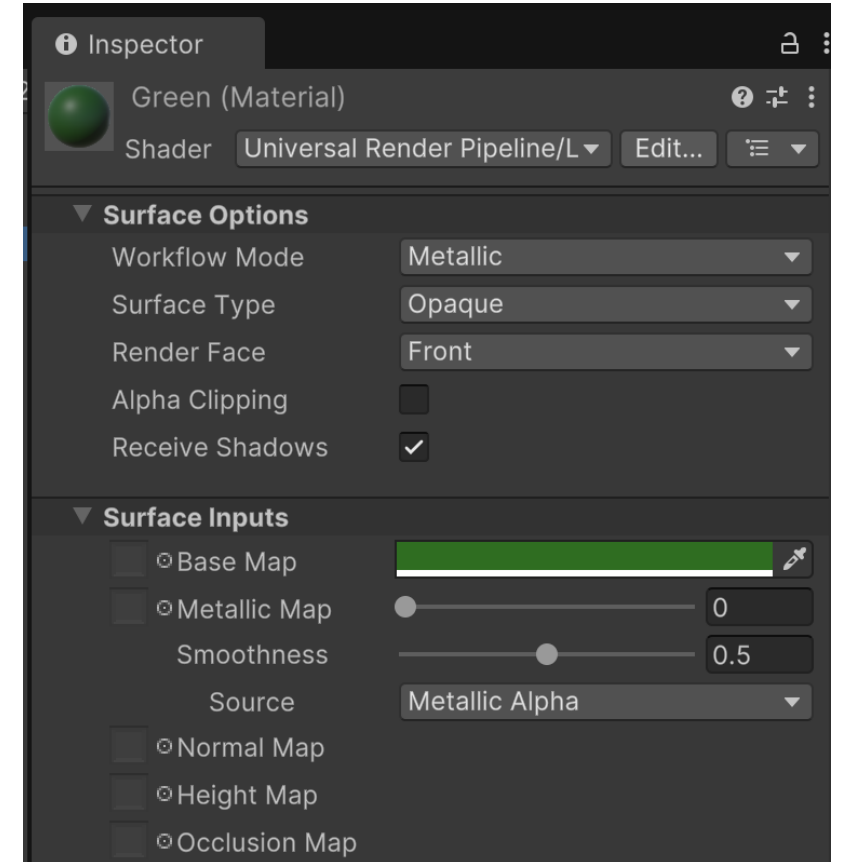
Set Up the Game Environment

- In the **Project** pane, create a folder called *Environment* under the *Assets* folder
- Inside *Environment*, create a material and call it *Green*



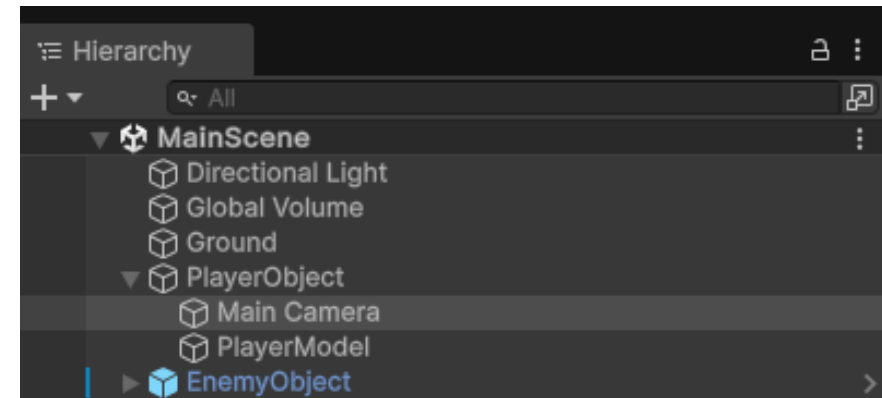
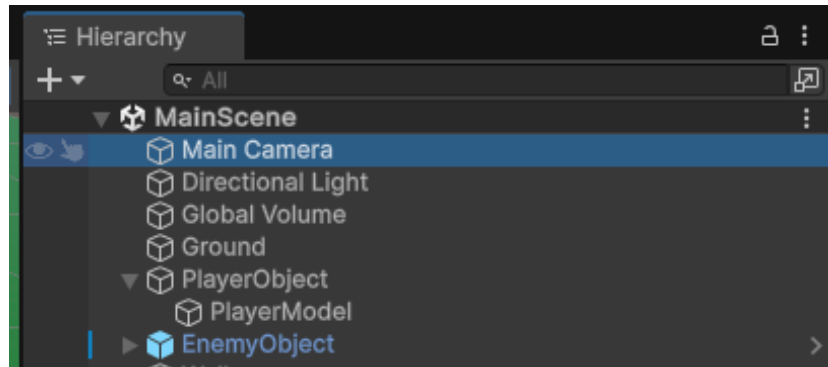
Set Up the Game Environment

- While selecting the created *Green* material, inside the **Inspector** pane, change its Base Map to **Green**
- Drag the *Green* material to the ground in the **Scene** pane



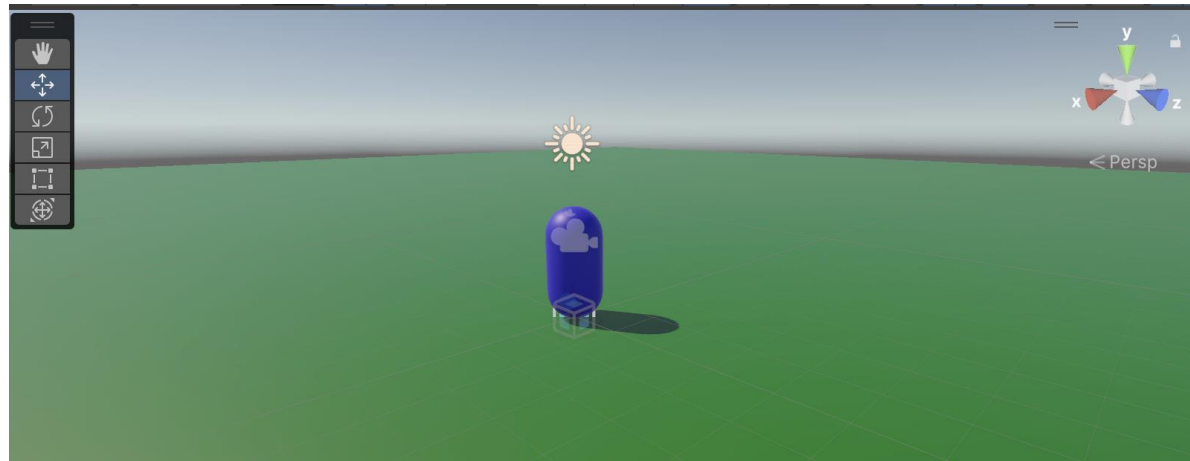
Create the Player

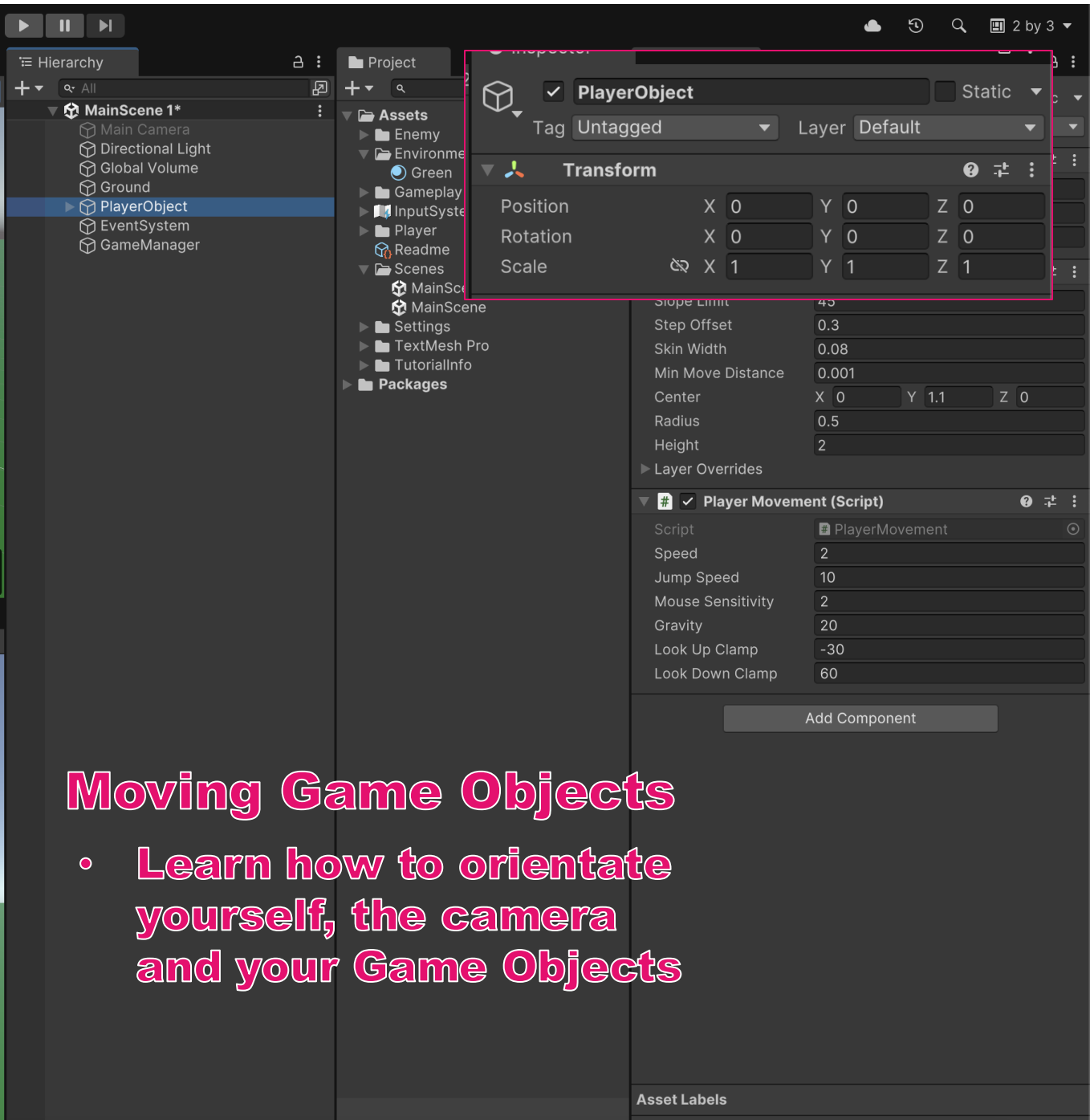
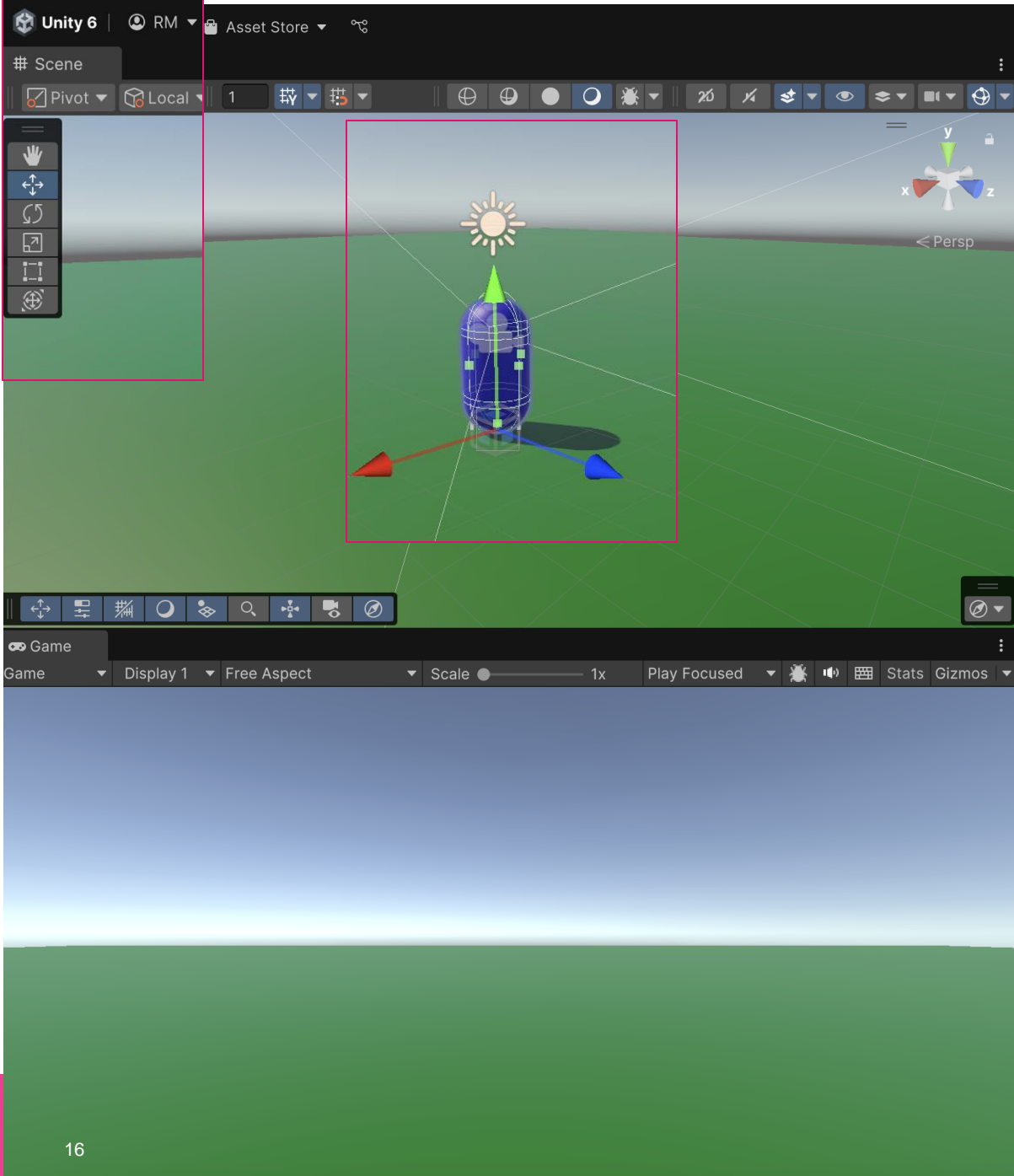
- Inside the **Hierarchy** pane, create an empty object called *PlayerObject*
 - Position: $x=0, y=0, z=0$
- Create a 3D Capsule object under *PlayerObject* and call it *PlayerModel*
 - Position: $x=0, y=1, z=0$
- Move the Main Camera into *PlayerObject*
 - Position: $x=0, y=1.5, z=0$



Create the Player

- Let's assign a color to the player
- In the **Project** pane, create a folder called *Player* under the *Assets* folder
- Create a material and call it *Blue* under *Player*
- Inside the **Inspector** pane, change its Base Map to **blue**
- Drag the *Blue* material to the player in the **Scene** pane





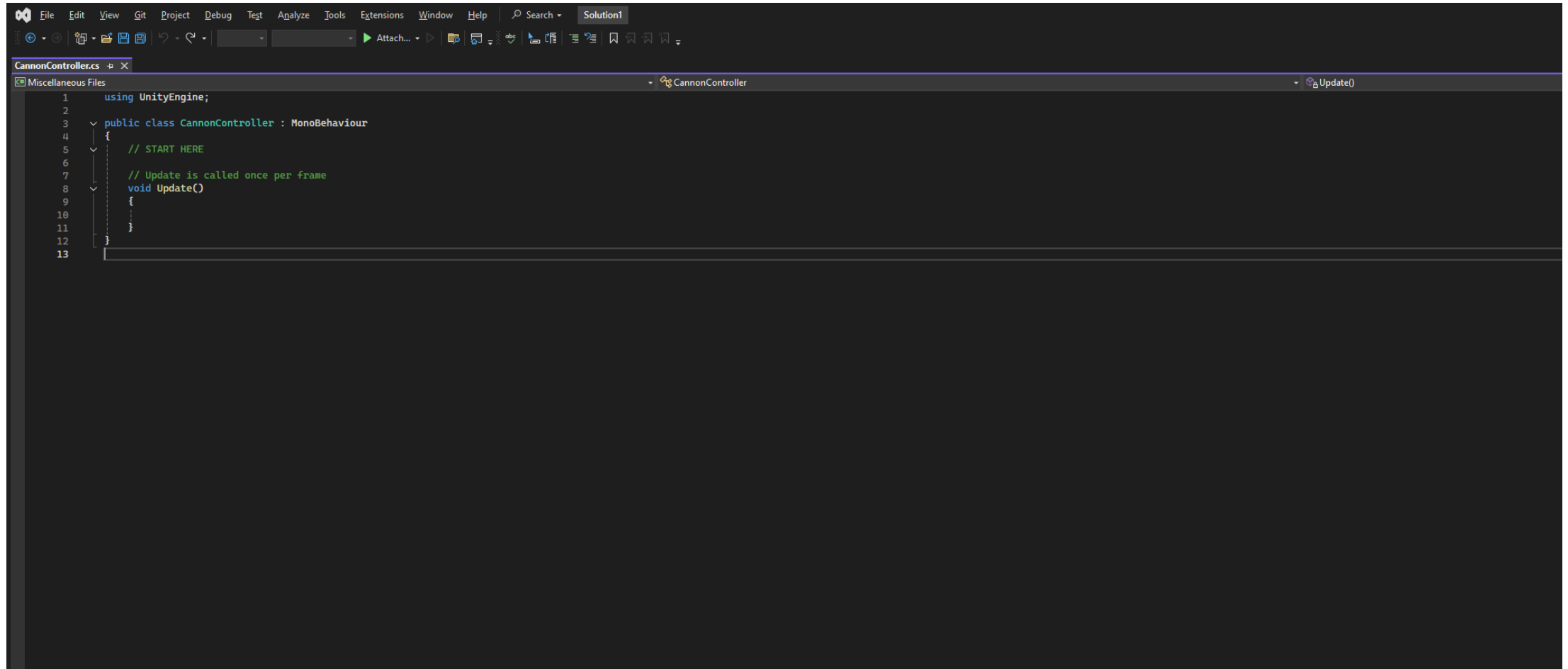
Moving Game Objects

- Learn how to orientate yourself, the camera and your Game Objects

Move the Player

- Let's add some code to the player so that we can move it
- In the **Project** pane, create a MonoBehaviour called *PlayerMovement* under the *Assets/Player* folder
- Double-click *PlayerMovement* to open it in **Visual Studio**

Visual Studio 2022



```
1 using UnityEngine;
2
3 public class CannonController : MonoBehaviour
4 {
5     // START HERE
6
7     // Update is called once per frame
8     void Update()
9     {
10
11     }
12 }
13
```

Move the Player

- Copy this and paste it into **Visual Studio**
 - https://github.com/ronaldkmo/hidden_seek/blob/main/Assets/Player/PlayerMovement.cs
- Make sure ***you commit your changes*** and go back to **Unity**
 - Ctrl + S to save your script, for example

Inside the Script

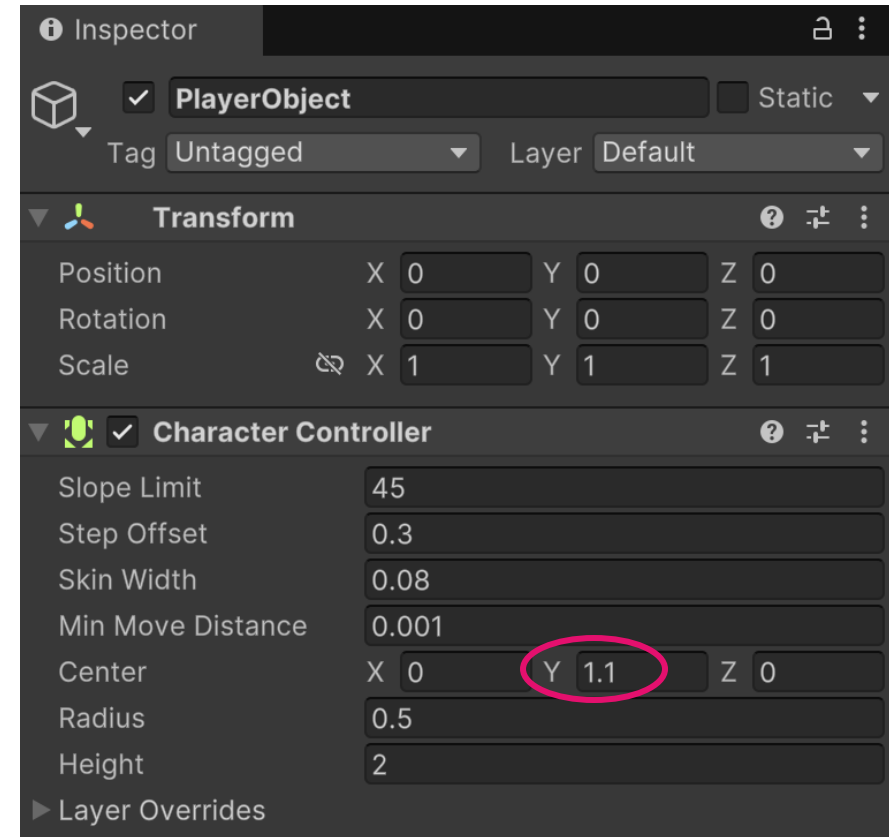
- `void Start()`
 - Executes when the game starts
 - Hides the cursor and deals with player control
- `void Update()`
 - Executes every frame
 - Call the `Movement()` and `RotateAndLook()` methods

Inside the Script

- `void Movement()`
 - Move the player by the keypresses
 - Make a jump
 - Something more 😊
- `void RotateAndLook()`
 - Rotate the camera

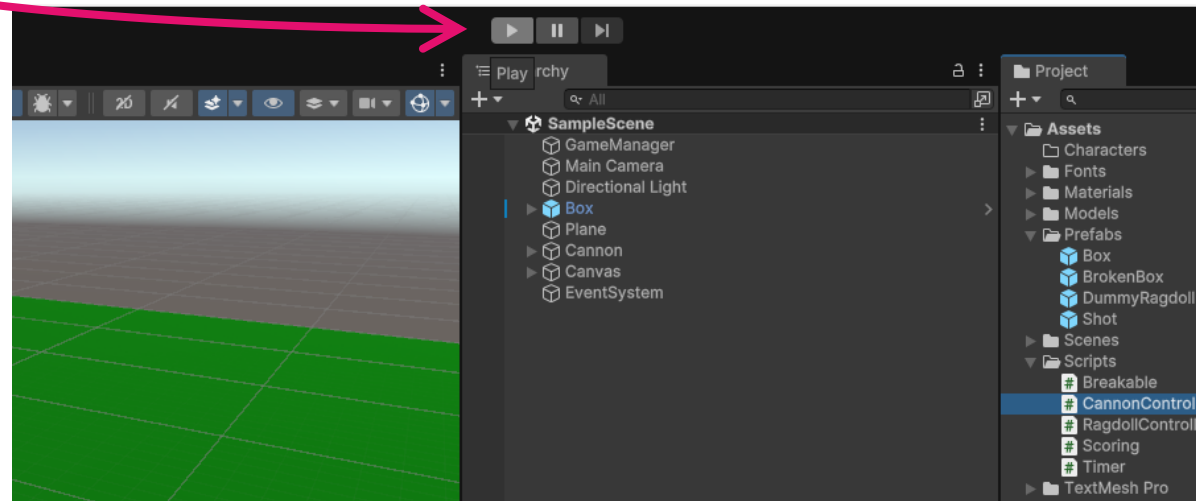
Move the Player

- Select *PlayerObject* in the **Hierarchy** pane
- Drag the *PlayerMovement* script into the **Inspector** plan
- Change the center of y in CharacterController to 1.1



Run the Game

- Click the *run* button to run the game



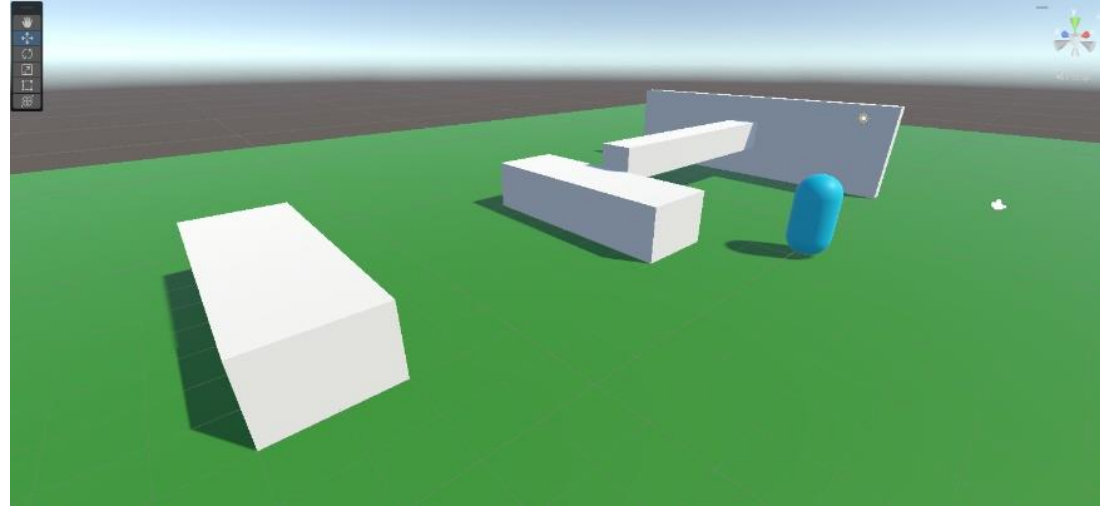
- You should be able to move the player by the arrow keys on your keyboard
- You can also jump using the space bar and change the camera angle with your mouse

What's Wrong?

- Even though you can move the player, it moves very slowly
- It can barely jump, too 😞
- There are two workarounds
 - Change the code which could be tedious
 - Change the settings in the Inspector which is a lot more intuitive
- See which settings you should modify

Next Up...

- Moving around on empty ground can be boring
- Why don't we put together some obstacles?
- You may right-click on the Hierarchy pane and add some 3D Game Objects (e.g., cube, sphere, capsule)
- Use your imagination to create your game environment 😊

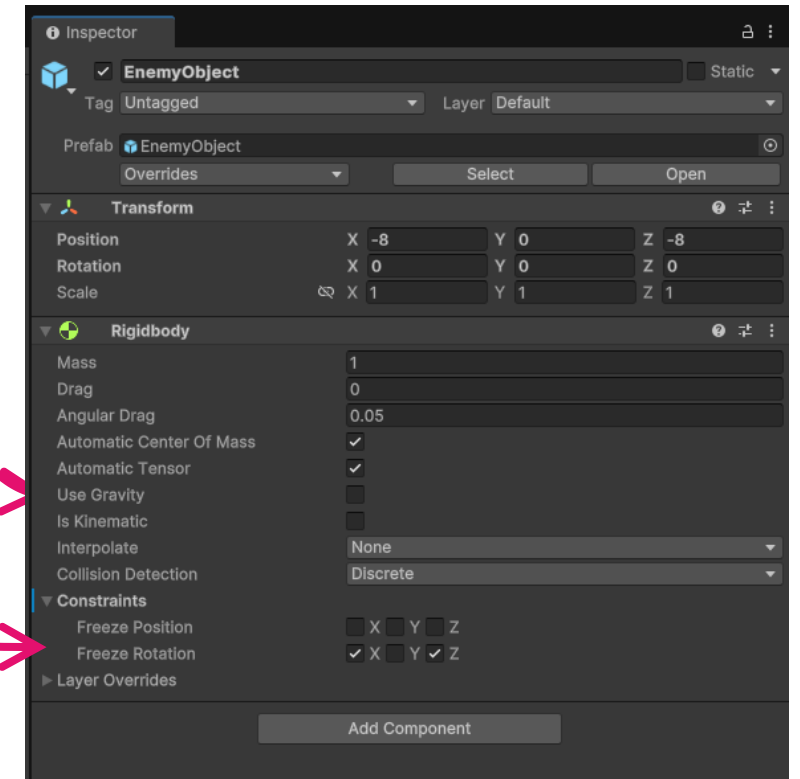


Enemies

- It's time to create some enemies
- Inside the **Hierarchy** pane, create an empty object called *EnemyObject*, and keep it far from the player
 - e.g., Position: $x=-5$, $y=0$, $z=-5$
- Create a 3D Capsule object under *EnemyObject* and call it *EnemyModel*
 - Position: $x=0$, $y=1$, $z=0$

Enemies

- While keeping *EnemyObject* selected, add a Rigidbody component in the **Inspector** pane
- Deselect Use Gravity
- In **Freeze Rotation**, check X and Y



Enemies

- In the **Project** pane, create a folder called *Enemy* under the *Assets* folder
- Create a material and call it *Red* under *Enemy*
- Inside the **Inspector** pane, change its Base Map to red
- Drag the *Red* material to the enemy in the **Scene** pane

Move the Enemy

- The enemy will chase the player upon spotting them
- In the **Project** pane, create a MonoBehaviour called *EnemyAttack* under the *Assets/Enemy* folder
- Double-click *PlayerMovement* to open it in **Visual Studio**
- Copy this code and paste it into **Visual Studio**
 - https://github.com/ronaldkmo/hidden_and_seek/blob/main/Assets/Enemy/EnemyAttack.cs
- Make sure ***you commit your changes*** and go back to **Unity**
 - Ctrl + S to save your script, for example

Inside the Script

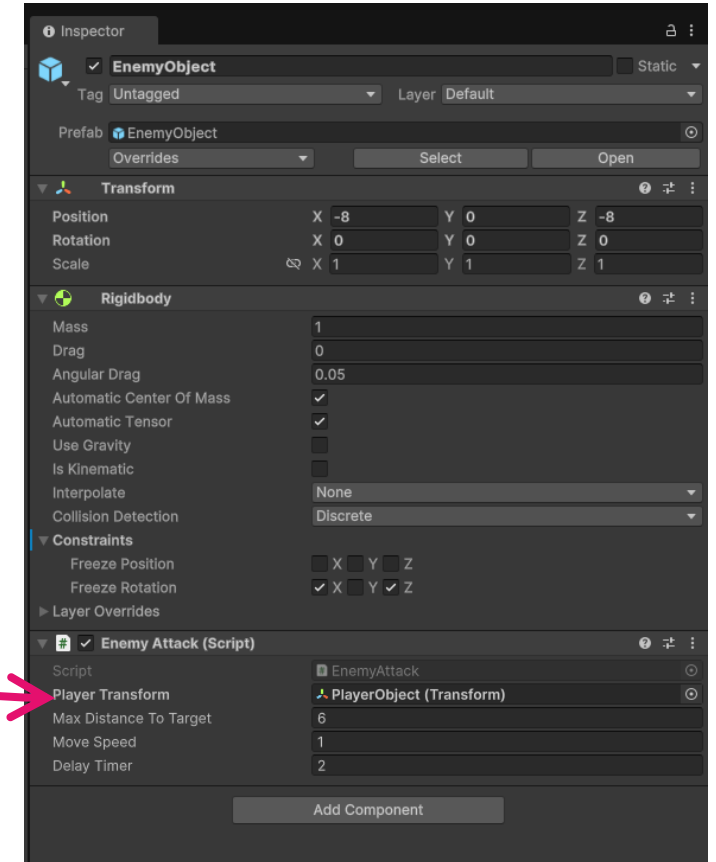
- `void Start()`
 - Executes when the game starts
 - Set the time and obtain the enemy game object
- `void Update()`
 - Executes every frame
 - Call the `Attack()` method

Inside the Script

- `void IsReadyToAttack()`
 - Check if the enemy is ready to attack
- `void Attack()`
 - Attack the player if the enemy sees them
 - The enemy will approach the player
- `void OnCollisionEnter()`
 - Check if the enemy has caught the player and terminate the game appropriately

Move the Enemy

- Select *EnemyObject* in the **Hierarchy** pane
- Drag the *EnemyAttack* script into the **Inspector** plan
- Drag *PlayerObject* in **Hierarchy** into the Player Transform field in **Inspector**

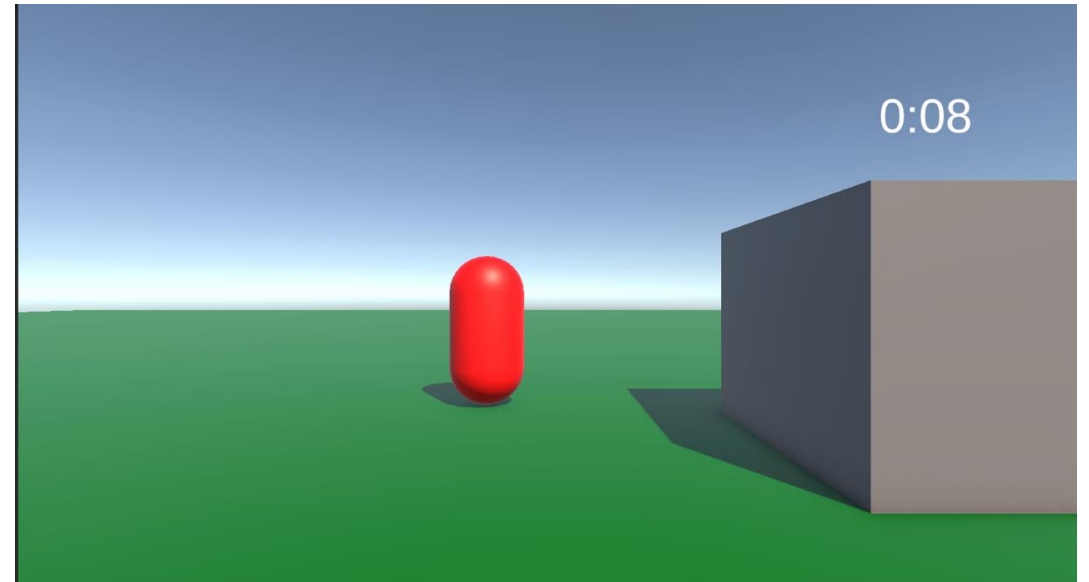


Run the Game (Again)

- Click the *run* button to run the game
- The enemy will chase you if you get too close
- However, they will stop if you move out of their sight
- The game ends if they catch you ☹️

What's More?

- To enhance the gameplay, you may
 - Add more enemies by copying and pasting *EnemyObject*
 - Change the setting in the script *EnemyAttack*
 - Play around with the settings in *Rigidbody* of *EnemyObject*
- Some potential improvements
 - Add a timer
 - Make the enemy jump perhaps



Join our Digital Community

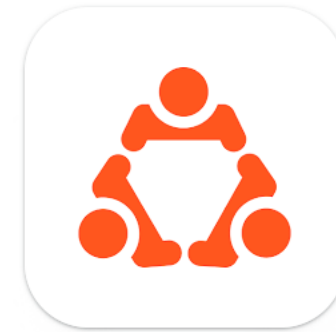
Join our exclusive Digital Community for University of Sunderland offer holders. Once you've joined, you'll be able to:

- Connect with future classmates
- Prepare for university life and learn more about what it's like to be a Sunderland student

Here's how to get started:

- Scan the QR code
- Download the app when prompted
- Get chatting!

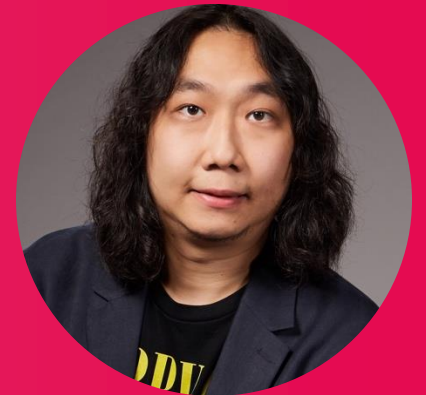
If you have used TAP before you can use your existing log in details.



Thank you

**WE ARE THE LIFE CHANGING
UNIVERSITY OF SUNDERLAND**

Dr. Ronald Mo
ronald.mo@sunderland.ac.uk



**University of
Sunderland**