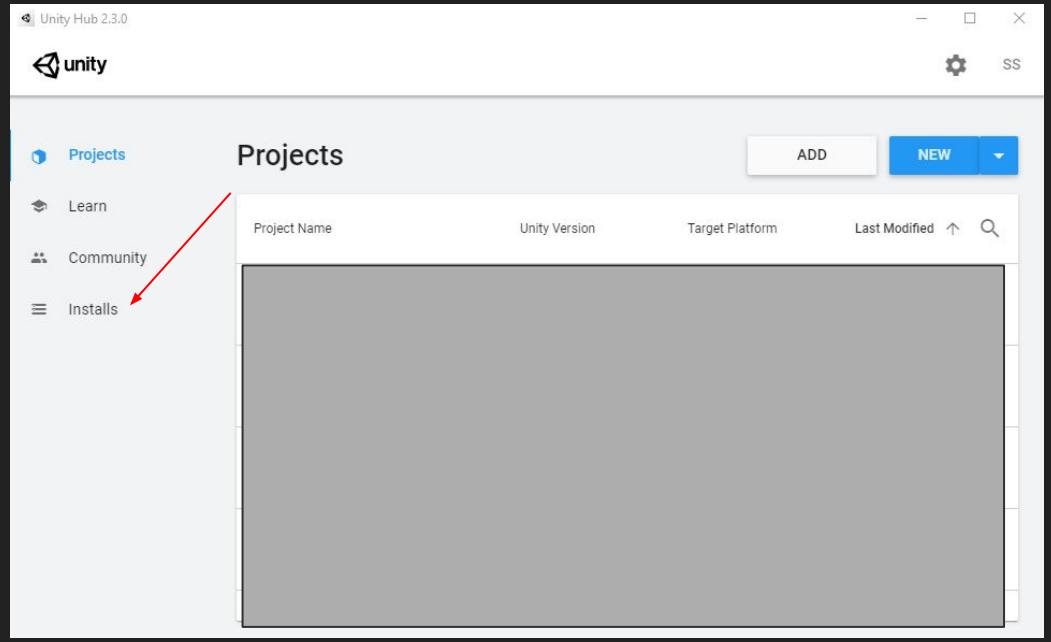
Intro to Unity

Spenser Solys

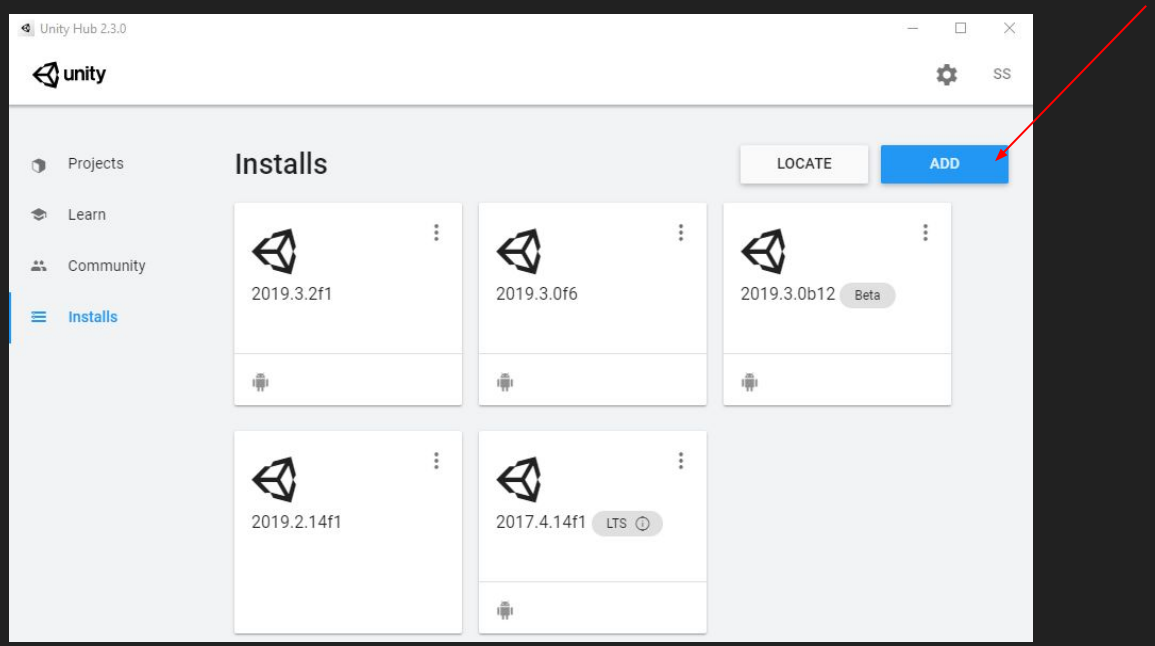
Downloads

UnityHub: https://unity3d.com/get-unity/download

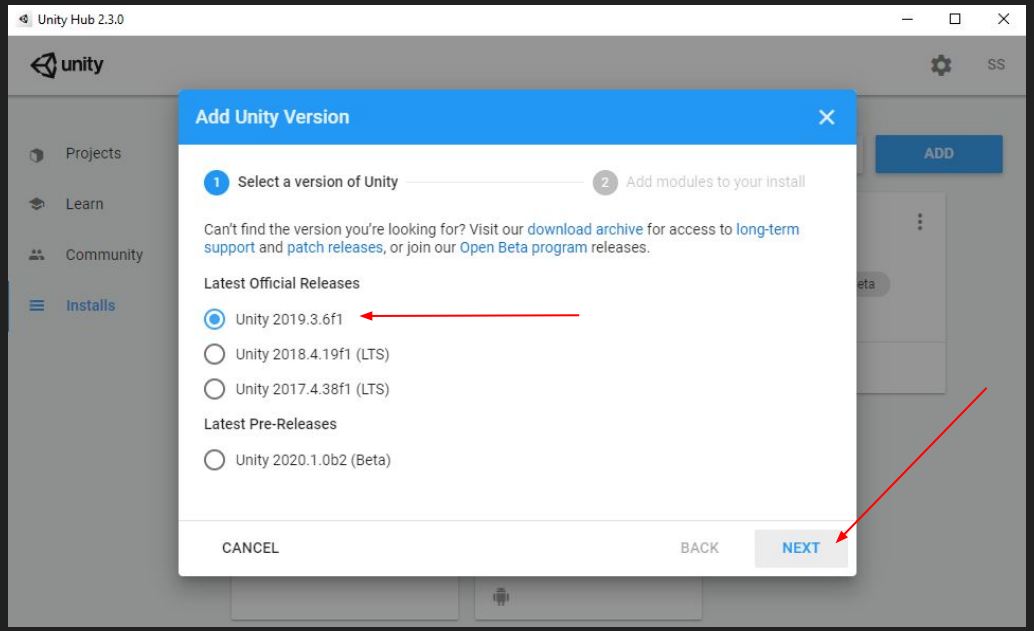
Install Unity 2019.3.6f1



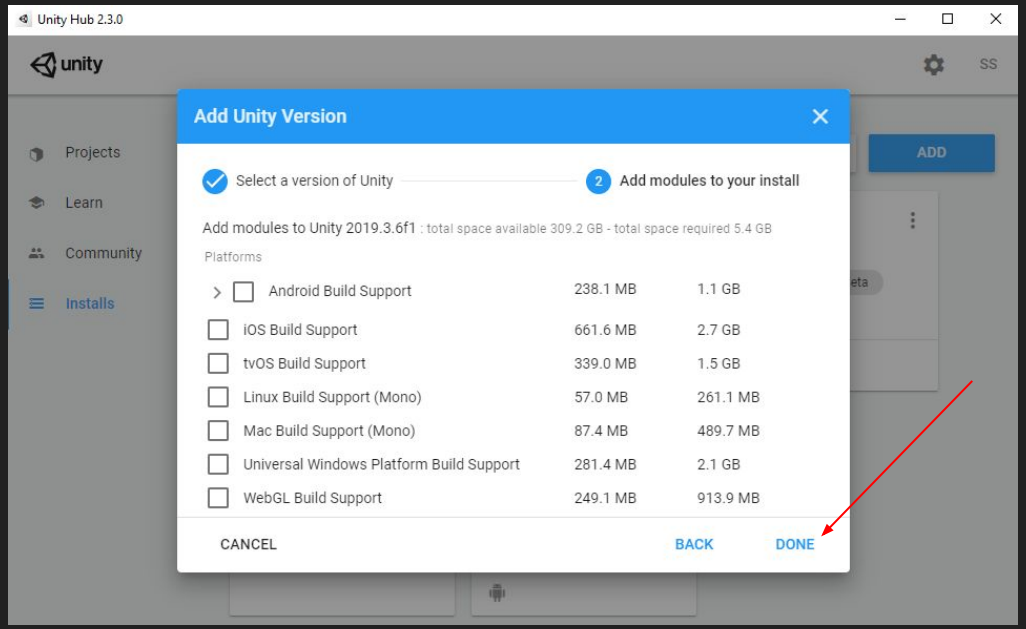
Install Unity 2019.3.6f1



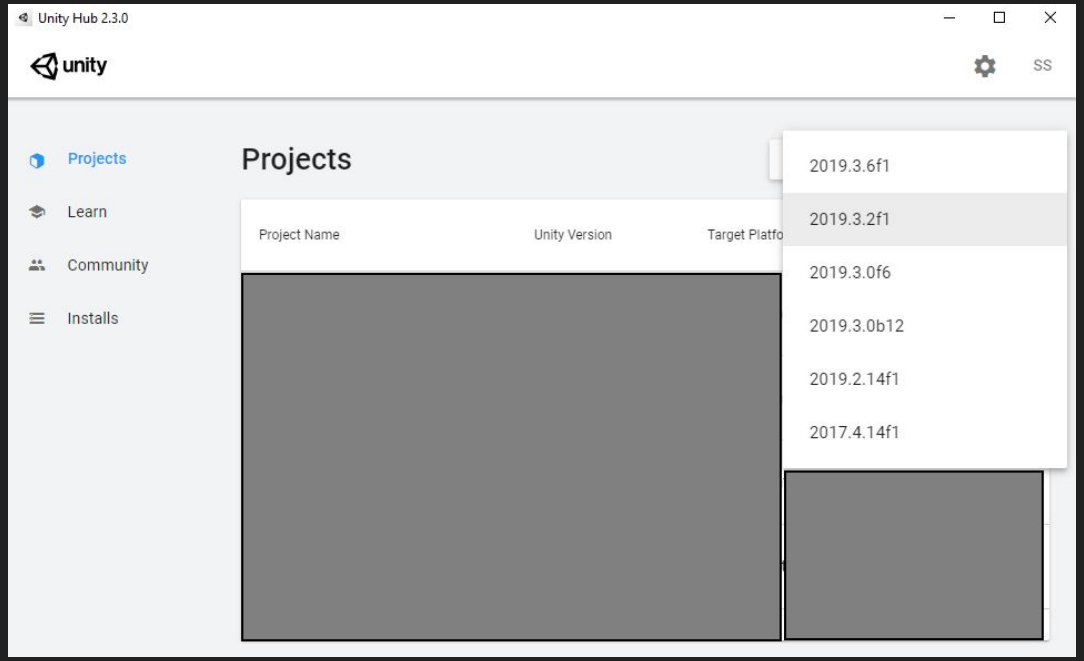
Install Unity 2019.3.6f1



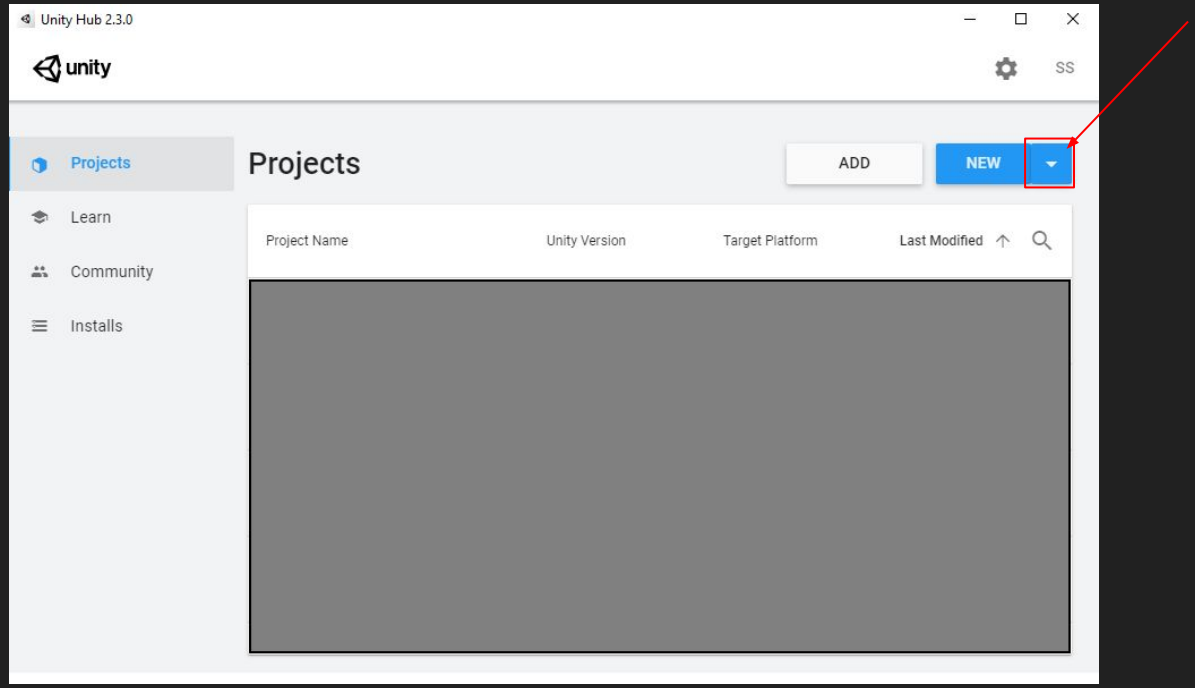
Install Unity 2019.3.6f1



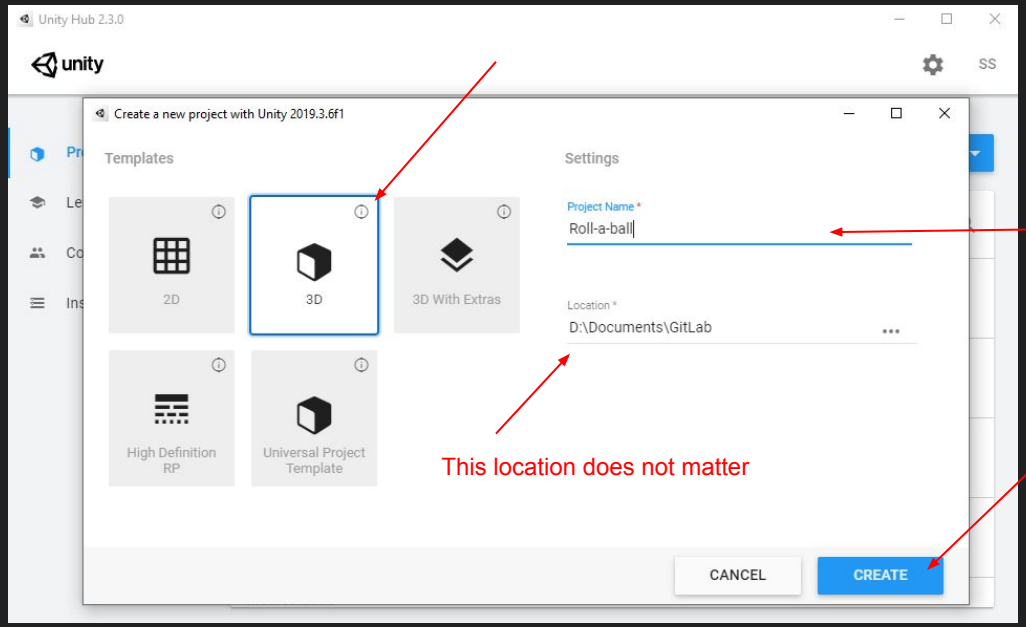
Create New Project



Create New Project



Create New Project



While we wait

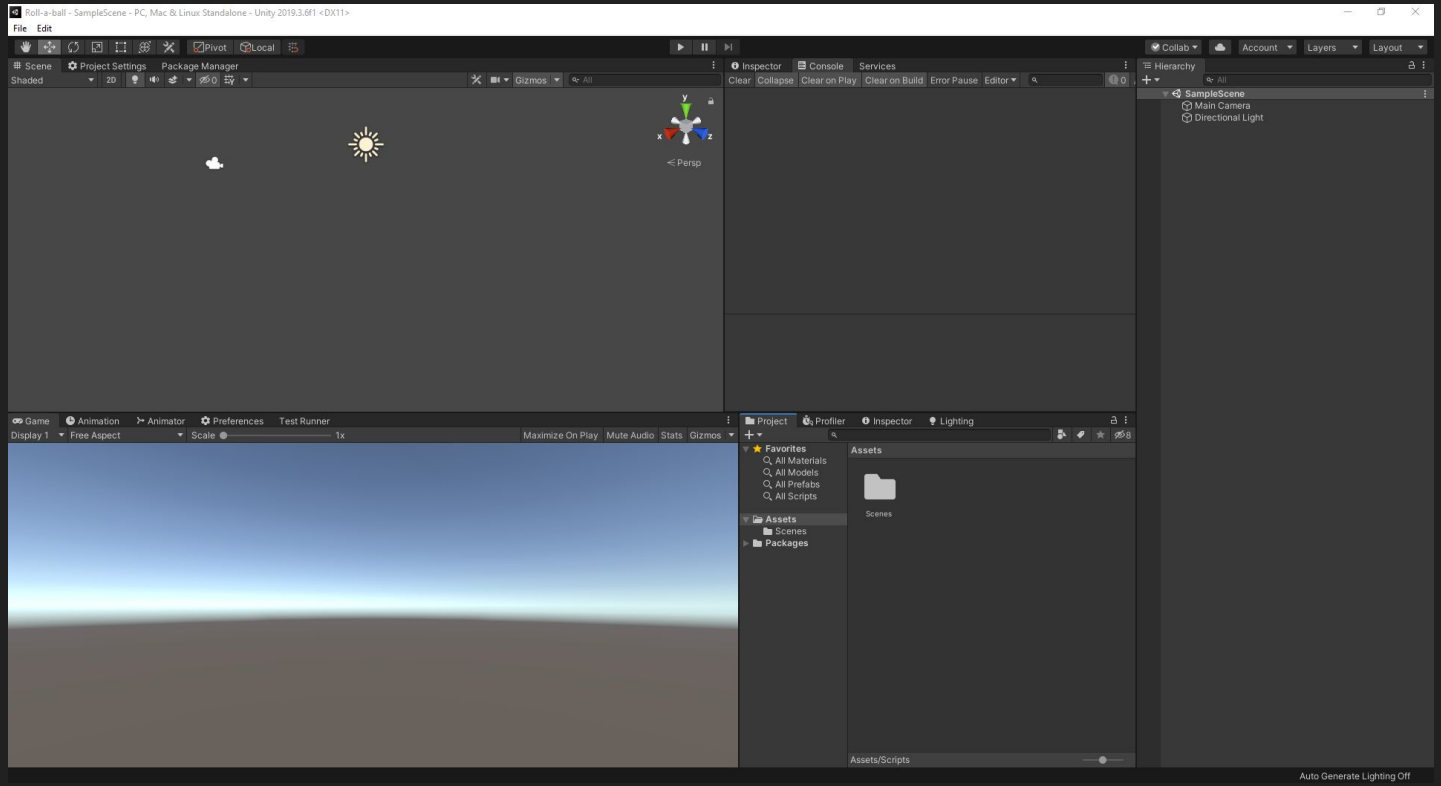
for it to create the project....

Get the most out of Unity

Check out Unity Learn for more tutorials: https://learn.unity.com

If you are a student the Unity Student Plan is included in the GitHub Student Developer Pack! More details here: https://education.github.com/pack

What we should see



First Step: Housekeeping

- Create a folder for our scripts

First Step: Housekeeping

- Create a folder for our scripts
- Create a folder for our prefabs

First Step: Housekeeping

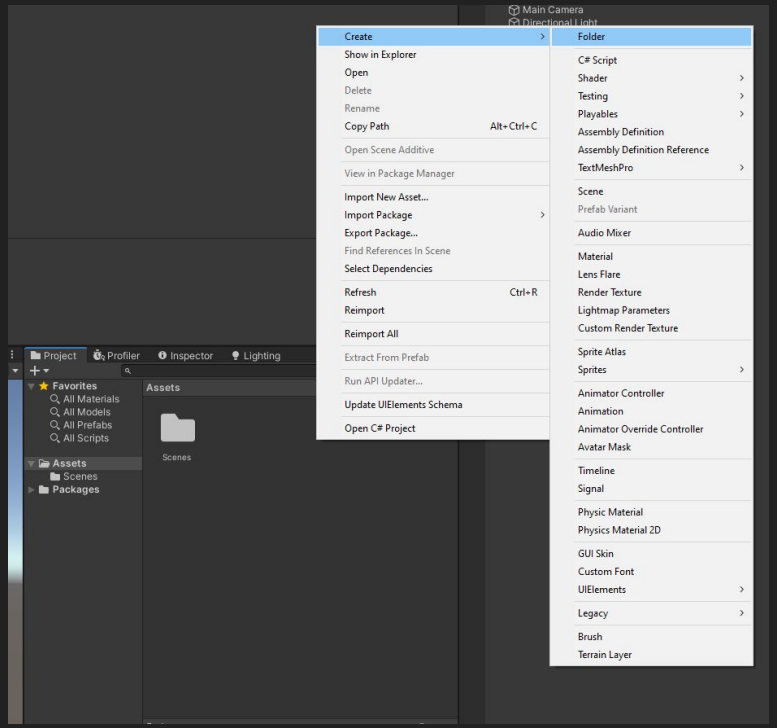
- Create a folder for our scripts
- Create a folder for our prefabs
- Create a folder for our materials

Creating Folders

- Find and click the project tab

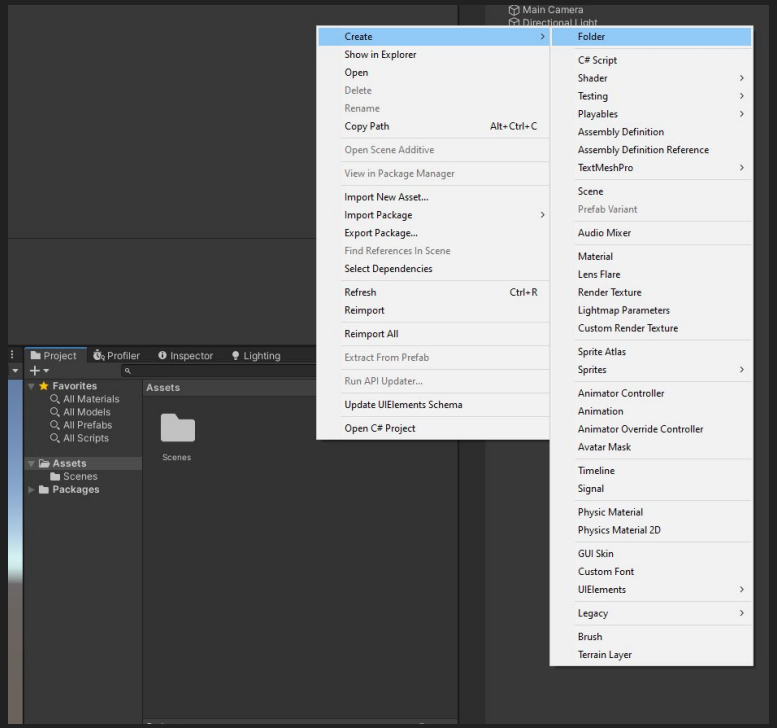
Creating Folders

- Find and click the project tab
- Right click inside the project window



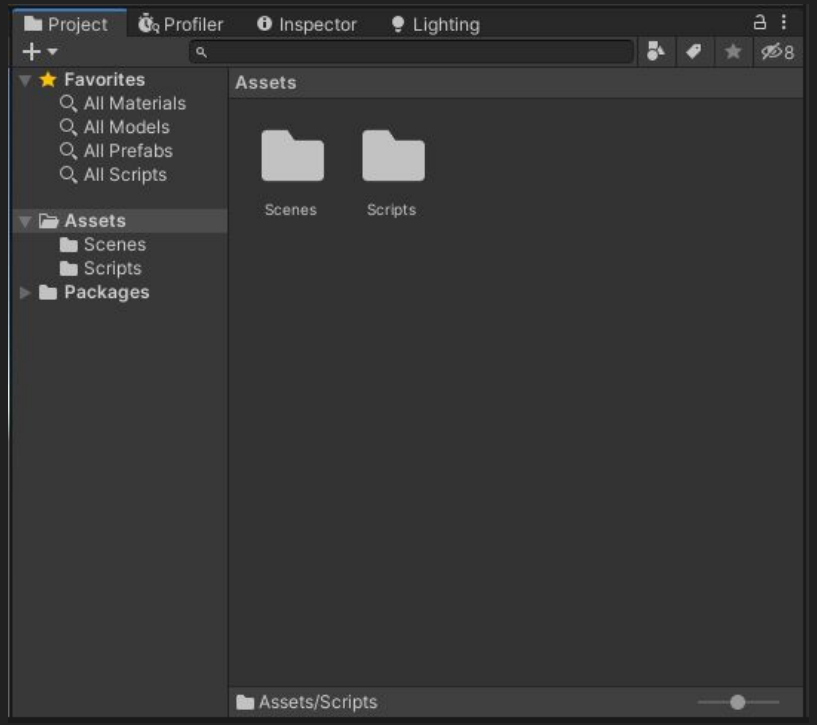
Creating Folders

- Find and click the project tab
- Right click inside the project window
- Click: Create → Folder



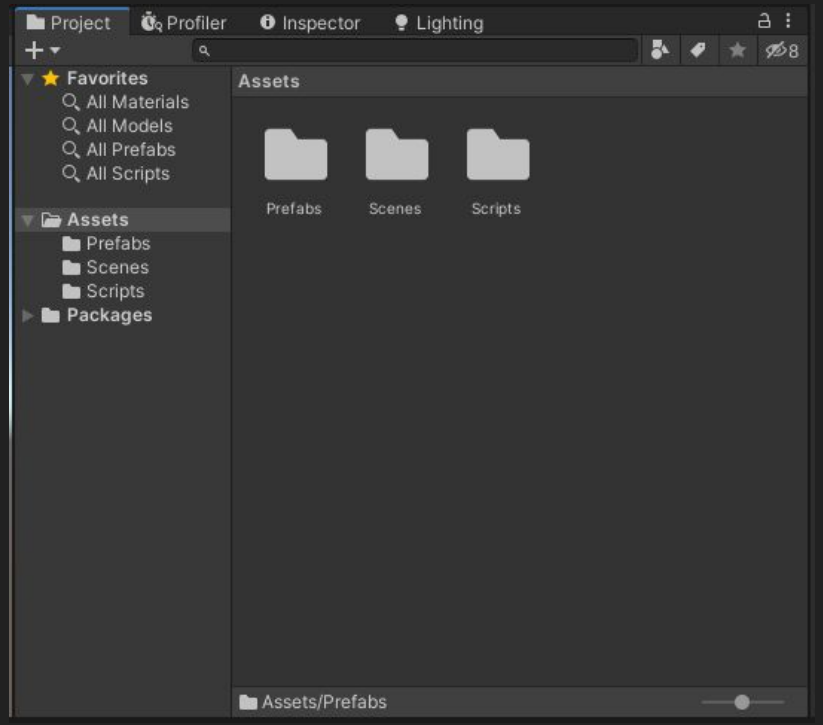
Creating Folders

- Find and click the project tab
- Right click inside the project window
- Click: Create → Folder
- Create a folder called Scripts



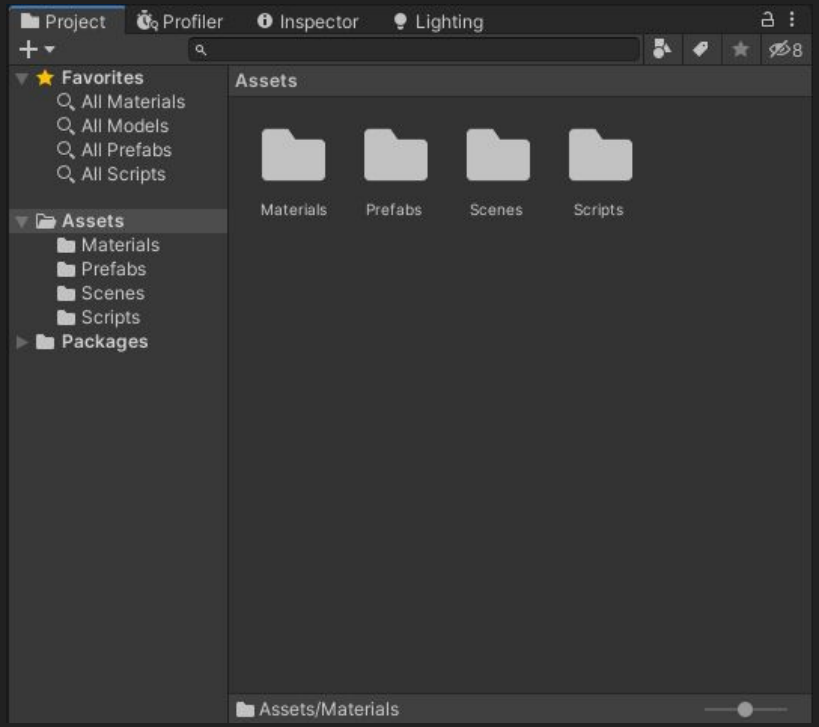
Creating Folders

- Find and click the project tab
- Right click inside the project window
- Click: Create → Folder
- Create a folder called Scripts
- Create a folder called Prefabs



Creating Folders

- Find and click the project tab
- Right click inside the project window
- Click: Create → Folder
- Create a folder called Scripts
- Create a folder called Prefabs
- Create a folder called Materials



Creating our game board

- Create a Plane

Creating our game board

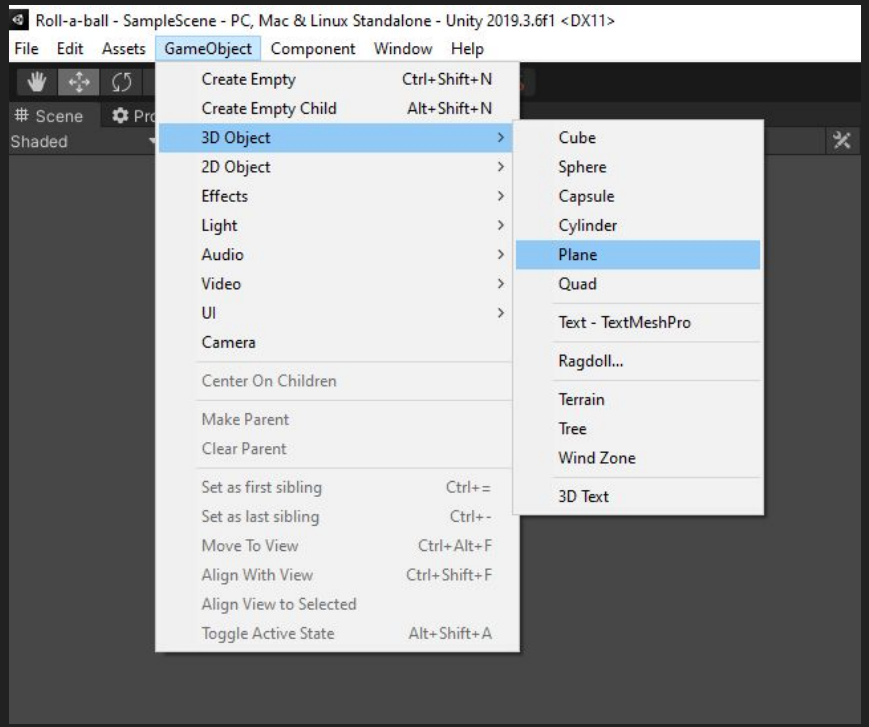
- Create a Plane
  - This is what our ball will roll around on

Creating a Plane

- Click on the GameObject tab on the top bar

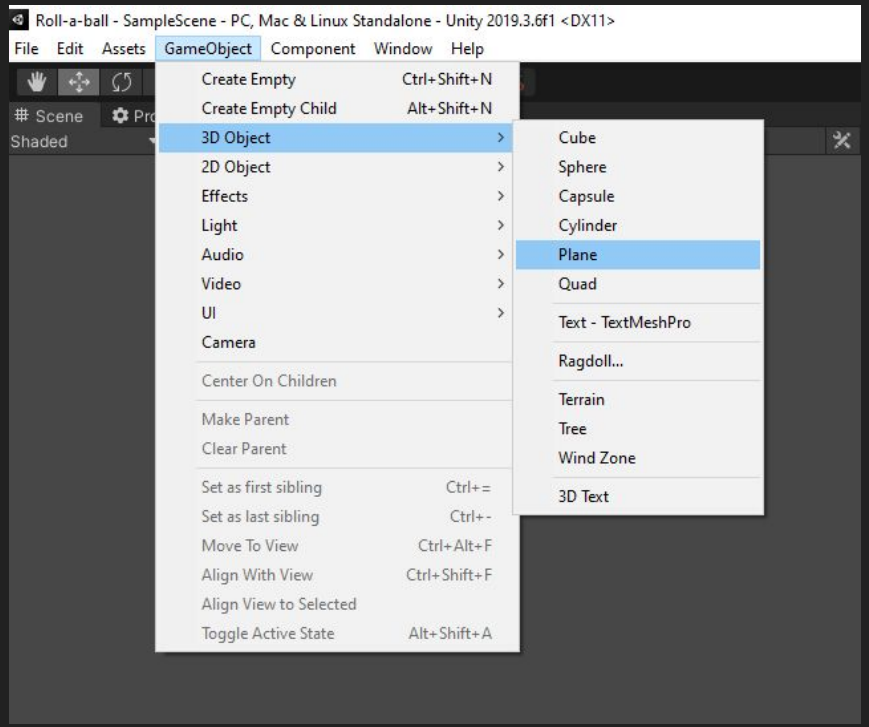
Creating a Plane

- Click on the GameObject tab on the top bar



Creating a Plane

- Click on the GameObject tab on the top bar
- Click: 3D Object → Plane

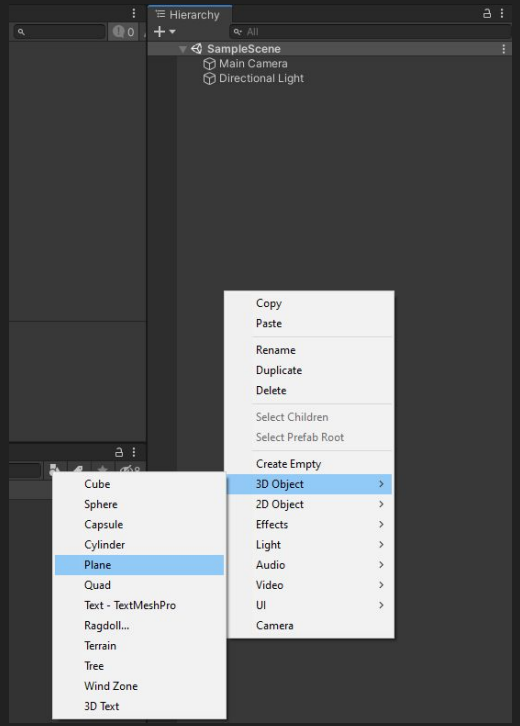


Creating a Plane (Alternative Method)

- Right click on the empty space in the Hierarchy window

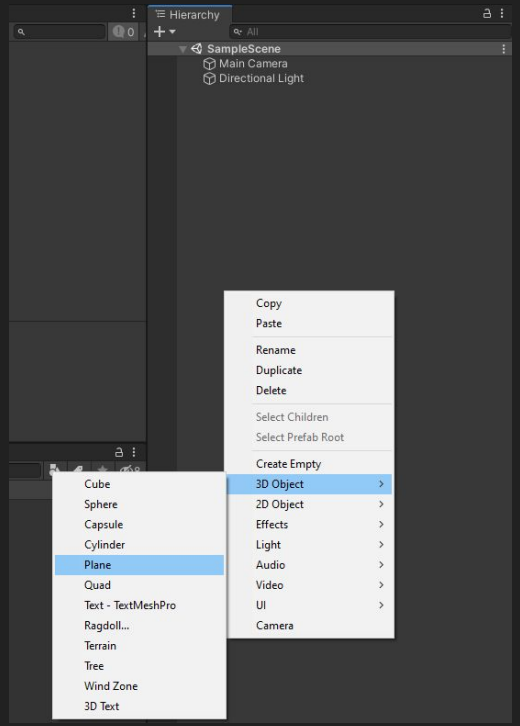
Creating a Plane (Alternative Method)

- Right click on the empty space in the Hierarchy window



Creating a Plane (Alternative Method)

- Right click on the empty space in the Hierarchy window
- Click: 3D Object → Plane

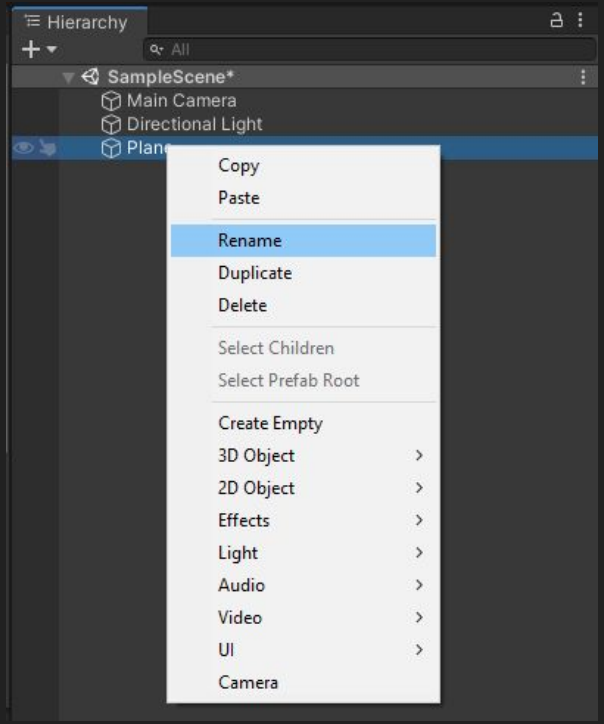


Renaming our Plane

- We want to rename our Plane gameobject to “Ground”

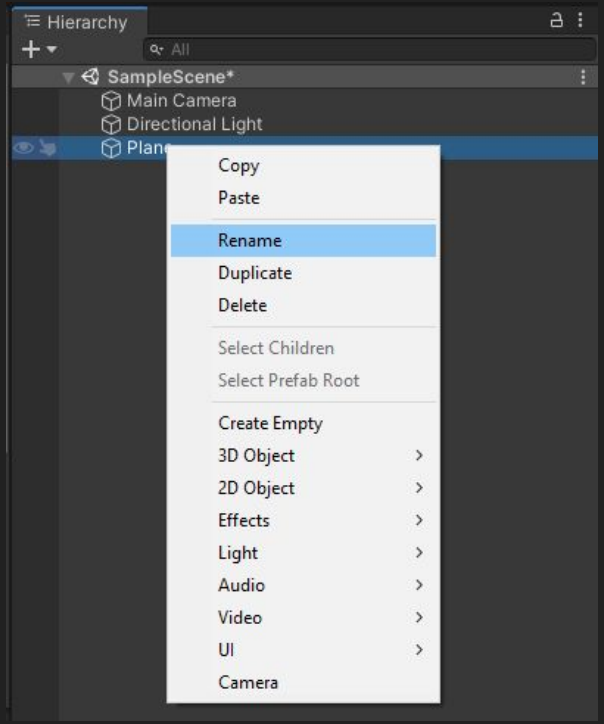
Renaming our Plane

- We want to rename our Plane gameobject to “Ground”
- Right click on the gameobject in the Hierarchy window



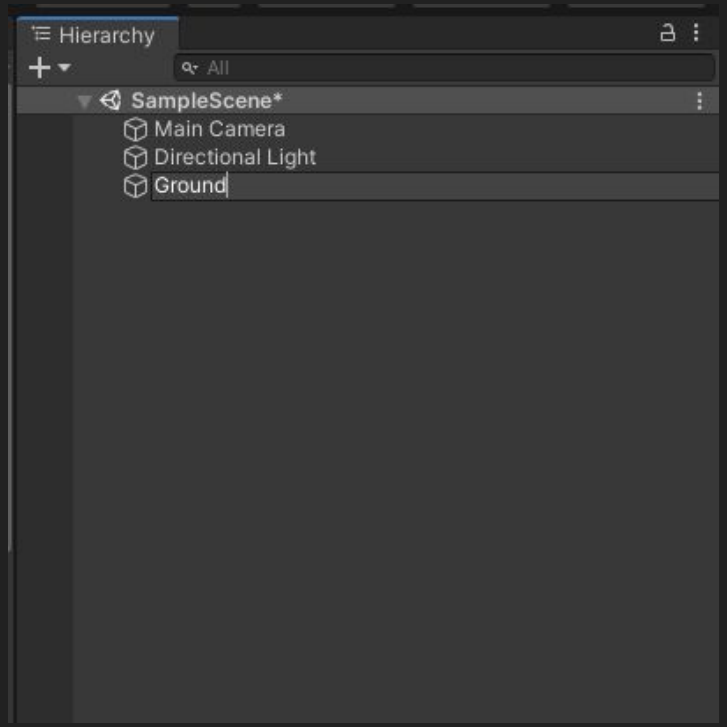
Renaming our Plane

- We want to rename our Plane gameobject to “Ground”
- Right click on the gameobject in the Hierarchy window
- Click: Rename



Renaming our Plane

- We want to rename our Plane gameobject to “Ground”
- Right click on the gameobject in the Hierarchy window
- Click: Rename
- Rename to: Ground

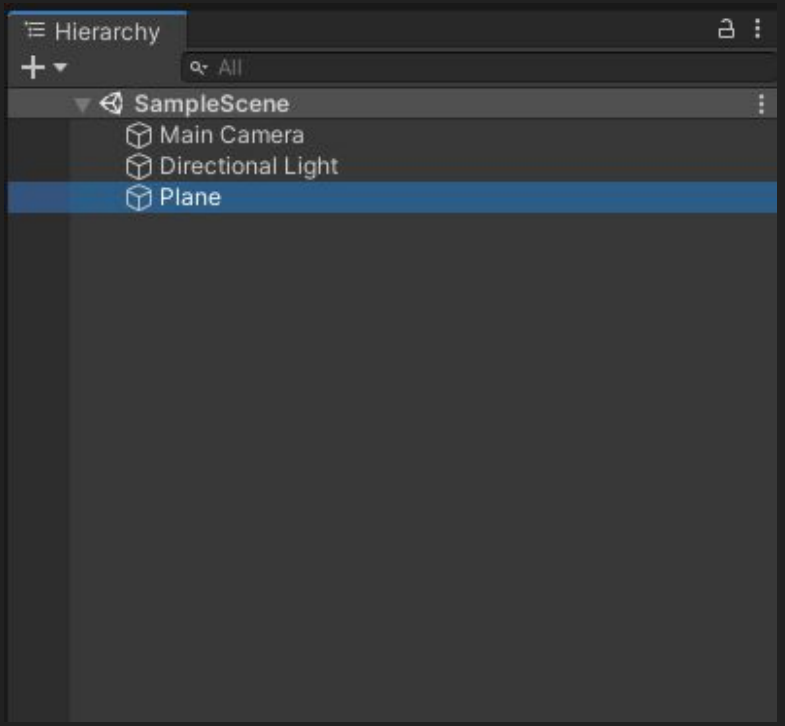


Renaming our Plane (Alternative Method)

- We want to rename our Plane gameobject to “Ground”

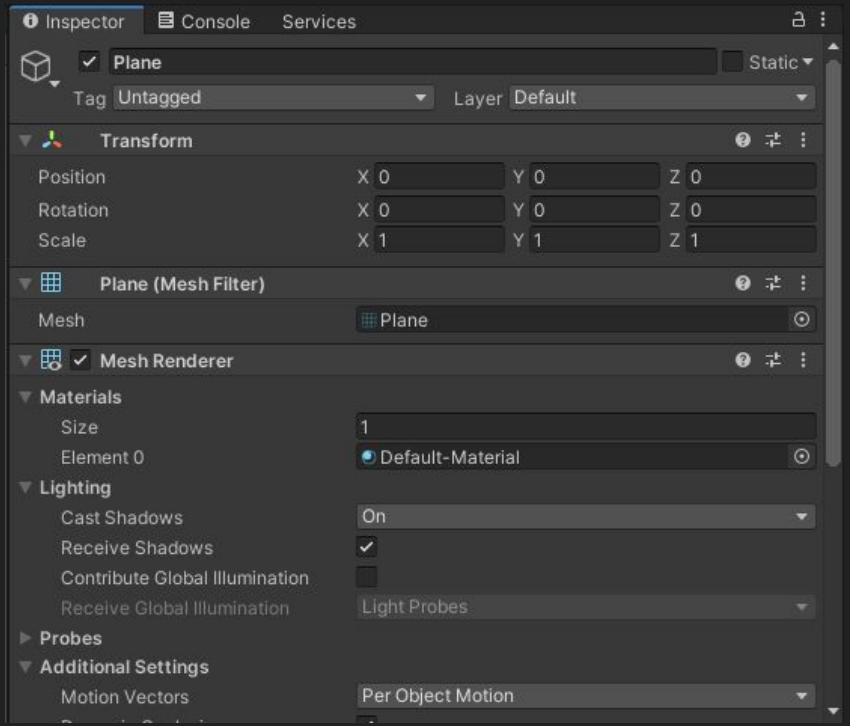
Renaming our Plane (Alternative Method)

- We want to rename our Plane gameobject to “Ground”
- Left click the gameobject in the Hierarchy window to select it



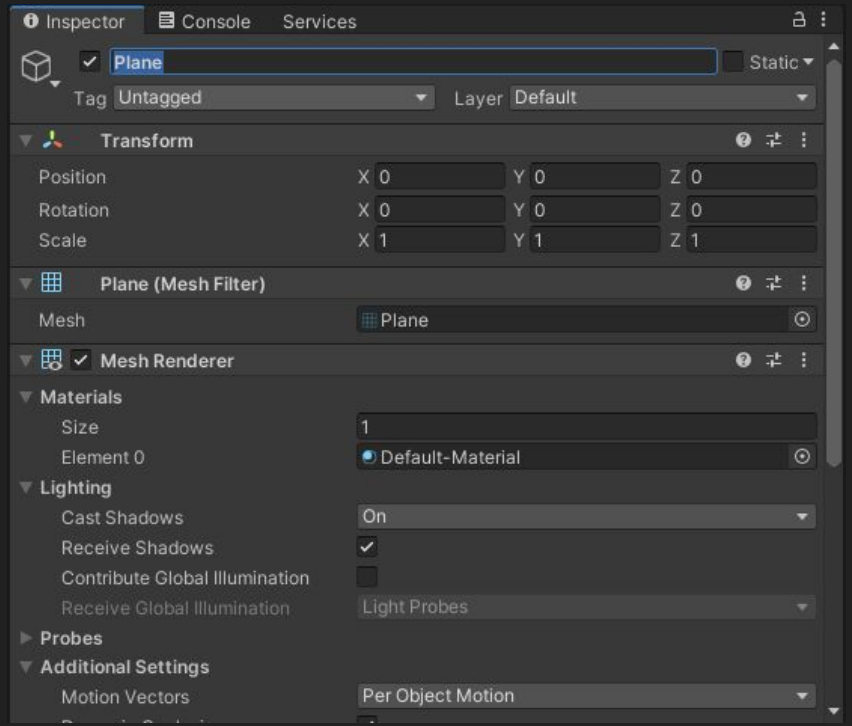
Renaming our Plane (Alternative Method)

- We want to rename our Plane gameobject to “Ground”
- Left click the gameobject in the Hierarchy window to select it
- Find and click on the Inspector tab



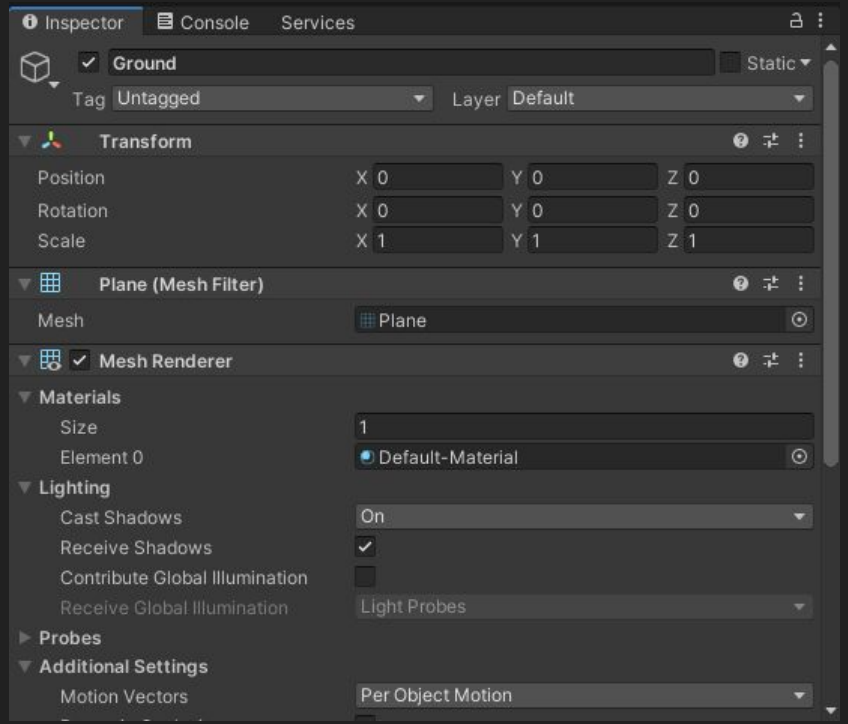
Renaming our Plane (Alternative Method)

- We want to rename our Plane gameobject to “Ground”
- Left click the gameobject in the Hierarchy window to select it
- Find and click on the Inspector tab
- Click on the input text box on the top



Renaming our Plane (Alternative Method)

- We want to rename our Plane gameobject to “Ground”
- Left click the gameobject in the Hierarchy window to select it
- Find and click on the Inspector tab
- Click on the input text box on the top
- Rename to: Ground



Centering the GameObject

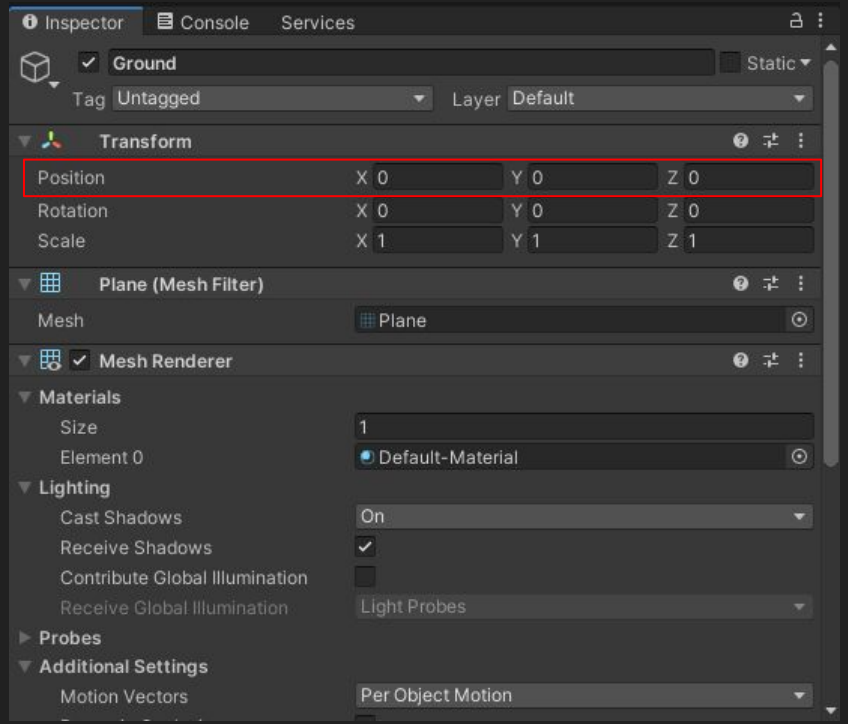
- If our Ground gameobject is at (0, 0, 0) of world space, skip this

Centering the GameObject

- If our Ground gameobject is at (0, 0, 0) of world space, skip this
- If not, we need to center it

Centering the GameObject

- If our Ground gameobject is at (0, 0, 0) of world space, skip this
- If not, we need to center it
- You can do this by manually setting the position in the inspector tab

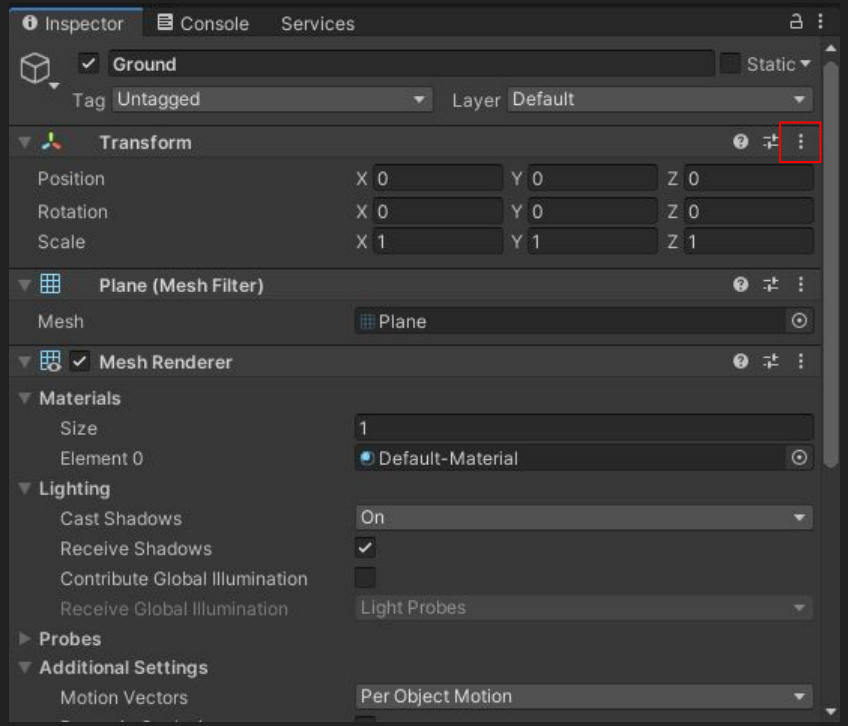


Resetting the Transform

- If you want to reset all values of a gameobject’s Transform

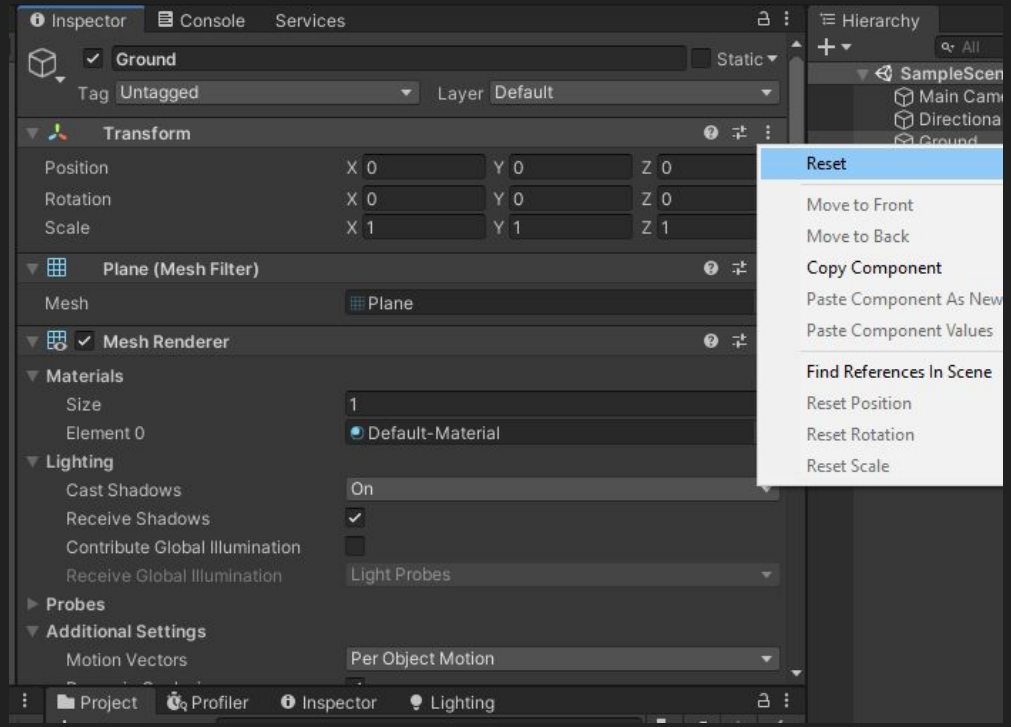
Resetting the Transform

- If you want to reset all values of a gameobject's Transform
- Left click on the three dots in the upper left corner of the Transform component



Resetting the Transform

- If you want to reset all values of a gameobject's Transform
- Left click on the three dots in the upper left corner of the Transform component
- Click: Reset

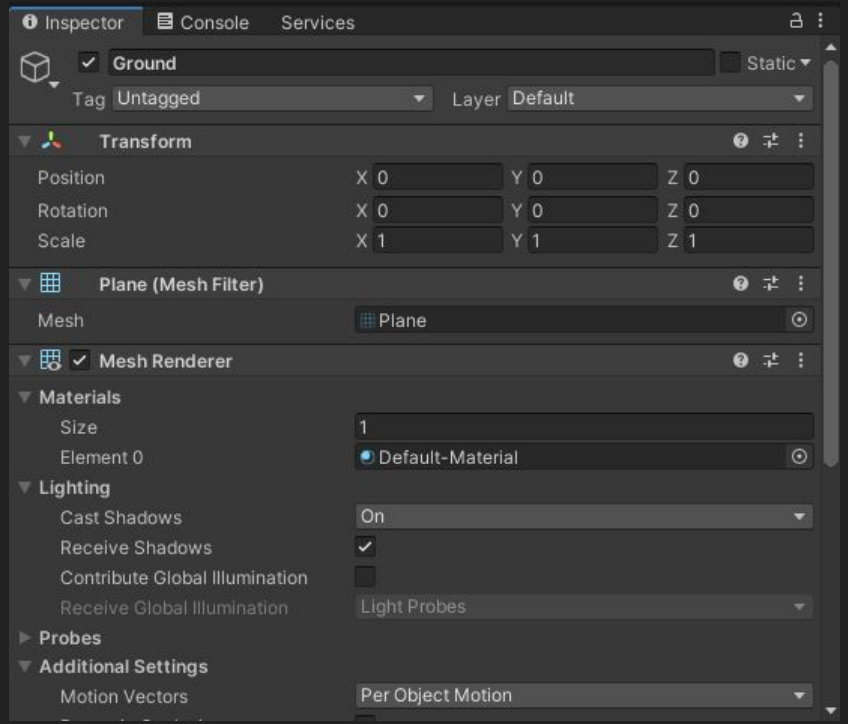


Resize our Ground

- We want to change the scale of our Ground gameobject

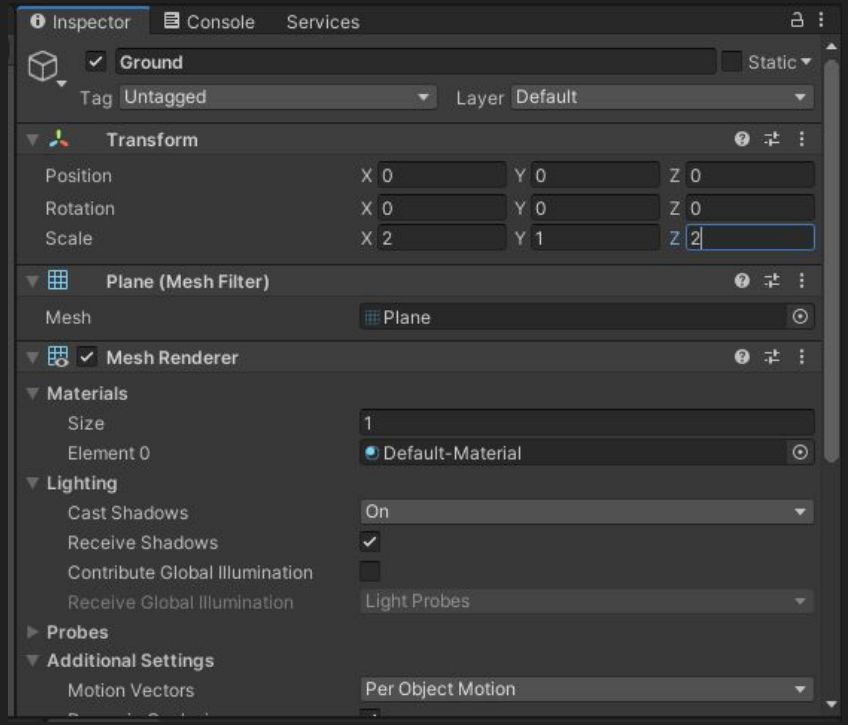
Resize our Ground

- We want to change the scale of our Ground gameobject
- In the Inspector window, find the Transform component



Resize our Ground

- We want to change the scale of our Ground gameobject
- In the Inspector window, find the Transform component
- Set the scale to be: (2, 1, 2) [Be sure to hit enter after typing the values into the boxes]

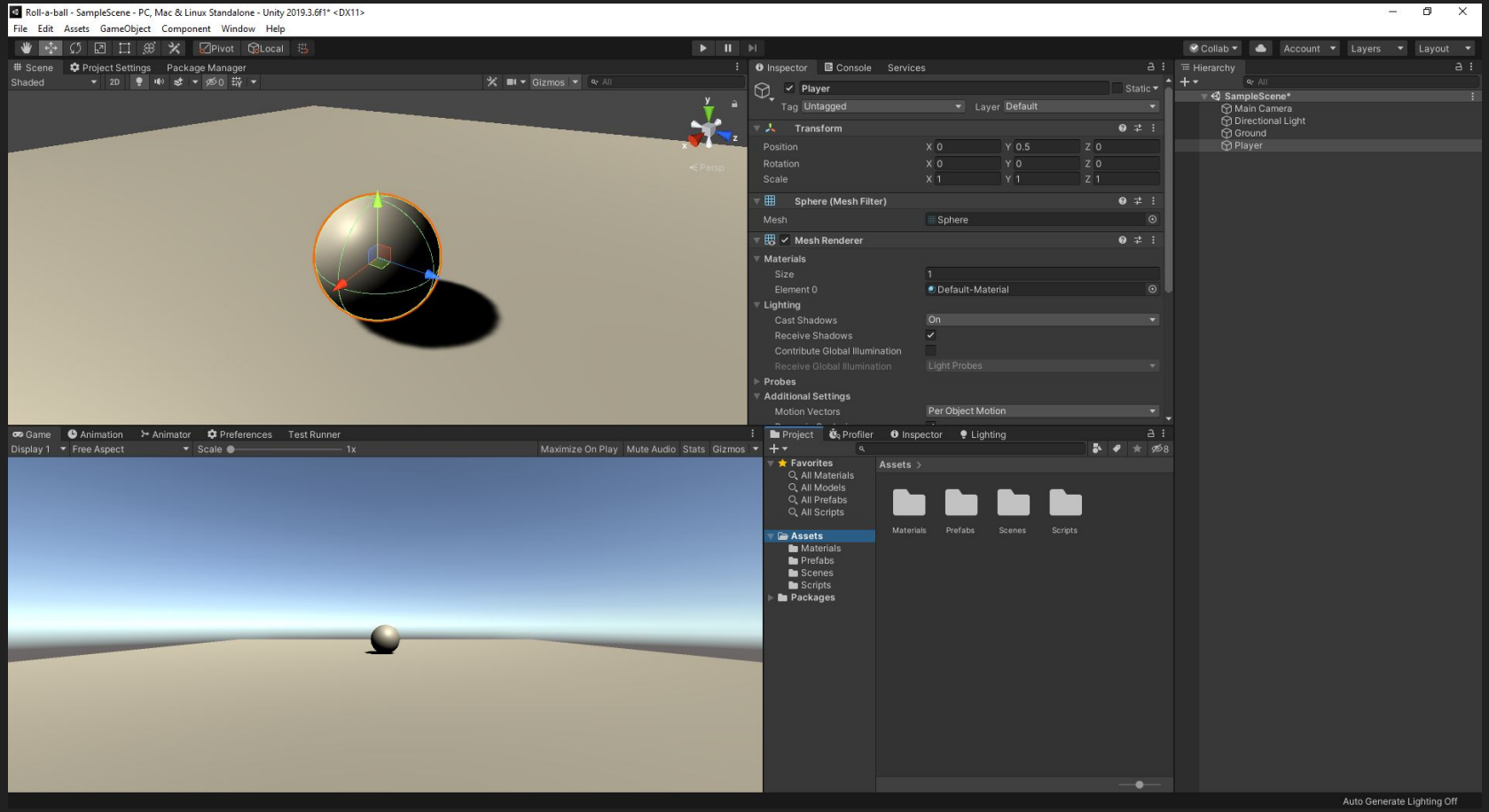


Creating the Player

- Create a Sphere and rename it to Player

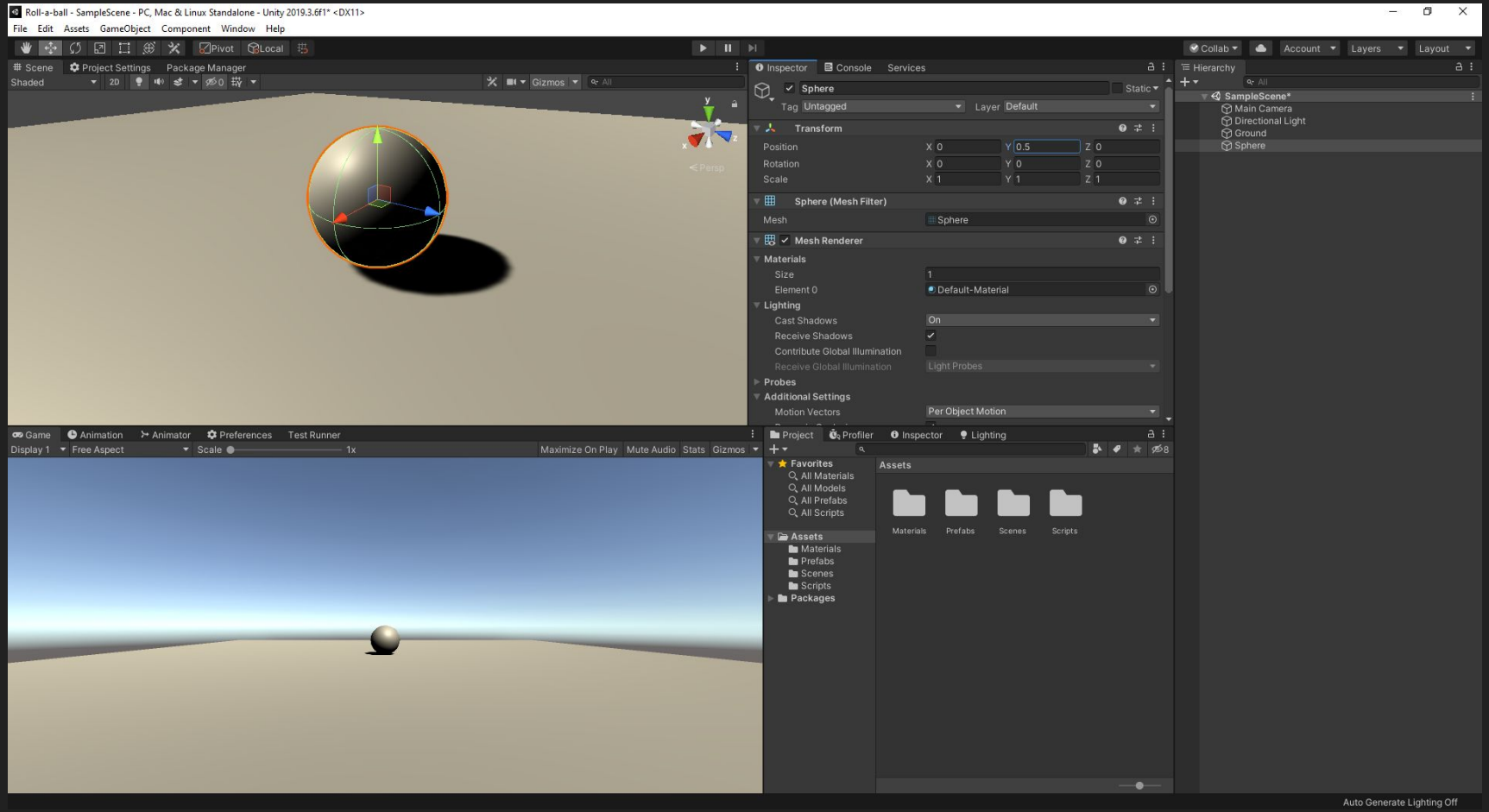
Creating the Player

- Create a Sphere and rename it to Player
- Reset the transform of the Player (Sphere)


Roll-a-ball - SampleScene - PC, Mac & Linux Standalone - Unity 2019.3.6f1* <DX11>
File Edit Assets GameObject Component Window Help
Pivot
Local
Collab
Account
Layers
Layout
Scene
Project Settings
Package Manager
Shaded
2D
Gizmos
Persp
Inspector
Console
Services
Player
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0.5 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Sphere (Mesh Filter)
Mesh Sphere
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows On
Receive Shadows
Contribute Global Illumination
Receive Global Illumination Light Probes
Probes
Additional Settings
Motion Vectors Per Object Motion
Hierarchy
SampleScene*
Main Camera
Directional Light
Ground
Player
Game
Animation
Animator
Preferences
Test Runner
Display 1
Free Aspect
Scale 1x
Maximize On Play
Mute Audio
Stats
Gizmos
Project
Profiler
Inspector
Lighting
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Materials
Prefabs
Scenes
Scripts
Packages
Auto Generate Lighting Off

Creating the Player

- Create a Sphere and rename it to Player
- Reset the transform of the Player (Sphere)
- Move the Player to (0, .5, 0) so it is sitting on top of the Ground


Roll-a-ball - SampleScene - PC, Mac & Linux Standalone - Unity 2019.3.6f1* <DX11>
File Edit Assets GameObject Component Window Help
Pivot Local
Collab Account Layers Layout
Scene Project Settings Package Manager
Shaded 2D Gizmos
Persp
Inspector Console Services
Sphere Static
Tag Untagged Layer Default
Transform
Position X 0 Y 0.5 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Sphere (Mesh Filter)
Mesh Sphere
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows On
Receive Shadows
Contribute Global Illumination
Receive Global Illumination Light Probes
Probes
Additional Settings
Motion Vectors Per Object Motion
Hierarchy
SampleScene*
Main Camera
Directional Light
Ground
Sphere
Game Animation Animator Preferences Test Runner
Display 1 Free Aspect Scale 1x Maximize On Play Mute Audio Stats Gizmos
Project Profiler Inspector Lighting
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Materials
Prefabs
Scenes
Scripts
Packages
Assets
Materials Prefabs Scenes Scripts
Auto Generate Lighting Off

Save Often!

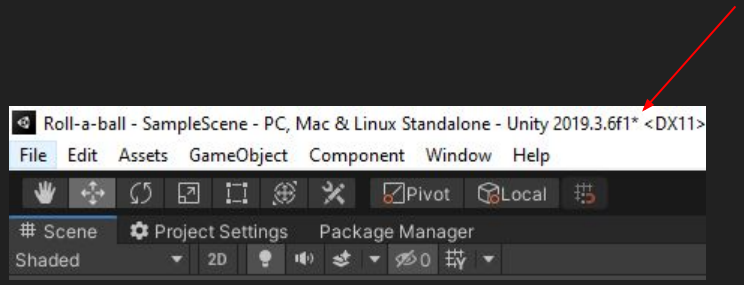
- Unity does not auto-save your work inside the scenes

Save Often!

- Unity does not auto-save your work inside the scenes
  - It actually does… but it doesn’t...

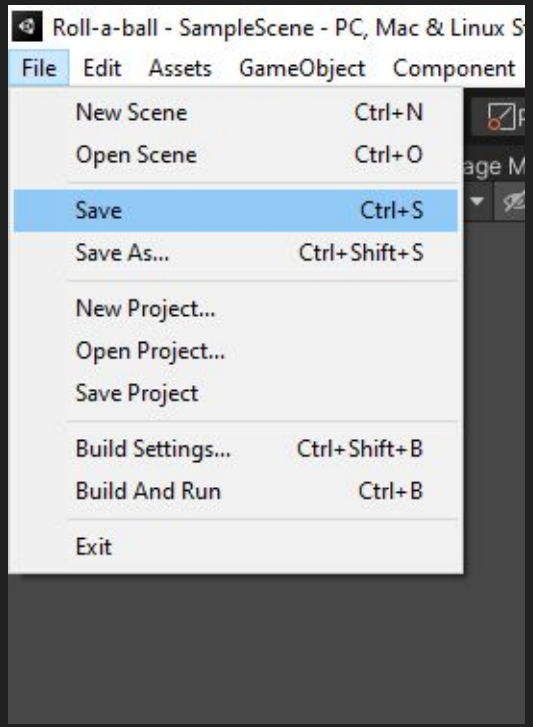
Save Often!

- Unity does not auto-save your work inside the scenes
  - It actually does… but it doesn't...
- * means that you have unsaved work



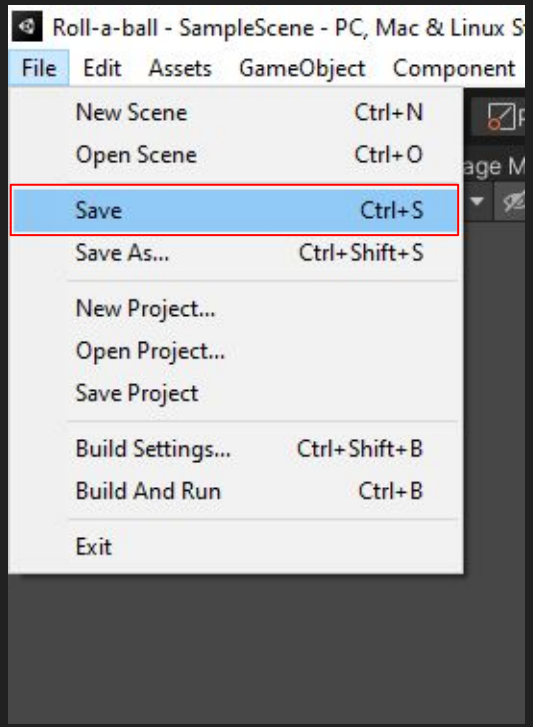
Save Often!

- Unity does not auto-save your work inside the scenes
  - It actually does… but it doesn't...
- * means that you have unsaved work
- Click File → Save



Save Often!

- Unity does not auto-save your work inside the scenes
  - It actually does… but it doesn't...
- * means that you have unsaved work
- Click File → Save
- Ctrl + S also saves



Creating Materials

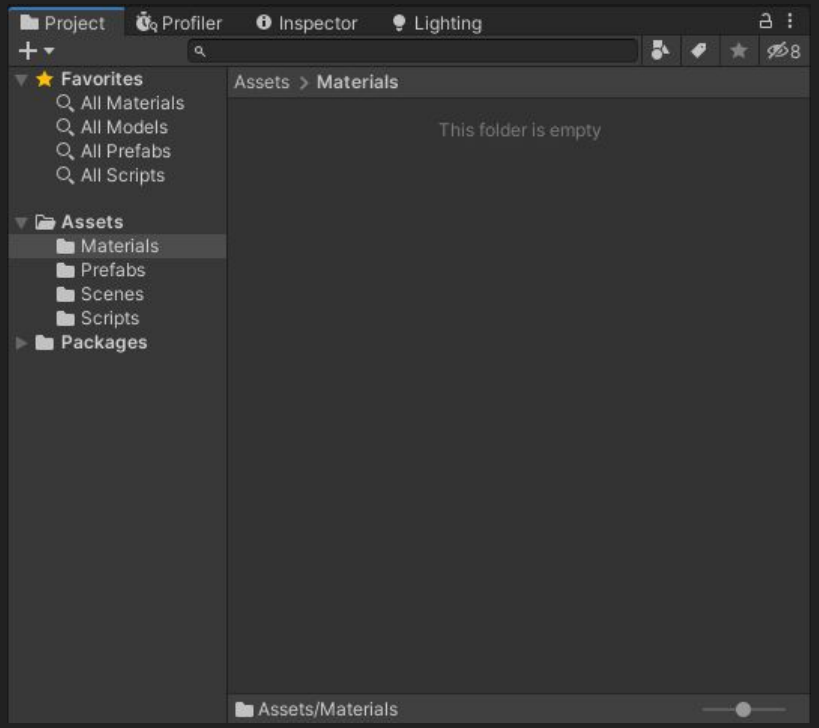
- Currently our player and ground are the same color

Creating Materials

- Currently our player and ground are the same color
- This is undesirable

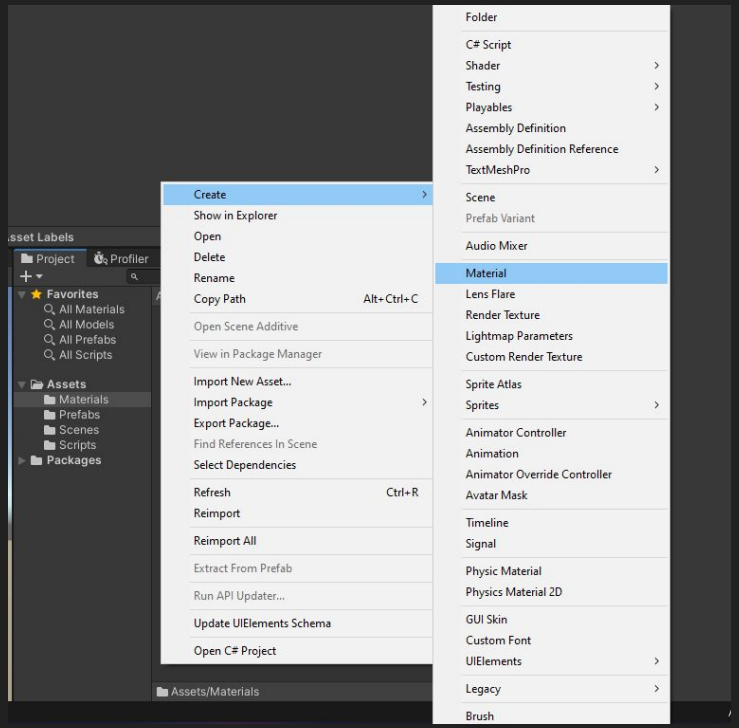
Creating Materials

- Currently our player and ground are the same color
- This is undesirable
- Navigate to the Materials folder



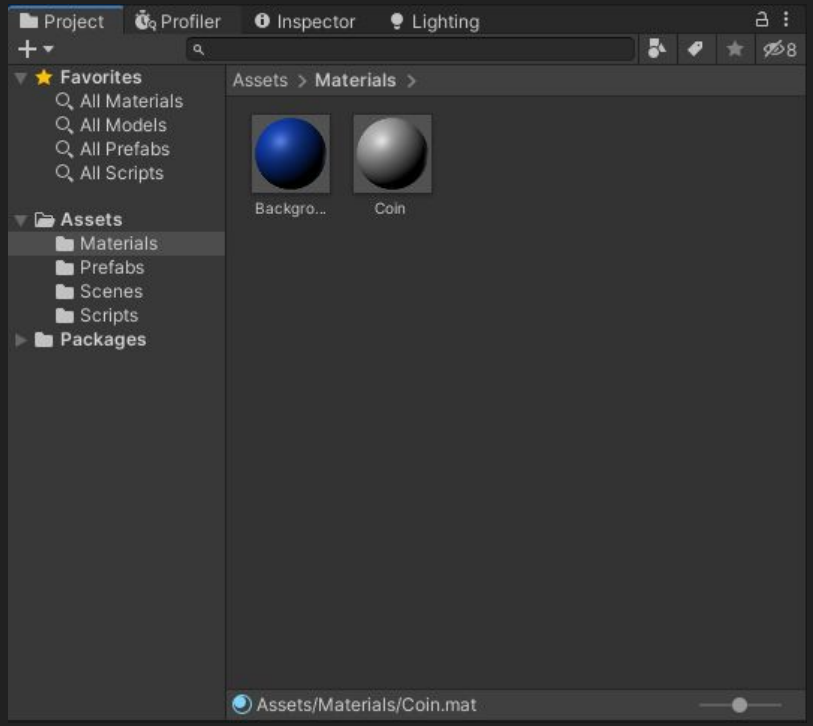
Creating Materials

- Currently our player and ground are the same color
- This is undesirable
- Navigate to the Materials folder
- Create a material called Background



Creating Materials

- Currently our player and ground are the same color
- This is undesirable
- Navigate to the Materials folder
- Create a material called Background
- Create a material called Coin

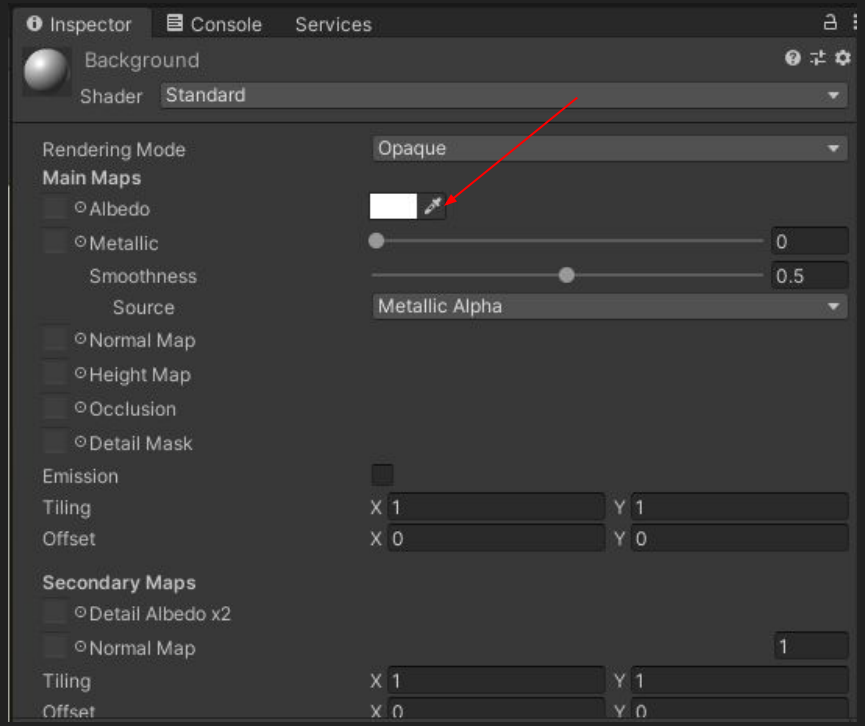


Editing Materials

- Select the material in the Project window

Editing Materials

- Select the material in the Project window
- Click on the eyedropper in the Inspector Window

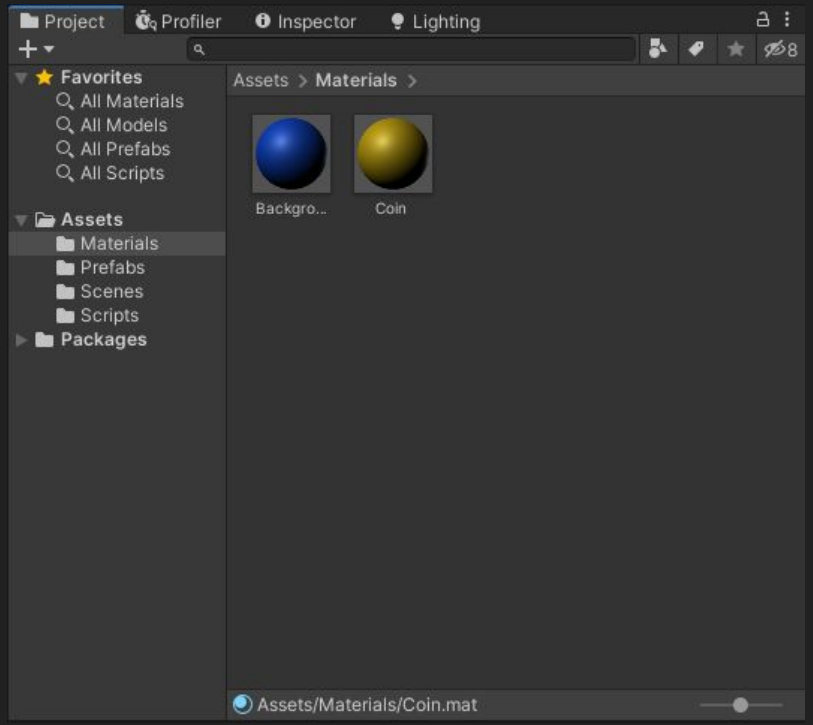


Editing Materials

- Select the material in the Project window
- Click on the eyedropper in the Inspector Window
- Choose a color

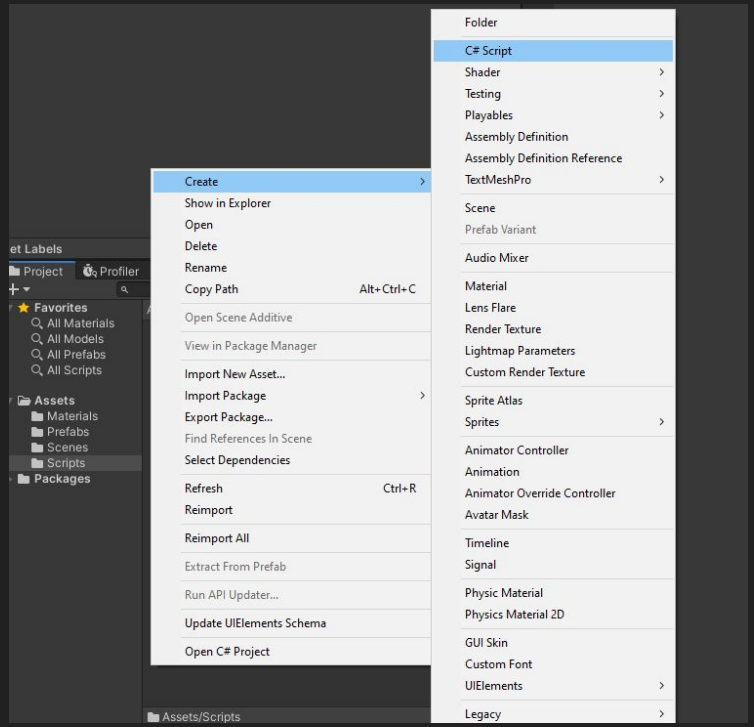
Editing Materials

- Select the material in the Project window
- Click on the eyedropper in the Inspector Window
- Choose a color
- Repeat for the other material



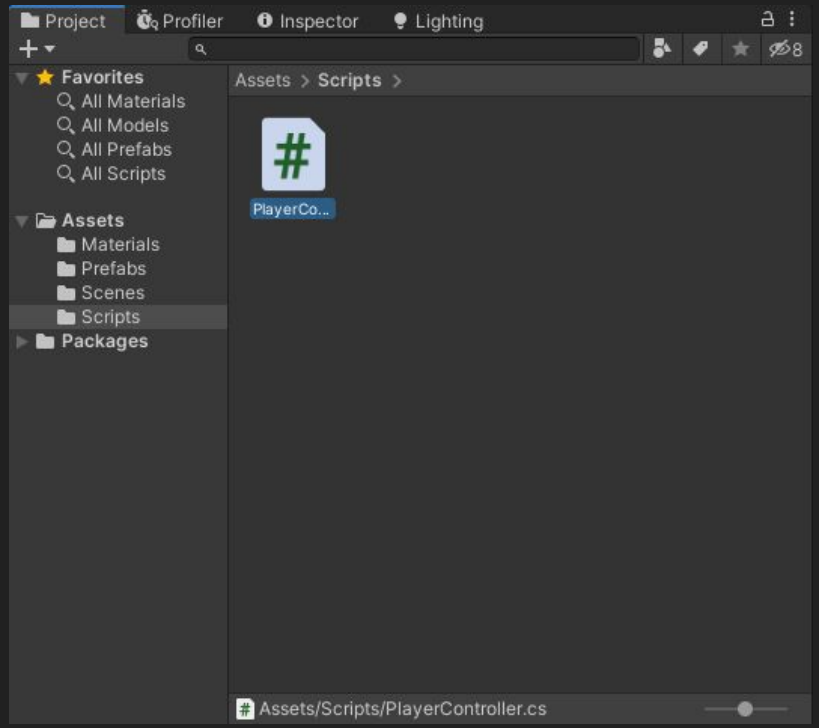
Player Controller

- Create a C# script to handle player movement



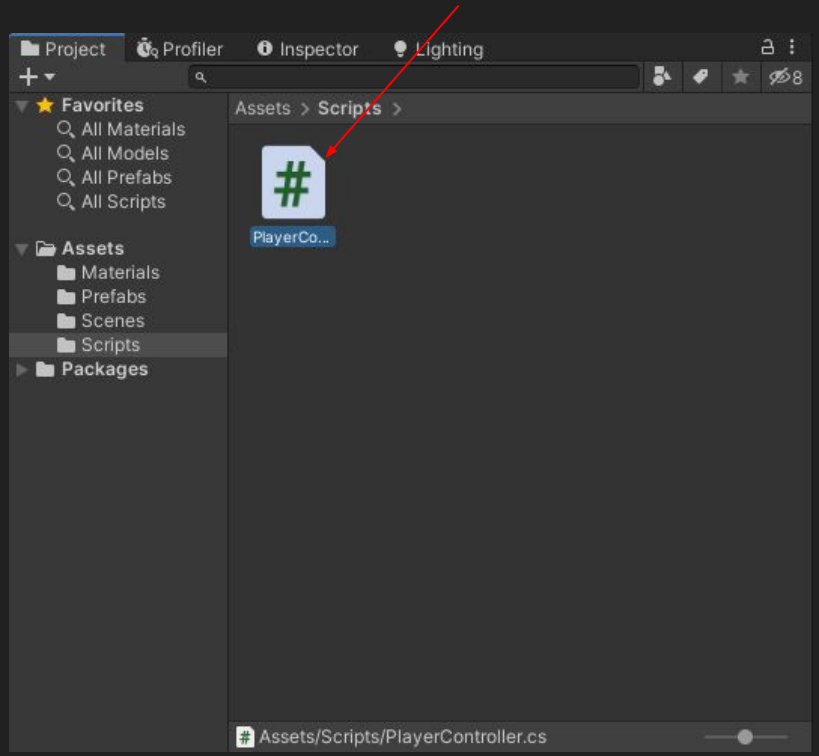
Player Controller

- Create a C# script to handle player movement
- Name it PlayerController.cs



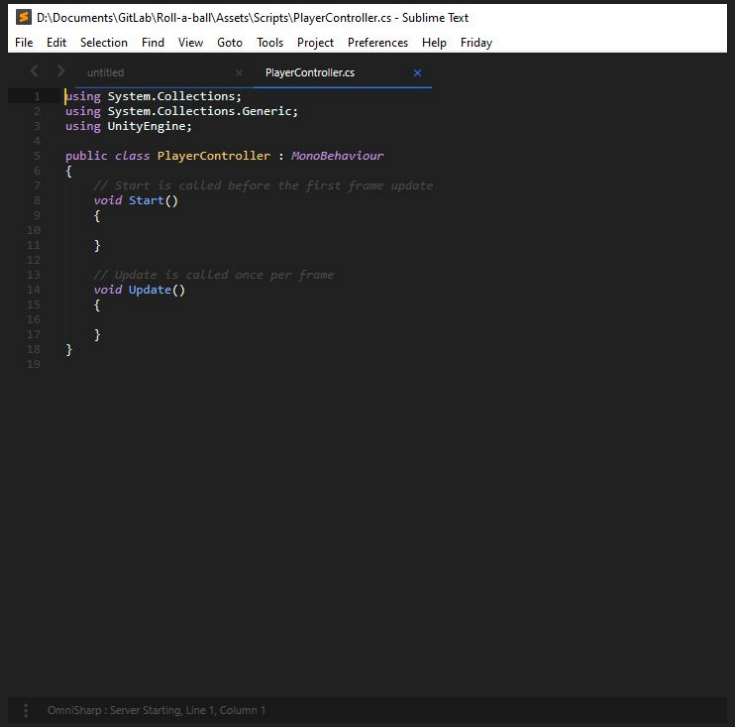
Player Controller

- Create a C# script to handle player movement
- Name it PlayerController.cs
- Double click on the file to open it



Player Controller

- Create a C# script to handle player movement
- Name it PlayerController.cs
- Double click on the file to open it
- I use Sublime Text as my IDE but any text editor will work for this



Player Controller

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

Player Controller

- Get a reference to the RigidBody component attached to the gameobject

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

Player Controller

- Get a reference to the RigidBody component attached to the gameobject
- Get user input from the keyboard and store as a float (Left, Right, Up, Down)

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

Player Controller

- Get a reference to the RigidBody component attached to the gameobject
- Get user input from the keyboard and store as a float (Left, Right, Up, Down)
- Combine the movement values into a Vector3 (gives our movement a direction)

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

Player Controller

- Get a reference to the RigidBody component attached to the gameobject
- Get user input from the keyboard and store as a float (Left, Right, Up, Down)
- Combine the movement values into a Vector3 (gives our movement a direction)
- Multiply by a variable (we can modify this value in the editor)

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

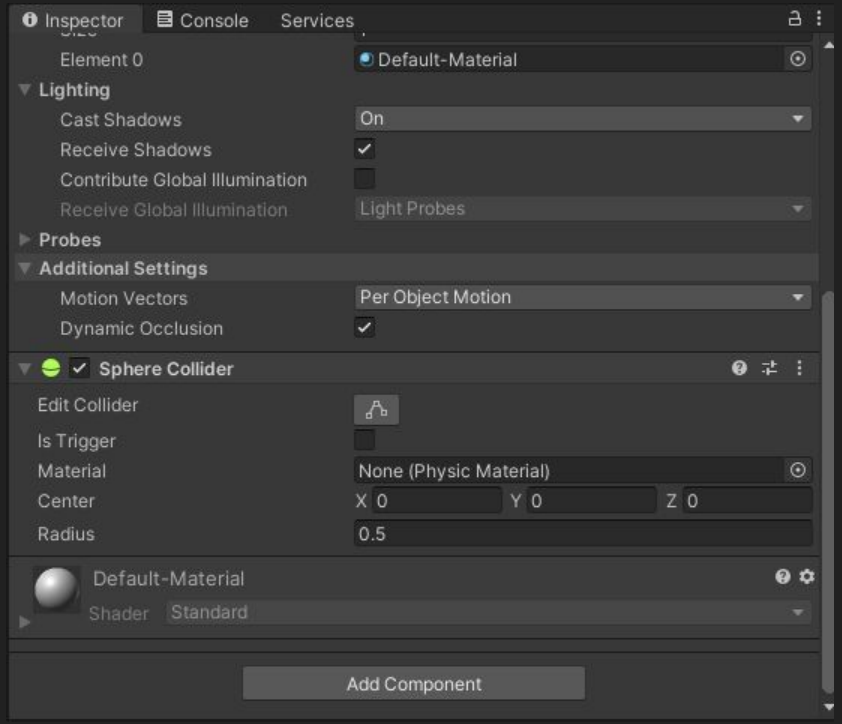
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

Adding the Script to the GameObject

- Select the player object in the Hierarchy

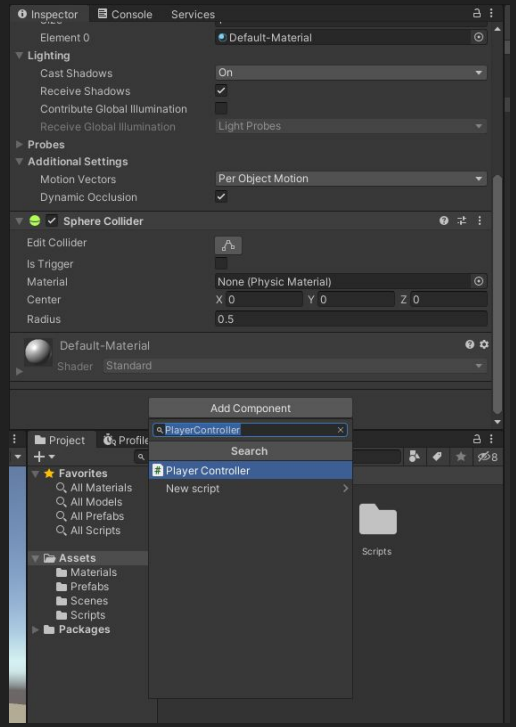
Adding the Script to the GameObject

- Select the player object in the Hierarchy
- In the Inspector window, scroll to the bottom



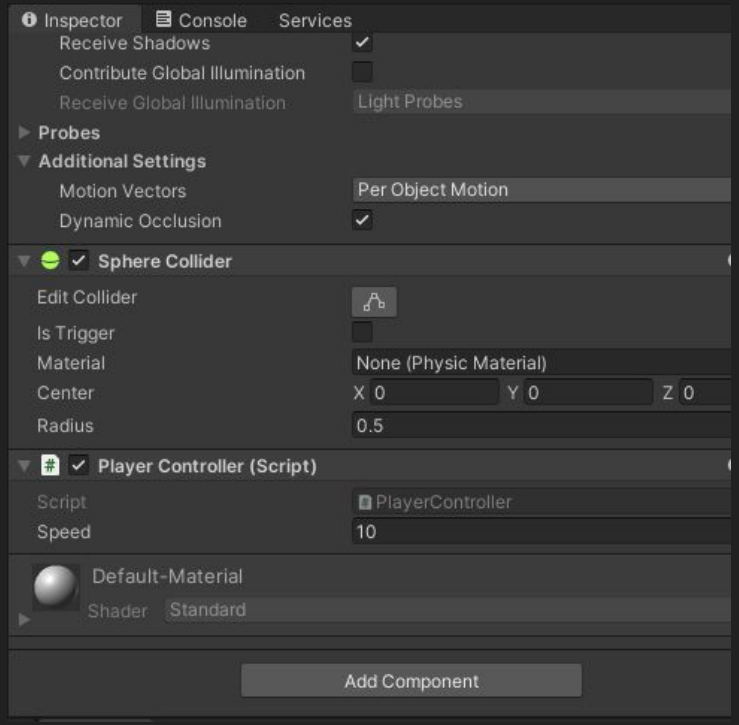
Adding the Script to the GameObject

- Select the player object in the Hierarchy
- In the Inspector window, scroll to the bottom
- Add the PlayerController component



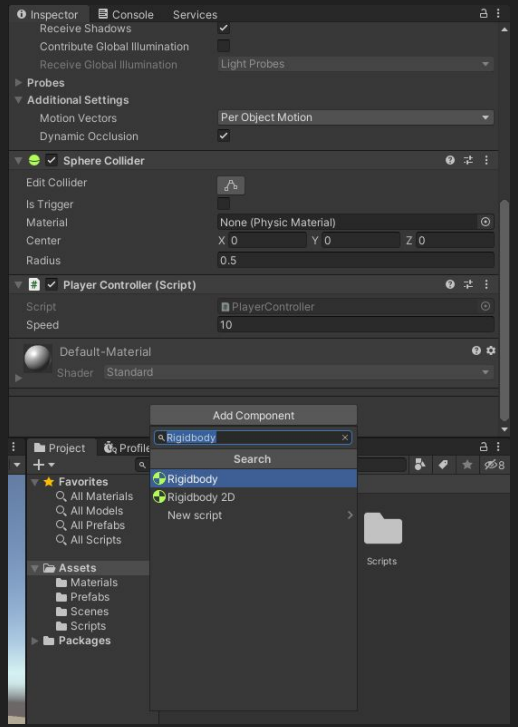
Adding the Script to the GameObject

- Select the player object in the Hierarchy
- In the Inspector window, scroll to the bottom
- Add the PlayerController component
- Set the value of the Speed variable



Adding the Script to the GameObject

- Select the player object in the Hierarchy
- In the Inspector window, scroll to the bottom
- Add the PlayerController component
- Set the value of the Speed variable
- Add the RigidBody component



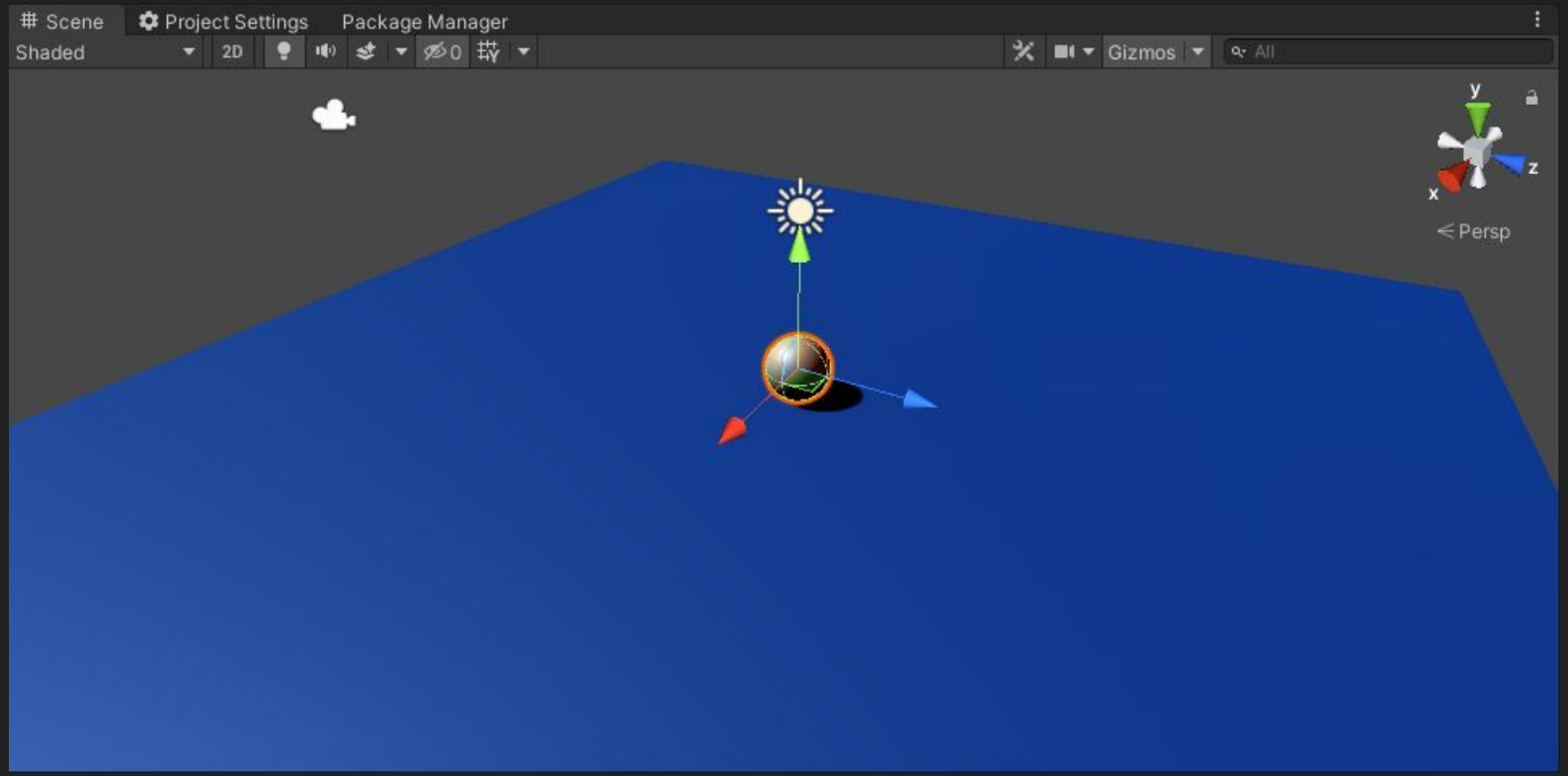
Ah!

- We forgot to give our ground object the material we made…

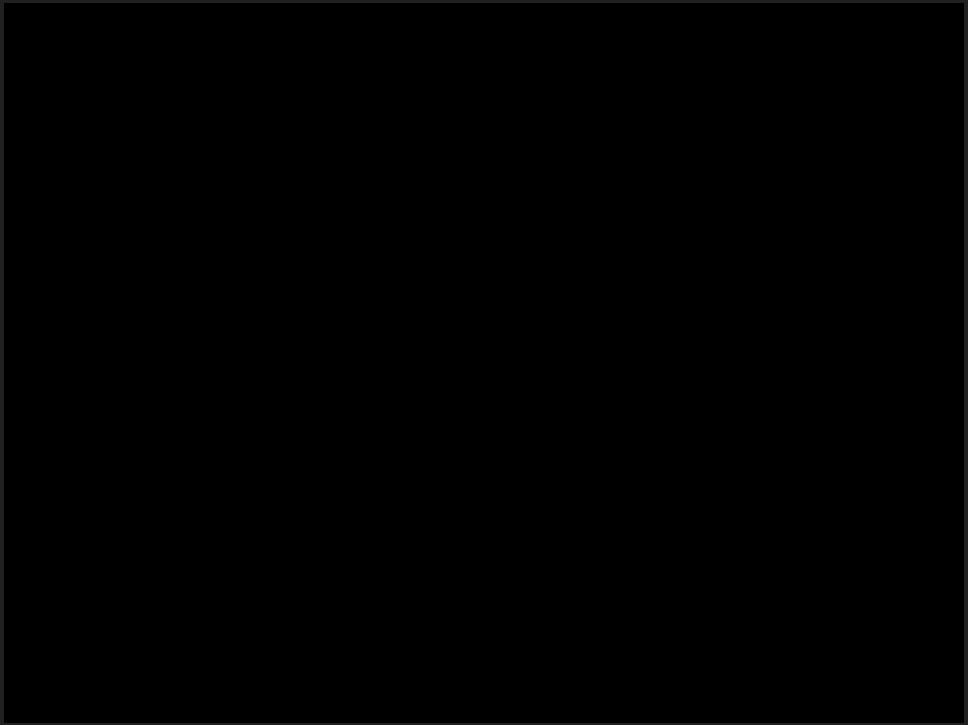
Ah!

- We forgot to give our ground object the material we made…
- Click and drag the Background material from the project window onto the desired gameobject in the Scene window to apply

Looks better



What we have so far:

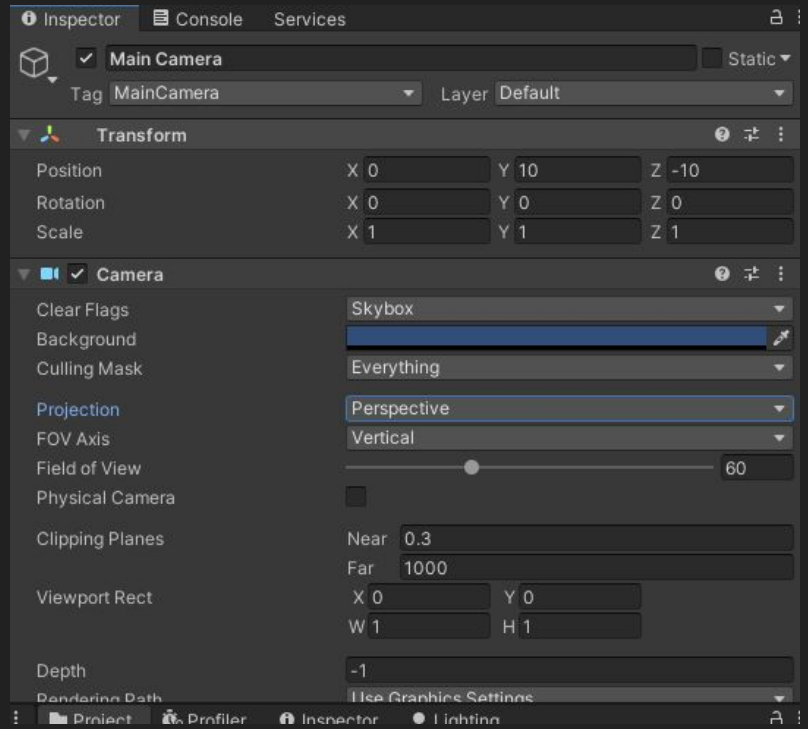


Reorient the Camera

- Select the Main Camera object in the Hierarchy

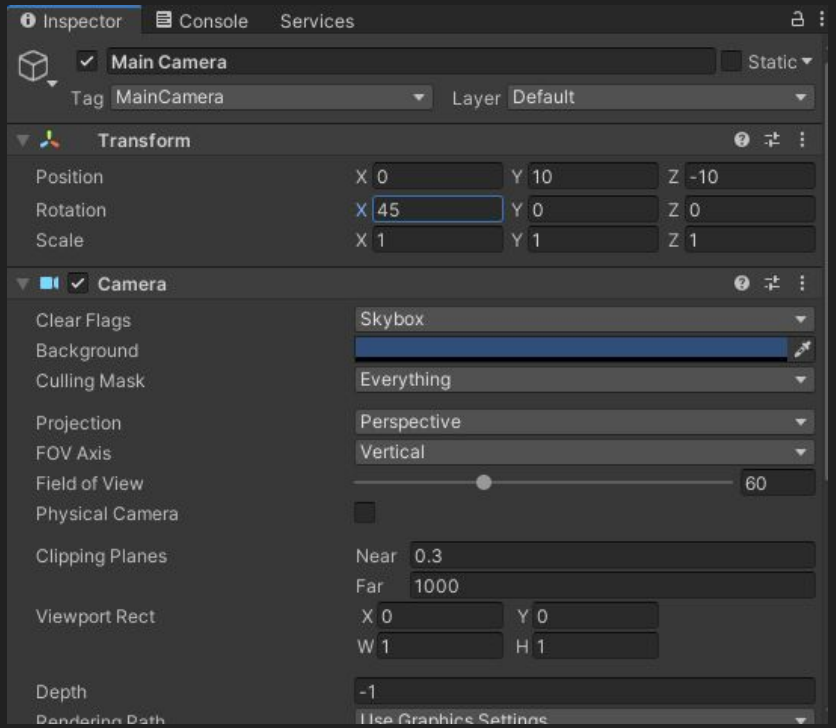
Reorient the Camera

- Select the Main Camera object in the Hierarchy
- Set the position of the camera to be (0, 10, -10)



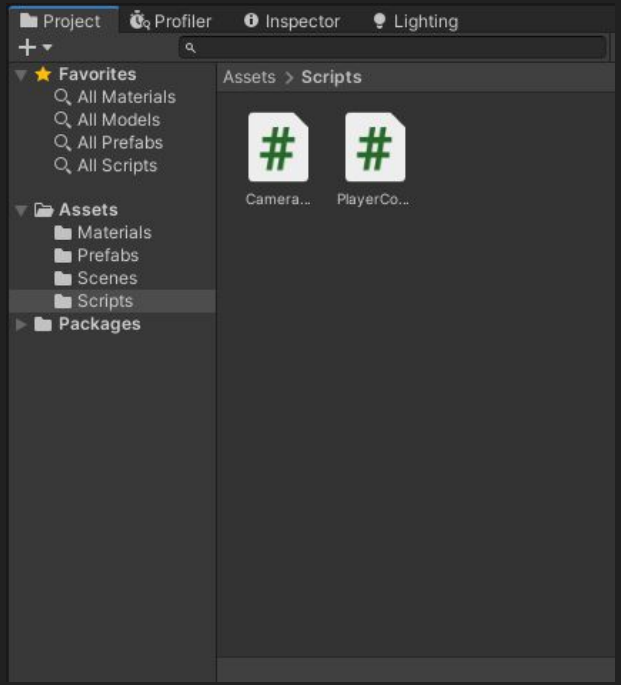
Reorient the Camera

- Select the Main Camera object in the Hierarchy
- Set the position of the camera to be (0, 10, -10)
- Set the rotation of the camera to be (45, 0, 0)



Camera Controller

- Create a C# file called CameraController



Camera Controller

- Create a C# file called CameraController
- Open the CameraController.cs file in your IDE

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\CameraController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs | CameraController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraController : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Camera Controller

- Create a C# file called CameraController
- Open the CameraController.cs file in your IDE
- Create a reference to a gameobject

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\CameraController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs | CameraController.cs

```
using UnityEngine;
using System.Collections;

public class CameraController : MonoBehaviour {

    public GameObject player;

    private Vector3 offset;

    void Start ()
    {
        offset = transform.position - player.transform.position;
    }

    void LateUpdate ()
    {
        transform.position = player.transform.position + offset;
    }
}
```

Camera Controller

- Create a C# file called CameraController
- Open the CameraController.cs file in your IDE
- Create a reference to a gameobject
- Store the offset vector of the camera and the player

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\CameraController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs | CameraController.cs

```
using UnityEngine;
using System.Collections;

public class CameraController : MonoBehaviour {

    public GameObject player;

    private Vector3 offset;

    void Start ()
    {
        offset = transform.position - player.transform.position;
    }

    void LateUpdate ()
    {
        transform.position = player.transform.position + offset;
    }
}
```

Camera Controller

- Create a C# file called CameraController
- Open the CameraController.cs file in your IDE
- Create a reference to a gameobject
- Store the offset vector of the camera and the player
- Update the camera's position to be the sum of the offset plus the "player's" current position

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\CameraController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs CameraController.cs

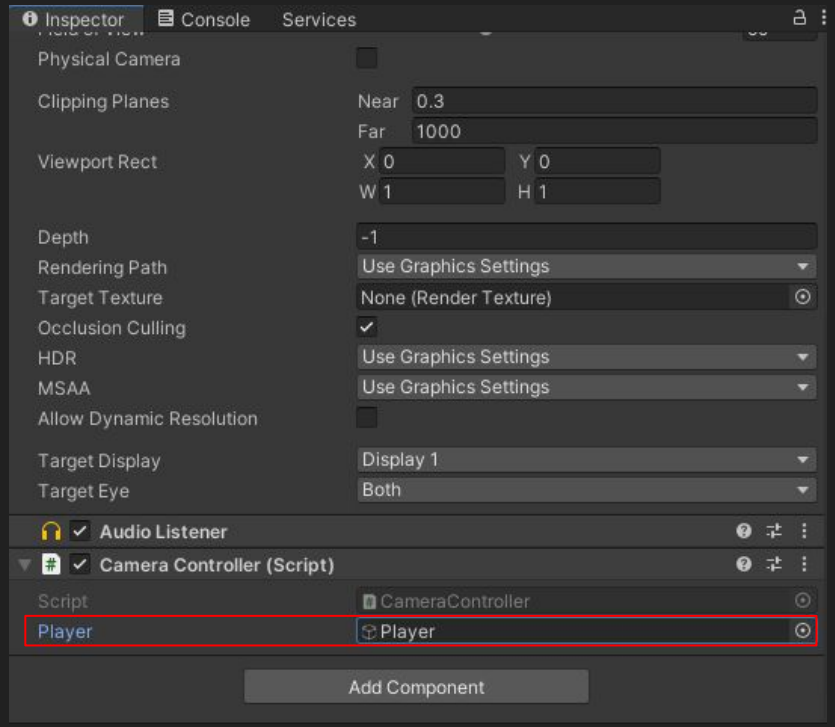
```
using UnityEngine;
using System.Collections;

public class CameraController : MonoBehaviour {

    public GameObject player;

    private Vector3 offset;

    void Start ()
    {
        offset = transform.position - player.transform.position;
    }

    void LateUpdate ()
    {
        transform.position = player.transform.position + offset;
    }
}
```

Camera Controller

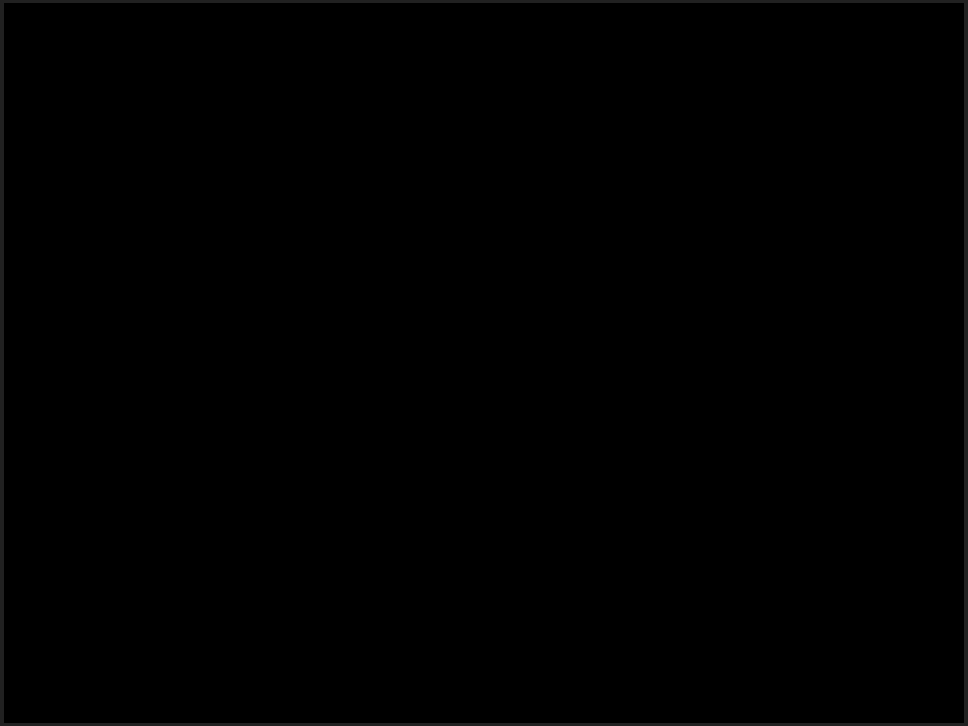
- Add the CameraController component to the Main Camera object

Camera Controller

- Add the CameraController component to the Main Camera object
- Click and drag the Player object from the Hierarchy

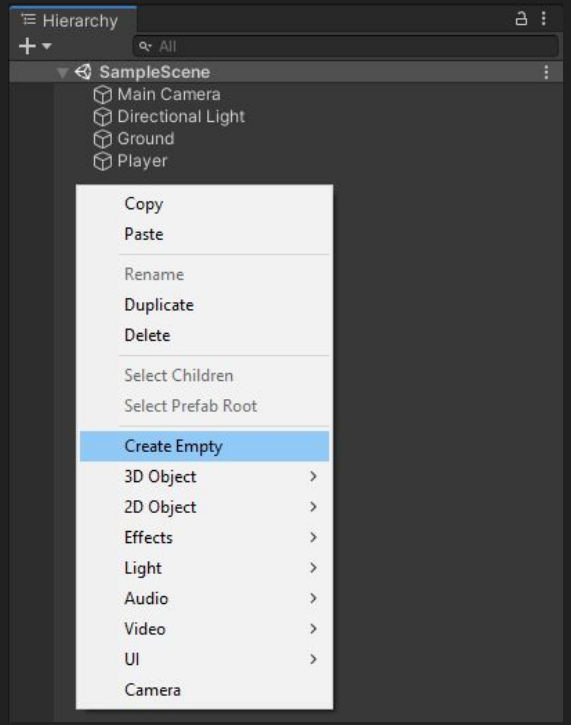


What we have so far:



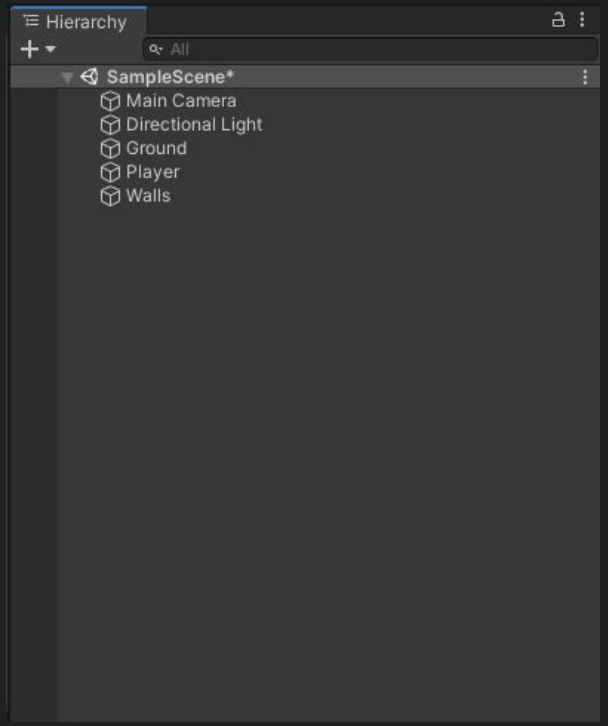
Adding Walls

- Create an Empty GameObject in the Hierarchy



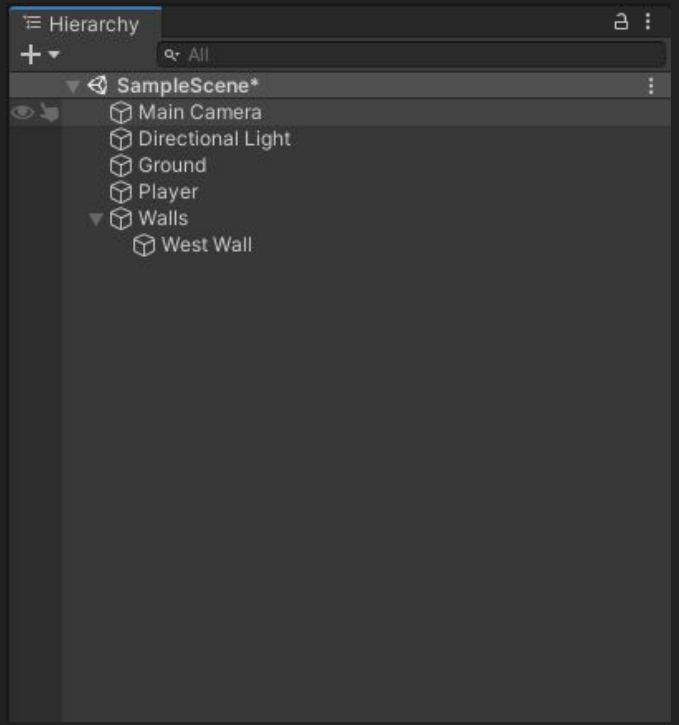
Adding Walls

- Create an Empty GameObject in the Hierarchy
- Name it “Walls” and reset it’s Transform



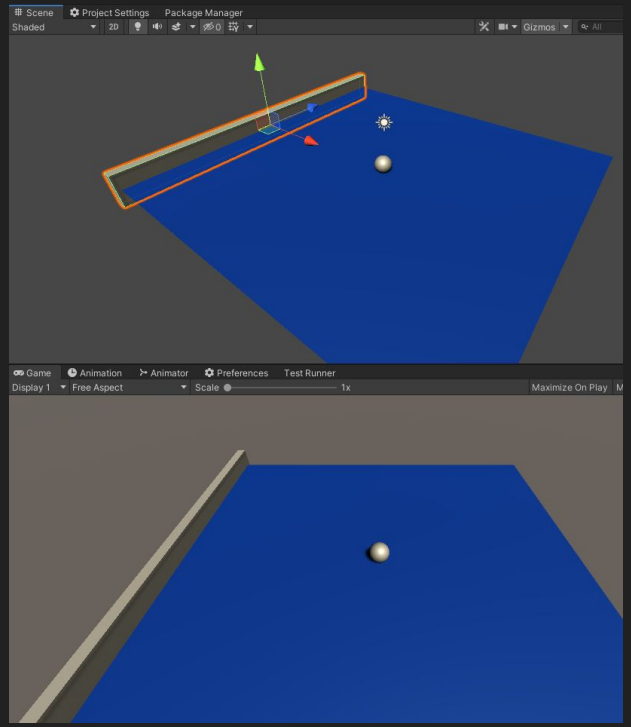
Adding Walls

- Create an Empty GameObject in the Hierarchy
- Name it “Walls” and reset it’s Transform
- Create a Cube, name it “West Wall”, reset it’s Transform, and make it a child of Walls



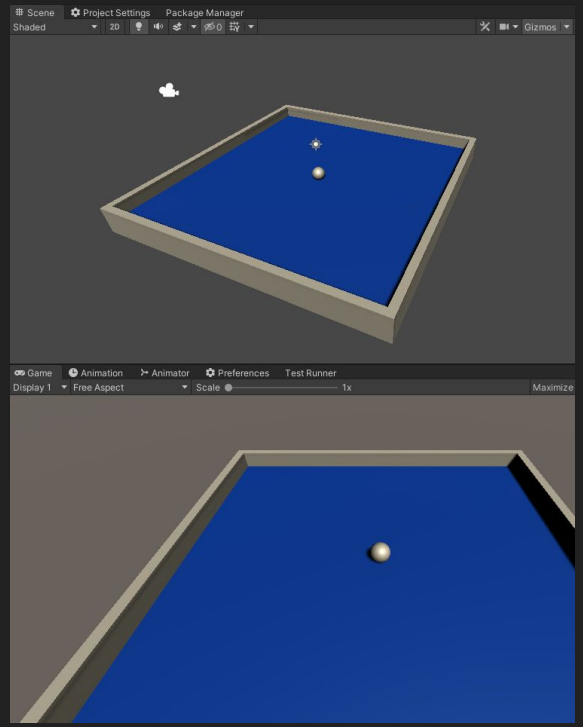
Adding Walls

- Create an Empty GameObject in the Hierarchy
- Name it “Walls” and reset it’s Transform
- Create a Cube, name it “West Wall”, reset it’s Transform, and make it a child of Walls
- Set the wall’s scale to be (.5, 2, 20.5) and make it’s position (-10, 0, 0)



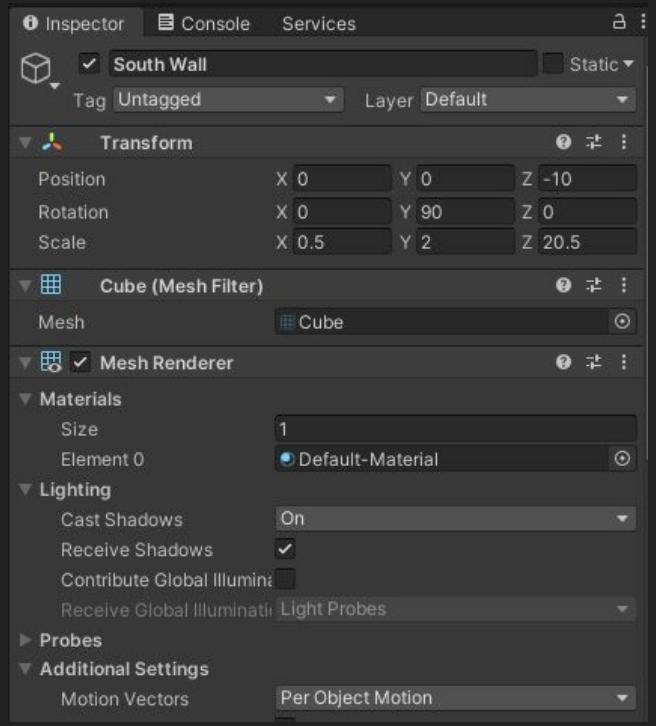
Adding Walls

- Create an Empty GameObject in the Hierarchy
- Name it “Walls” and reset it’s Transform
- Create a Cube, name it “West Wall”, reset it’s Transform, and make it a child of Walls
- Set the wall’s scale to be (.5, 2, 20.5) and make it’s position (-10, 0, 0)
- Do this for the other three walls



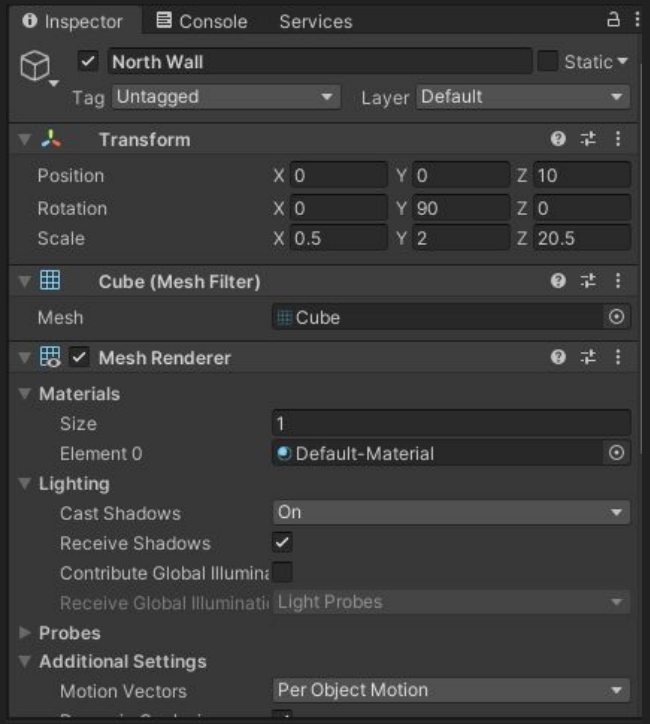
Adding Walls

- Create an Empty GameObject in the Hierarchy
- Name it “Walls” and reset it’s Transform
- Create a Cube, name it “West Wall”, reset it’s Transform, and make it a child of Walls
- Set the wall’s scale to be (.5, 2, 20.5) and make it’s position (-10, 0, 0)
- Do this for the other three walls
    - For the north and south walls you will need to rotate them by 90° on the Y-axis



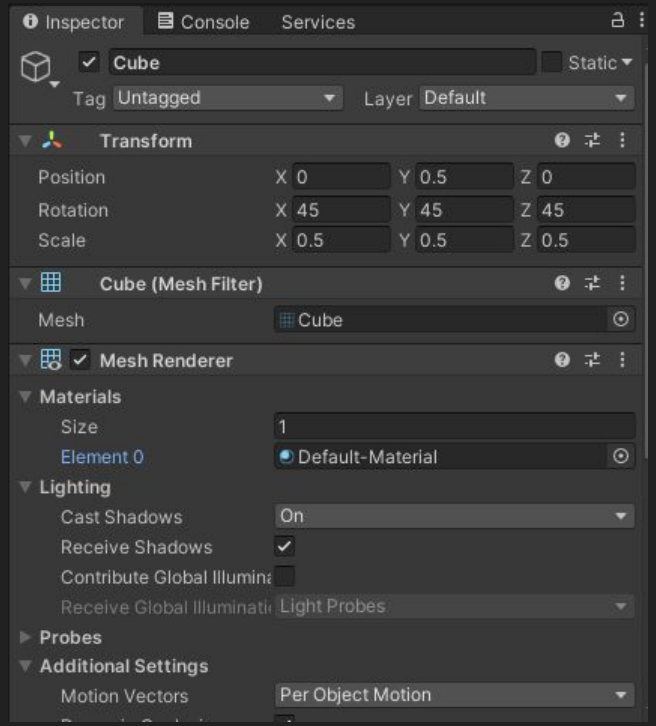
Adding Walls

- Create an Empty GameObject in the Hierarchy
- Name it “Walls” and reset it’s Transform
- Create a Cube, name it “West Wall”, reset it’s Transform, and make it a child of Walls
- Set the wall’s scale to be (.5, 2, 20.5) and make it’s position (-10, 0, 0)
- Do this for the other three walls
  - For the north and south walls you will need to rotate them by 90° on the Y-axis
  - You also need to translate them on their Z-axes instead of the X



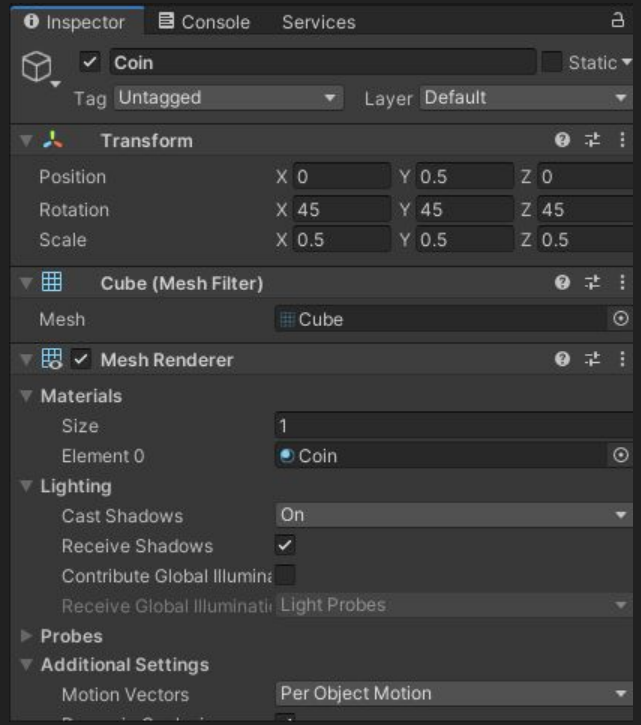
Creating “Coins”

- Create a Cube, set it’s position to (0, .5, 0), rotation to (45, 45, 45), and scale to (.5, .5, .5)



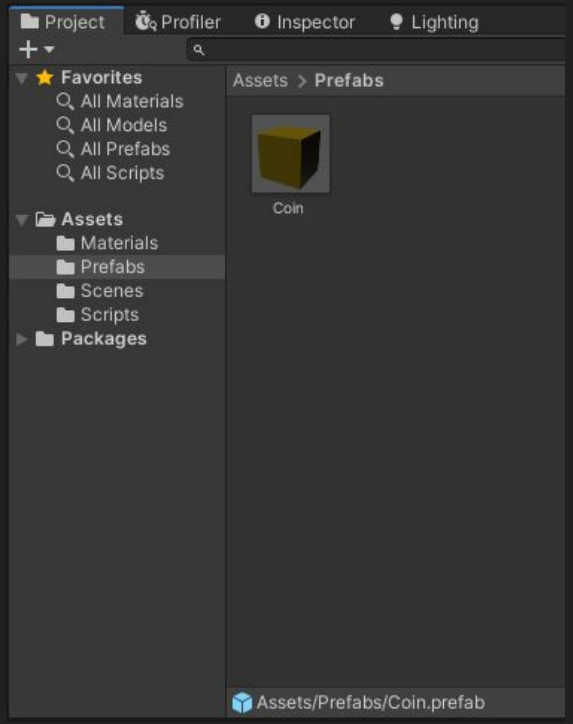
Creating “Coins”

- Create a Cube, set it’s position to (0, .5, 0), rotation to (45, 45, 45), and scale to (.5, .5, .5)
- Name it “Coin” and apply the Coin material to it



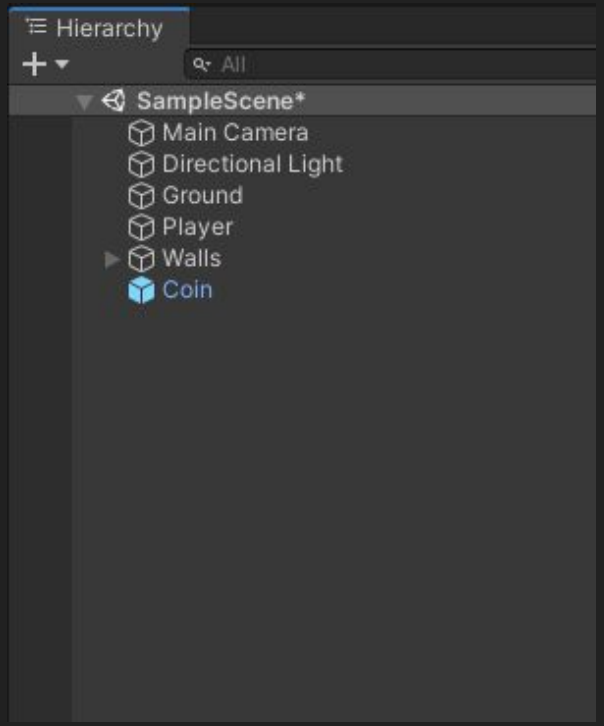
Creating “Coins”

- Create a Cube, set it’s position to (0, .5, 0), rotation to (45, 45, 45), and scale to (.5, .5, .5)
- Name it “Coin” and apply the Coin material to it
- Click and drag from the Hierarchy into the Prefabs folder



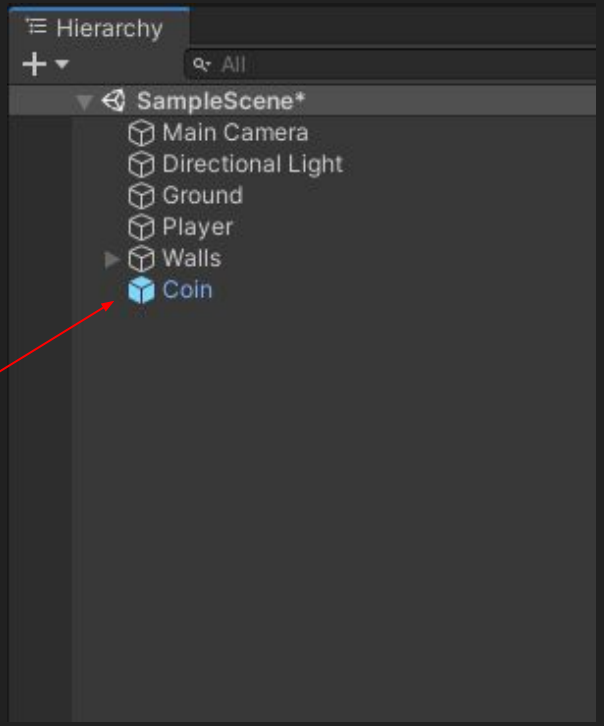
Creating “Coins”

- Create a Cube, set it’s position to (0, .5, 0), rotation to (45, 45, 45), and scale to (.5, .5, .5)
- Name it “Coin” and apply the Coin material to it
- Click and drag from the Hierarchy into the Prefabs folder
- We just created a Prefab



Creating “Coins”

- Create a Cube, set it’s position to (0, .5, 0), rotation to (45, 45, 45), and scale to (.5, .5, .5)
- Name it “Coin” and apply the Coin material to it
- Click and drag from the Hierarchy into the Prefabs folder
- We just created a Prefab
  - This icon denotes a Prefab



Creating a Rotator Script

- Currently, our “Coins” will just exist in the game

Creating a Rotator Script

- Currently, our “Coins” will just exist in the game
- This is undesirable

Creating a Rotator Script

- Currently, our “Coins” will just exist in the game
- This is undesirable
- Create Rotator.cs in the Scripts folder

Rotator.cs

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\Rotator.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

Rotator.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotator : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Rotator.cs

- We can remove the start method

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\Rotator.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

Rotator.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotator : MonoBehaviour
{
    // Update is called once per frame
    void Update ()
    {
        transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime);
    }
}
```

Rotator.cs

- We can remove the start method
- In the Update method we will rotate the gameobject this script is attached to

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\Rotator.cs - Sublime Text

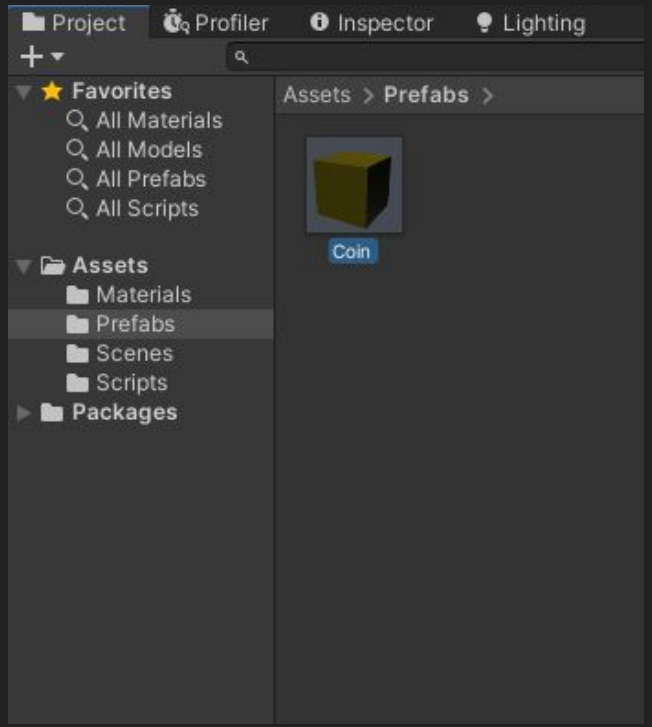
File Edit Selection Find View Goto Tools Project Preferences Help Friday

Rotator.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotator : MonoBehaviour
{
    // Update is called once per frame
    void Update ()
    {
        transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime);
    }
}
```

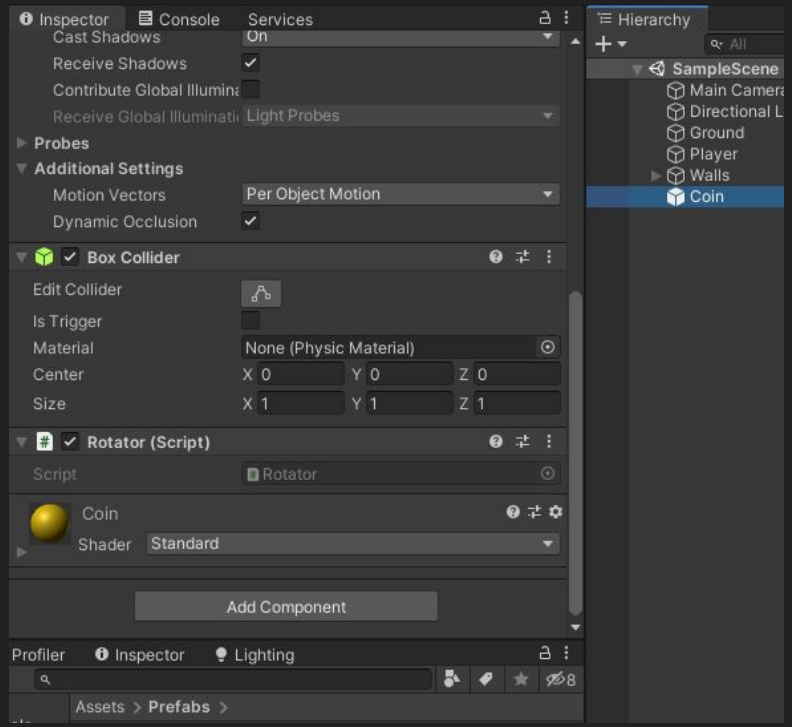
Adding the Rotator Component

- Find the Coin prefab in the Project Window



Adding the Rotator Component

- Find and select the Coin prefab in the Project Window
- Add the Rotator script from the Inspector Window
- Notice that the Coin currently in the scene now has the Rotator component too

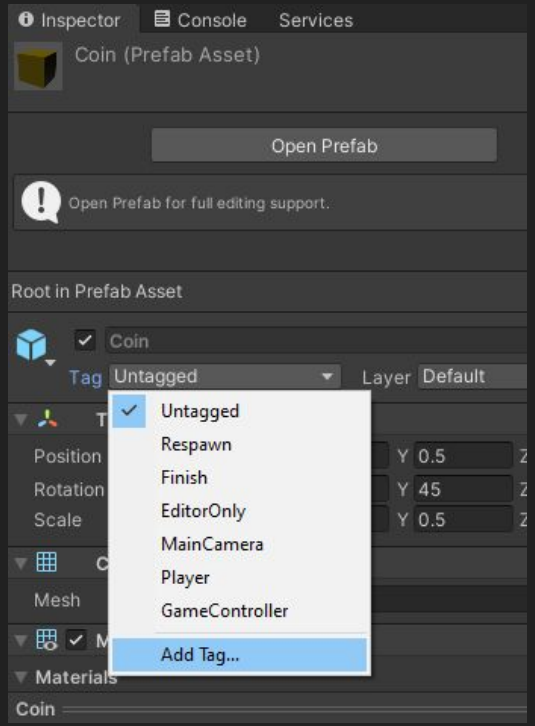


Tagging our Prefab

- We want to know when we have collided with a coin vs when we have collided with something else

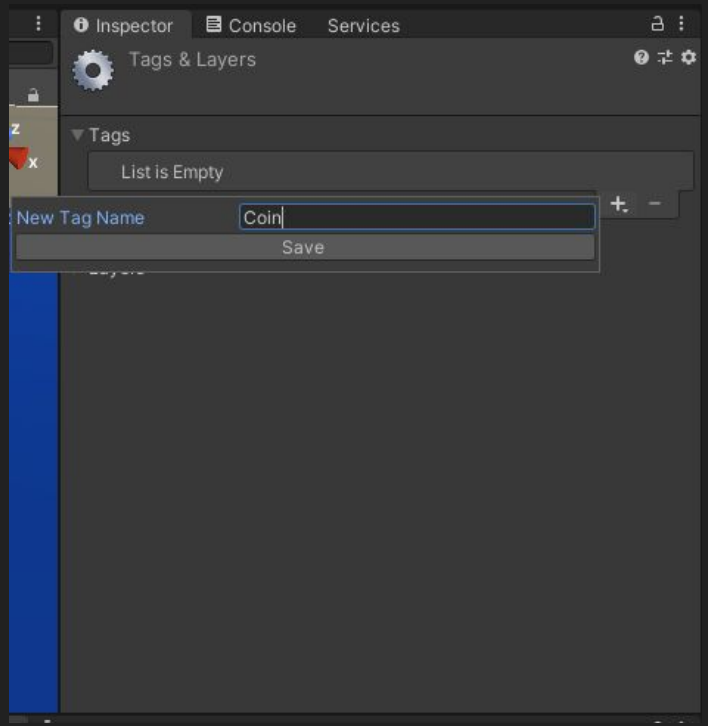
Tagging our Prefab

- We want to know when we have collided with a coin vs when we have collided with something else
- To do this we need to give our prefab a tag



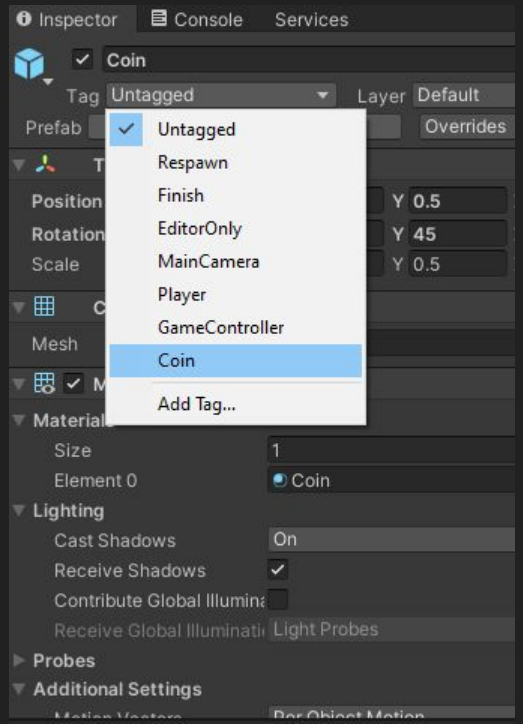
Tagging our Prefab

- We want to know when we have collided with a coin vs when we have collided with something else
- To do this we need to give our prefab a tag
- Create a tag called “Coin”



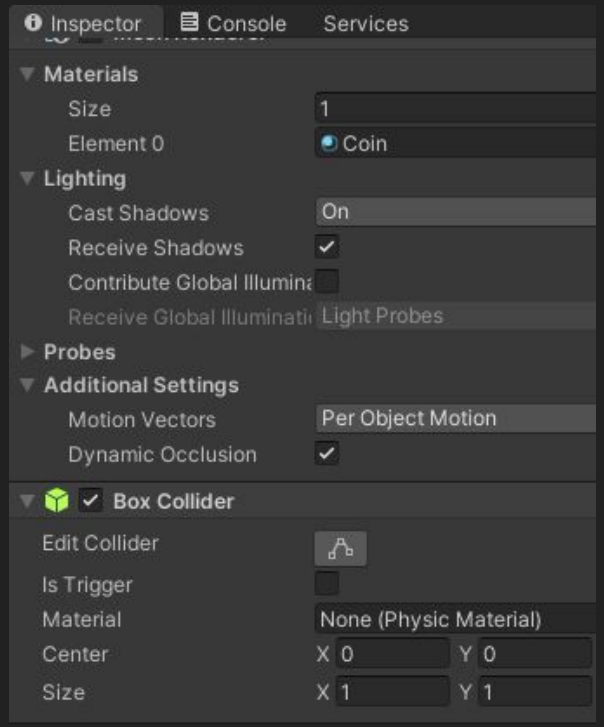
Tagging our Prefab

- We want to know when we have collided with a coin vs when we have collided with something else
- To do this we need to give our prefab a tag
- Create a tag called “Coin”
- Now we can actually give our prefab the tag



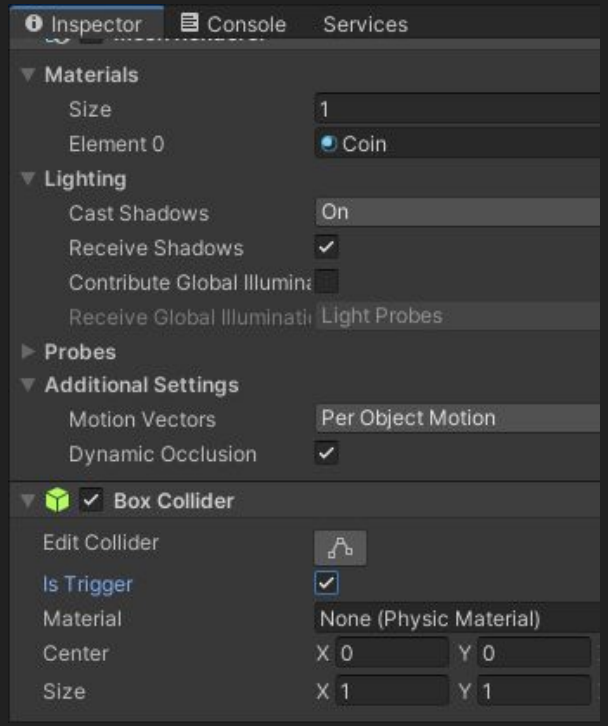
Modifying the Coin Collider

- Current our Coin prefab has a BoxCollider with “Is Trigger” set to false



Modifying the Coin Collider

- Current our Coin prefab has a BoxCollider with “Is Trigger” set to false
- Set it to True



Placing the Coins

- We need to place multiple coins to make the game fun

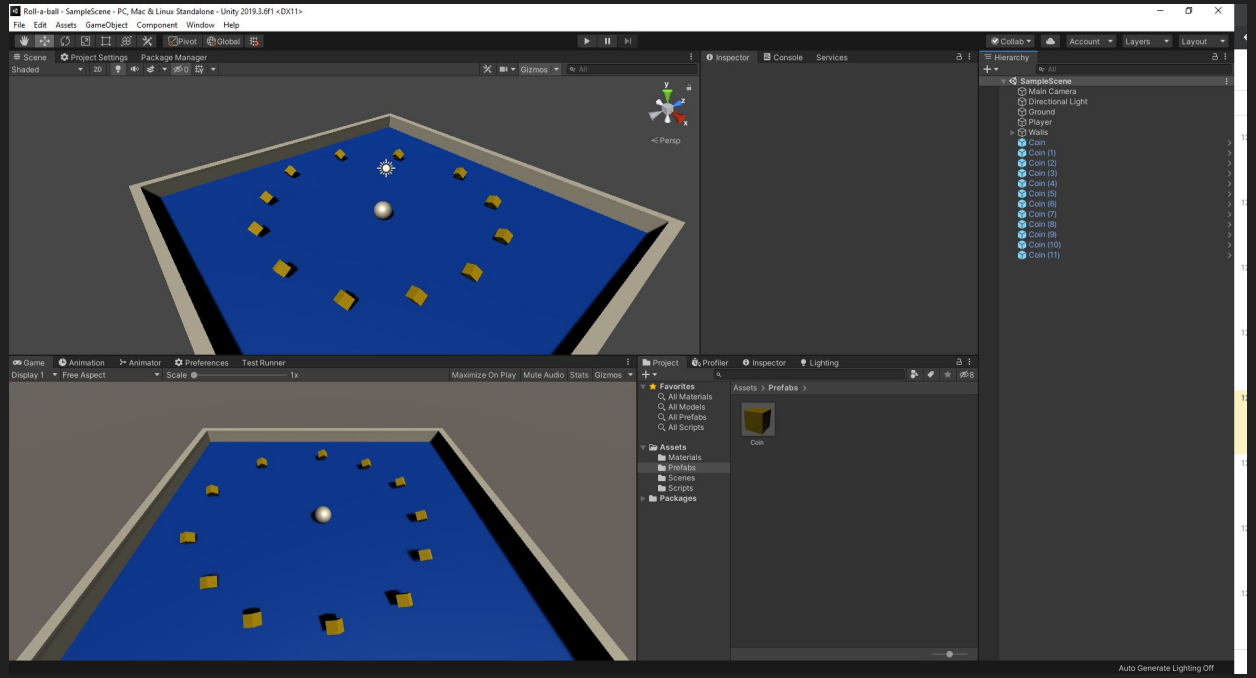
Placing the Coins

- We need to place multiple coins to make the game fun
- You can duplicate a gameobject by pressing Ctrl + d after selecting it from the Hierarchy window

Placing the Coins

- We need to place multiple coins to make the game fun
- You can duplicate a gameobject by pressing Ctrl + d after selecting it from the Hierarchy window
- Let’s make 12 coins and arrange them in a circle around the player

What we should have



Handling Collisions

- Open PlayerController.cs

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

Handling Collisions

- Open PlayerController.cs
- Add the OnTriggerEnter method

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag ("Coin"))
        {
            other.gameObject.SetActive (false);
        }
    }
}
```

Handling Collisions

- Open PlayerController.cs
- Add the OnTriggerEnter method
- We only care about collisions with objects tagged as coins

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag ("Coin"))
        {
            other.gameObject.SetActive (false);
        }
    }
}
```

Handling Collisions

- Open PlayerController.cs
- Add the OnTriggerEnter method
- We only care about collisions with objects tagged as coins
- By calling SetActive(false) on a gameobject we “hide” it from the scene

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag ("Coin"))
        {
            other.gameObject.SetActive (false);
        }
    }
}
```

Visualizing the Score

- Add references to UI Text elements that will display our current score and text for when we win

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    public Text countText;
    public Text winText;
    private int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
        winText.text = "";
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
```

Visualizing the Score

- Add references to UI Text elements that will display our current score and text for when we win
- To use them, we need to import UI info

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    public Text countText;
    public Text winText;
    private int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
        winText.text = "";
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
```

Visualizing the Score

- Add references to UI Text elements that will display our current score and text for when we win
- To use them, we need to import UI info
- Create a variable to keep track of the score

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    public Text countText;
    public Text winText;
    private int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
        winText.text = "";
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
```

Visualizing the Score

- Add references to UI Text elements that will display our current score and text for when we win
- To use them, we need to import UI info
- Create a variable to keep track of the score
- On start up we want our count to be zero and the text for when we win to be empty

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    public Text countText;
    public Text winText;
    private int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
        winText.text = "";
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
```

Visualizing the Score

- Add references to UI Text elements that will display our current score and text for when we win
- To use them, we need to import UI info
- Create a variable to keep track of the score
- On start up we want our count to be zero and the text for when we win to be empty
- We also want the text that displays our score to be updated

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Friday

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    public Text countText;
    public Text winText;
    private int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
        winText.text = "";
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
```

Visualizing the Score (Cont.)

- When we collide with a coin, we want to increment our score

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Fric

PlayerController.cs

```
        Vector3 movement = new Vector3 (moveHorizontal, 0.(

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag ("Coin"))
        {
            other.gameObject.SetActive (false);
            count = count + 1;
            SetCountText ();
        }
    }

    void SetCountText ()
    {
        countText.text = "Count: " + count.ToString ();
        if (count >= 12)
        {
            winText.text = "You Win!";
        }
    }
}
```

Visualizing the Score (Cont.)

- When we collide with a coin, we want to increment our score
- We also need to update the text displaying it

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Fric

PlayerController.cs

```
        Vector3 movement = new Vector3 (moveHorizontal, 0.(

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag ("Coin"))
        {
            other.gameObject.SetActive (false);
            count = count + 1;
            SetCountText ();
        }
    }

    void SetCountText ()
    {
        countText.text = "Count: " + count.ToString ();
        if (count >= 12)
        {
            winText.text = "You Win!";
        }
    }
}
```

Visualizing the Score (Cont.)

- When we collide with a coin, we want to increment our score
- We also need to update the text displaying it
- The method that handles updating the UI Text

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Fric

PlayerController.cs

```
28
29          Vector3 movement = new Vector3 (moveHorizontal, 0.(
30
31          rb.AddForce (movement * speed);
32      }
33
34      void OnTriggerEnter(Collider other)
35      {
36          if (other.gameObject.CompareTag ("Coin"))
37          {
38              other.gameObject.SetActive (false);
39              count = count + 1;
40              SetCountText ();
41          }
42      }
43
44      void SetCountText ()
45      {
46          countText.text = "Count: " + count.ToString ();
47          if (count >= 12)
48          {
49              winText.text = "You Win!";
50          }
51      }
52  }
```

Visualizing the Score (Cont.)

- When we collide with a coin, we want to increment our score
- We also need to update the text displaying it
- The method that handles updating the UI Text
- Display our current score

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Fric

PlayerController.cs

```
28
29          Vector3 movement = new Vector3 (moveHorizontal, 0.(
30
31          rb.AddForce (movement * speed);
32      }
33
34      void OnTriggerEnter(Collider other)
35      {
36          if (other.gameObject.CompareTag ("Coin"))
37          {
38              other.gameObject.SetActive (false);
39              count = count + 1;
40              SetCountText ();
41          }
42      }
43
44      void SetCountText ()
45      {
46          countText.text = "Count: " + count.ToString ();
47          if (count >= 12)
48          {
49              winText.text = "You Win!";
50          }
51      }
52  }
```

Visualizing the Score (Cont.)

- When we collide with a coin, we want to increment our score
- We also need to update the text displaying it
- The method that handles updating the UI Text
- Display our current score
- If we have collected enough coins, display that we have won the game

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Fric

PlayerController.cs

```
28
29          Vector3 movement = new Vector3 (moveHorizontal, 0.(
30
31          rb.AddForce (movement * speed);
32      }
33
34      void OnTriggerEnter(Collider other)
35      {
36          if (other.gameObject.CompareTag ("Coin"))
37          {
38              other.gameObject.SetActive (false);
39              count = count + 1;
40              SetCountText ();
41          }
42      }
43
44      void SetCountText ()
45      {
46          countText.text = "Count: " + count.ToString ();
47          if (count >= 12)
48          {
49              winText.text = "You Win!";
50          }
51      }
52  }
```

Visualizing the Score (Cont.)

- When we collide with a coin, we want to increment our score
- We also need to update the text displaying it
- The method that handles updating the UI Text
- Display our current score
- If we have collected enough coins, display that we have won the game
  - The amount can be whatever you want

D:\Documents\GitLab\Roll-a-ball\Assets\Scripts\PlayerController.cs - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help Fric

PlayerController.cs

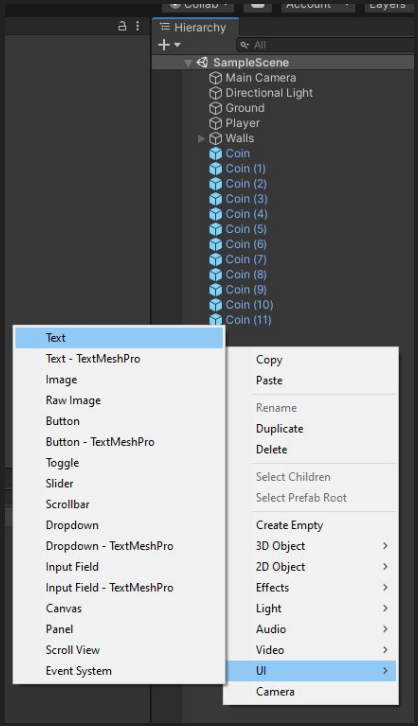
```
        Vector3 movement = new Vector3 (moveHorizontal, 0.(

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag ("Coin"))
        {
            other.gameObject.SetActive (false);
            count = count + 1;
            SetCountText ();
        }
    }

    void SetCountText ()
    {
        countText.text = "Count: " + count.ToString ();
        if (count >= 12)
        {
            winText.text = "You Win!";
        }
    }
}
```

Creating UI Elements

- We need to create 2 UI Text elements

Creating UI Elements

- We need to create 2 UI Text elements
- Create one to start



Creating UI Elements

- We need to create 2 UI Text elements
- Create one to start
- Notice that we just added 3 total objects to our scene

Creating UI Elements

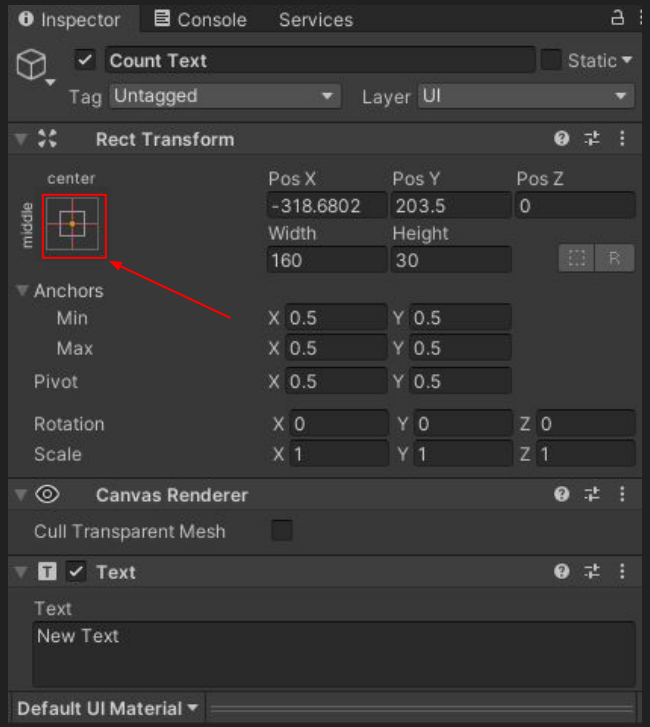
- We need to create 2 UI Text elements
- Create one to start
- Notice that we just added 3 total objects to our scene
- We can ignore both Canvas and EventSystem for now

Creating UI Elements

- We need to create 2 UI Text elements
- Create one to start
- Notice that we just added 3 total objects to our scene
- We can ignore both Canvas and EventSystem for now
- Rename the Text object to “Count Text”

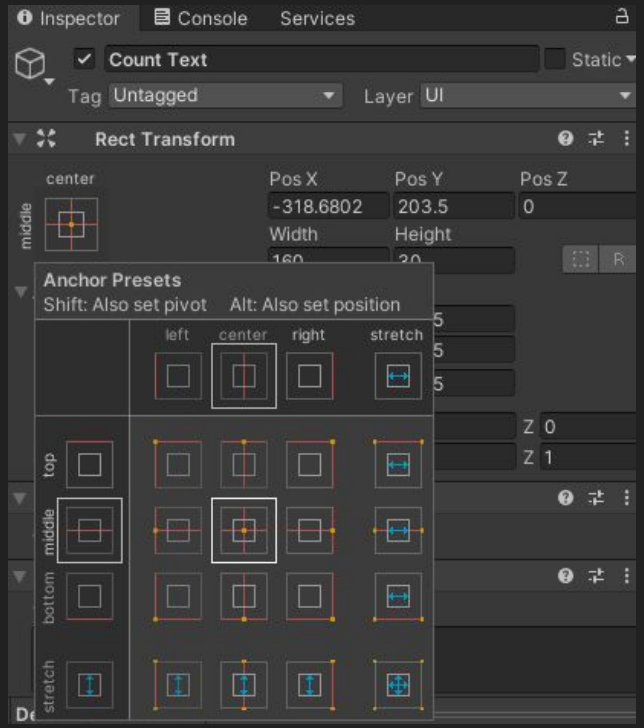
Setting the Transform

- First we need to change the anchor
- Click this



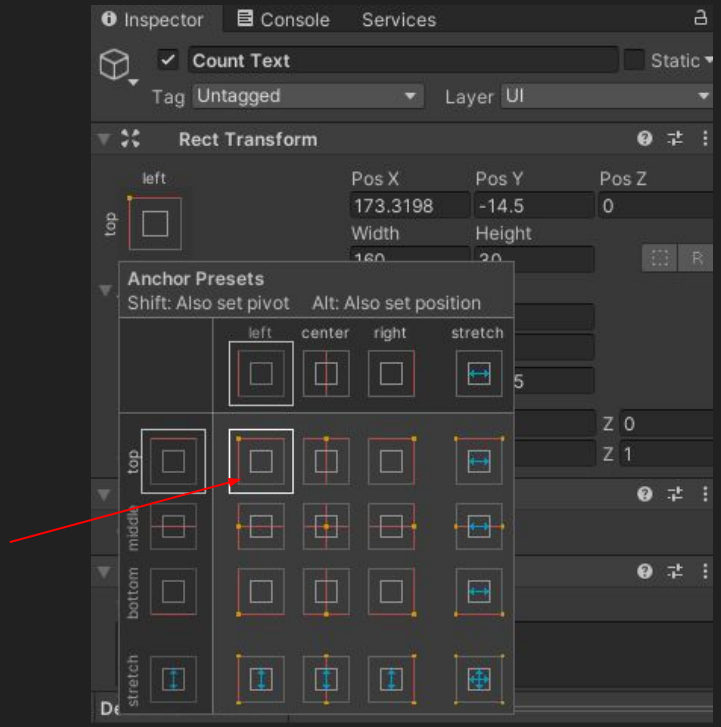
Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this



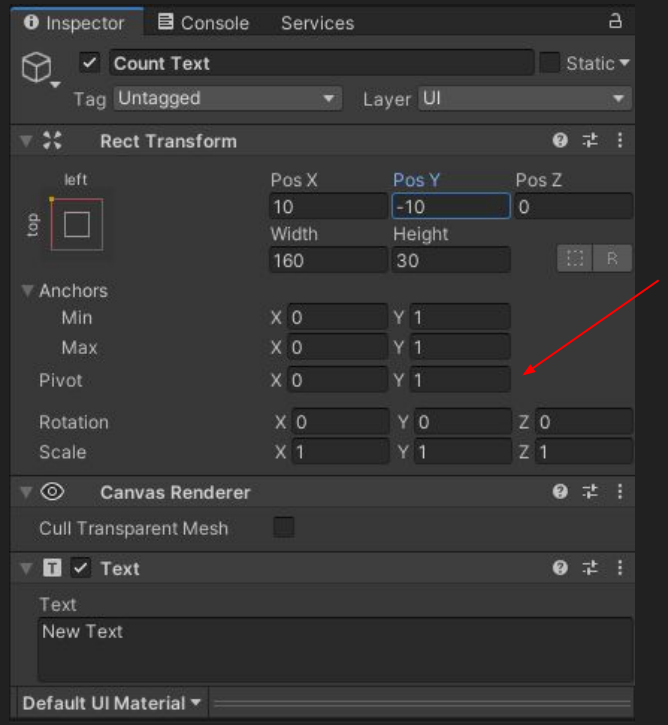
Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this
- Click this



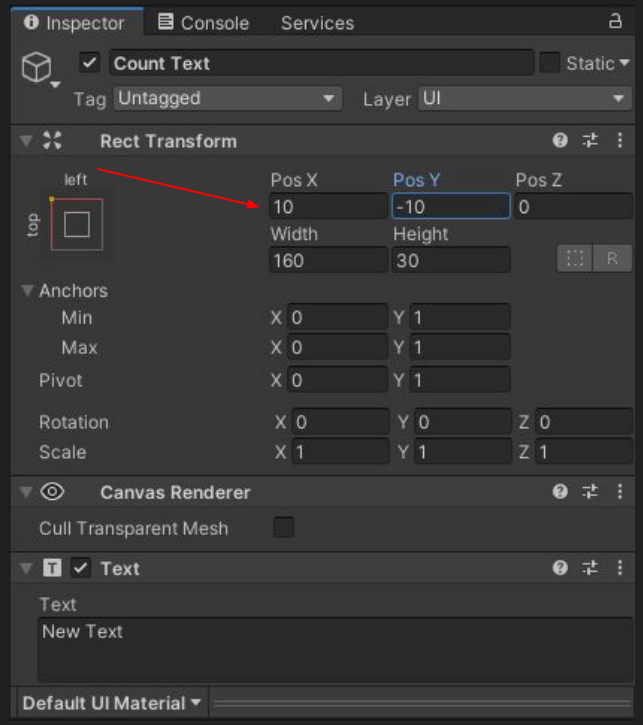
Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this
- Click this
- Change your pivot to (0, 1)



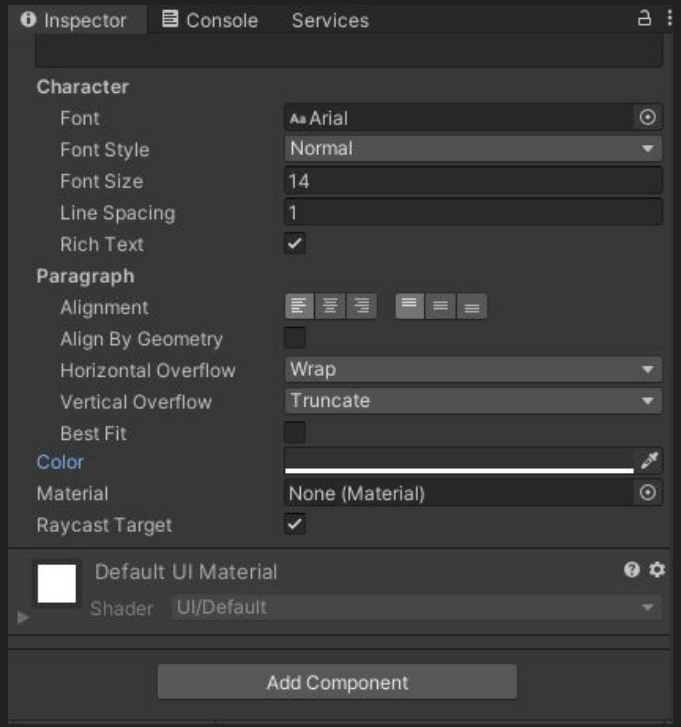
Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this
- Click this
- Set the pivot to (0, 1)
- Then set the position to (10, -10, 0)



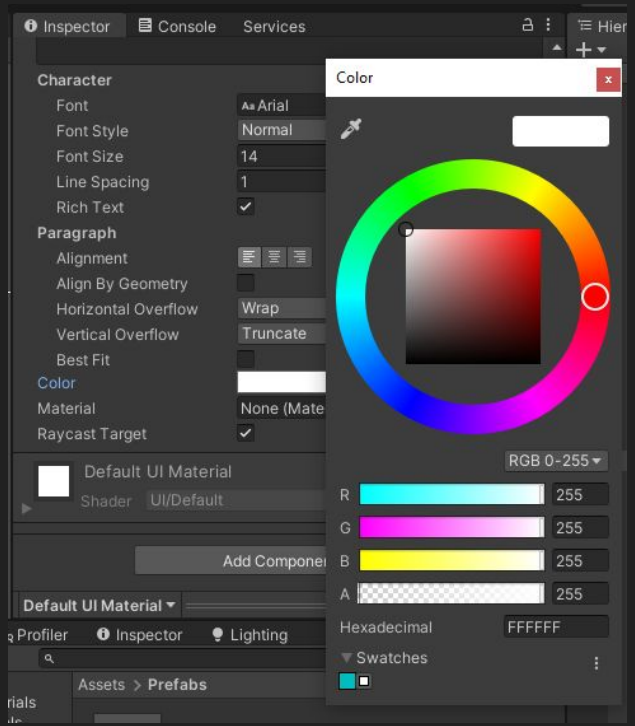
Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this
- Click this
- Set the pivot to (0, 1)
- Then set the position to (10, -10, 0)
- Scroll down in the inspector window



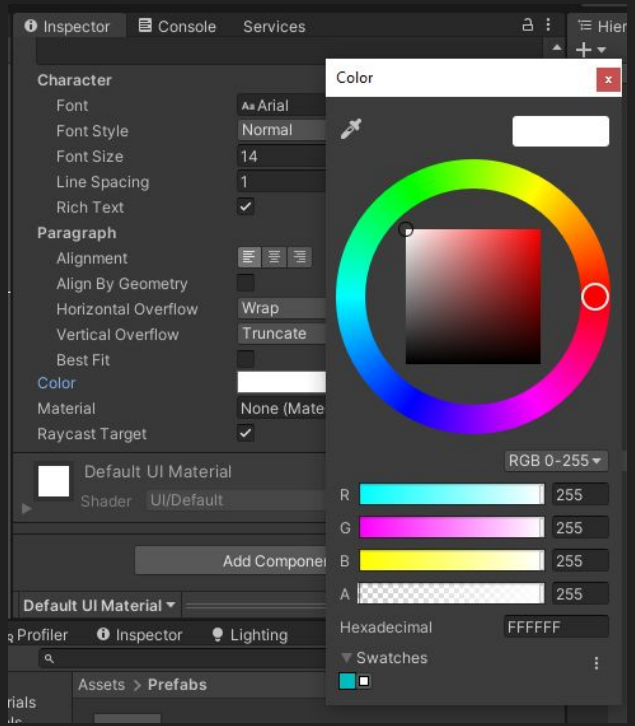
Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this
- Click this
- Set the pivot to (0, 1)
- Then set the position to (10, -10, 0)
- Scroll down in the inspector window
- Click the color box to change the text color

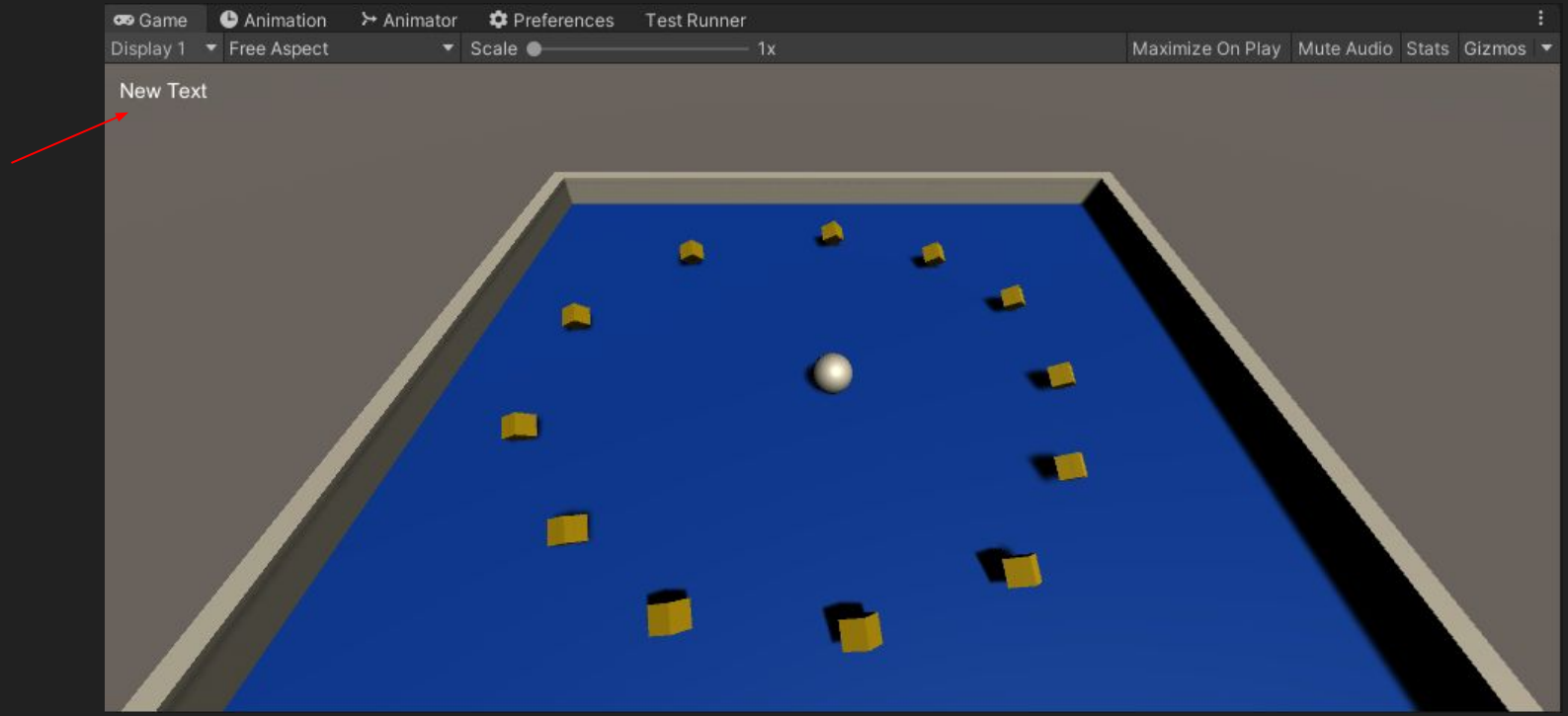


Setting the Transform

- First we need to change the anchor
- Click this
- You'll see this
- Click this
- Set the pivot to (0, 1)
- Then set the position to (10, -10, 0)
- Scroll down in the inspector window
- Click the color box to change the text color
- Let's go with white



What the Game Window Should Display

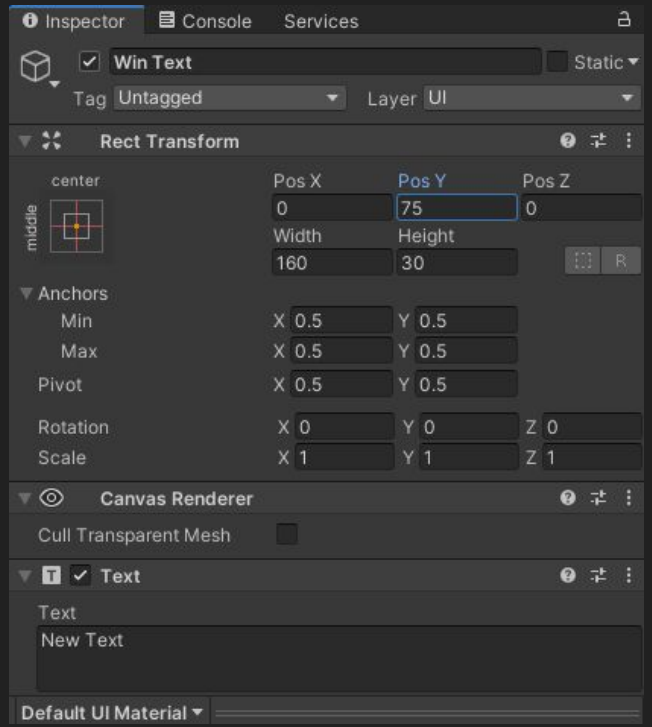


Create Another UI Text

- Create a second UI text element called “Win Text” as a child of the Canvas

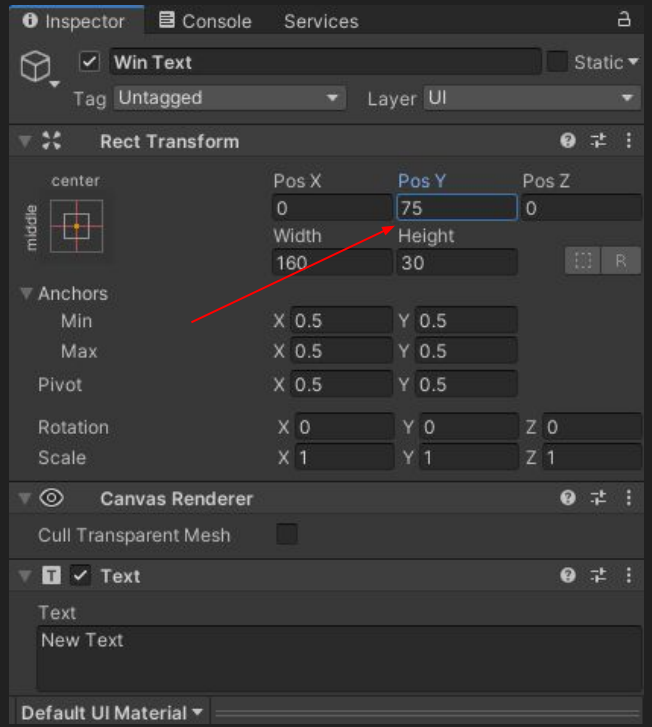
Create Another UI Text

- Create a second UI text element called “Win Text” as a child of the Canvas
- We’ll want this element to have a middle center anchor



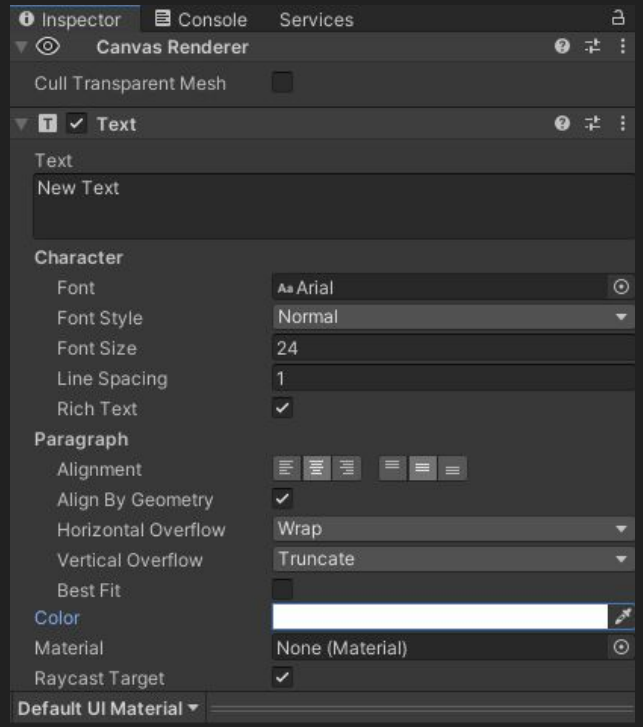
Create Another UI Text

- Create a second UI text element called “Win Text” as a child of the Canvas
- We’ll want this element to have a middle center anchor
- But the Y position should be 75



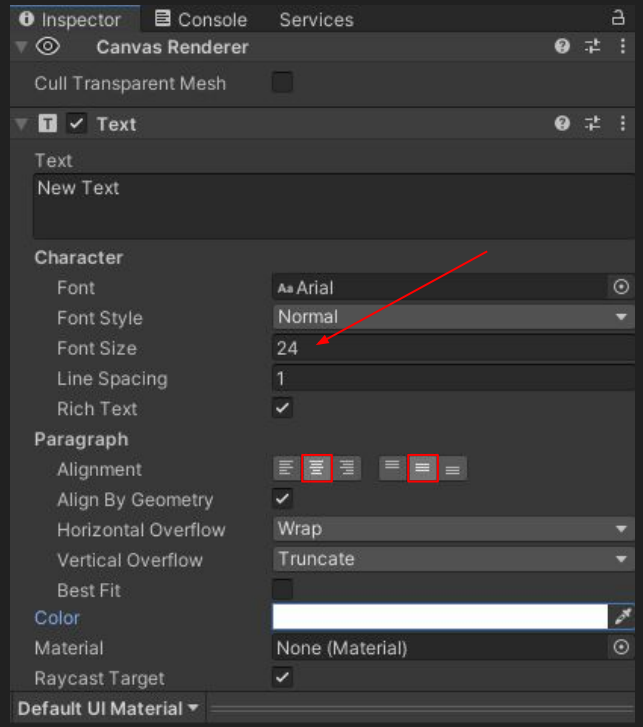
Create Another UI Text

- Create a second UI text element called “Win Text” as a child of the Canvas
- We’ll want this element to have a middle center anchor
- But the Y position should be 75
- We’ll also want this text to be white

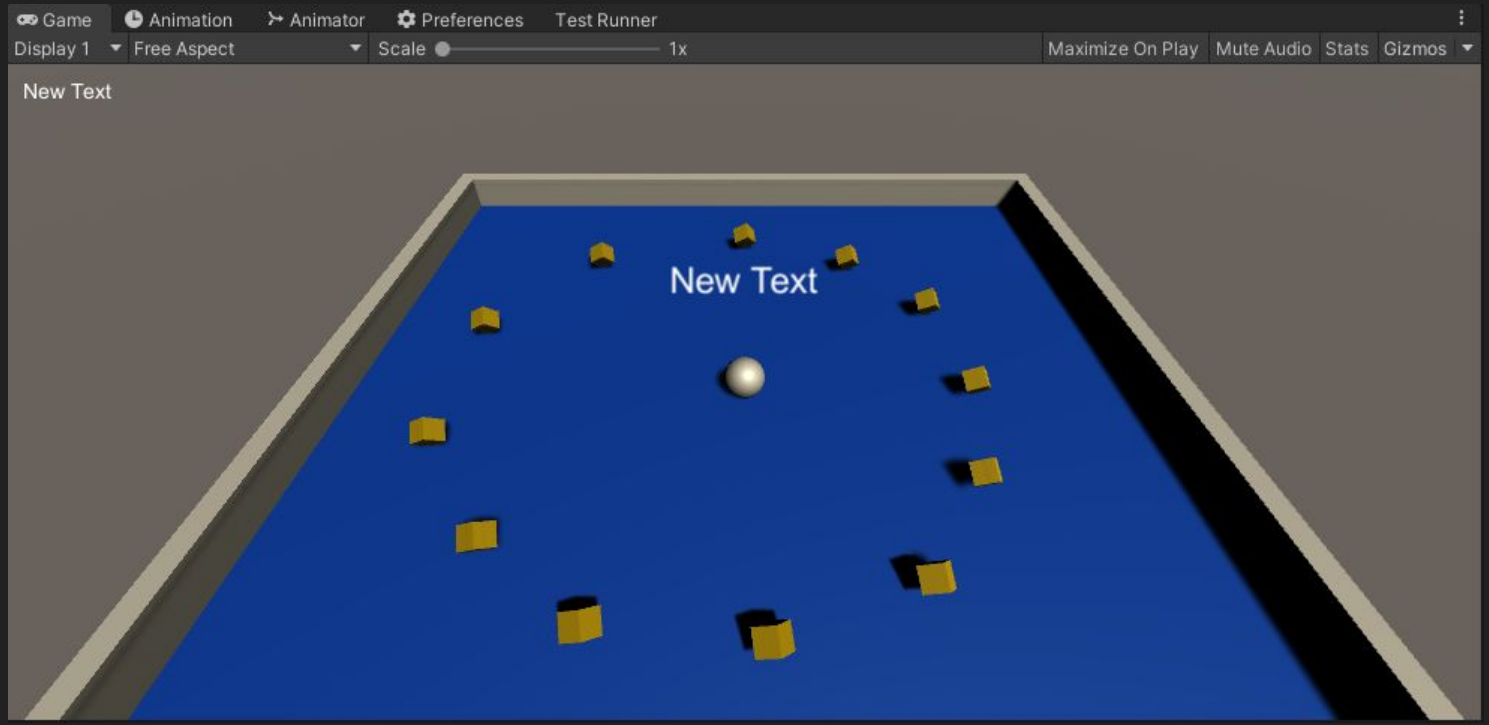


Create Another UI Text

- Create a second UI text element called “Win Text” as a child of the Canvas
- We’ll want this element to have a middle center anchor
- But the Y position should be 75
- We’ll also want this text to be white
- But we need to increase the size and adjust the alignments



What the Game Window Should Display

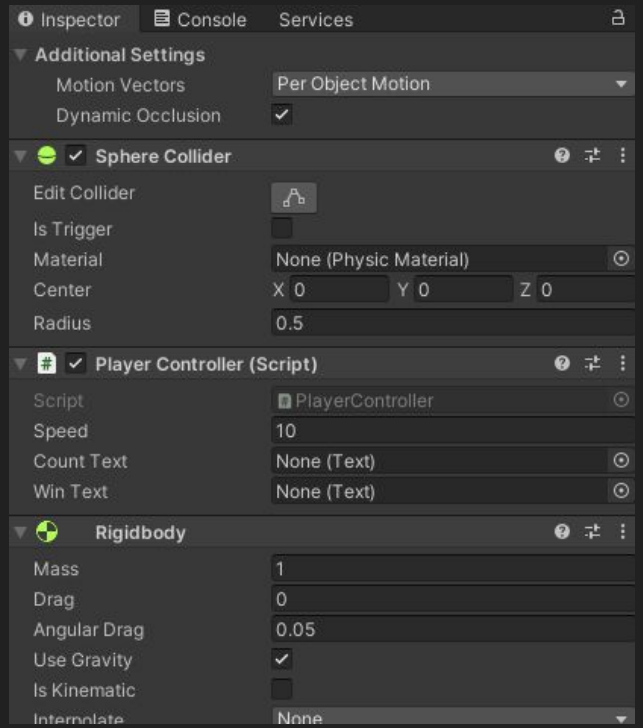


Linking to PlayerController

- Almost done!

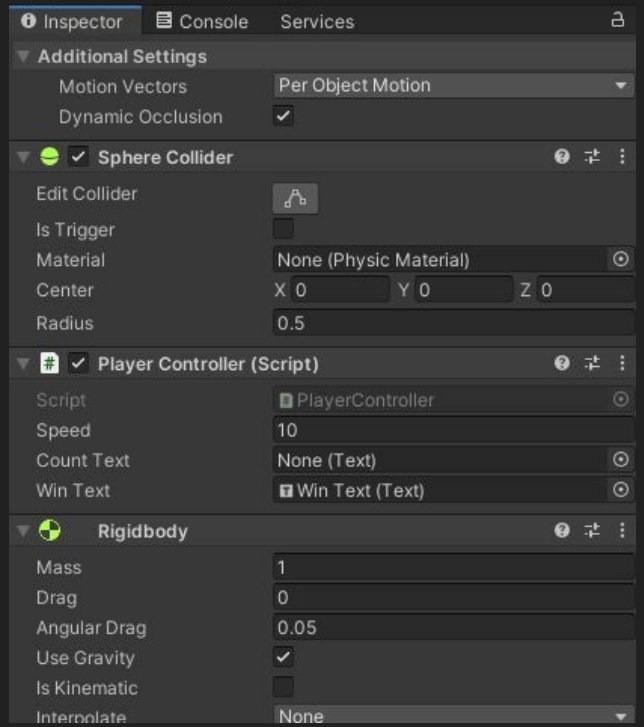
Linking to PlayerController

- Almost done!
- Select the Player object and find the PlayerController component in the Inspector tab



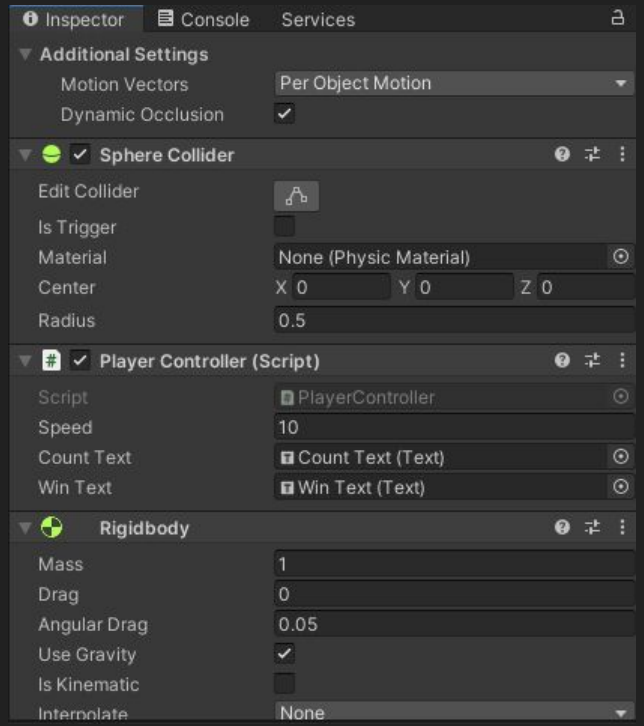
Linking to PlayerController

- Almost done!
- Select the Player object and find the PlayerController component in the Inspector tab
- Click and drag the Win Text object onto the Win Text field

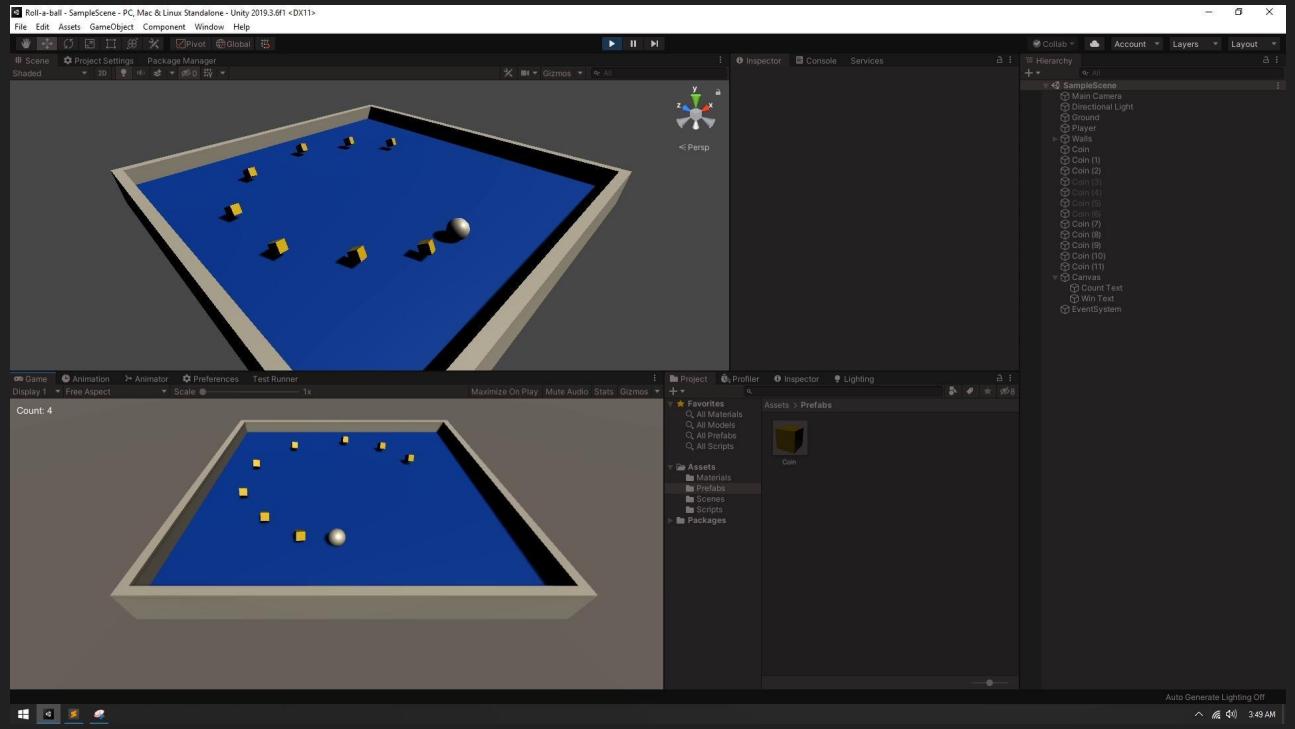


Linking to PlayerController

- Almost done!
- Select the Player object and find the PlayerController component in the Inspector tab
- Click and drag the Win Text object onto the Win Text field
- Click and drag the Count Text object onto the Count Text field

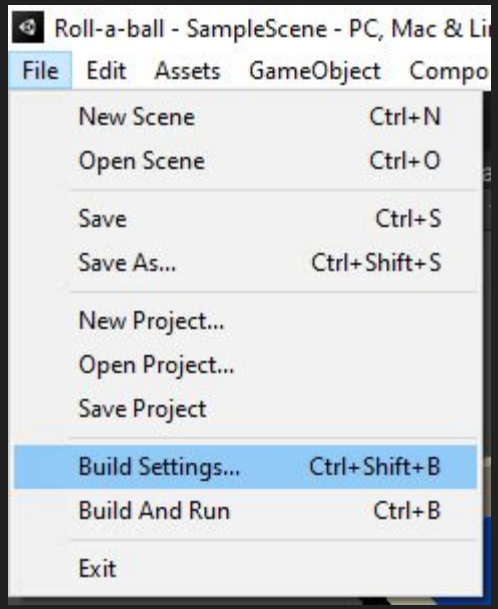


What We Have So Far



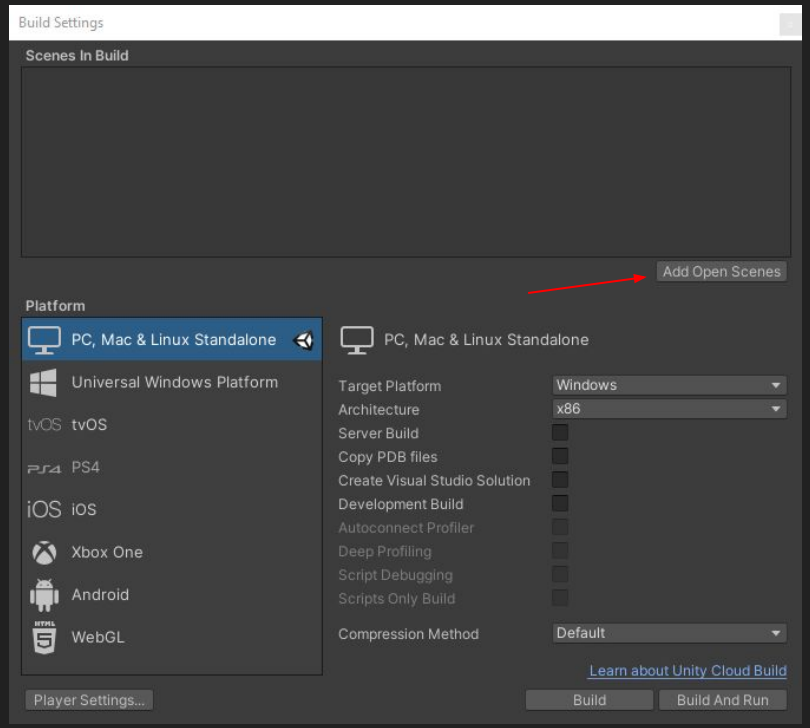
Last Step: Build!

- Click File → Build Settings



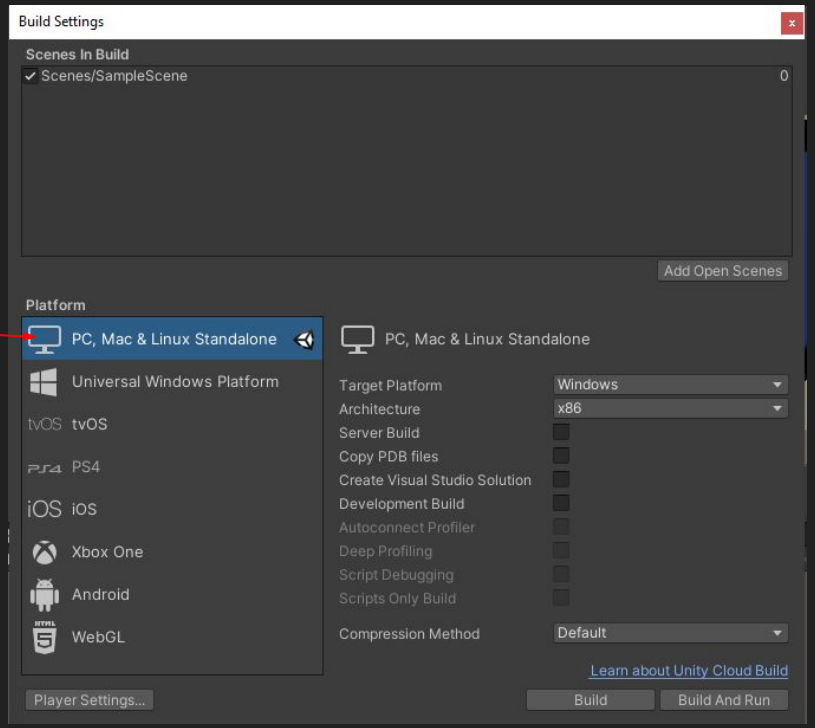
Last Step: Build!

- Click File → Build Settings
- Click “Add Open Scenes”



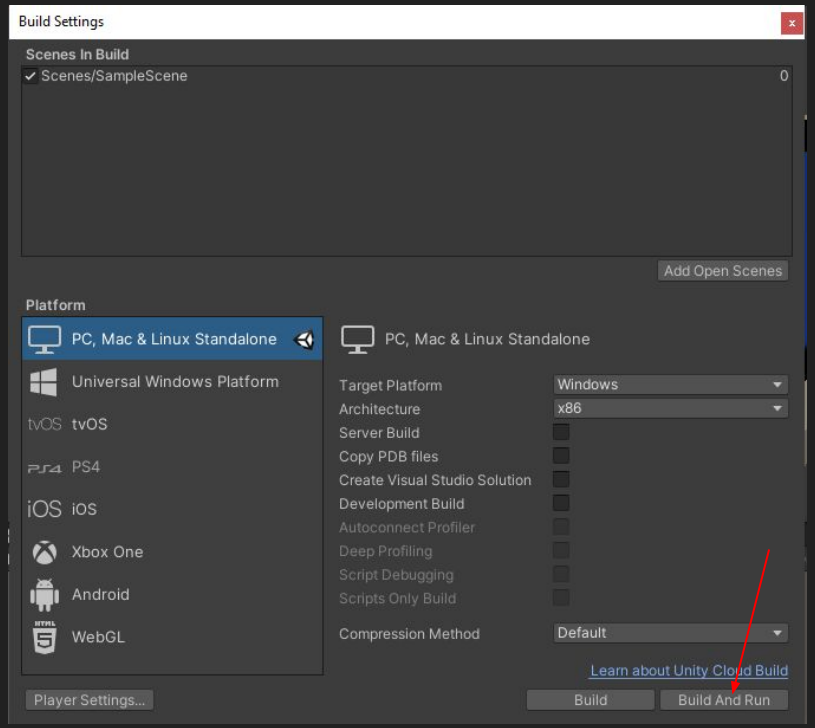
Last Step: Build!

- Click File → Build Settings
- Click “Add Open Scenes”
- Make sure you have the correct build platform



Last Step: Build!

- Click File → Build Settings
- Click “Add Open Scenes”
- Make sure you have the correct build platform
- Click “Build And Run”



Last Step: Build!

- Click File → Build Settings
- Click “Add Open Scenes”
- Make sure you have the correct build platform
- Click “Build And Run”
- Select the folder you want the game to be built in

Done