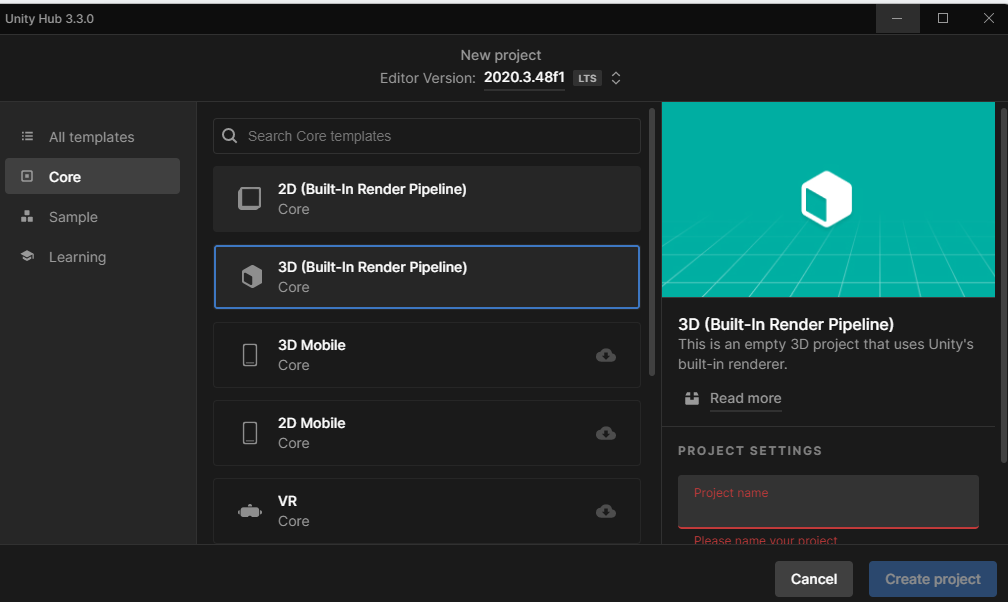
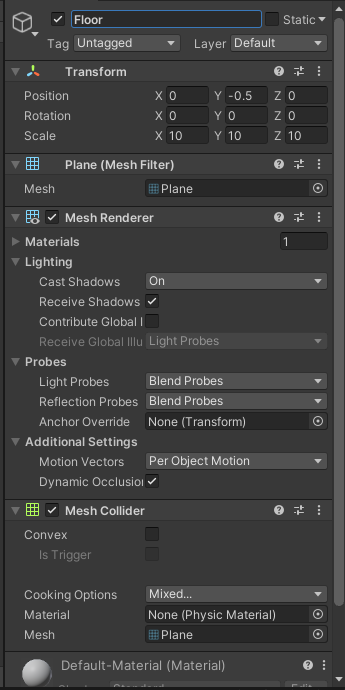
420 541\_Lab 1 : Getting to know the Editor

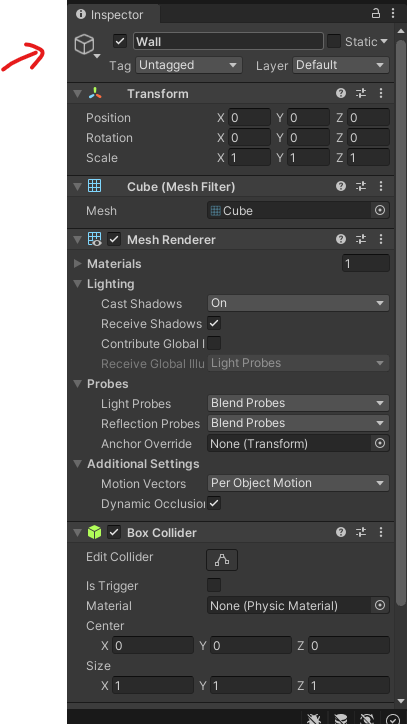
#### **Task 1: Create a Scene and Add Basic Cubes**

1. **Open Unity:** Start by launching Unity and creating a new 3D project.
   * Call your new Project 420-541\_Lab1\_[yourstudentid]
   * Where [yourstudentid] is your student ID.
   * Your project template need to be the 3D template.



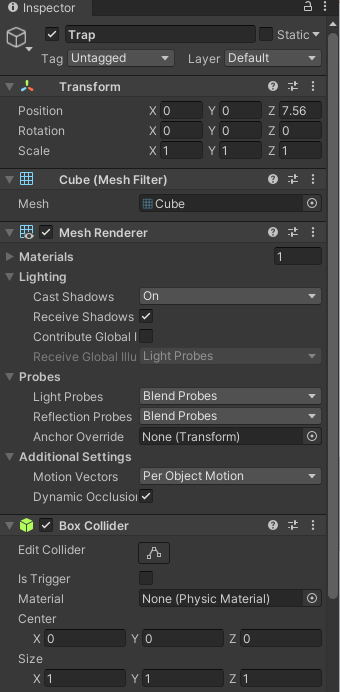
1. **Create a New Scene:** In the project panel, go to File > New Scene to create a new scene.
   * Call your scene “Main Scene”.
   * Save it in the folder Scene in the Project Panel
2. **Add a floor :** 
   * In the Hierarchy panel, right-click and select 3D Object > Plane to add a cube to your scene.
   * Place it at this position by modifying it’s transform component ( Position = X : 0 , Y : - 0.5 Z : 0.0f)
   * Scale it up in all direction by modifying it’s transform component ( Scale : X : 10 , Y : 10 Z : 10)



1. **Add Cubes ( Walls ) :**
   * In the Hierarchy panel, right-click and select 3D Object > Cube to add a cube to your scene.
   * Repeat this step to add more cubes. You can duplicate cubes by selecting a cube in the Hierarchy and pressing Ctrl + D (Windows) or Cmd + D (Mac).
   * Name those cube **Wall** in the Inspector
   * 

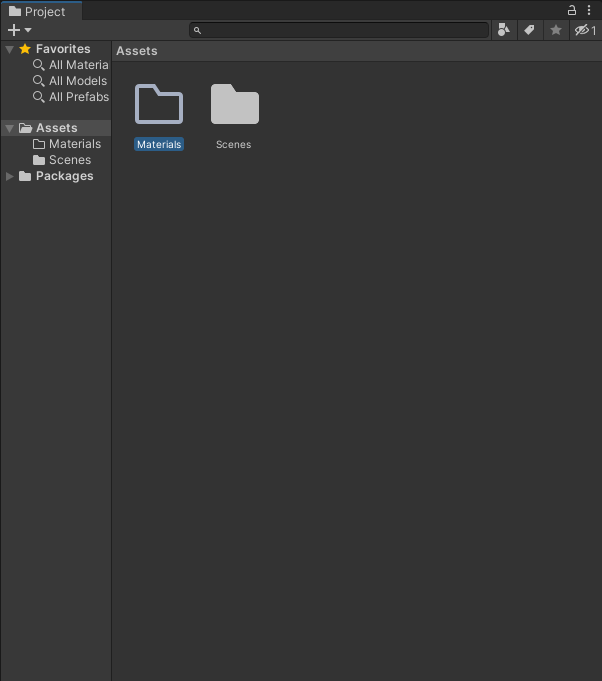
4 **Add Cubes ( Traps) :**

In the Hierarchy panel, right-click and select 3D Object > Cube to add a cube to your scene.

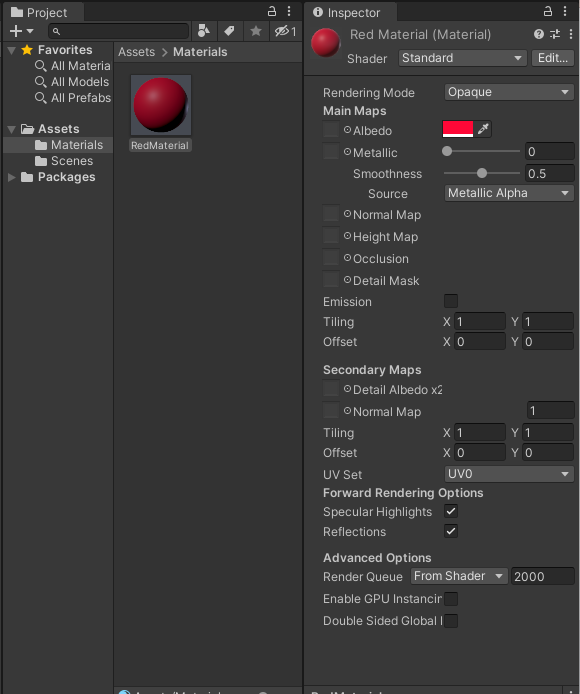
1. Repeat this step to add more traps. You can duplicate cubes by selecting a cube in the Hierarchy and pressing Ctrl + D (Windows) or Cmd + D (Mac).
2. Name those cube **Trap** in the Inspector
3. 
4. **Arrange your Cubes:** Use the Move, Rotate, and Scale tools in the Unity toolbar to arrange the cubes into a small level or labyrinth.
   * To move, select the cube and use the Move tool (shortcut W).
   * To rotate, use the Rotate tool (shortcut E).
   * To scale, use the Scale tool (shortcut R).
     1. Place the cubes so that there is no gab between the floor and the cubes.
     2. You can scale the cubes to make longuer walls
     3. You can use CTRL to more have the object snap to the grid.
     4. You can also use CTRL and the WASD keys to move the view around.

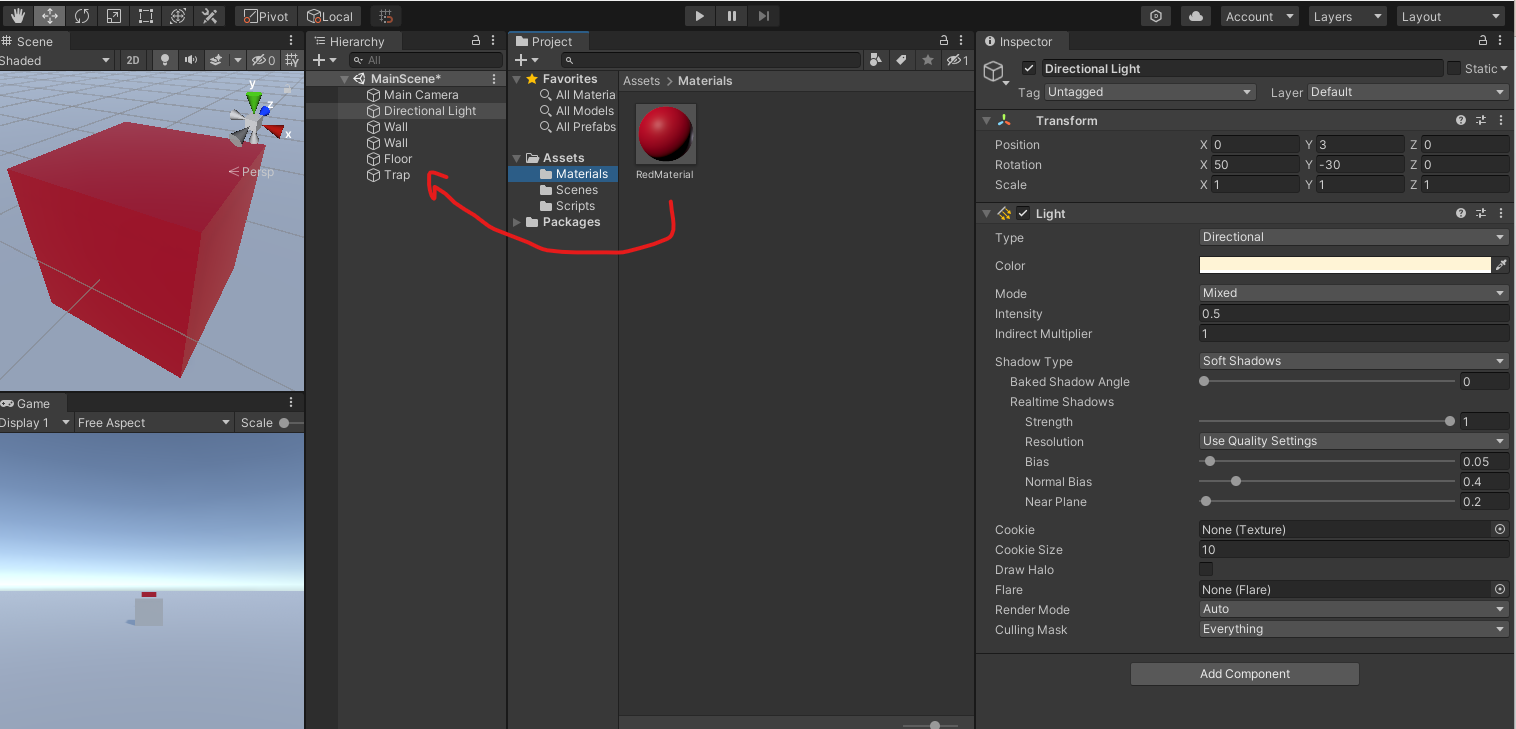
#### **Task 2: Create and Apply a New Material**

1. **Create a New Material:**
   * In the Project Panel, right-click and select Create > Folder.
   * Name your folder Materials
   * Open the folder in the Project panel
   * In the Project panel, right-click and select Create > Material.
   * Name the material (e.g., RedMaterial).



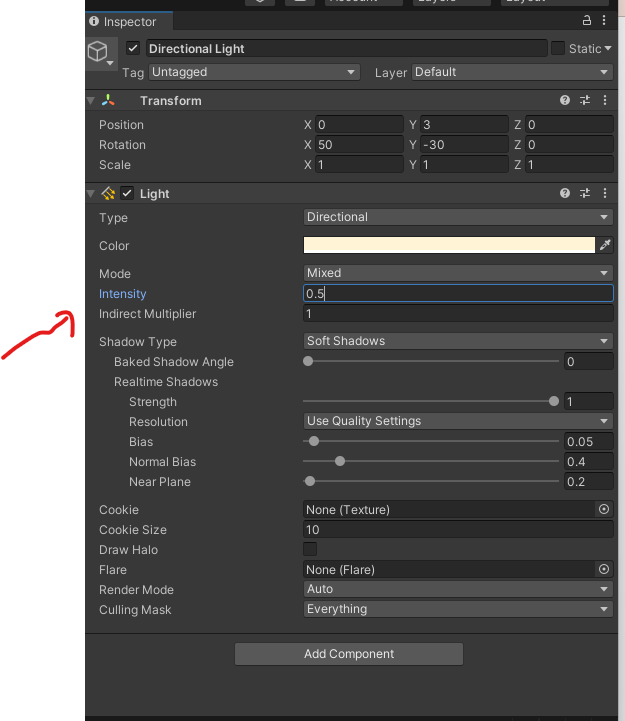
1. **Change Material Color:**
   * With the material selected, go to the Inspector panel.
   * Under the Main Maps > Albedo section, click the color box and choose a new color.



1. **Apply the Material:**
   * Drag and drop the material from the Project panel onto the cubes in the scene that you want to change.
   * Drag it to one of your traps 

#### **Task 3: Add Different Types of Light**

1. **Dim the Directional Light :** 
   * In the Hierarchy panel , Select the object Called DIrectional Light
   * Then in it’s inspector set it’s intensity to 0.5.



1. **Add a Spot Light:**
   * In the Hierarchy panel,**Select** on of your traps
   * Then right-click and select Light > Spot Light.
     1. It will add the light as child of the trap.
   * Position and rotate the light as desired using the Move and Rotate tools.
2. **Add a Point Light:**
   * In the Hierarchy panel, right-click and select Light > Point Light.
   * Position the light in your scene where you want to add localized lighting.

#### **Task 4: Create a Player Movement Script**

1. **Create a New Script:**
   * In the project panel, right-click and Select Create > Folder.
   * Name your folder Scripts
   * Double click on the folder to move into it.
   * In the Project panel, right-click and select Create > C# Script.
   * Name the script (e.g.,PlayerController).
2. **Edit the Script:**
   * Double-click the script to open it in your code editor.

Write a script to move the camera based on player input. Below is an example:

using UnityEngine;

public class PlayerController : MonoBehaviour

{

public float speed = 10.0f;

void Update()

{

float h = Input.GetAxis("Horizontal");

float v = Input.GetAxis("Vertical");

Vector3 movement = new Vector3(h, 0, v);

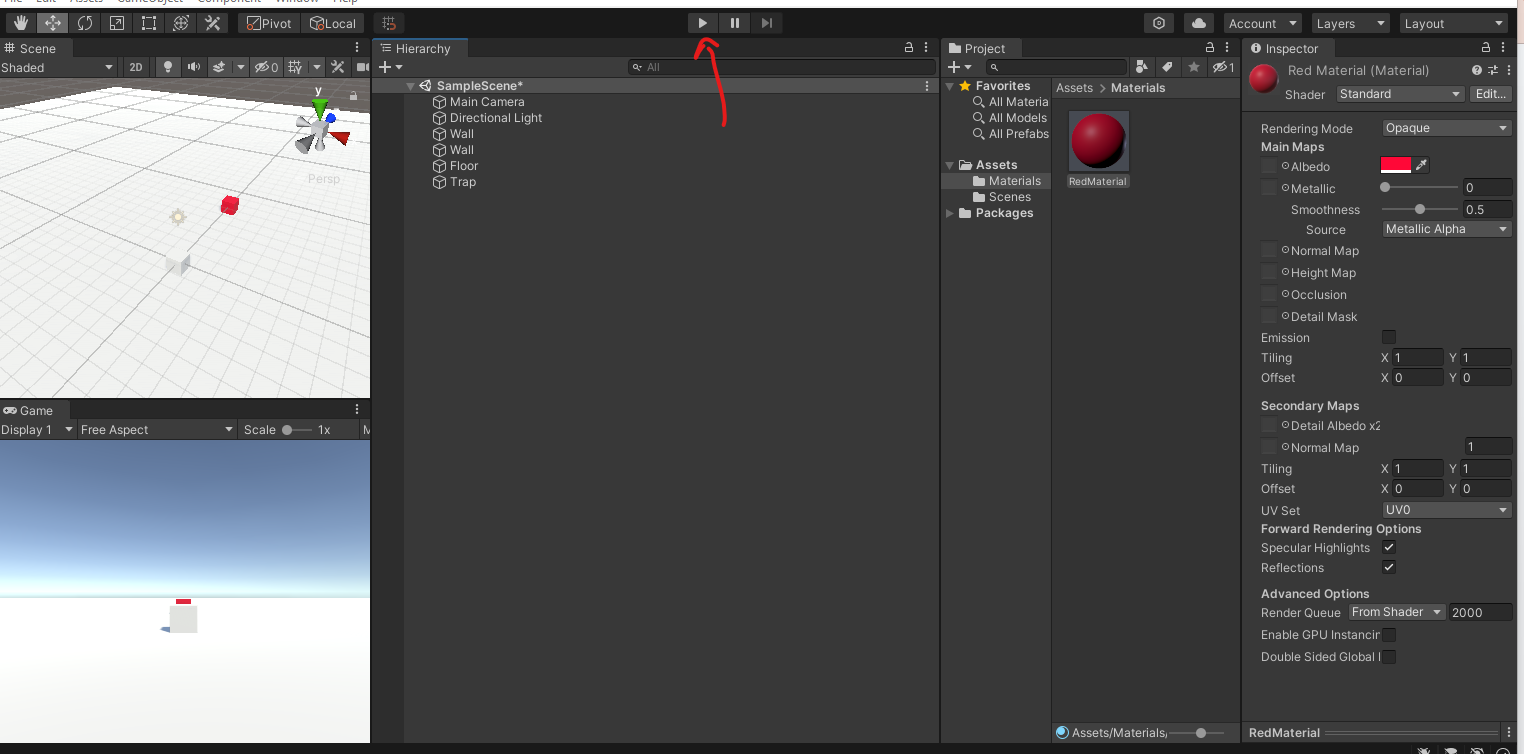
transform.Translate(movement \* speed \* Time.deltaTime);

}

}

1. **Attach the Script to the Camera:**
   * In the Hierarchy panel, select the Main Camera object.
   * Rename the MainCamera to Player
   * Drag the PlayerController script from the Project panel onto the Inspector panel of the camera.

Now Play your game using the Play option



Try to move the camera throught one of your walls. What do you notice ? Write down your answer here :

The camera moves through the walls. Based on the script when the frame updates along with the movement, the horizontal argument as -0.03f, while the camera going to left and vertical being at 0 since it should stay within the same height as the camera is moving forward until it goes through the wall.

#### **Task 5: Create an Object Movement Script**

1. **Create a New Script:**
   * In the Project panel, right-click and select Create > C# Script.
   * Name the script (e.g., ObjectMover).
2. **Edit the Script:**
   * Double-click the script to open it in your code editor.

Write a script to move an object back and forth. Here’s an example:

using UnityEngine;

public class ObjectMover : MonoBehaviour

{

public float timer = 0.0f;

public float directionChangeDelay = 2.0f;

public Vector3 movement;

void Start()

{

}

void Update()

{

timer += Time.deltaTime;

if (timer > directionChangeDelay)

{

movement \*= -1.0f;

timer = 0.0f;

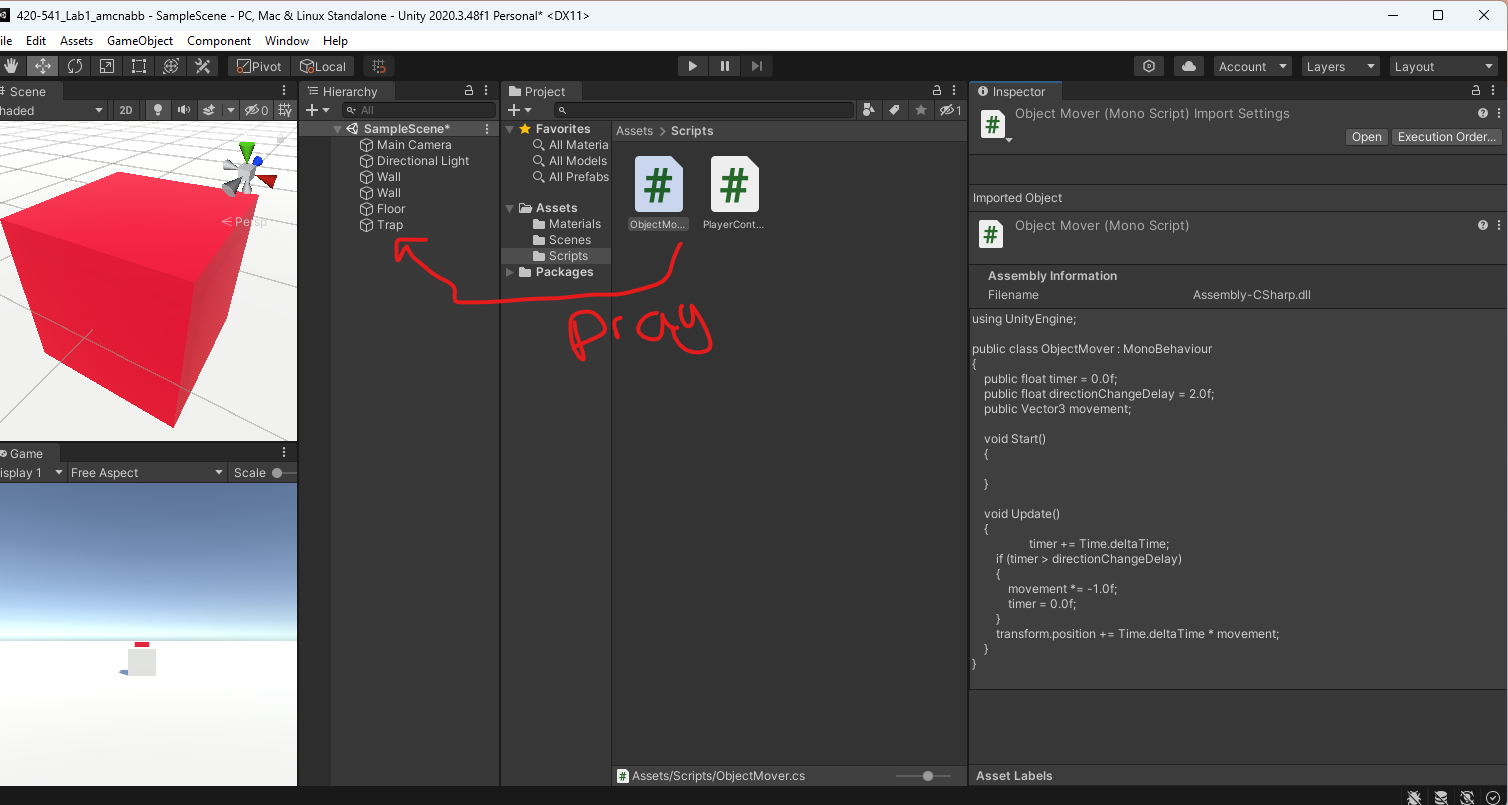
}

transform.position += Time.deltaTime \* movement;

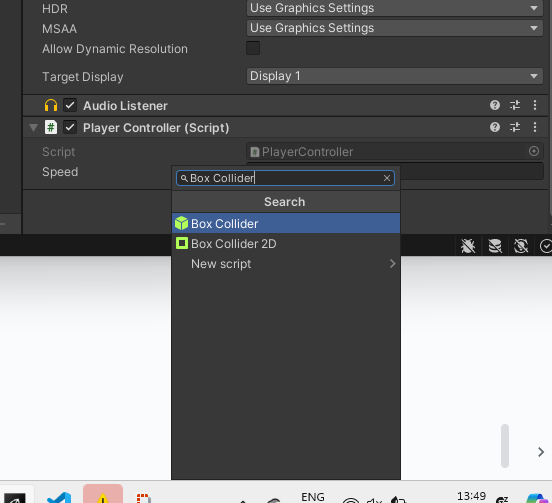
}

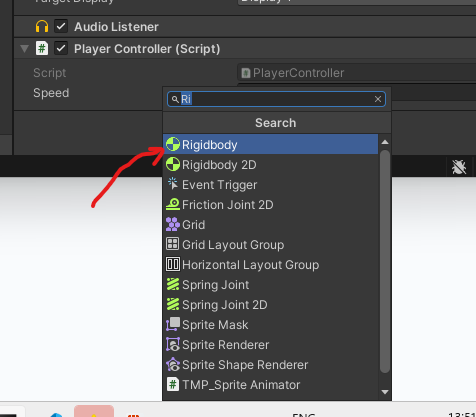
}

1. **Attach the Script to an Object:**
   * In the Hierarchy panel, select one of your traps that you want to move on it’s own.
   * Drag the ObjectMover script onto the object’s Inspector panel.



**Task 6: Lets add some collision to your player**

* + Click on your Player Game Object ( IE your Camera )
  + In the **Inspector** click Add component
  + Search **Box Colider**
    - Then double click on the option
    - 

* + Click on Add Component again
  + Search for **Rigid Body**
  + Double click on the first option
  + 
  + Your camera should now not be able to walk through your walls.

#### 

#### 

#### 

#### **Submission:**

1. **Compress Your Project Folder:**
   * In your file explorer, navigate to the root folder of your Unity project.
   * Add this text document with your written answer for step 5.
   * Right-click the folder and select Compress or Send to > Compressed (zipped) folder.
2. **Submit Through LÉA:**
   * Log in to LÉA, go to the submission section for this lab, and upload the compressed project folder.