



introduction to unity – roll-a-ball

(Lab) Wen-Jie Tseng 26. 11. 2020

Labs

- 
- 19.11 Website setup (hugo)
 - 26.11 Introduction to Unity (roll-a-ball)
 - 03.12 VR in Unity + selection
 - 10.12 locomotion
 - 17.12 cybersickness reduction techniques
 - (holidays)
 - 2021 TBA

today's topics

- I. Basics of Unity
- II. Implement the roll-a-ball game

I. Basics of Unity

- Online resources
- A simple guide to the Unity IDE

Online resources

- There are many materials ranging from the beginner to advance on
 - Unity projects (<https://learn.unity.com/projects>)
 - Unity tutorial (<https://learn.unity.com/tutorials>)

FEATURED PROJECTS

Introduction to Terrain Editor

Project • Beginner • 3 Hours 40 Mins • 346

[Start now](#)[Save for Later](#)

Created By Unity Technologies



Projects

Projects

Search

Level ▾

Topics ▾

Industry ▾

Duration ▾

English ✖

Most Recent ▾



LEARN LIVE ®

Beginning Audio in Unity

Project • Beginner • 5 Hours



Creating a Puzzle Platformer

Project • Intermediate • 5 Hours 30 Mins



LEGO® Microgame

Project • Beginner • 45 Mins

FEATURED TUTORIALS

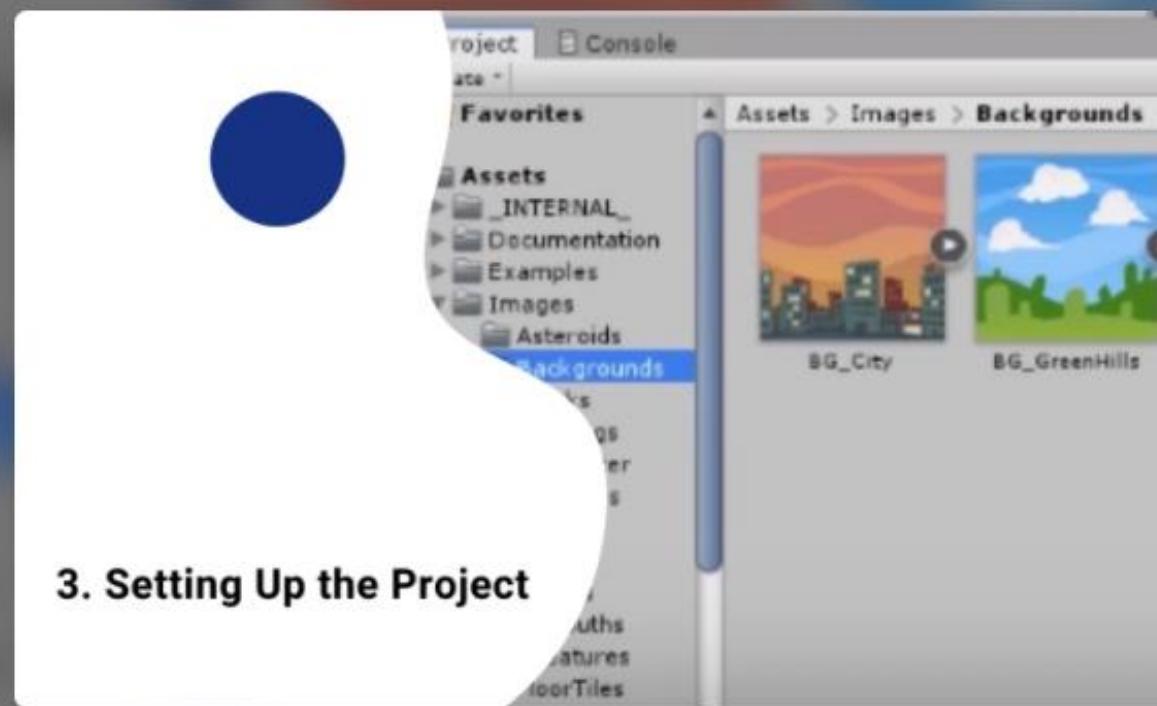
Setting Up the Project

Tutorial • Beginner • 40 Mins • 120

[Start now](#)[Save for Later](#)

Created By Unity Technologies

3. Setting Up the Project



Tutorials

Tutorials

Search

Level ▾

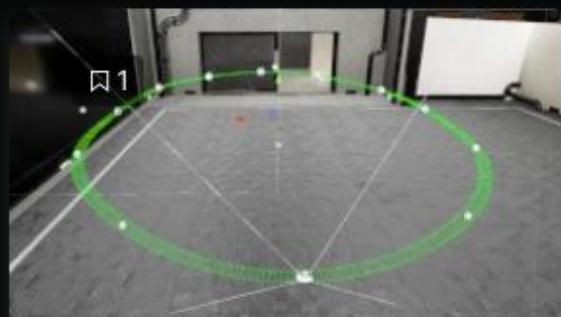
Topics ▾

Industry ▾

Duration ▾

English x

Most Recent ▾



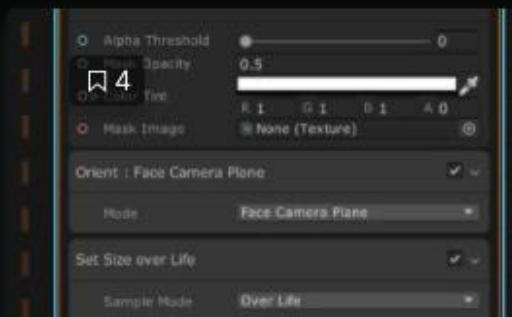
Cinemachine Dolly
Cart and Track - 2018

Tutorial • Beginner • 30 Mins



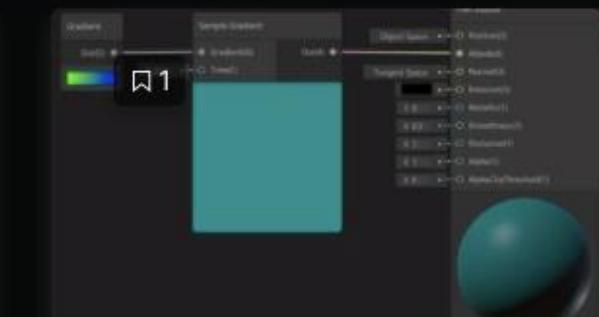
Simulating Weather
with the VFX Graph

Tutorial • Intermediate • 20 Mins



Integrate Shader
Graph into Visual

Tutorial • Intermediate • 20 Mins



Challenge: Using the
Gradient Node to

Tutorial • Intermediate • 10 Mins

Unity Technologies

Part of: [Working with Cinemachine Cams](#)

Unity Technologies

Part of: [Making Snow with VFX Graph](#)

Unity Technologies

Part of: [Creating fireworks with VFX Graph](#)

Unity Technologies

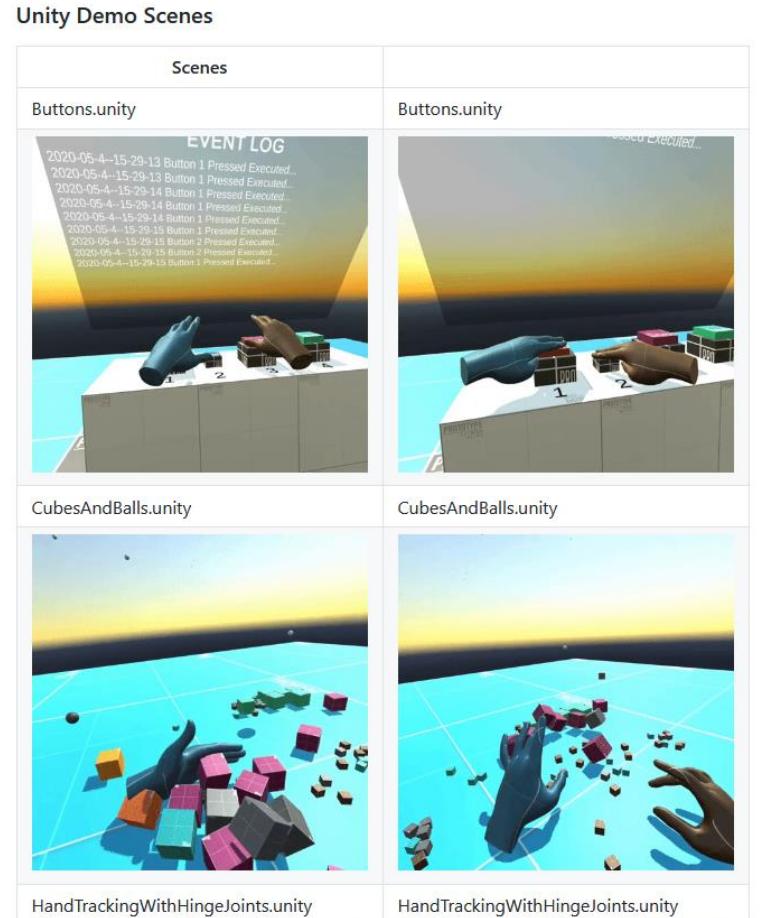
Part of: [Make a Flag Wave with Shaders](#)

How can I learn more?

- There are many materials ranging from the beginner to advance on
 - Unity projects (<https://learn.unity.com/projects>)
 - Unity tutorial (<https://learn.unity.com/tutorials>)
 - **GitHub, YouTube**, etc.

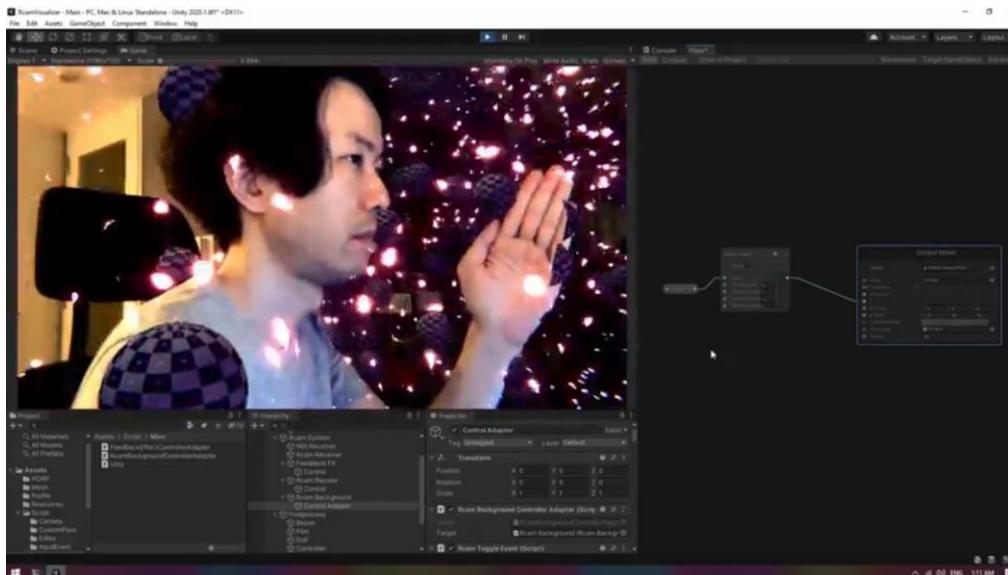
Projects built upon Unity

- Oculus Quest Hand Tracking Physic Examples



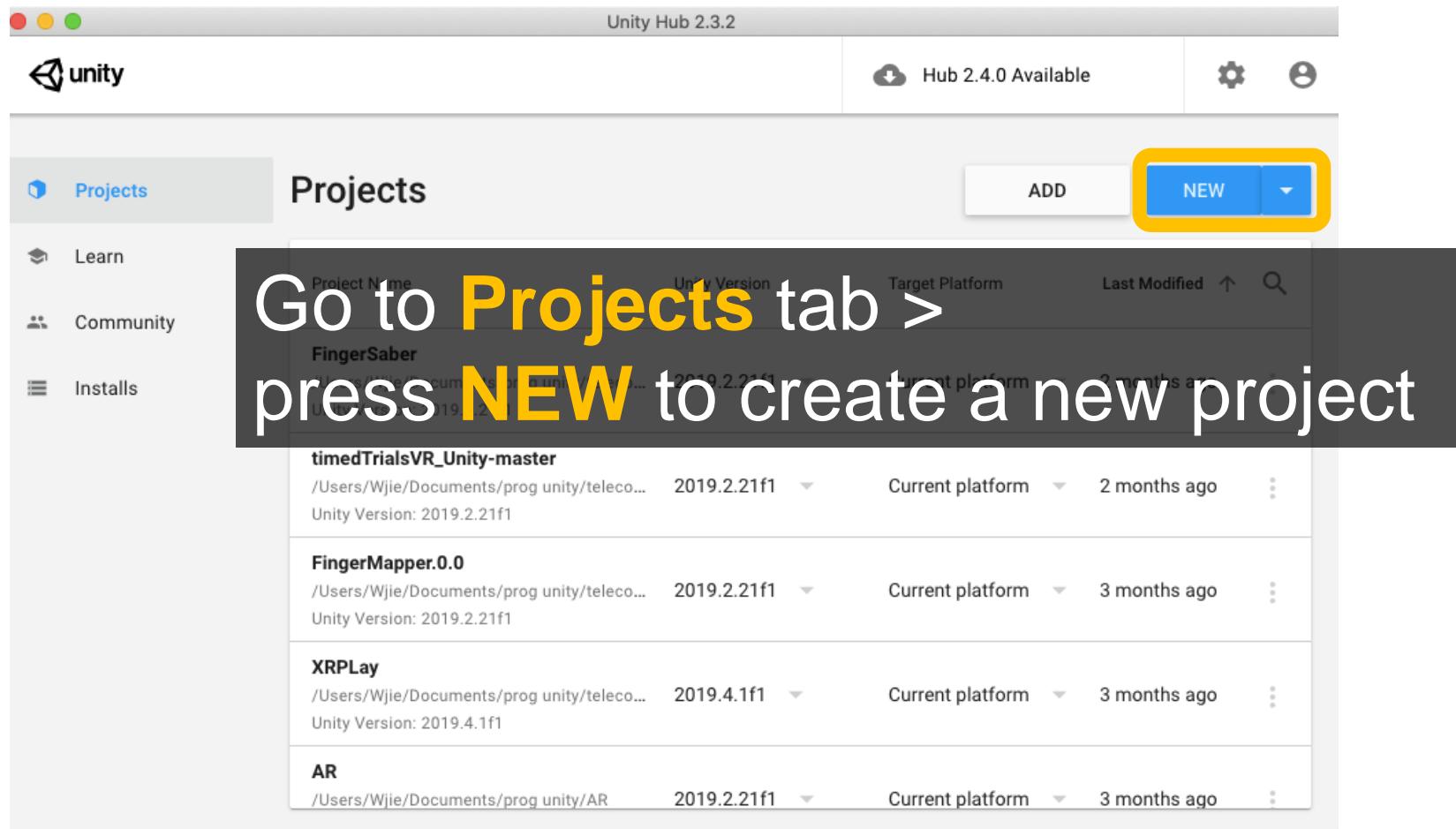
Projects built upon Unity

- Video jockey (VJ) project
 - [Link1\(github\)](#) [Link2\(blog, VJ-ing with Unity\)](#)
- AR
 - [Keijiro](#) (a unity developer does cool stuff)



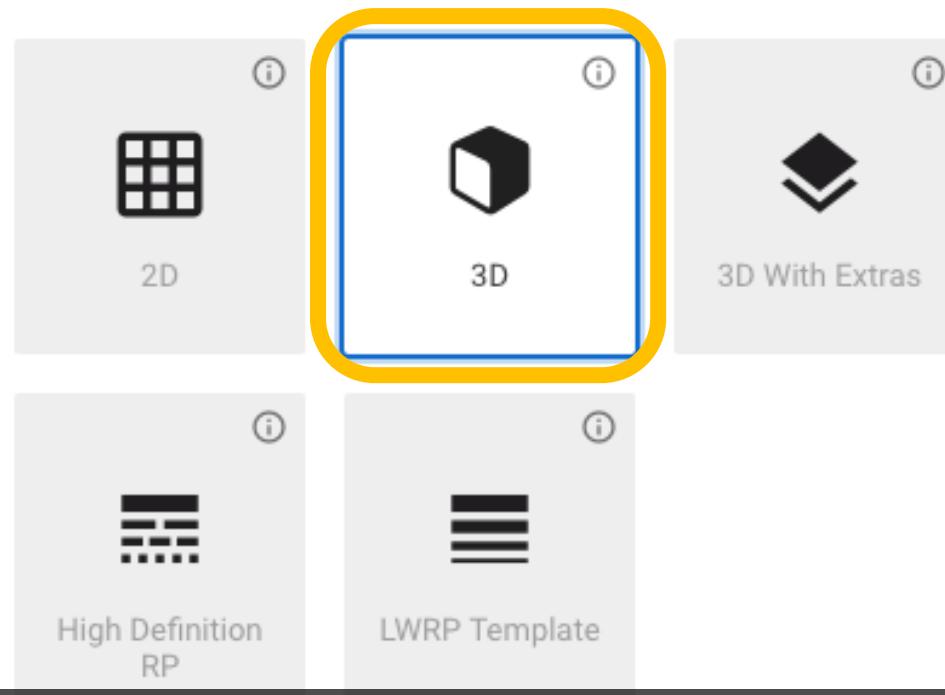
The Unity IDE

- Create a new 3D project from UnityHub.





Templates



Settings

Project Name *

TSP-VAR-HCI-Tutorial

Location *

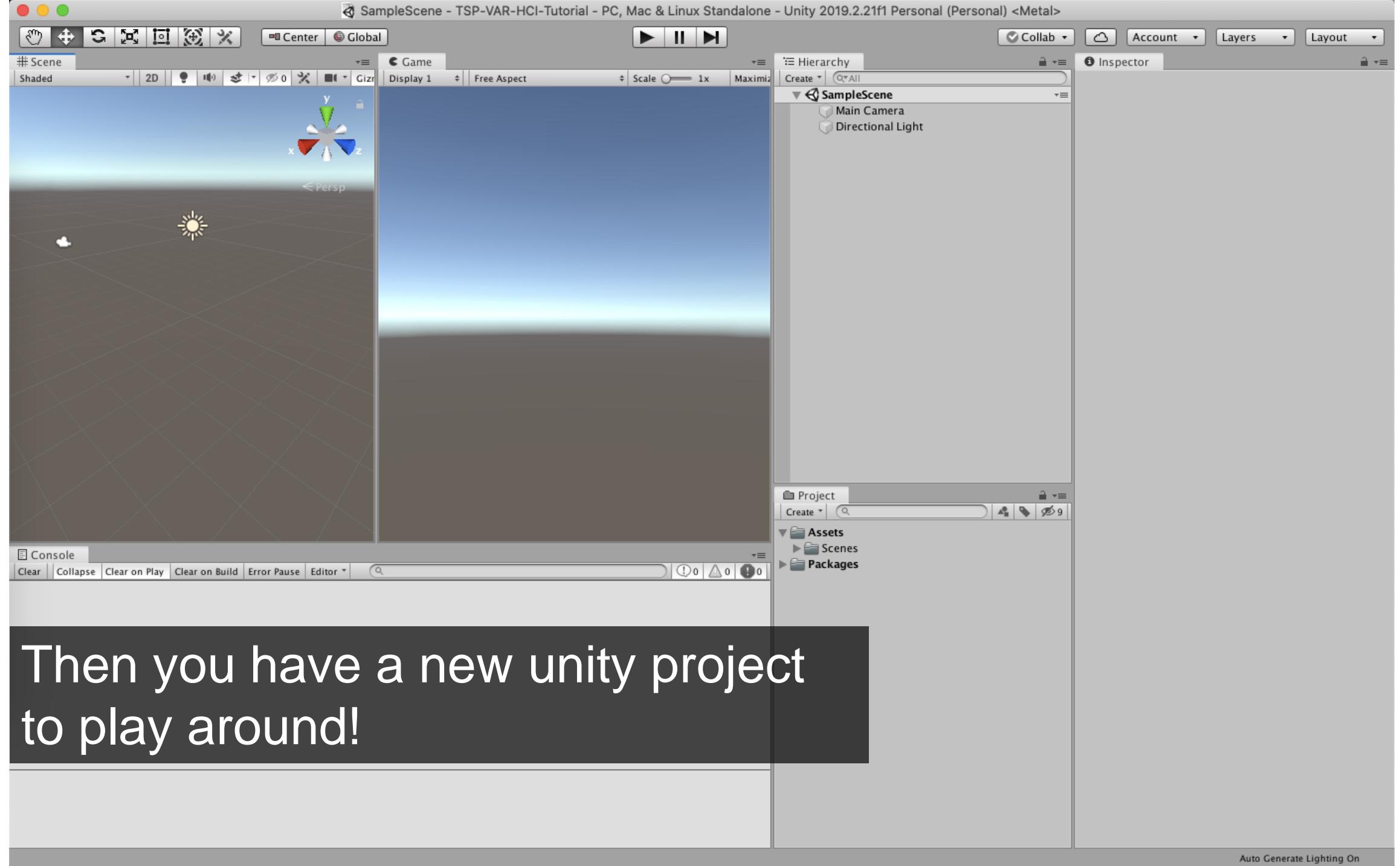
/Users/Wjie/Desktop

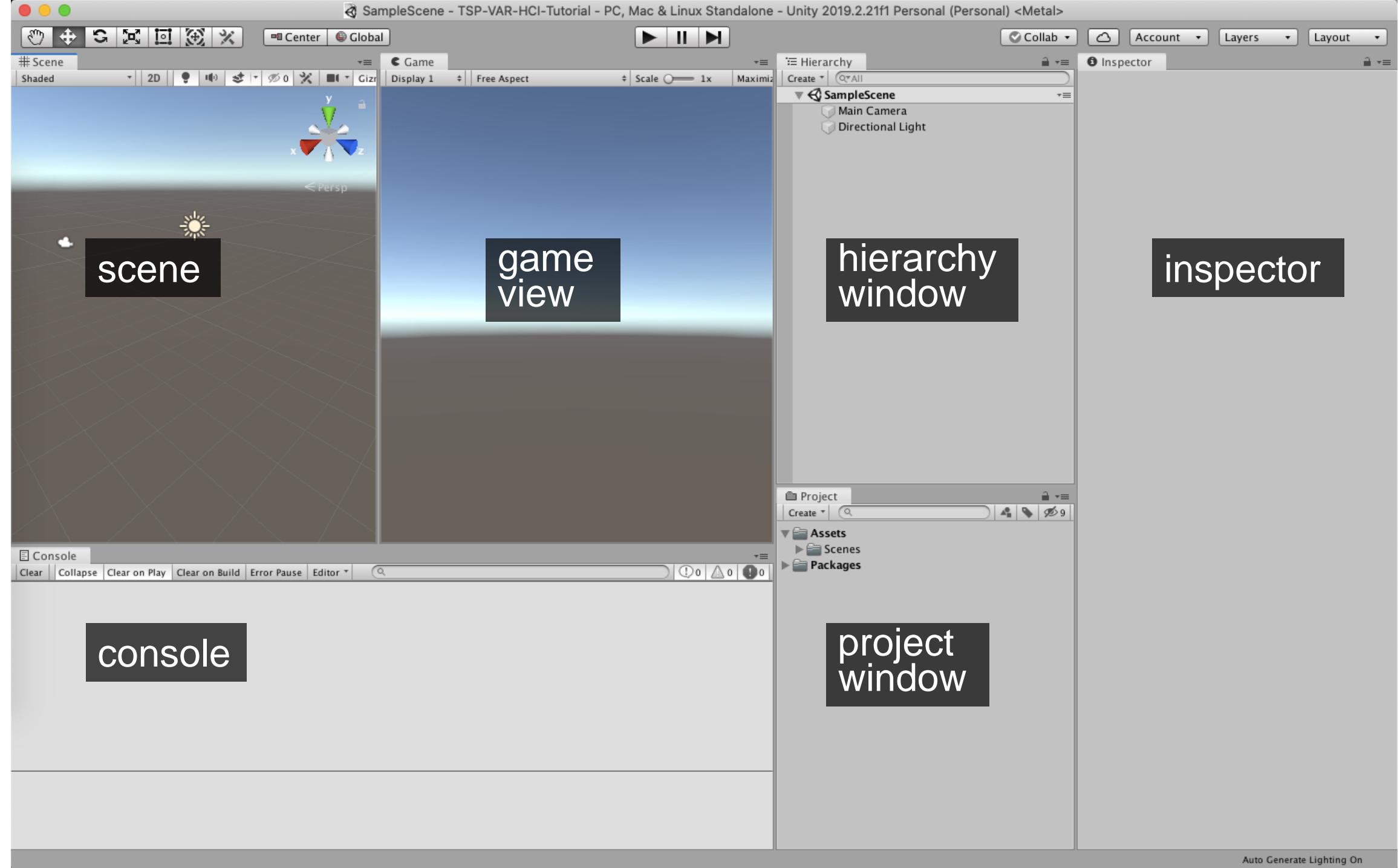
...

Select **3D** >
Enter project names and location >
press **CREATE**

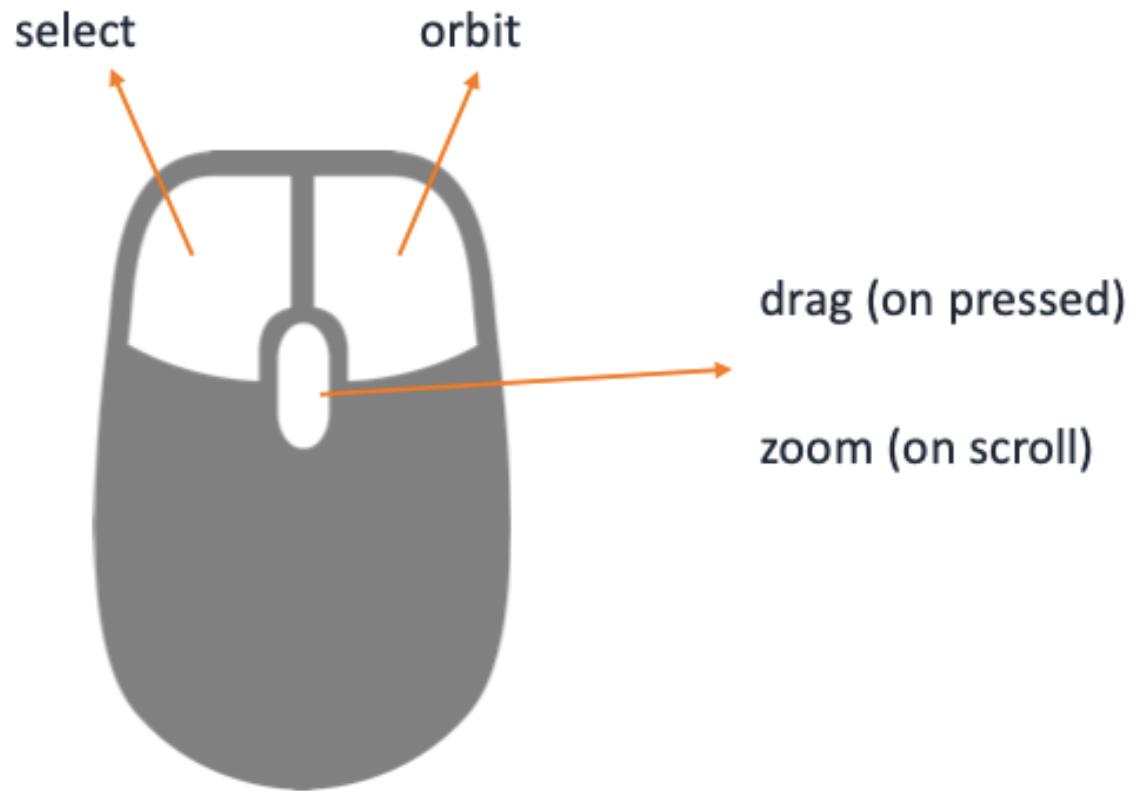
CANCEL

CREATE





Scene navigation

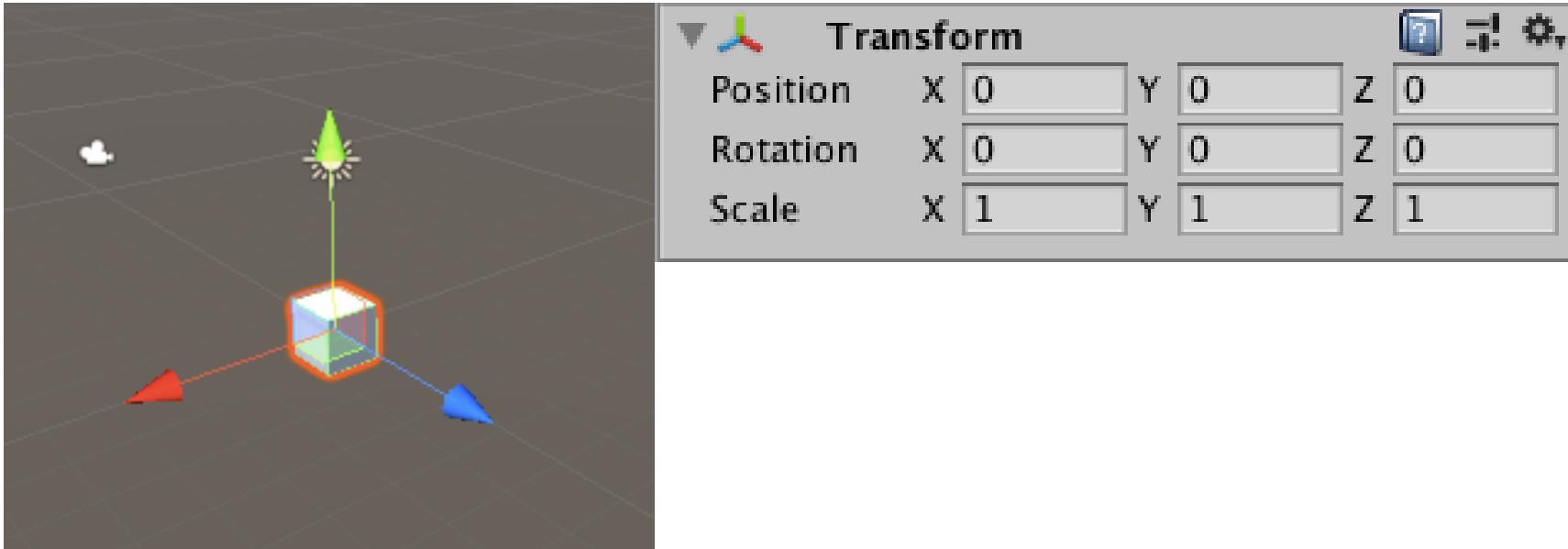


Flythrough mode

Right click + WASD EQ

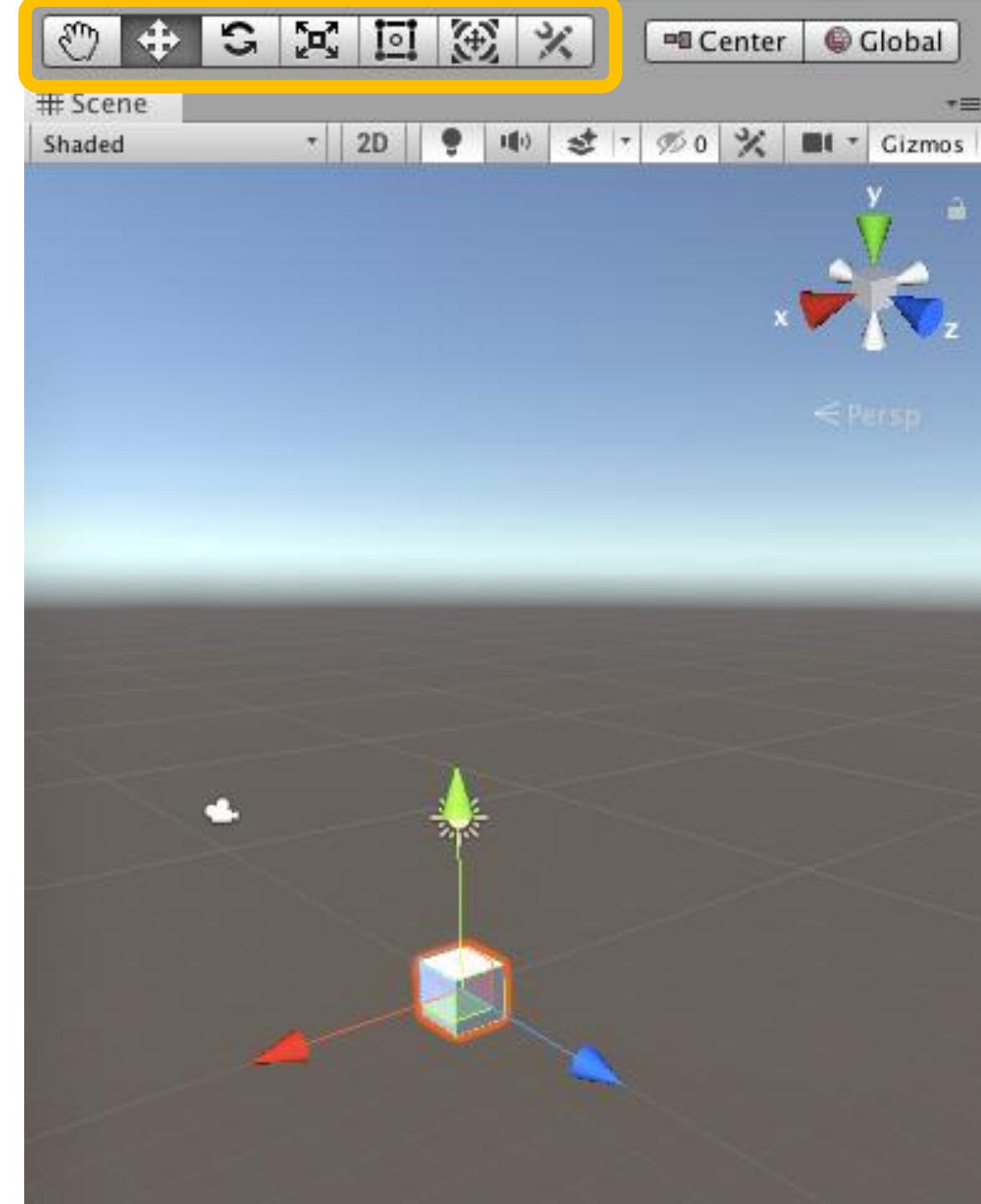
Transform

- The GameObject has Components that determine their behavior
- All objects have the **Transform** component to determine the object's
 - Position
 - Rotation
 - Scale



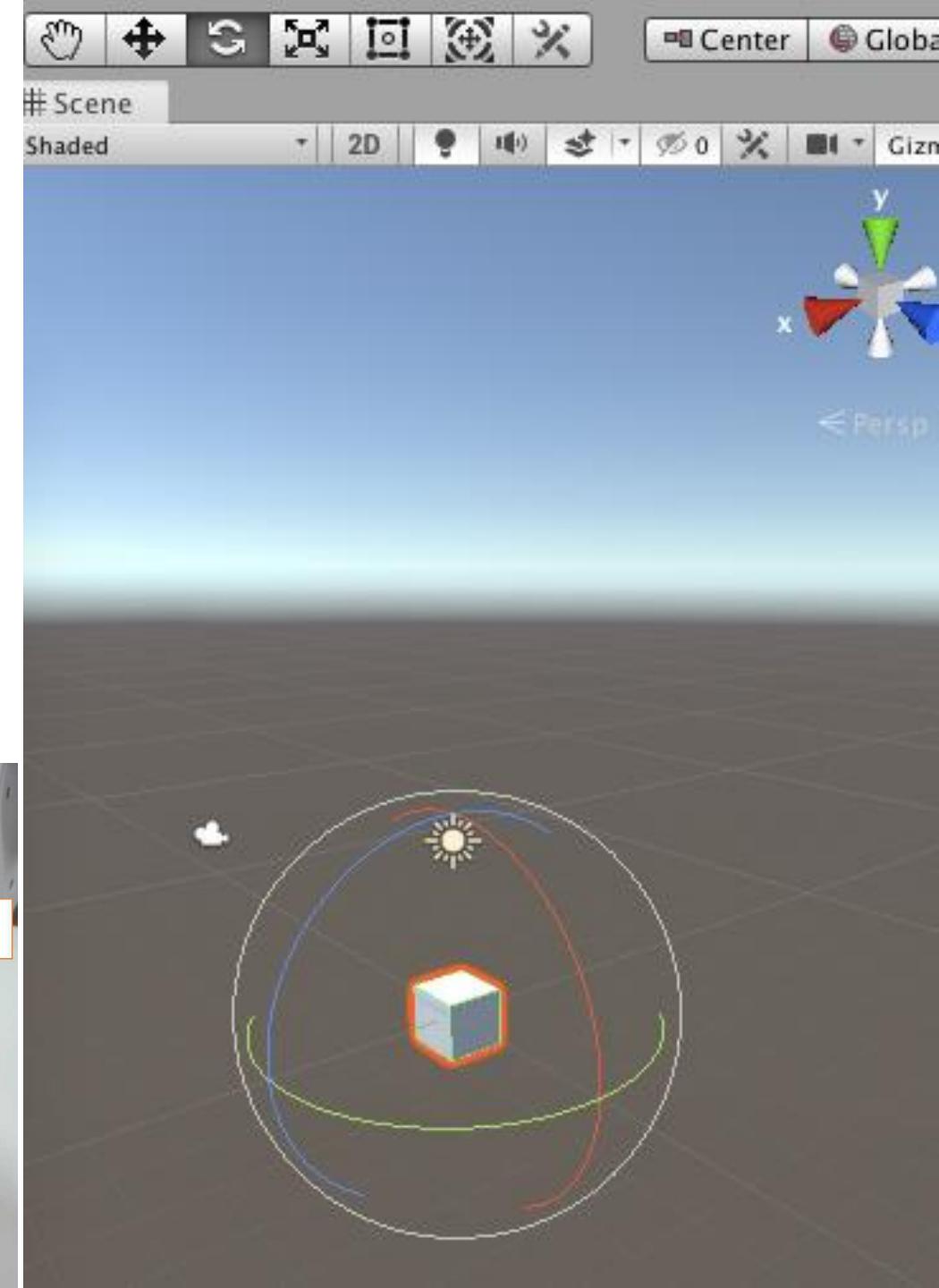
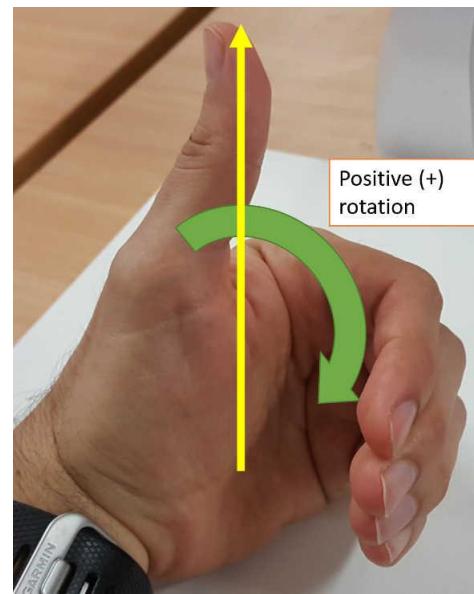
Coordinate system

- Change position, rotation, scales
in the scene view



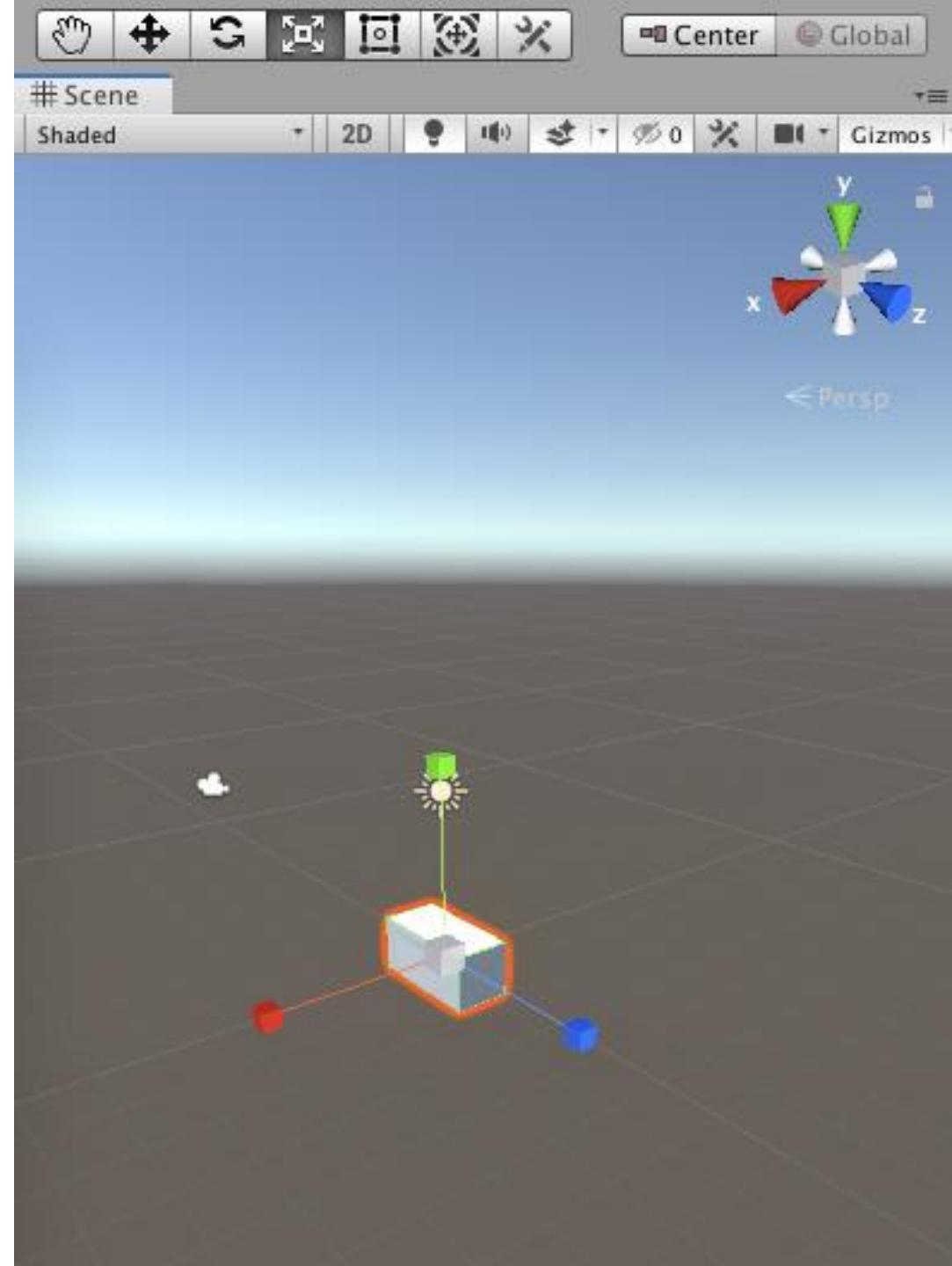
Rotation

- Can be expressed in EulerAngles (x, y, z)
- Positive and negative rotation using left hand coordinate
- Rotates the GameObject's local axis

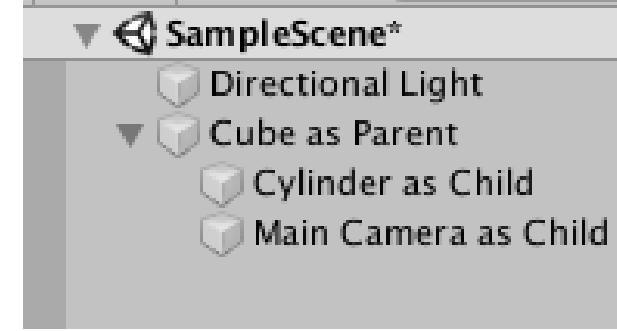


Scale

- Scale > 1 : increase the size of an object
- Scale < 1 : reduce the size of an object



Parent – Child



- The transform of the child is relative to the parent's
- For example:
 - Character carries the Camera
 - GameObject made of many GameObjects
 - Grouping things together
- To remove this relationship, just drag child out of the hierarchy of parent.

II. The roll-a-ball

Some projects to start with

- Unity projects (<https://learn.unity.com/projects>)
- Roll-a-Ball (<https://learn.unity.com/project/roll-a-ball?uv=2019.4>)

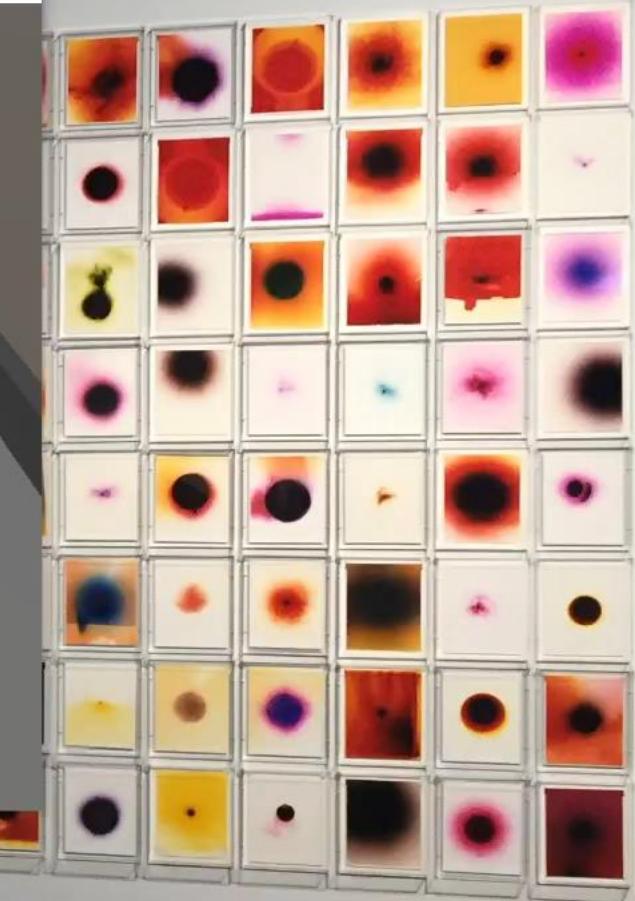
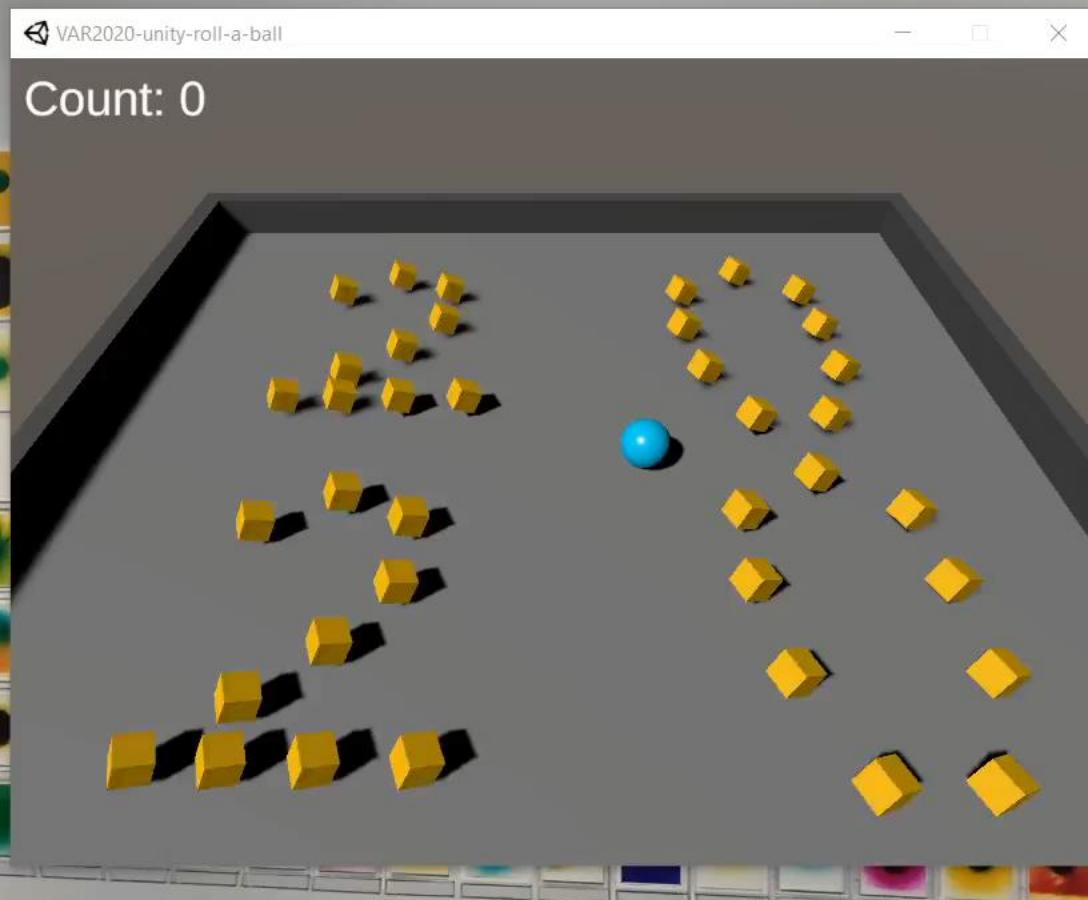
The image shows two project cards from the Unity Learn website:

- Roll-a-Ball**: Project • Beginner • 2 Hours 10 Mins • 404 views. Unity Technologies. Overview Details
- Survival Shooter Training Day Phases**: Tutorial • Beginner • 3 Hours 30 Mins • 115 views. Unity Technologies. Overview Tutorial Materials Details

Your progress is listed for both projects.

Language is set to English.

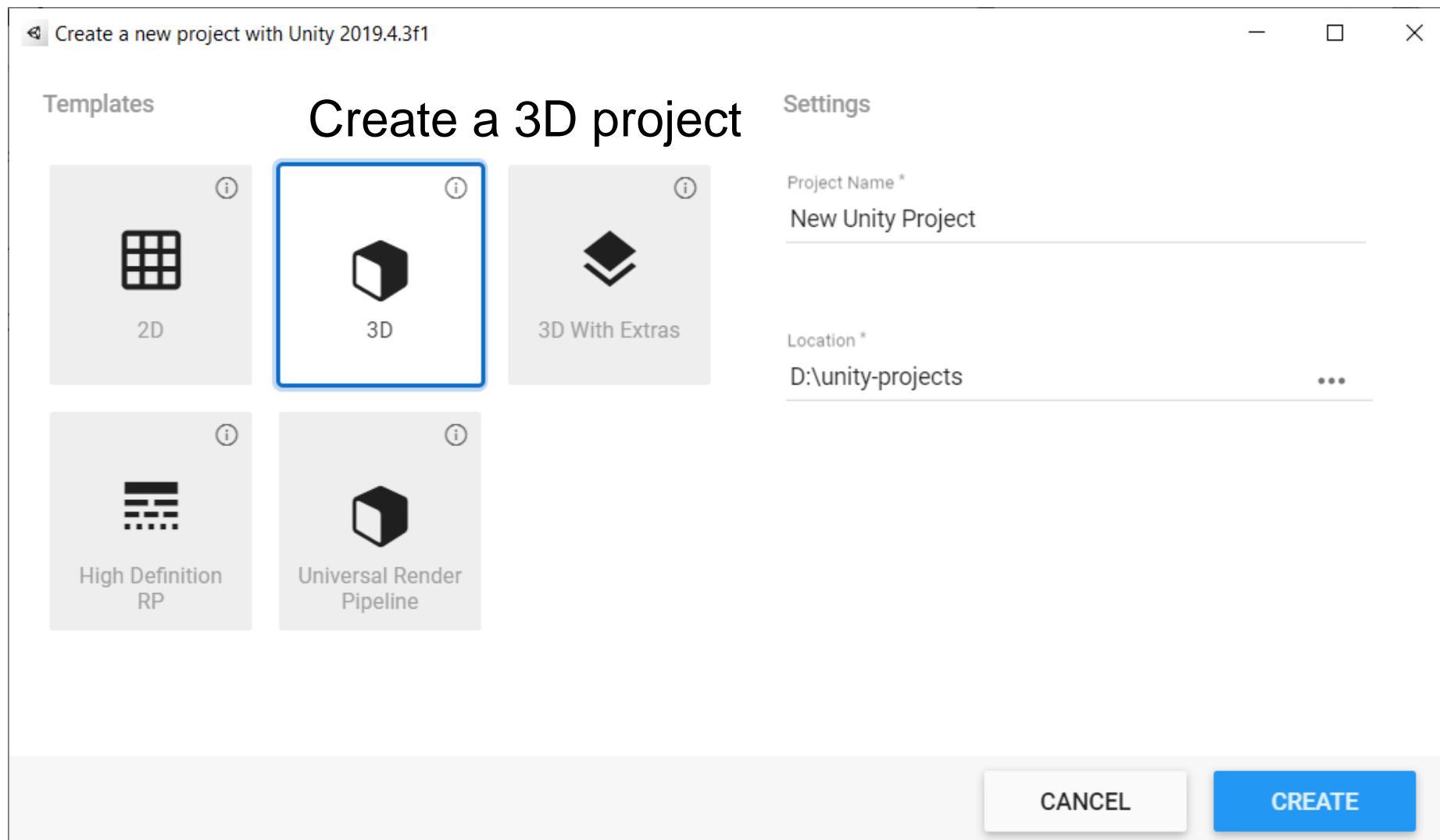
Page navigation: Select your Unity version (4.x) and a downward arrow icon.



Our goal in this game:

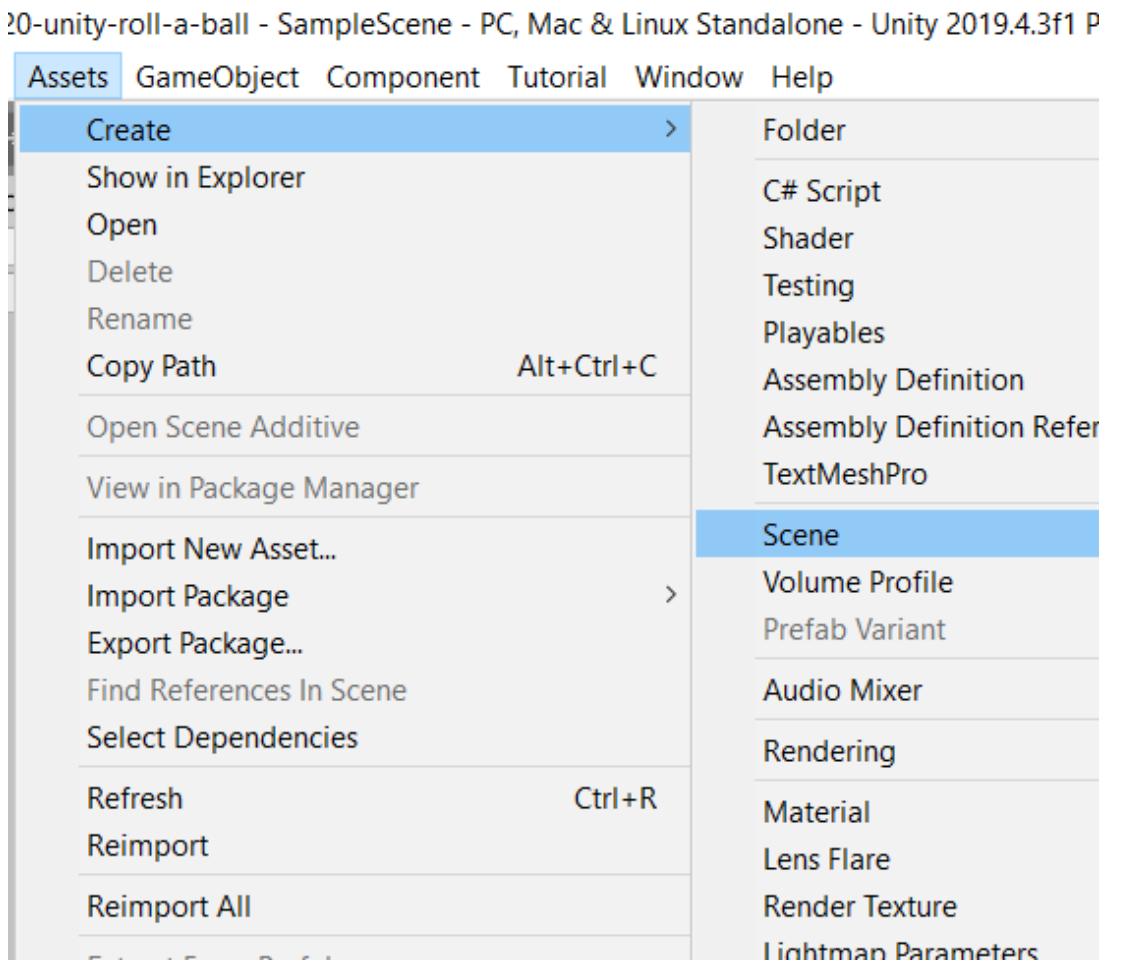
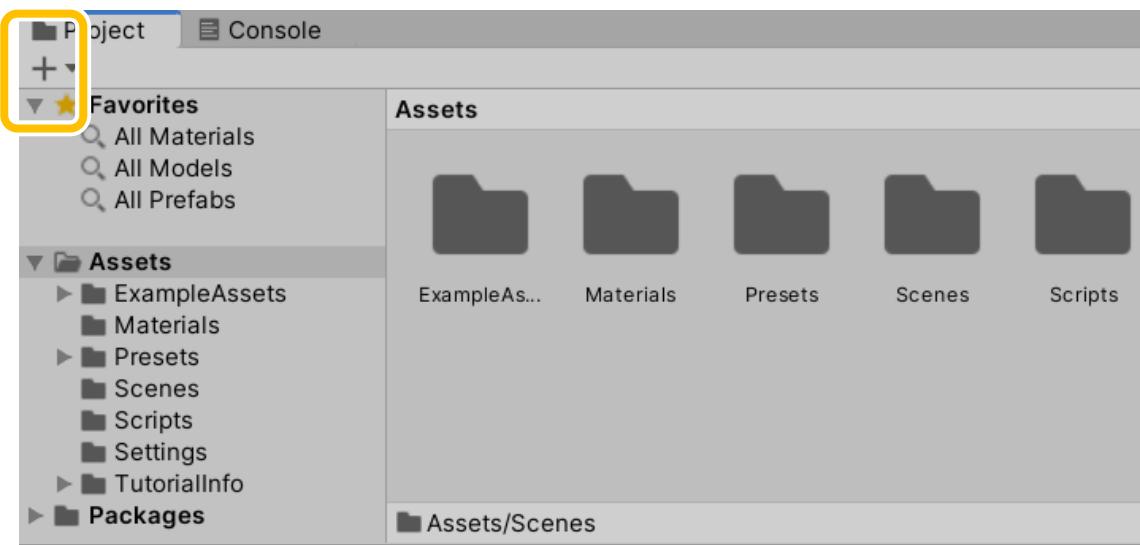
- when **Player** hits a **Pick-up**, the **Pick-up** disappears and increase the score.
- if score > X, win.

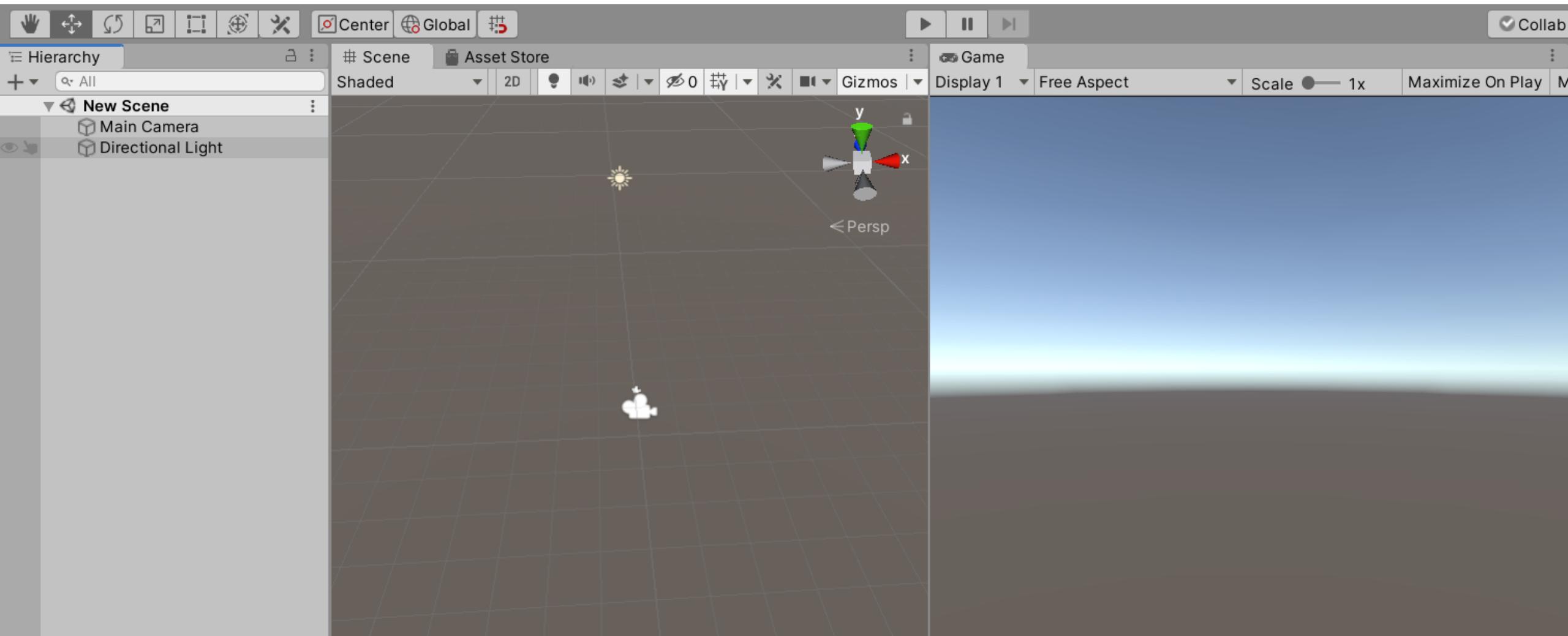
Create a new project



create a new Scene

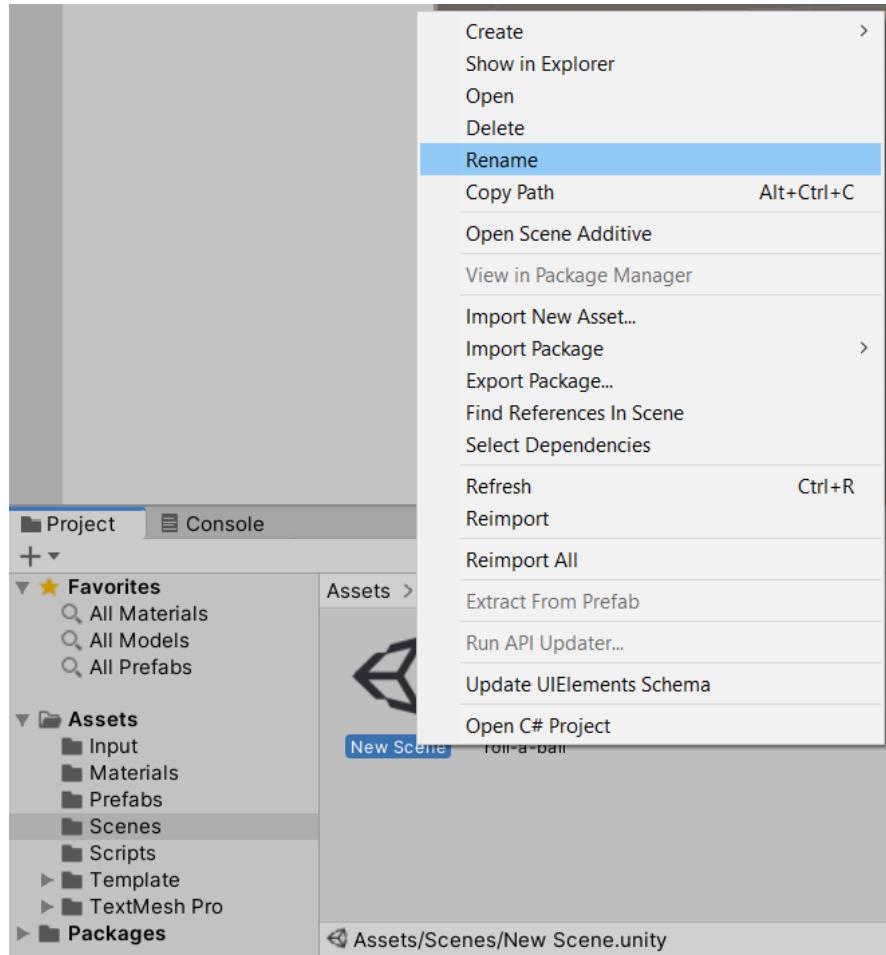
- Assets > create > scene
- Use project window





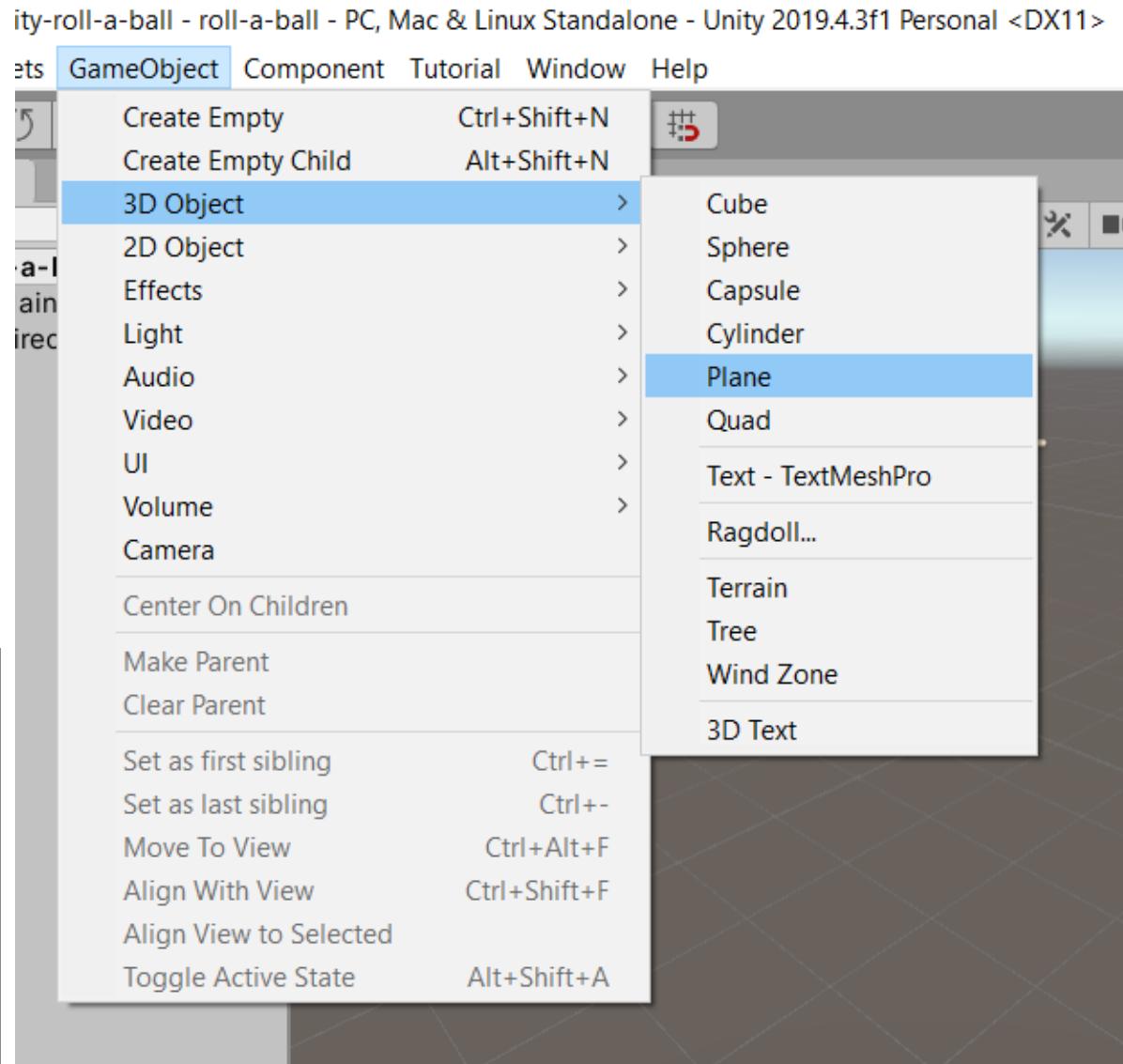
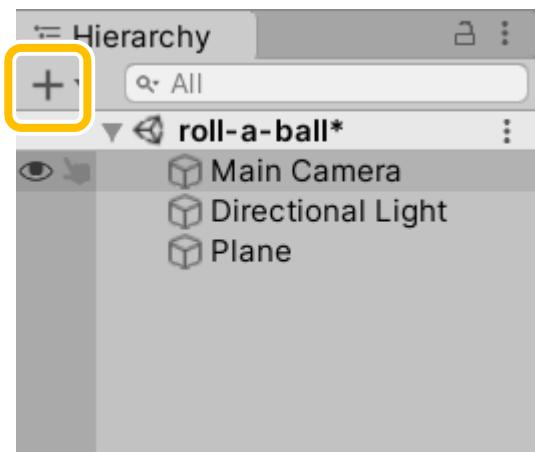
rename New Scene

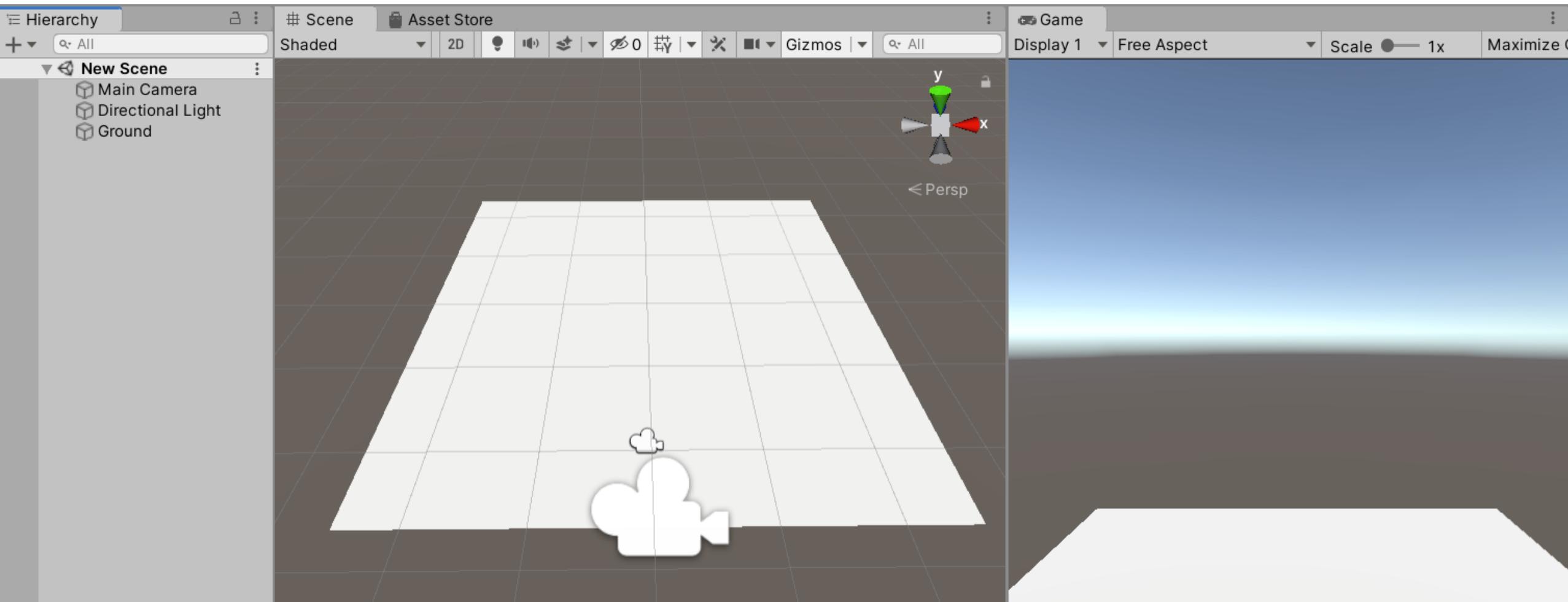
- right click the Scene you created in the Project panel
- select the Rename



create the ground

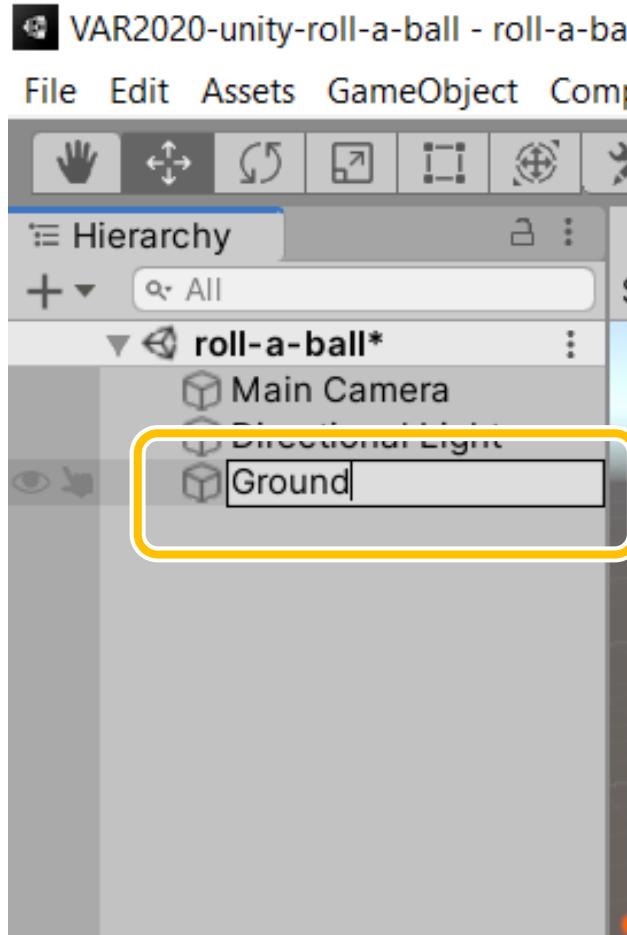
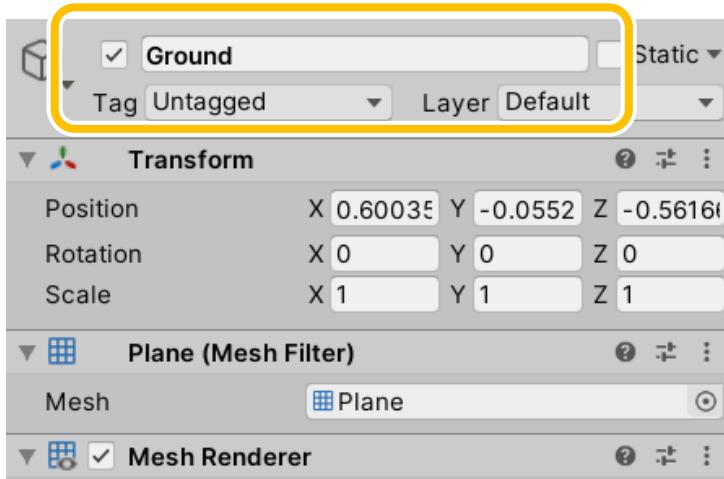
- GameObject > 3D Object > Plane





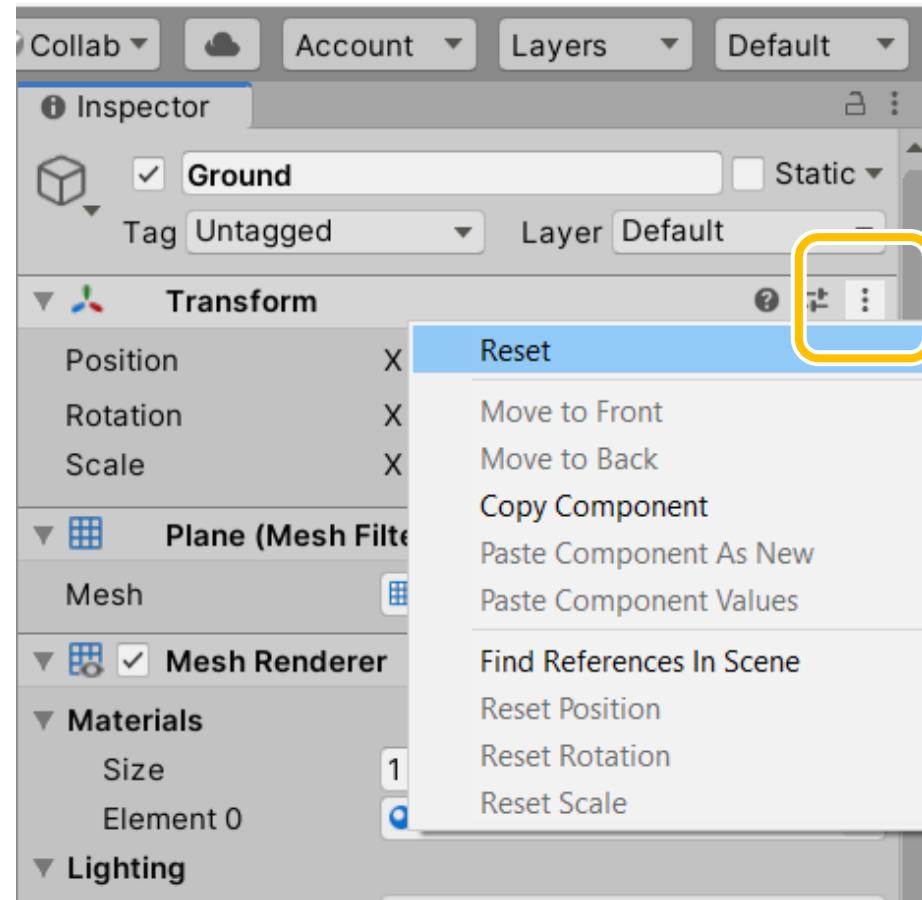
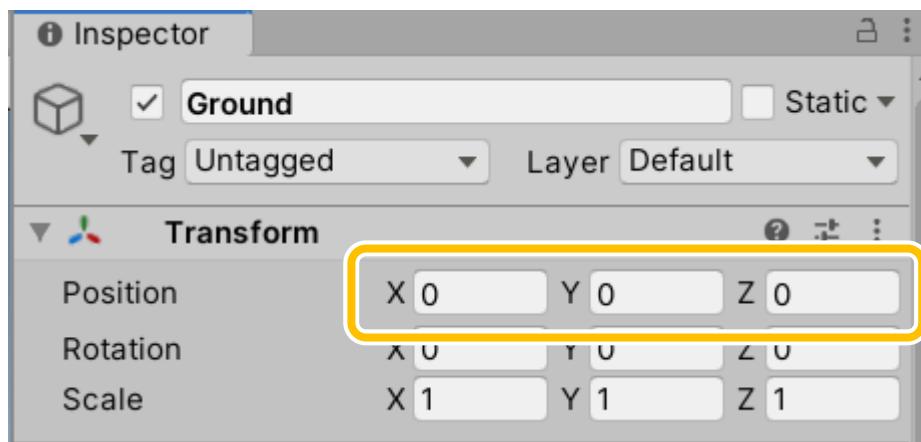
rename the Plane

- right click it in the hierarchy / in the inspector



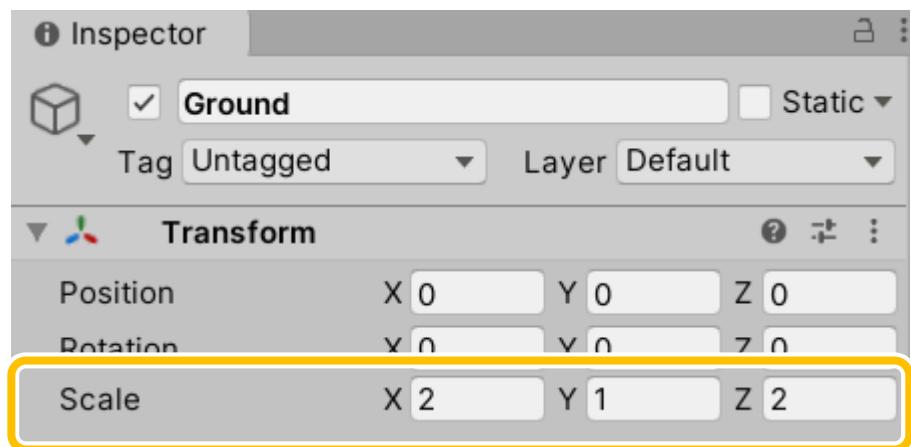
set Ground at (0,0,0)

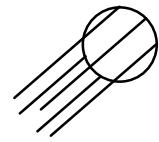
- Inspector > Transform



Change scale to (2,1,2)

- Use inspector





Directional Light
Transform
Light

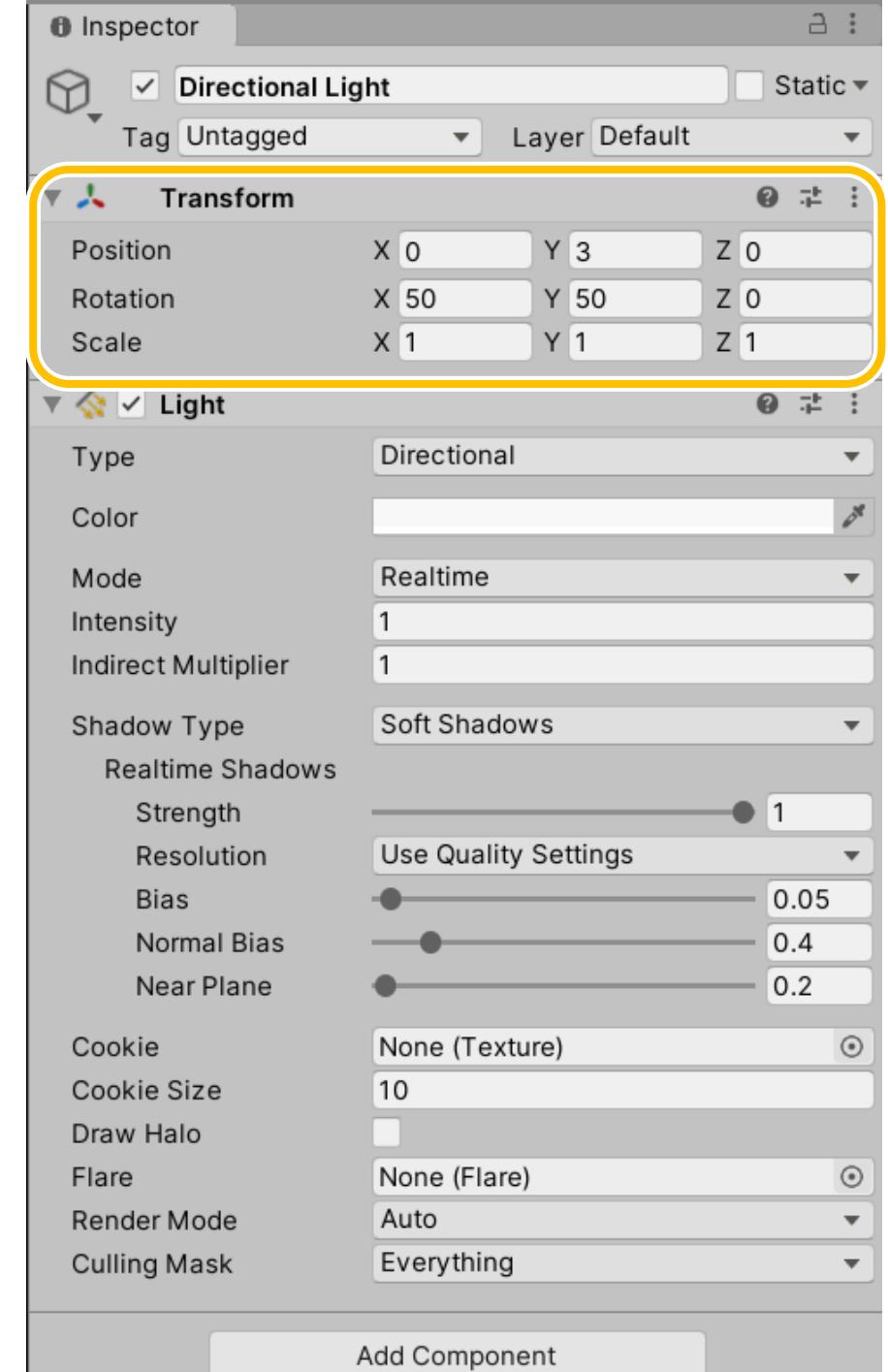
Main Camera
Transform
Camera



Ground
Transform
Renderer
Collider

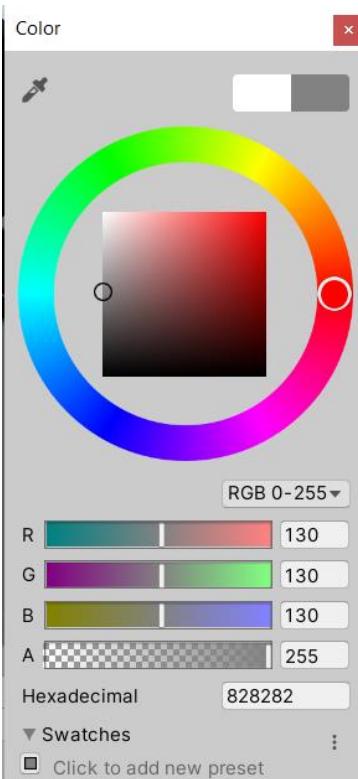
In the Directional Light

- Adjust Transform



In the Directional Light

- Adjust Transform
- Change the color of light to white



Inspector

Directional Light Static
Tag Untagged Layer Default

Transform

Position X 0 Y 3 Z 0
Rotation X 50 Y 50 Z 0
Scale X 1 Y 1 Z 1

Light

Type **Directional**

Color (highlighted with a yellow box)

Mode Realtime
Intensity 1
Indirect Multiplier 1

Shadow Type Soft Shadows
Realtime Shadows

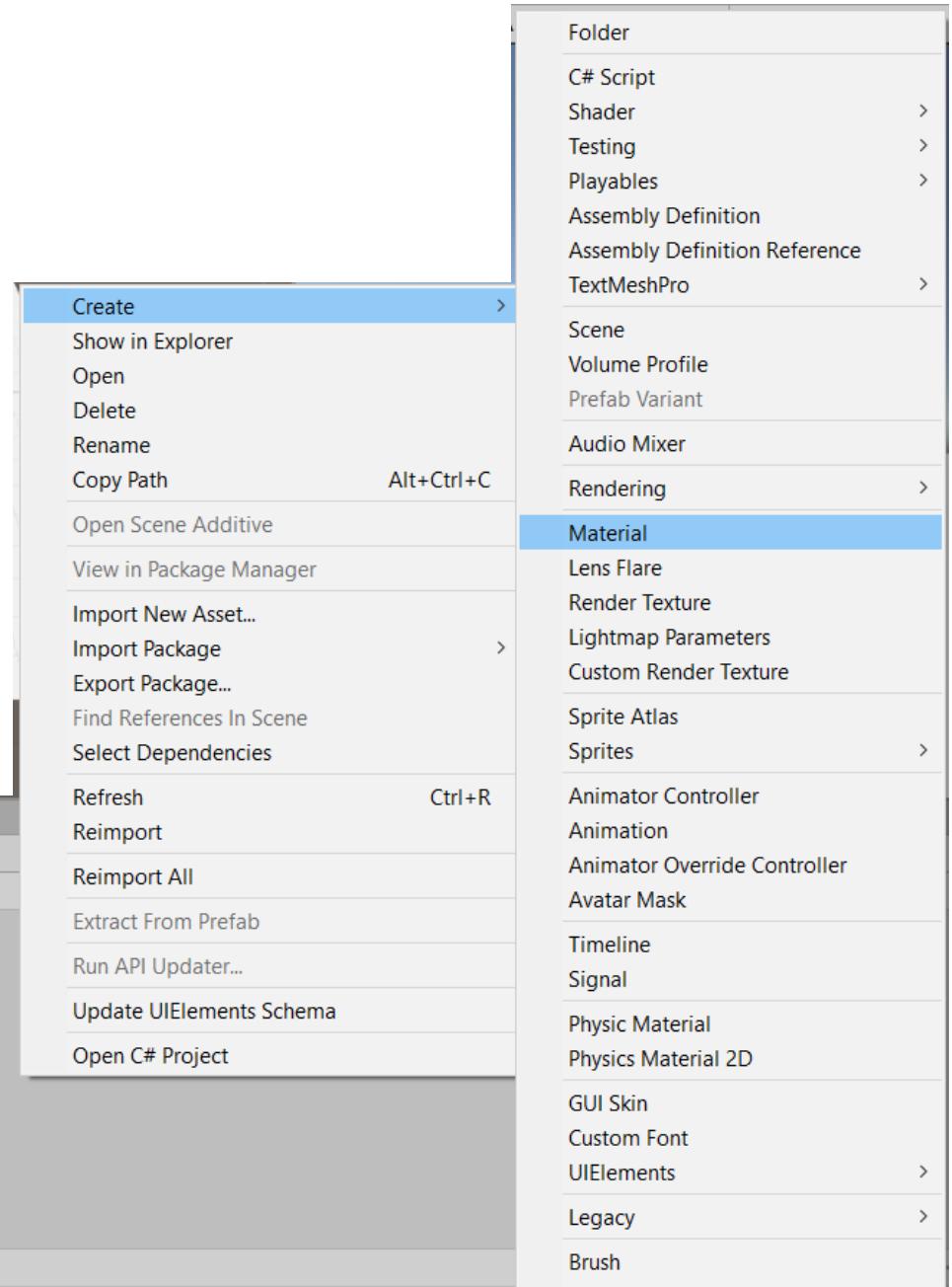
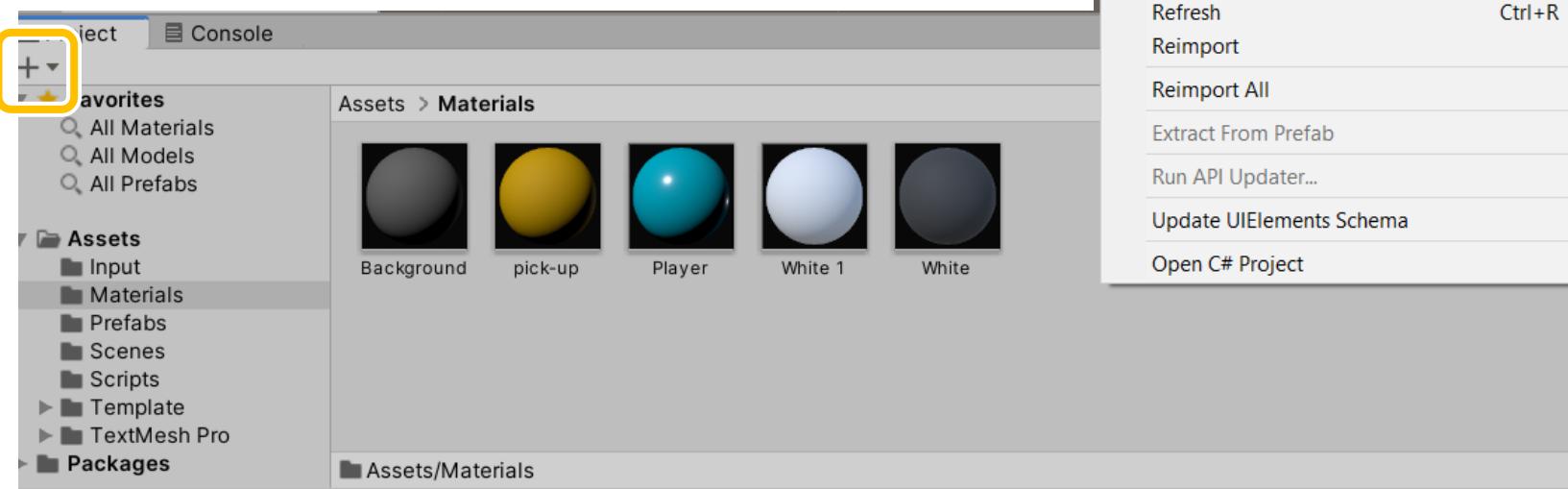
Strength 1
Resolution Use Quality Settings
Bias 0.05
Normal Bias 0.4
Near Plane 0.2

Cookie None (Texture)
Cookie Size 10
Draw Halo
Flare None (Flare)
Render Mode Auto
Culling Mask Everything

Add Component

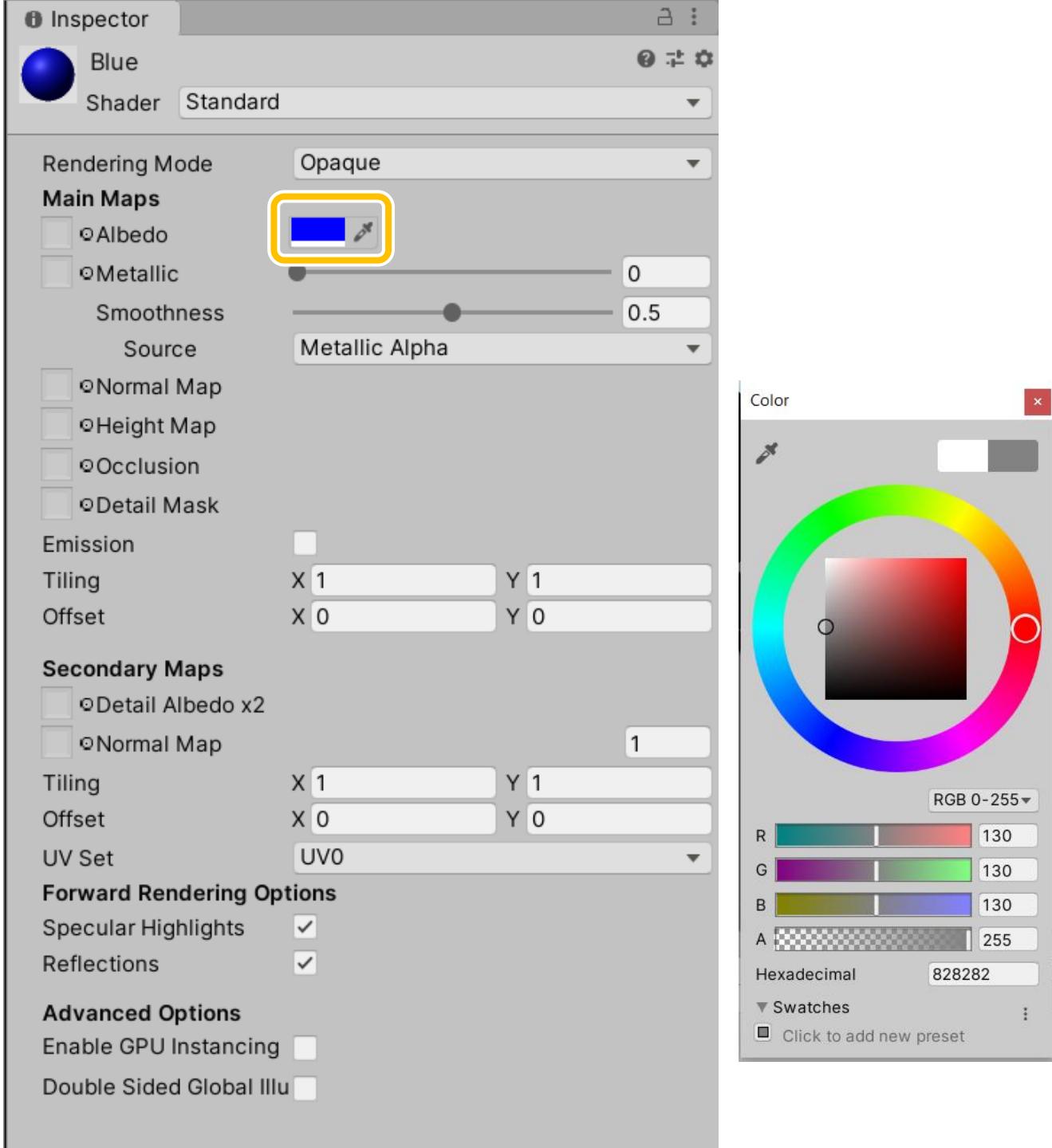
create Materials

- In Assets > Create > Material

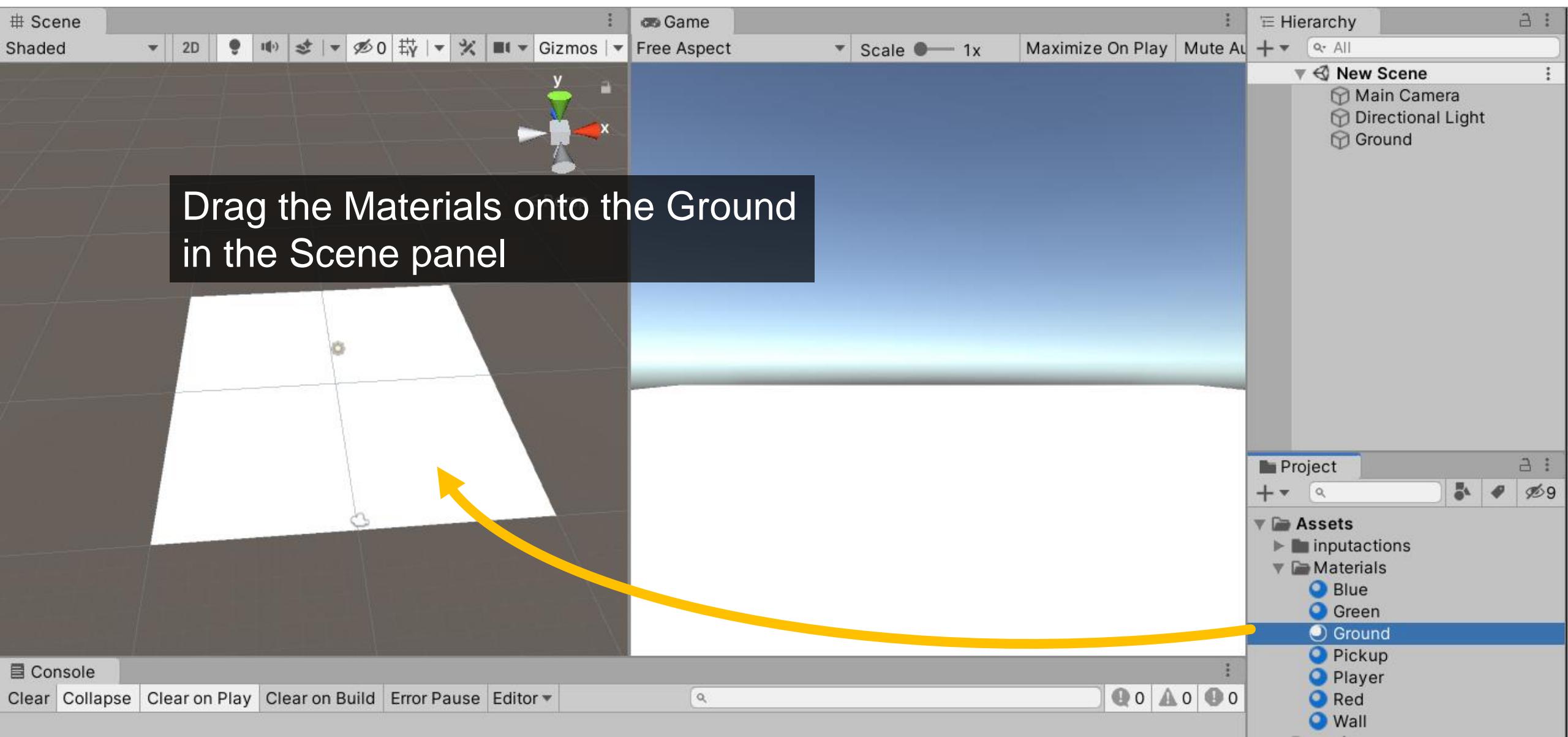


change Color

- Select the Material
- Change color in the Inspector

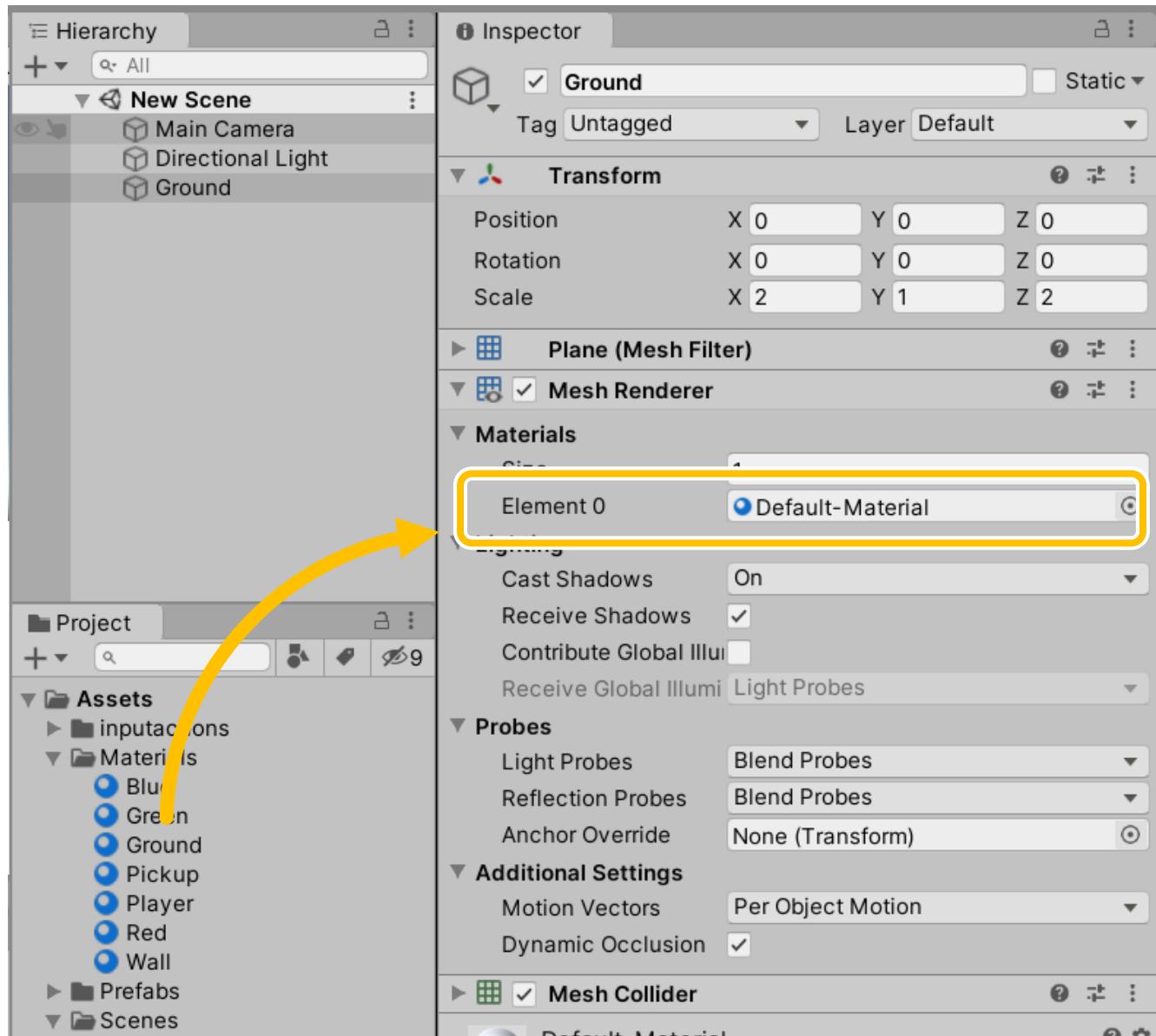


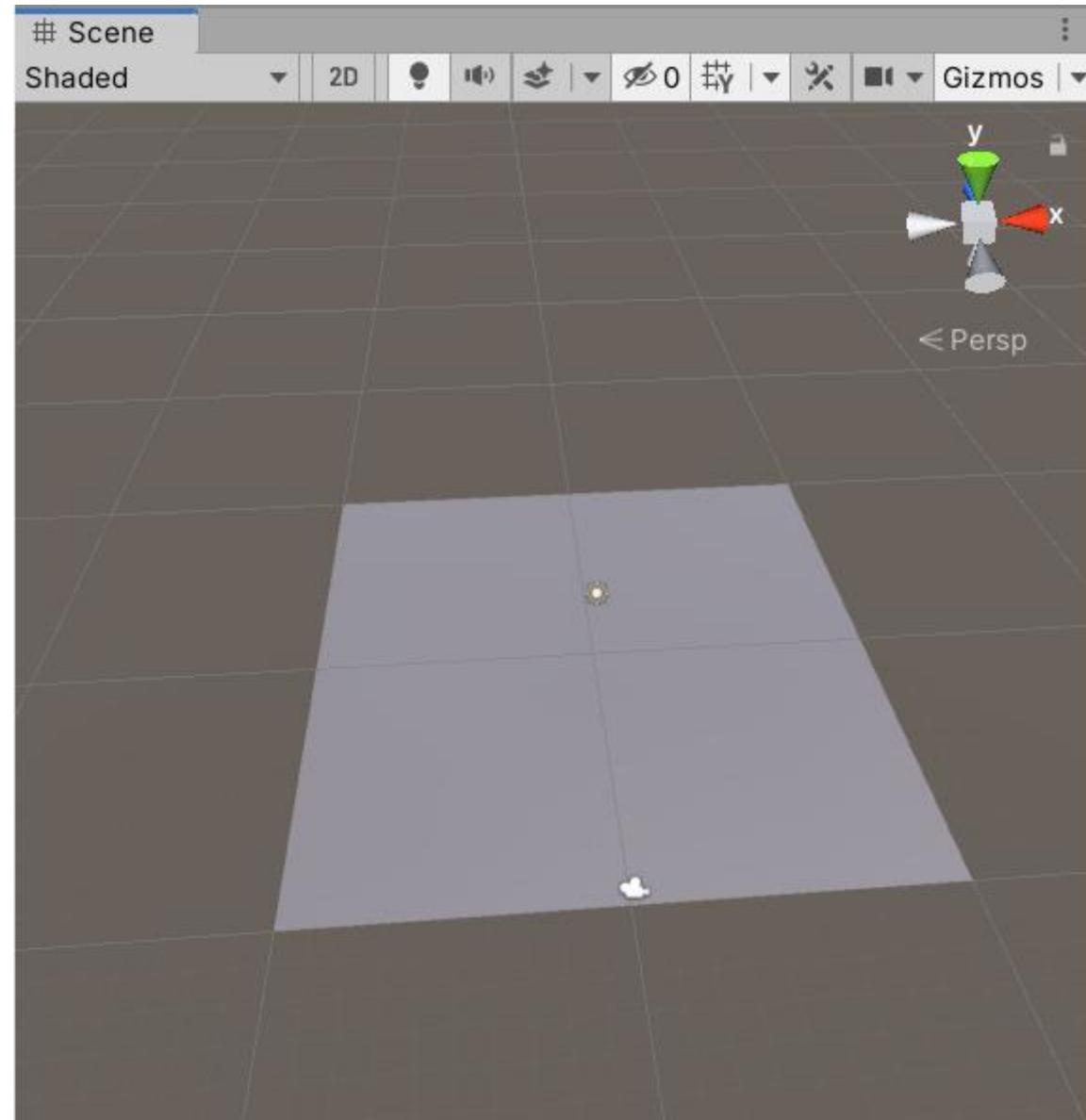
change Color

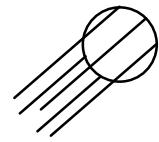


change Color

- You can also drag into the inspector of the gameobject.





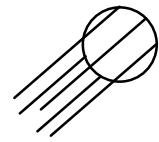


Directional Light
Transform
Light

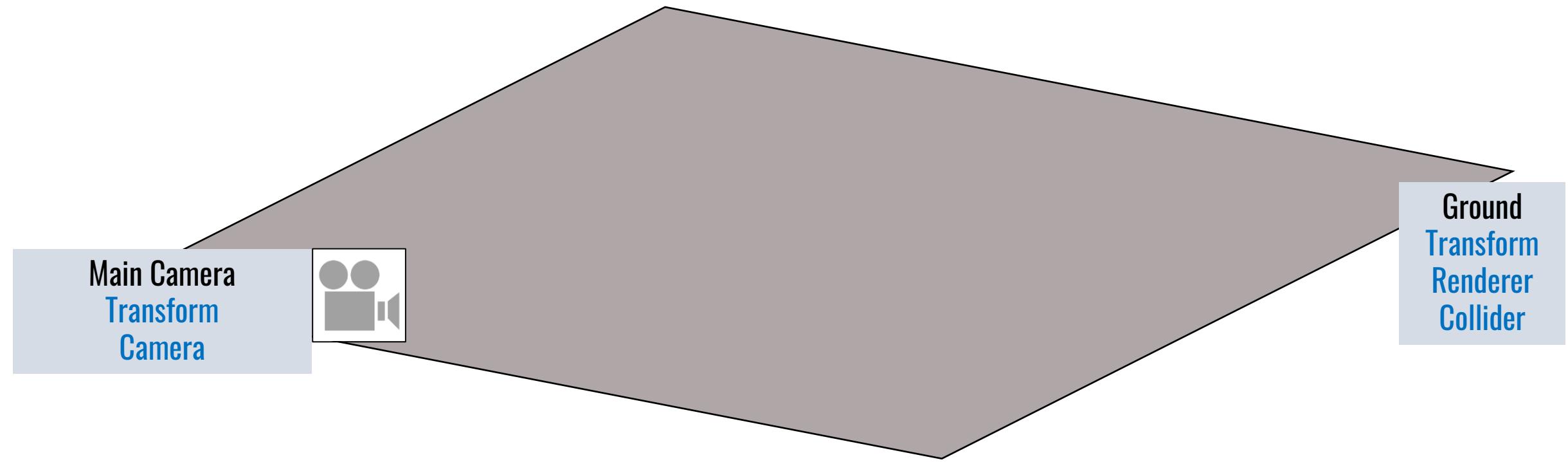
Main Camera
Transform
Camera



Ground
Transform
Renderer
Collider

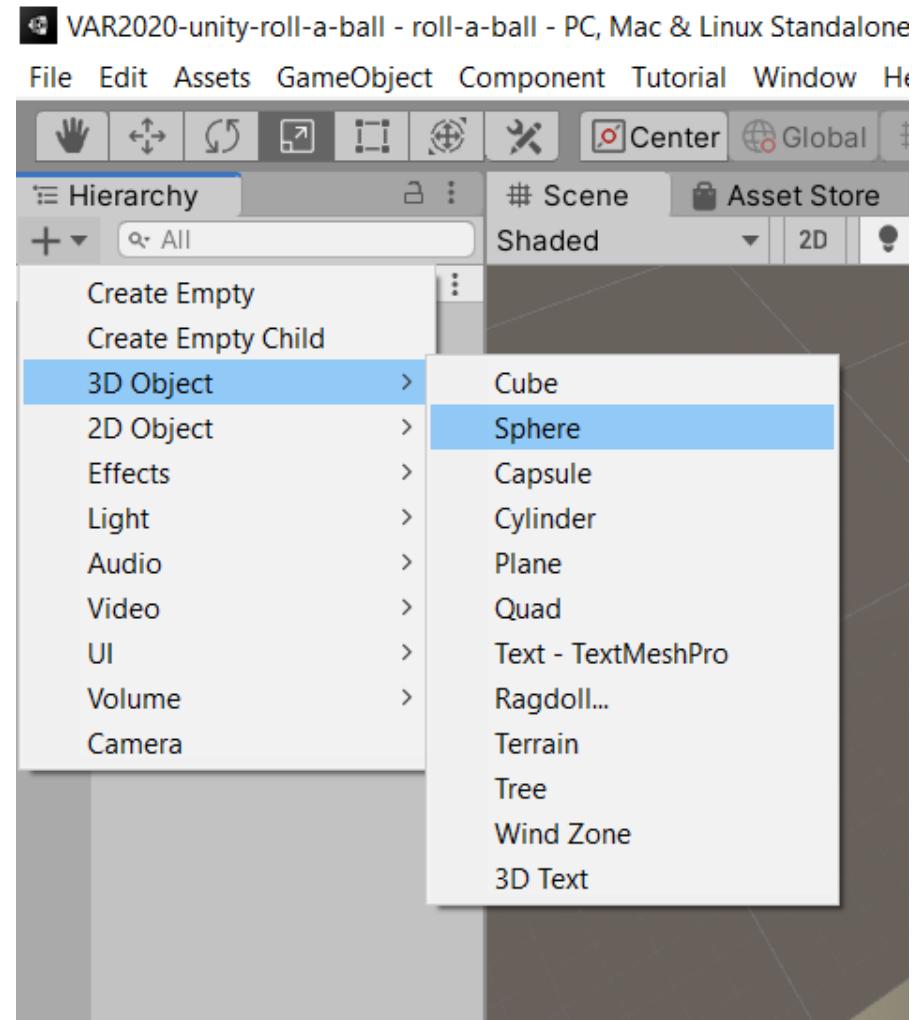


Directional Light
Transform
Light



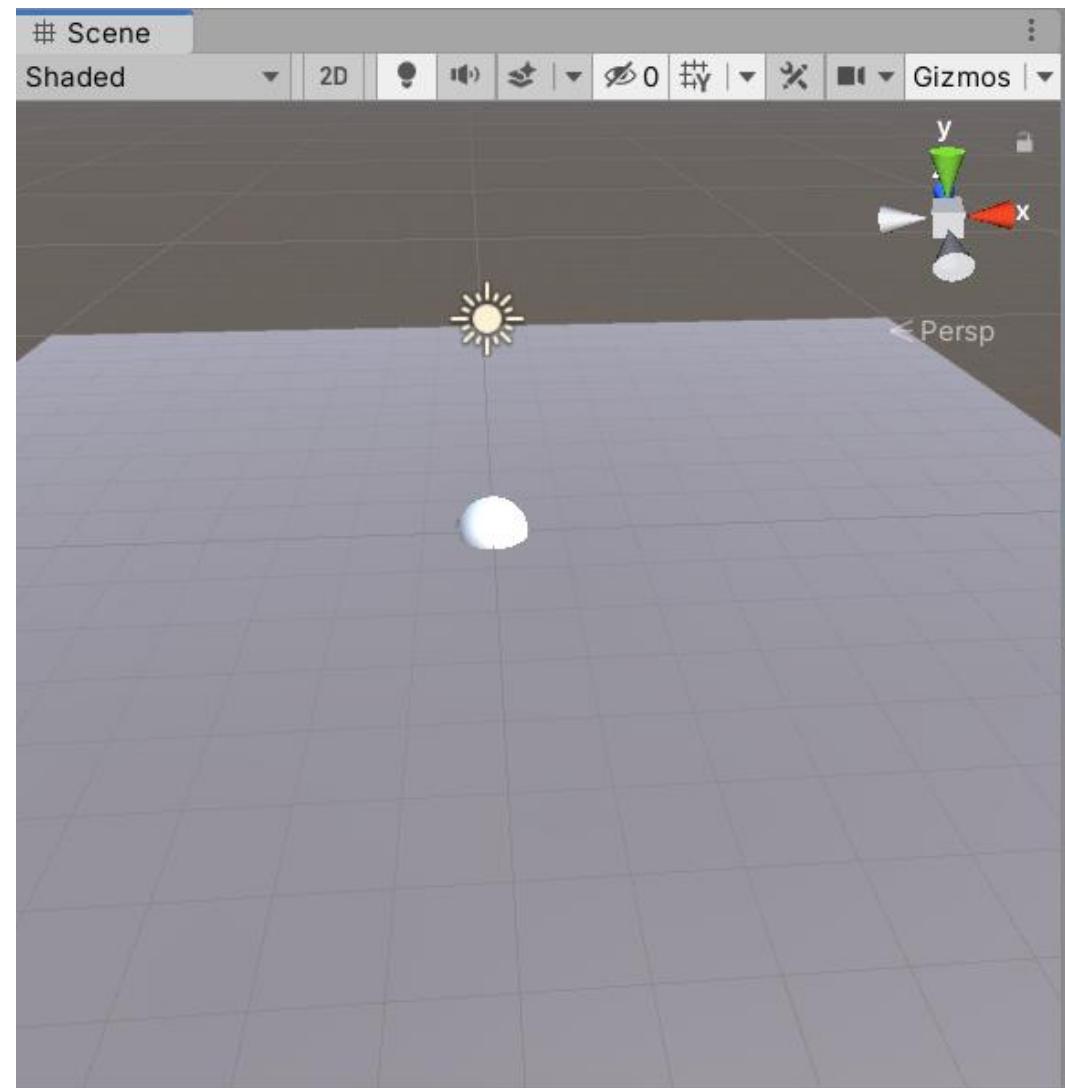
Create a sphere

- Reset to (0, 0, 0)
- Name it as “Player”

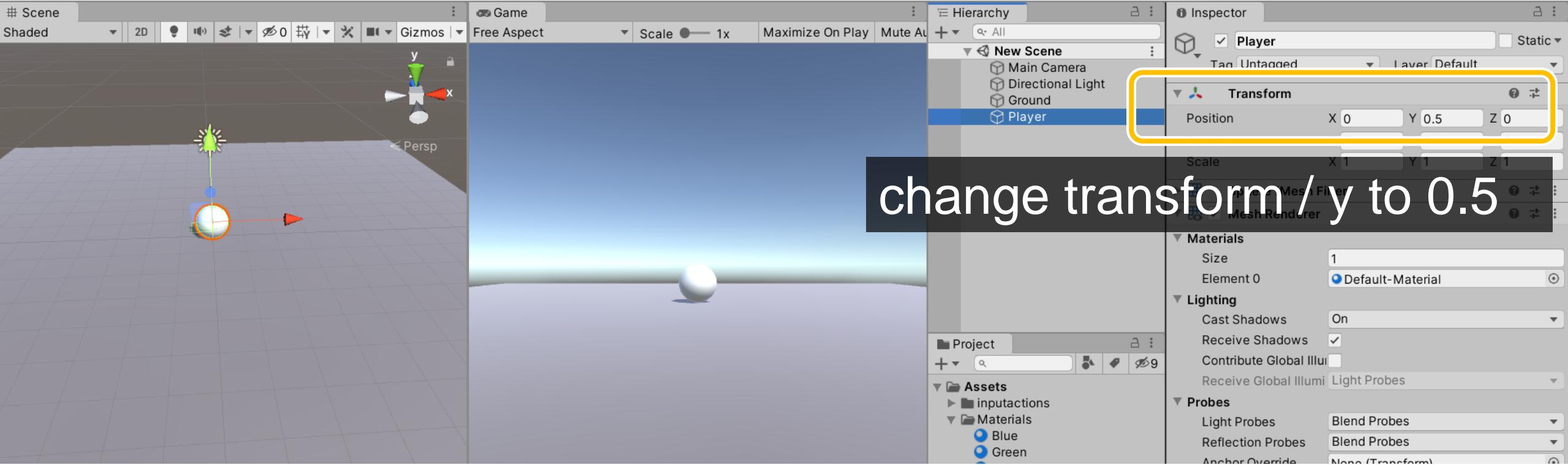


Create a sphere

- The player is in the Plane...

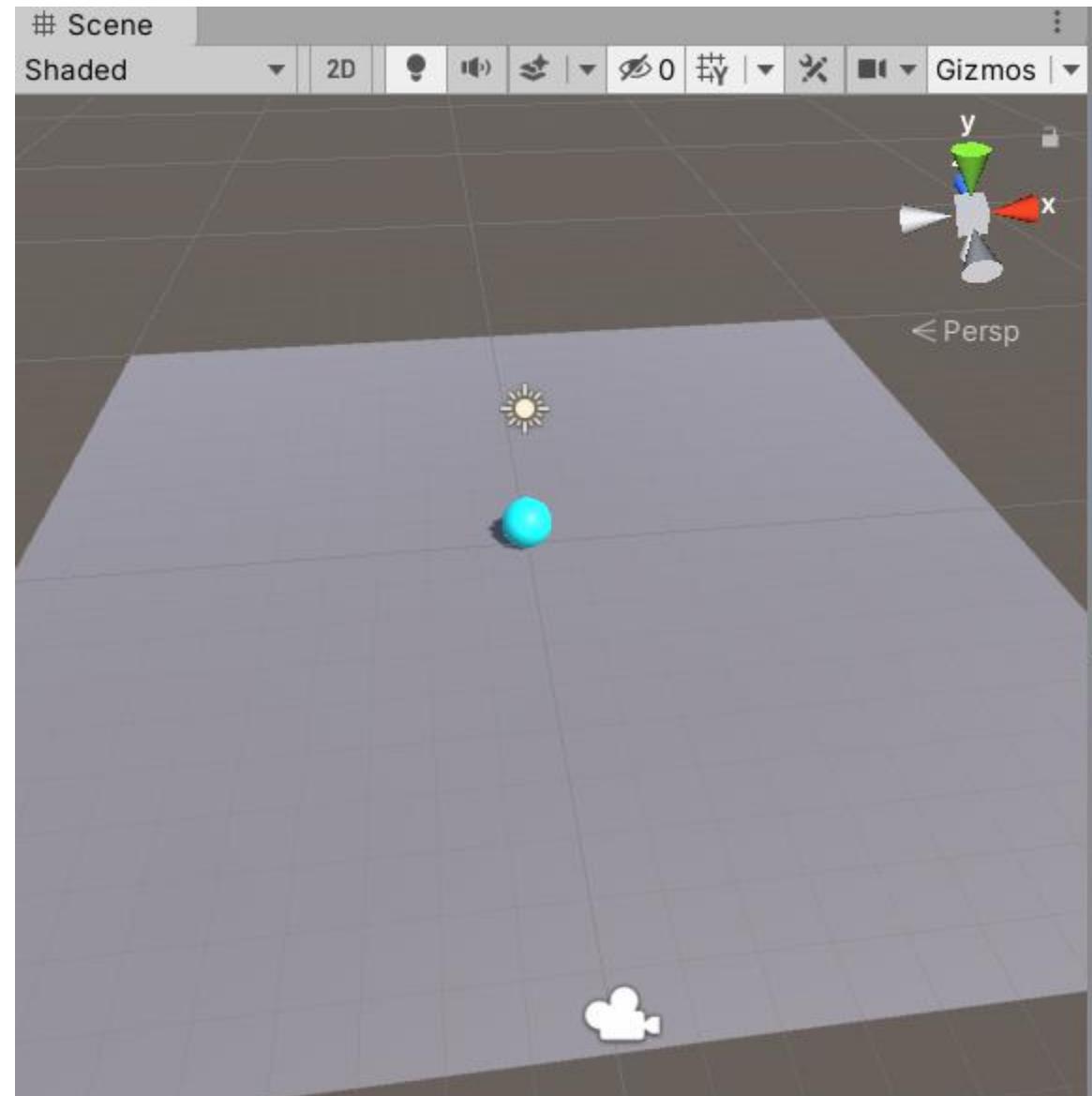


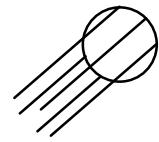
adjust Transform



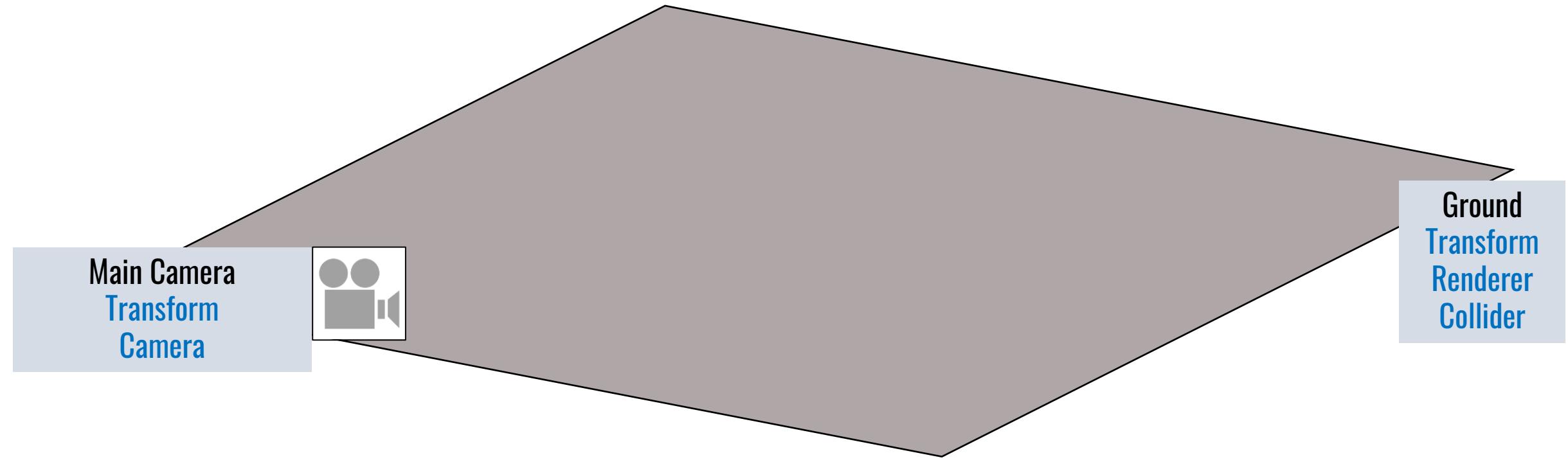
adjust Transform

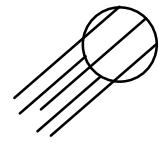
- Create a new color for Player



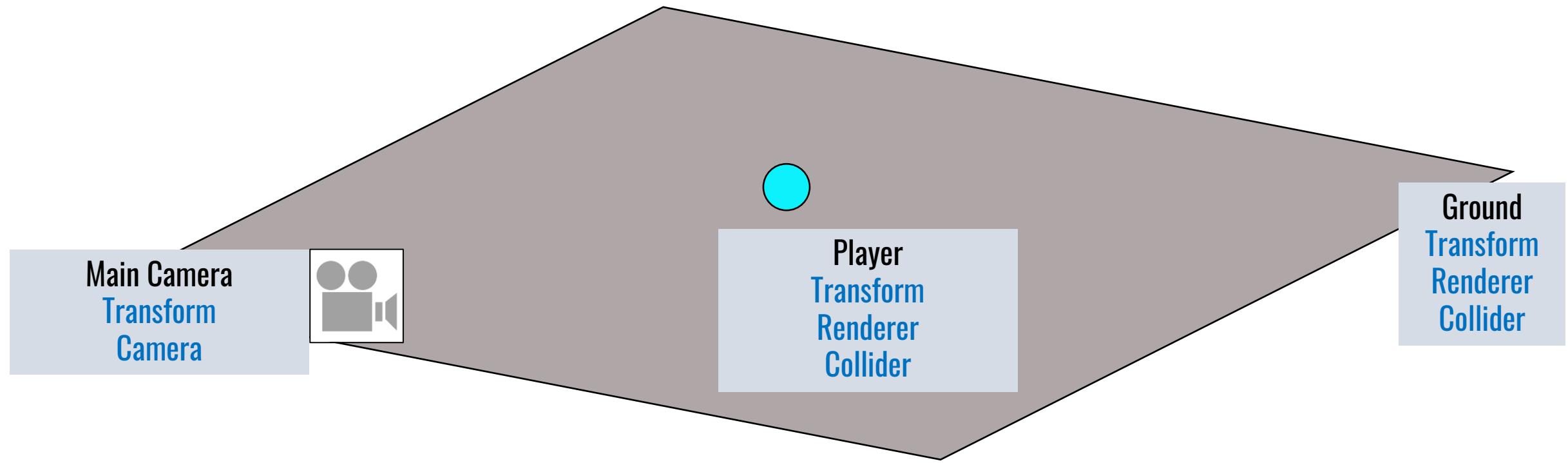


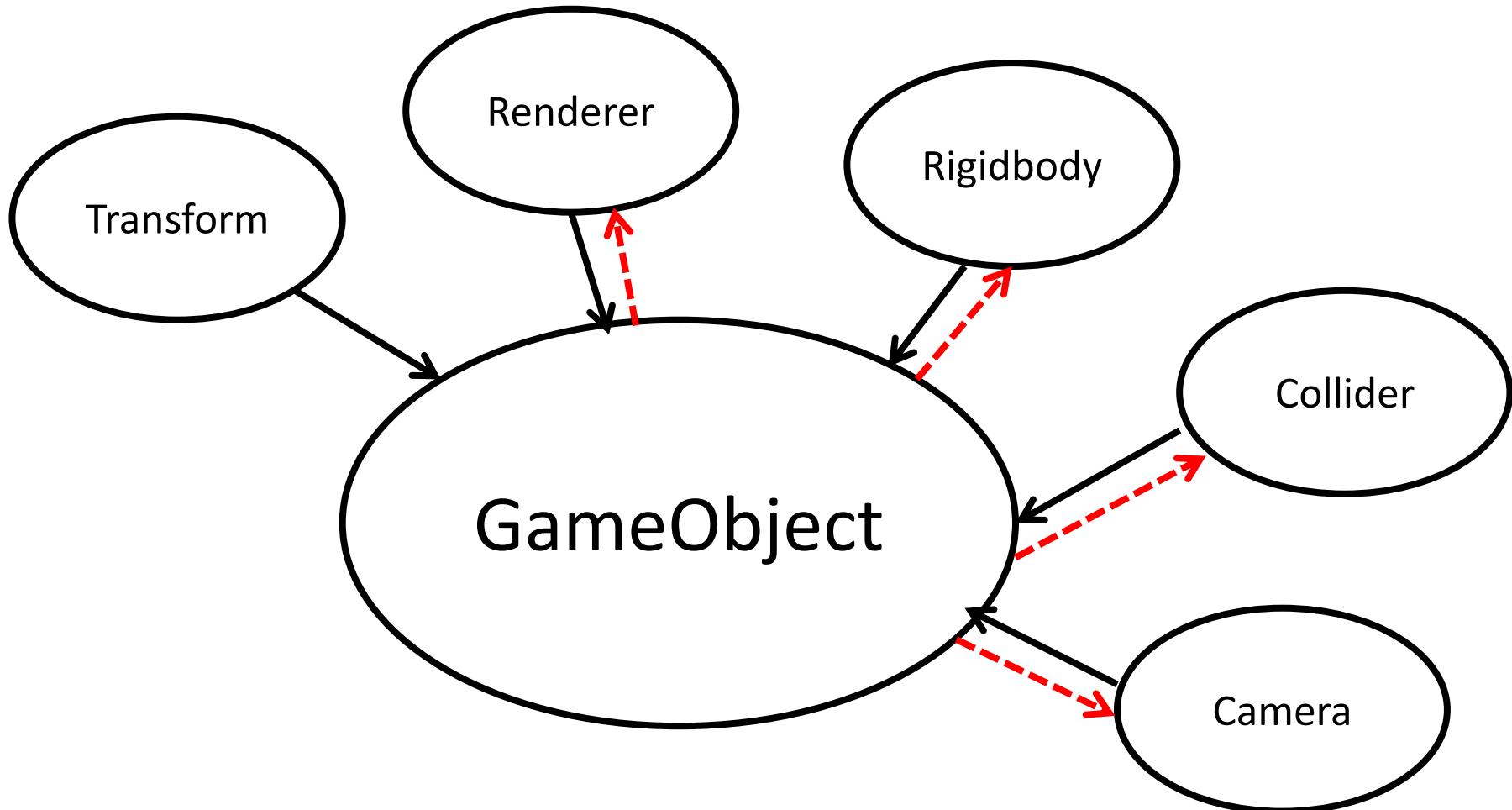
Directional Light
Transform
Light





Directional Light
Transform
Light

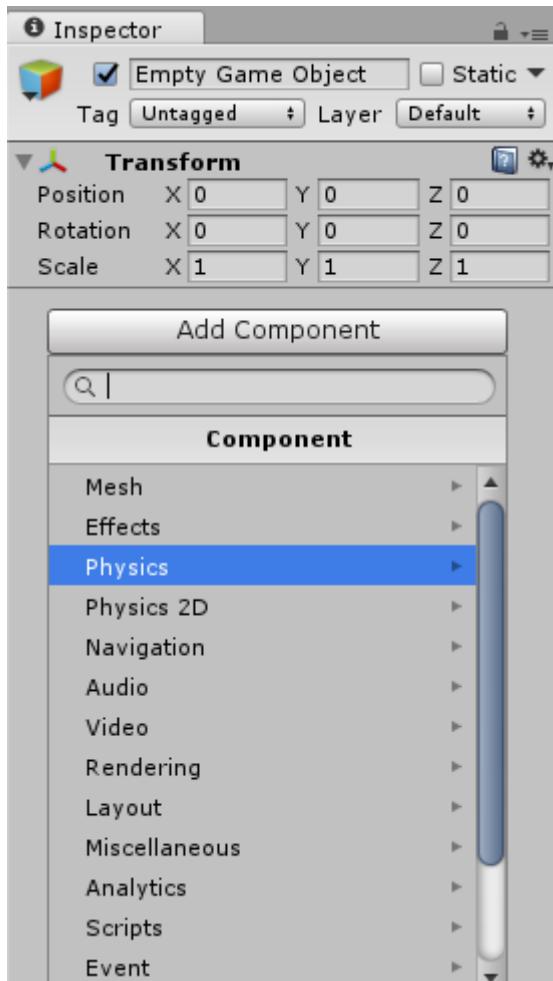




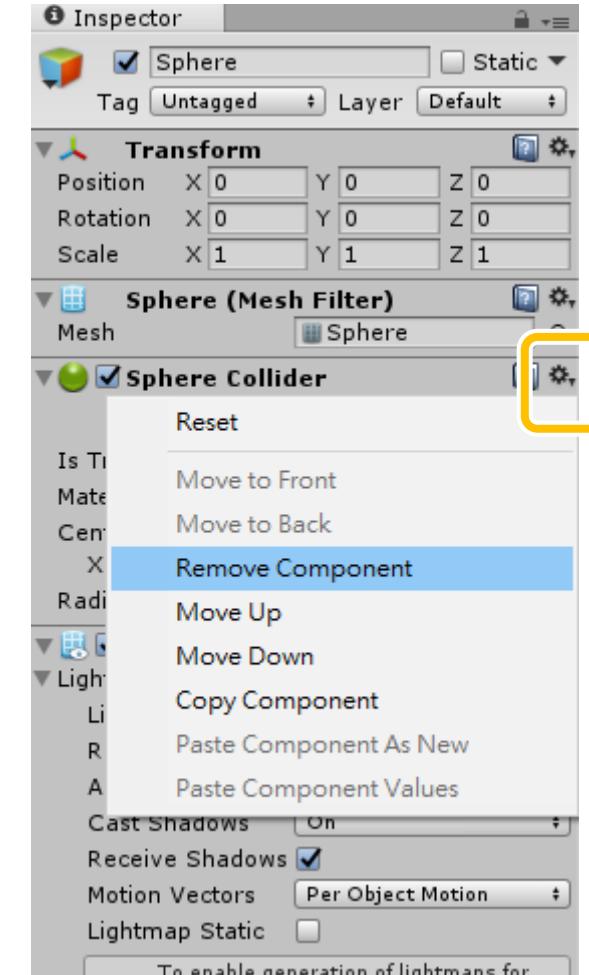
A **GameObject** has many components.
They all can be attached or removed in Unity or by script.

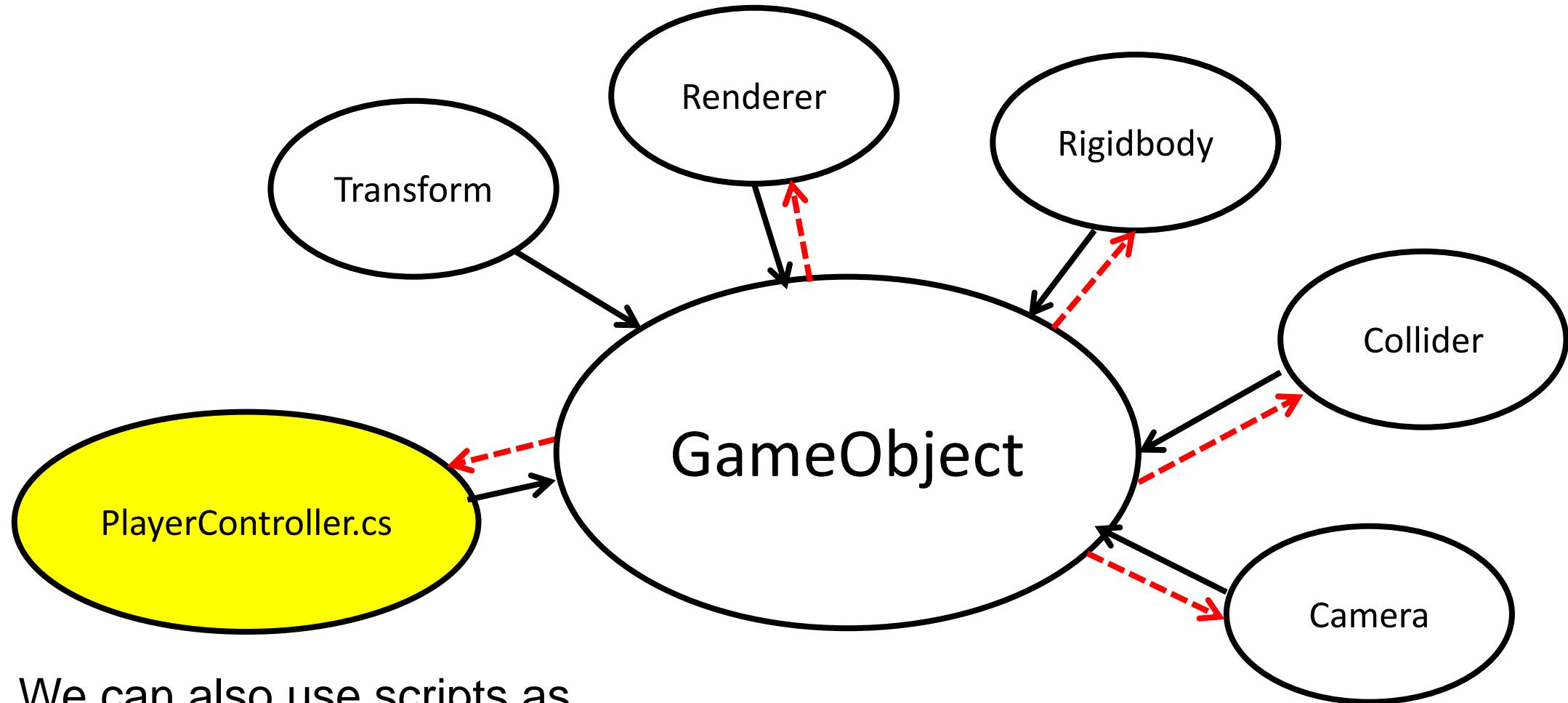
Components

- Add Component:



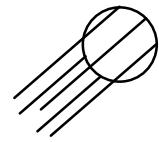
- Remove Component:



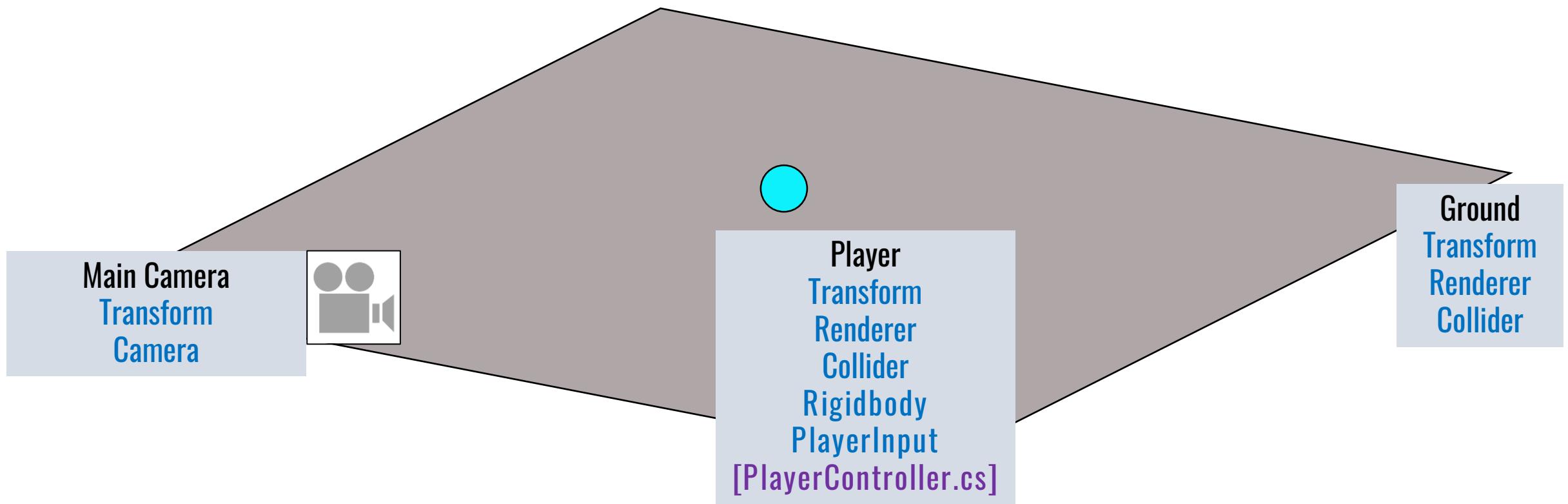


We can also use scripts as
a Component to control
the GameObject.

In this project, we use Unity Input System
to control Player with arrow keys.

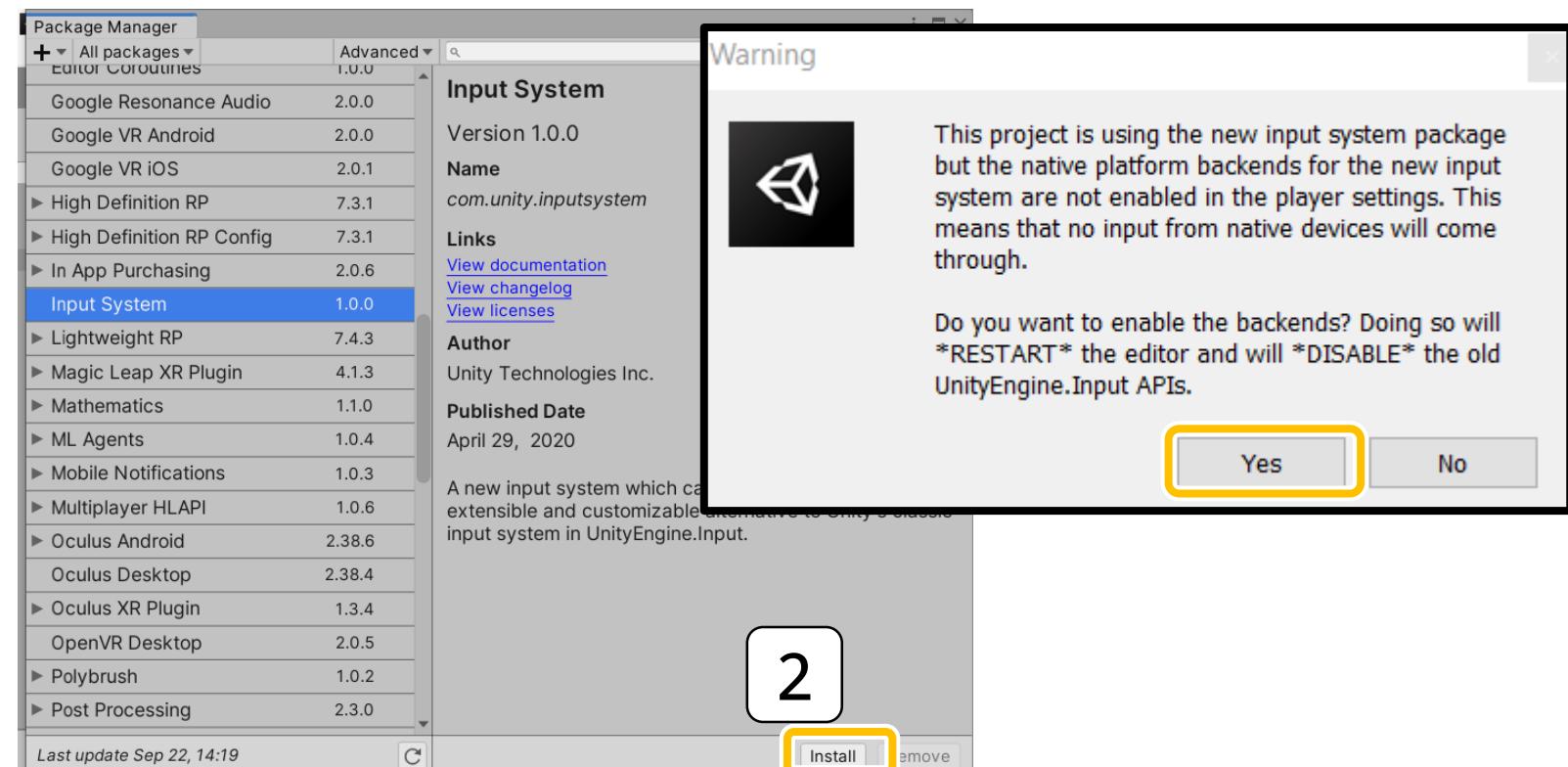
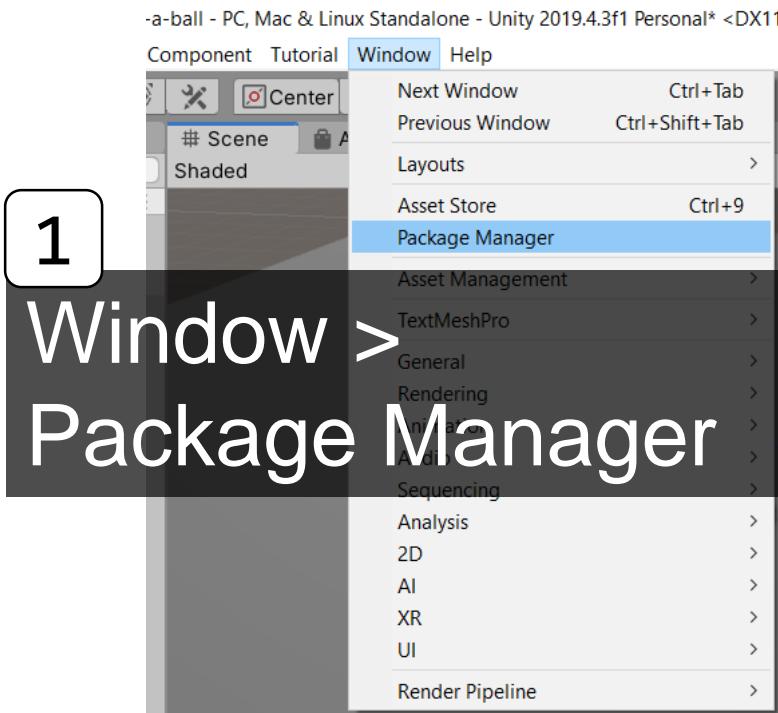


Directional Light
Transform
Light



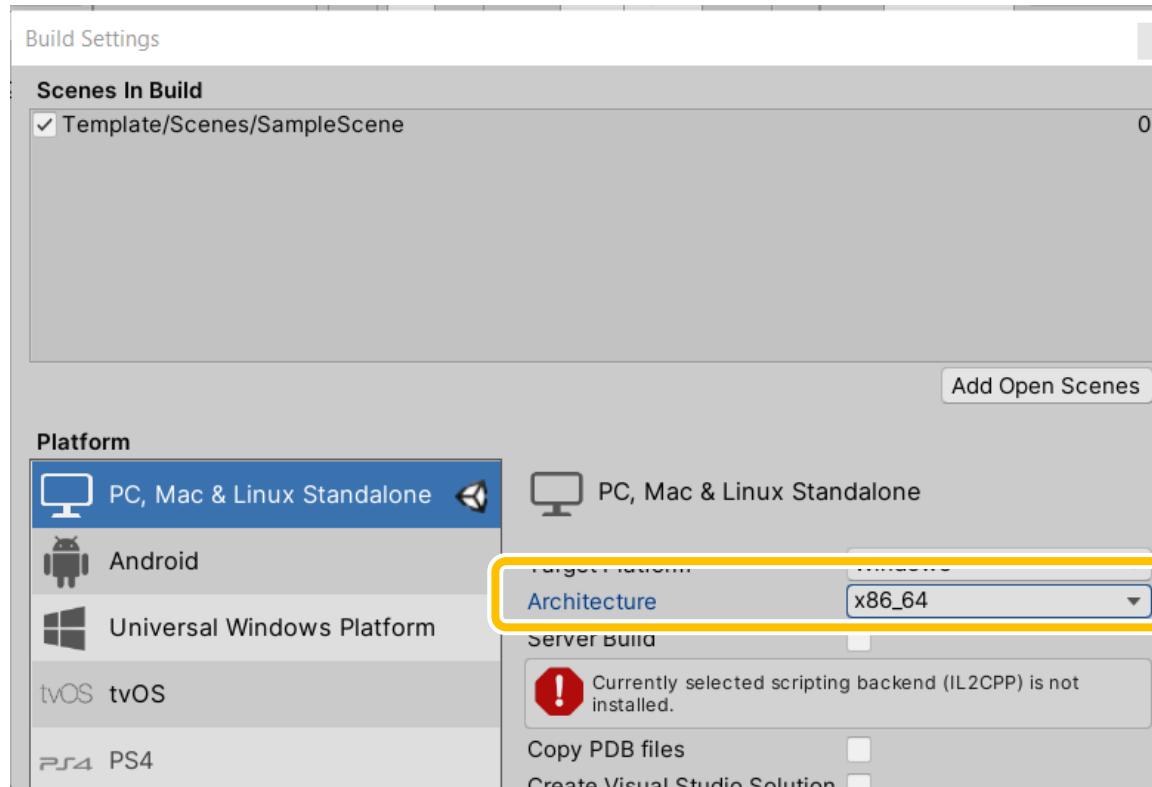
Input System 1.0.0

- Unity uses Input System Package to manage input



Open File > Build Setting

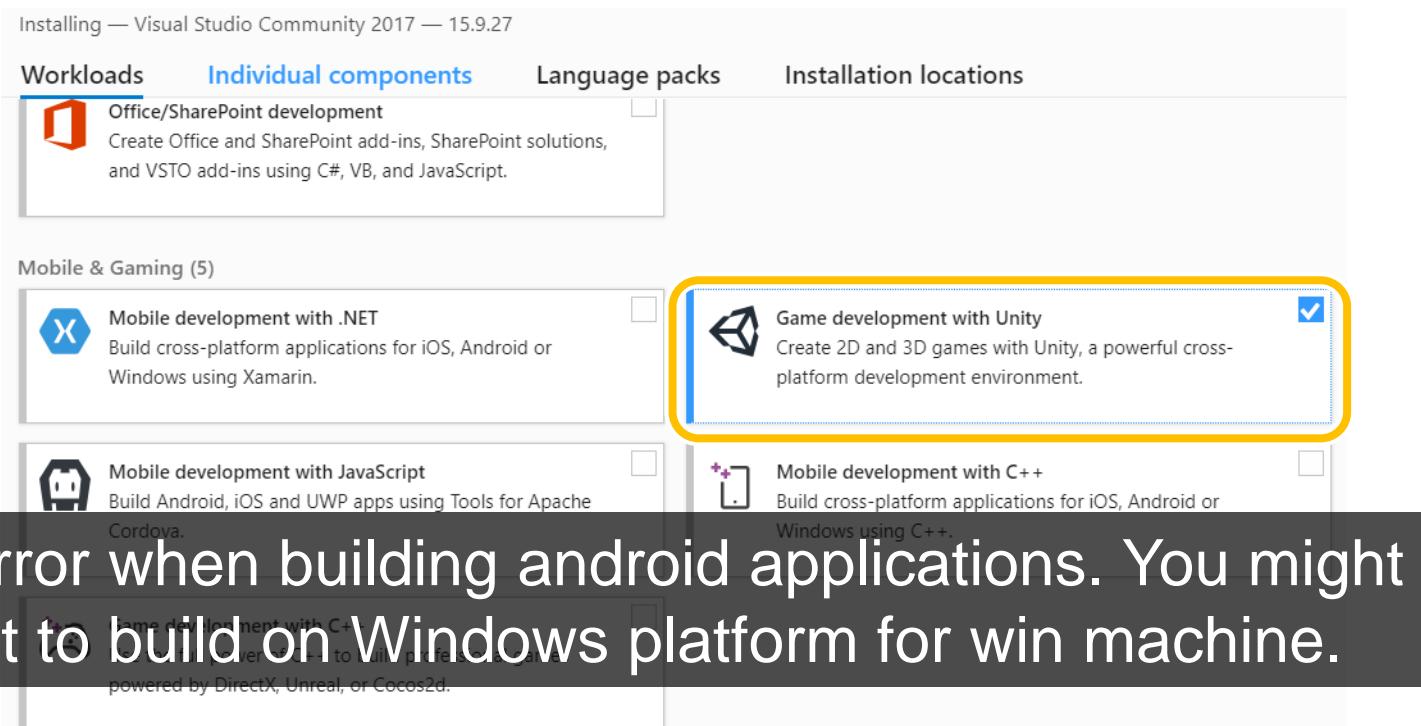
- Windows user: change architecture to x86_64



an error?

Build error

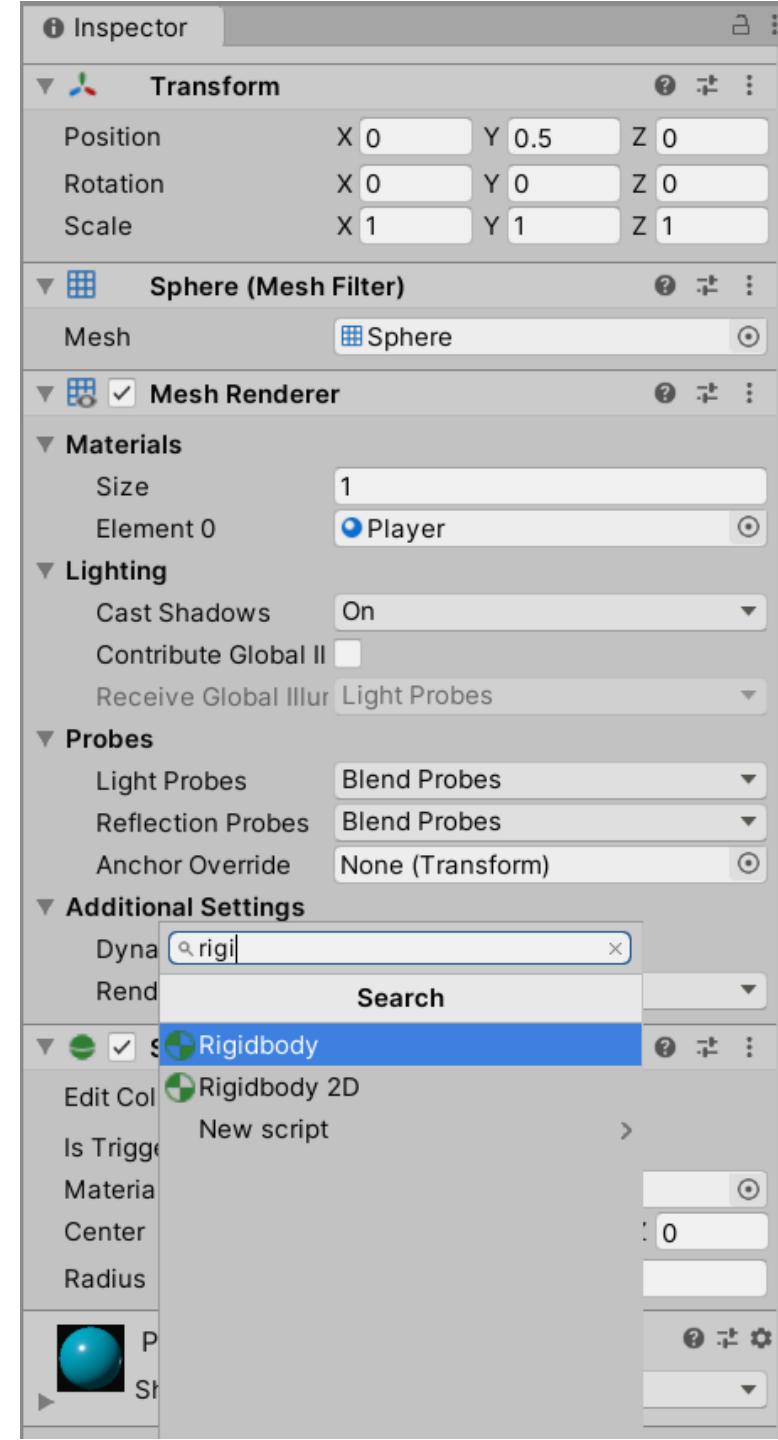
- Unity needs **Visual Studio 2017** and **Win 10 SDK** for building applications on Windows platform.
- <https://my.visualstudio.com/Downloads?q=visual%20studio%202017>



I did not have this error when building android applications. You might need this if you want to build on Windows platform for win machine.

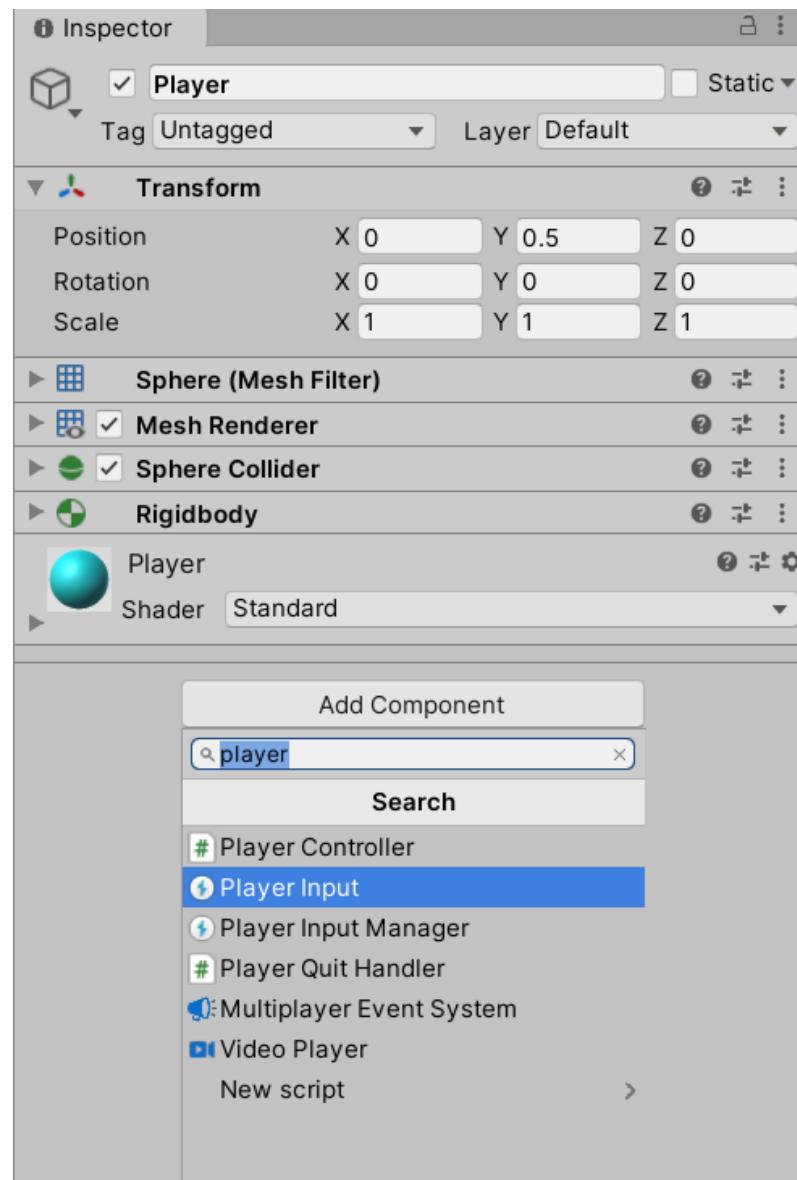
Add Component

- Select Rigidbody



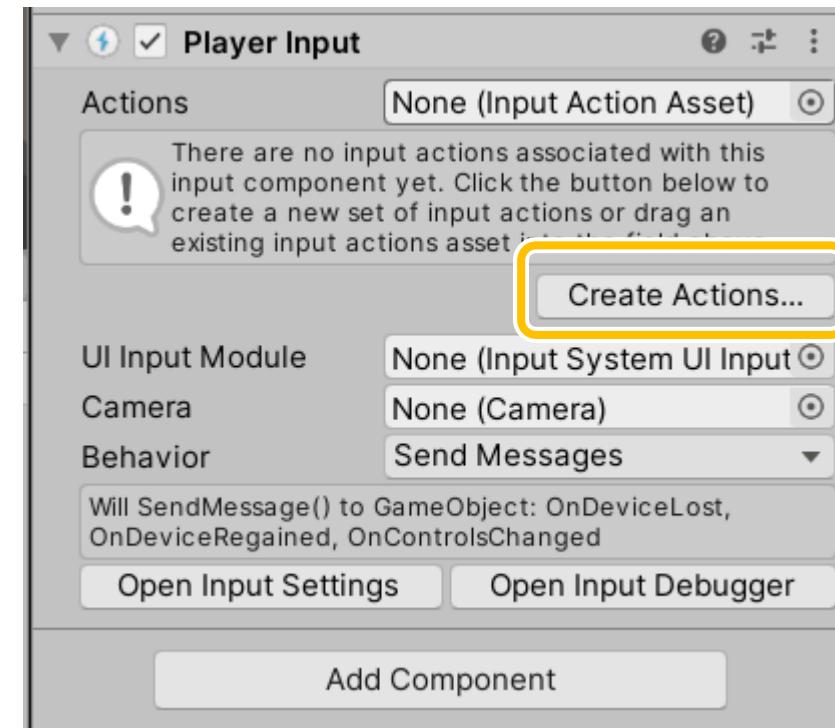
Add Component

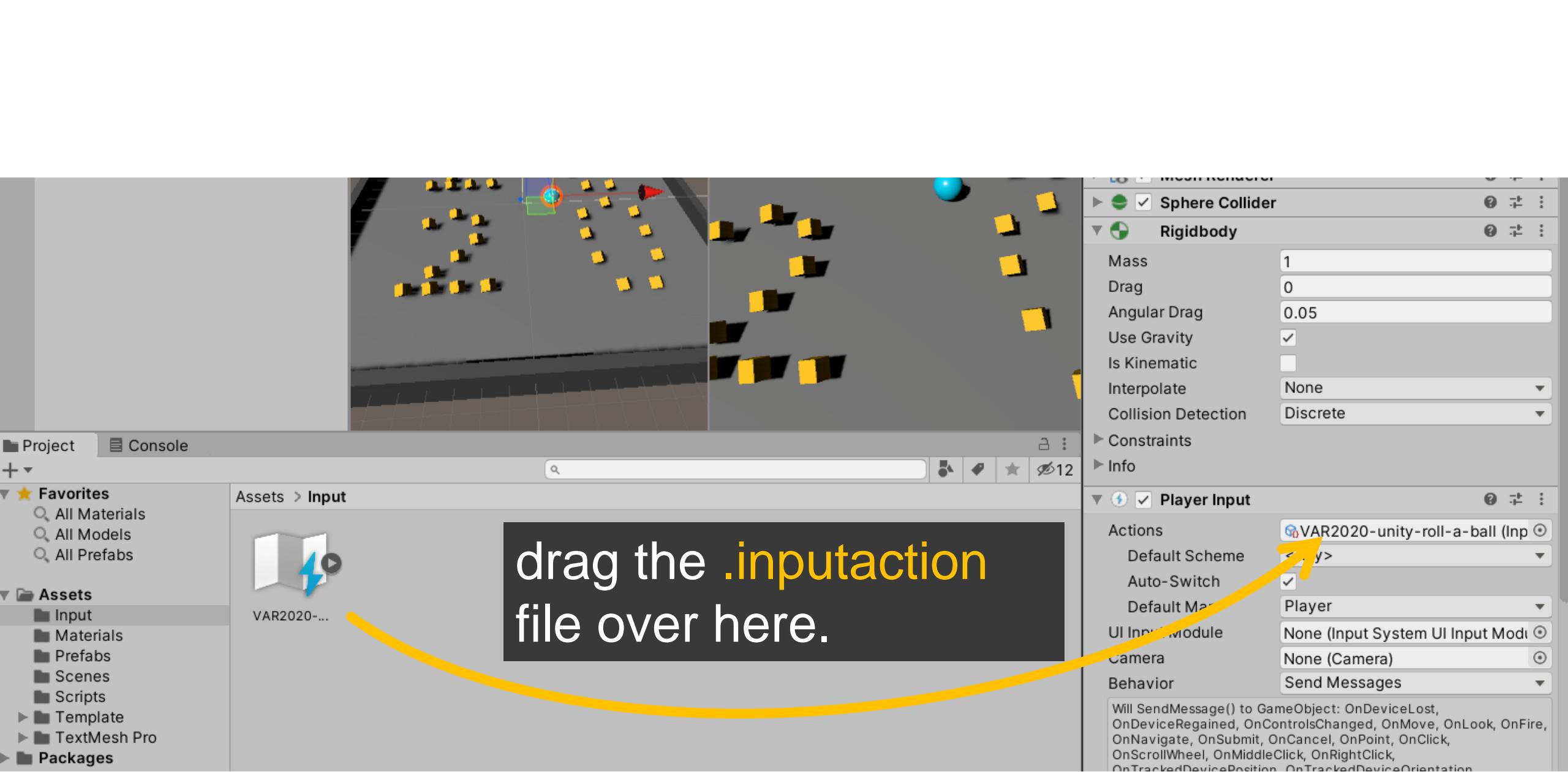
- Select Player Input



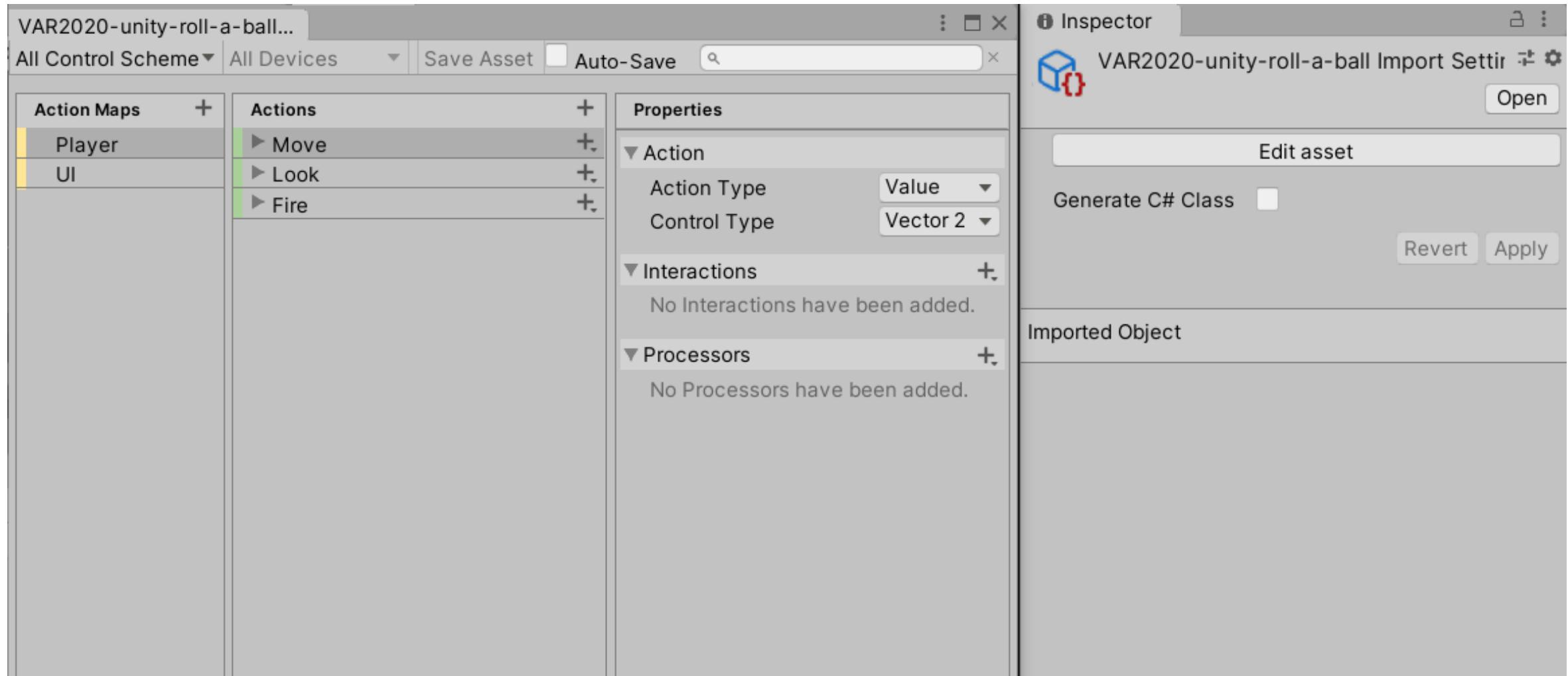
Create .inputaction file

- In the PlayerInput Component, click “Create Actions...”

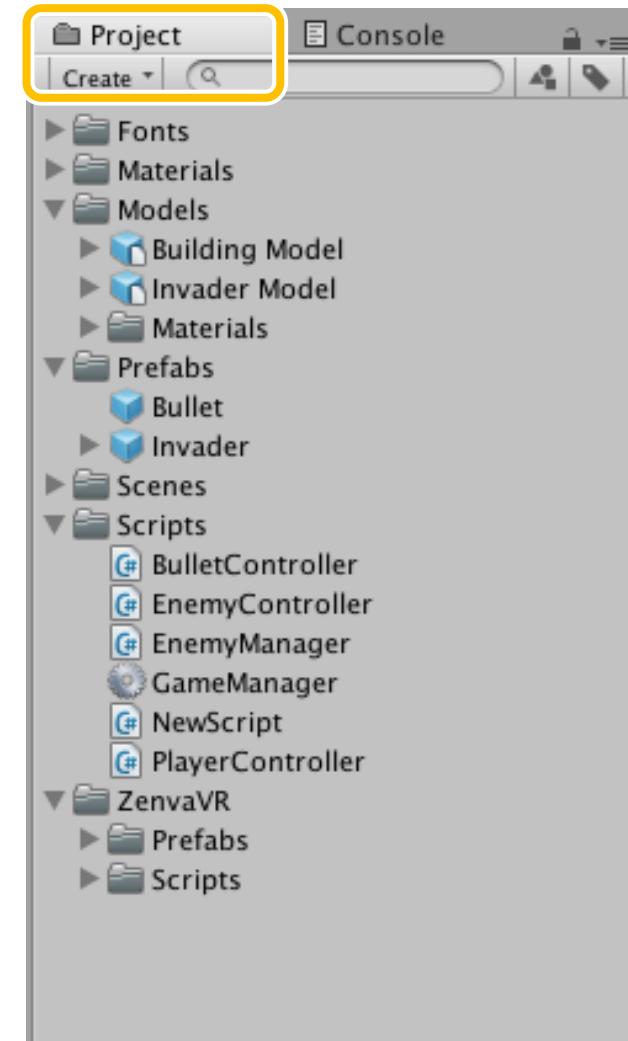
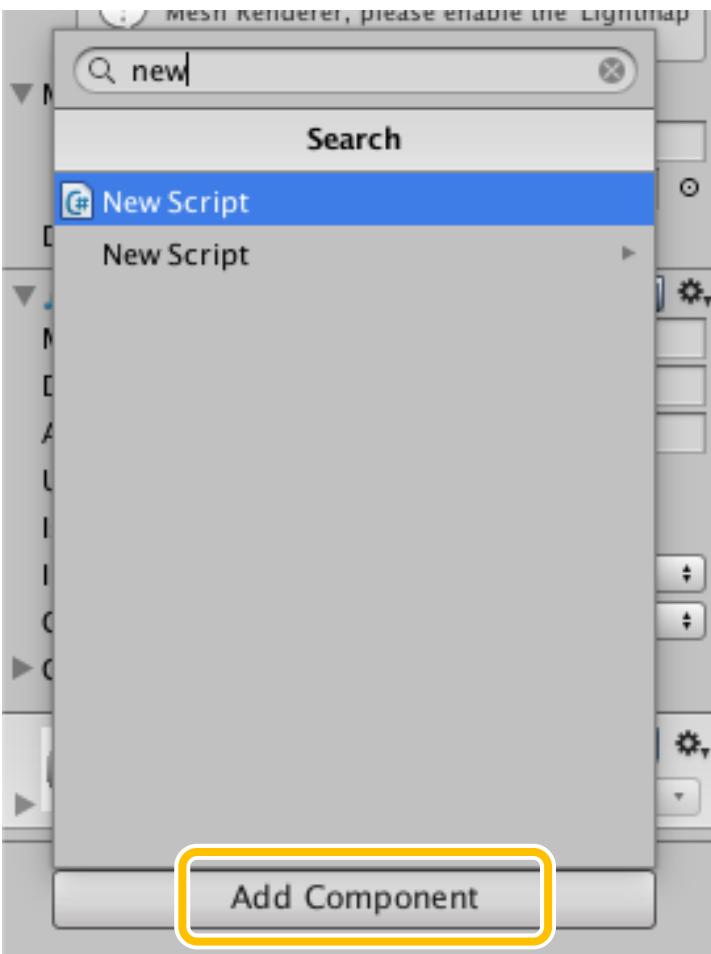




In the Inspector of an .inputaction file

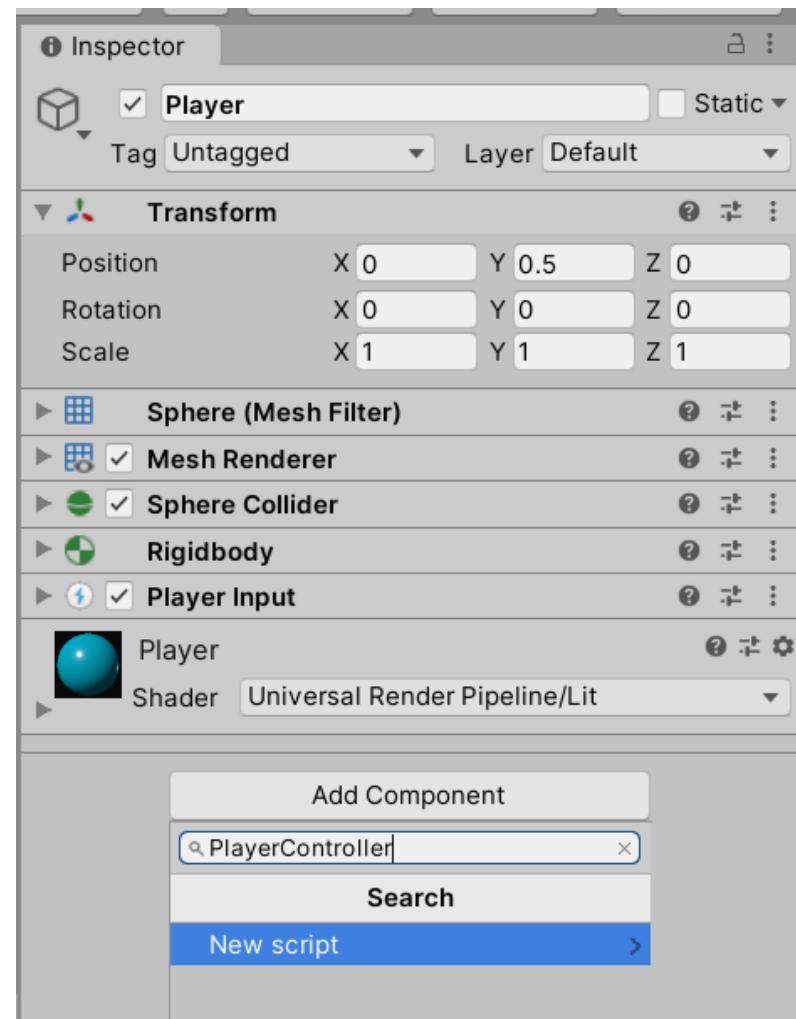
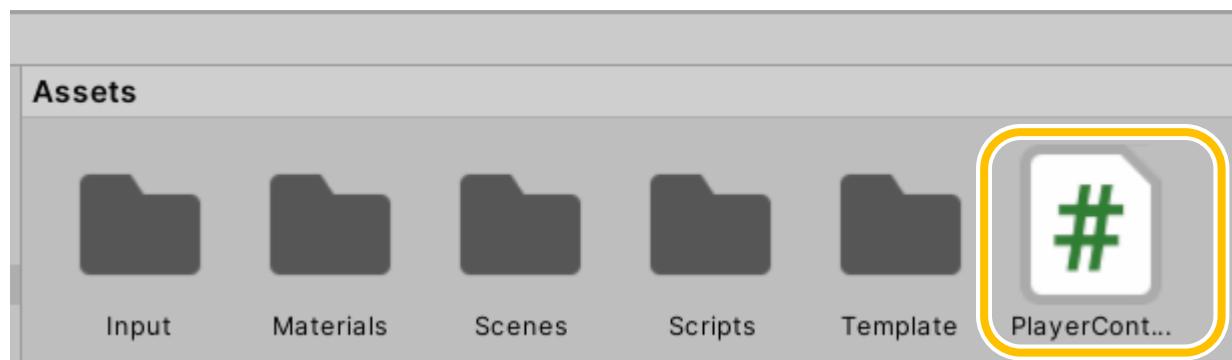


Create a new C# script

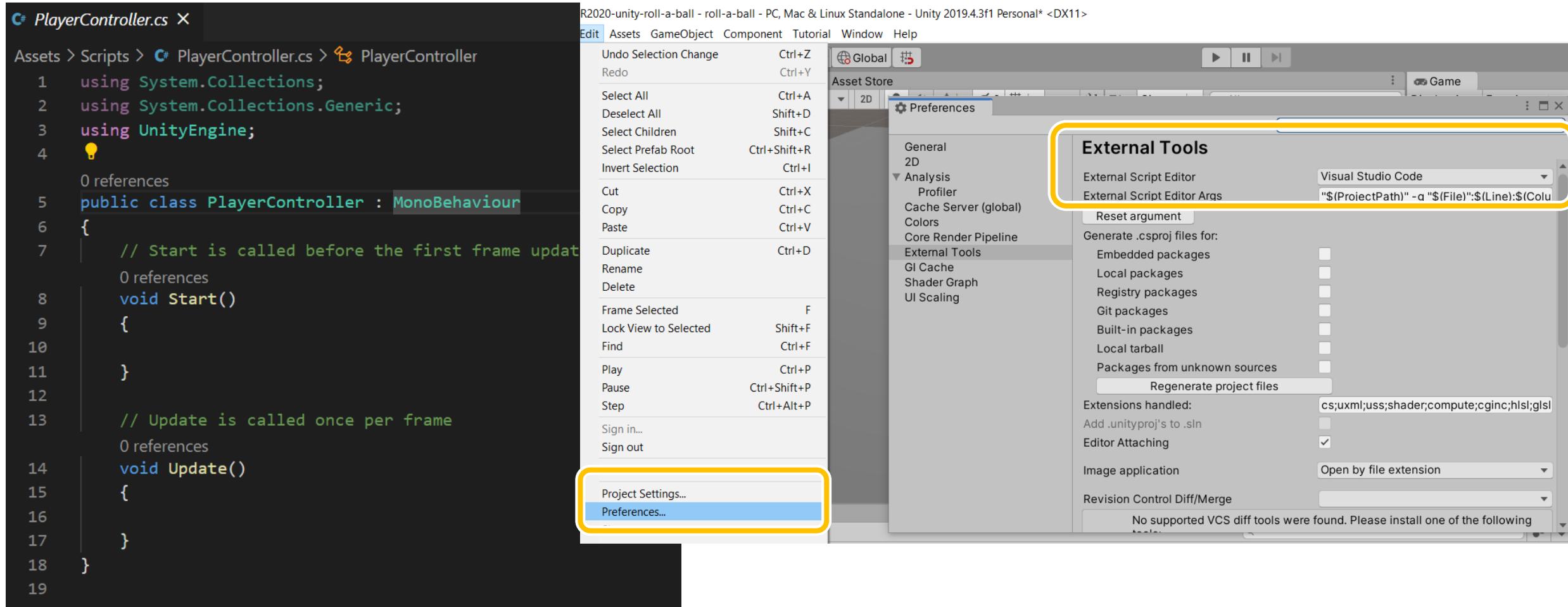


Create a new script PlayerController.cs

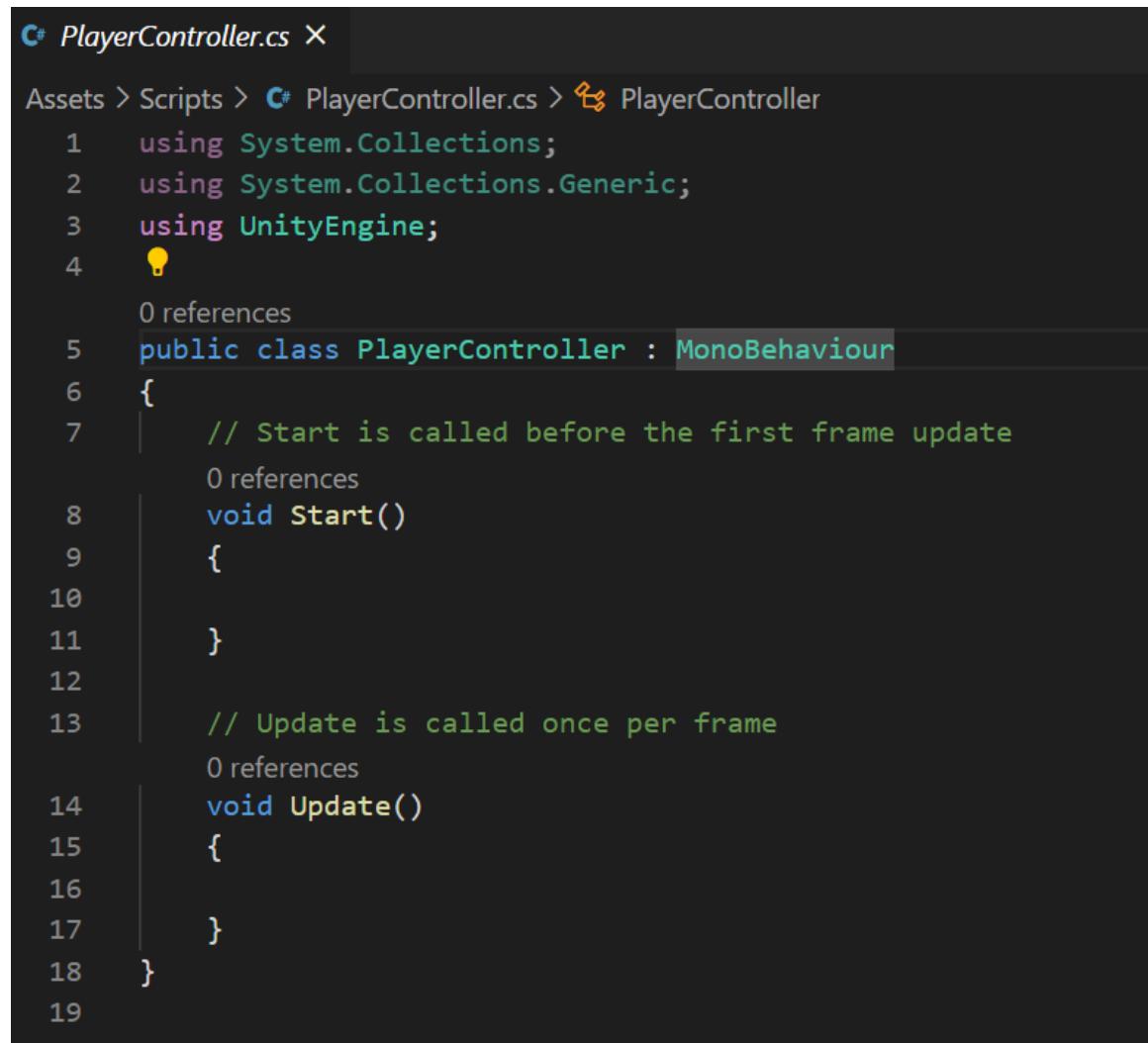
- Add Component > type PlayerController
> New script



Double click PlayerController.cs



Double click PlayerController.cs

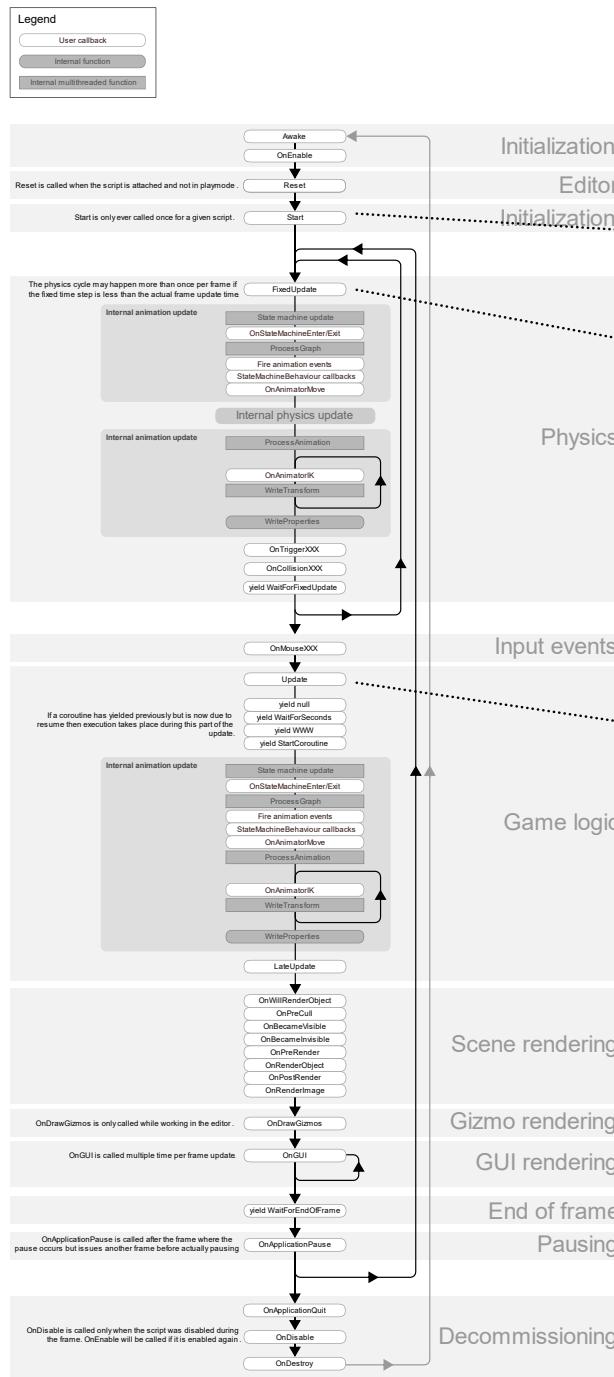


The screenshot shows a code editor window with the title "C# PlayerController.cs X". The file path is "Assets > Scripts > C# PlayerController.cs > PlayerController". The code itself is as follows:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11  }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17  }
18
19 }
```

The code is a simple Unity MonoBehaviour script with empty Start() and Update() methods.

Unity Lifecycle



Start: called once at the beginning

FixedUpdate: for physics (e.g., Rigidbody)

Update: we usually put everything here except physics

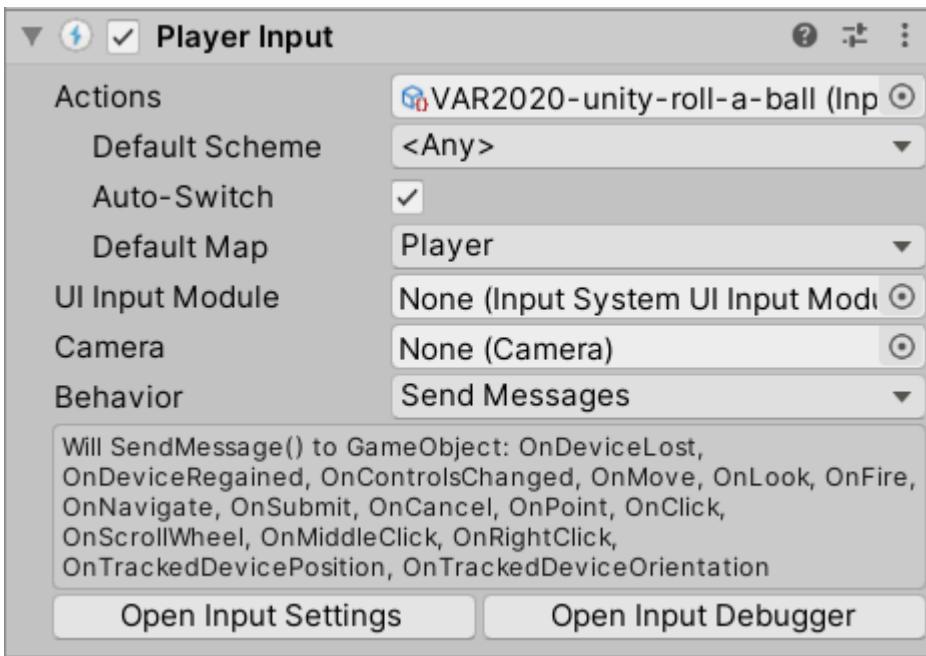
PlayerController.cs

1. reference the namespace
 2. global variables
 3. reference the rigidbody in the script

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 using TMPro;
6
7 0 references
8 public class PlayerController : MonoBehaviour
9 {
10 public float speed = 0;
11 public TextMeshProUGUI countText;
12 public GameObject winTextObject;
13 private Rigidbody rb;
14 private float movementX;
15 private float movementY;
16 private int count;
17
18 0 references
19 void Start()
20 {
21 rb = this.GetComponent<Rigidbody>();
22 count = 0;
23 SetCountText();
24 winTextObject.SetActive(false);
25 }
26 }

PlayerController.cs

- OnMove(): access input value (here are the arrow keys)

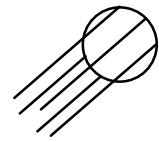


```
24
25     0 references
26     void Update()
27     {
28     }
29
30     0 references
31     void OnMove(InputValue movementValue)
32     {
33         Vector2 movementVector = movementValue.Get<Vector2>();
34
35         movementX = movementVector.x;
36         movementY = movementVector.y;
37     }
38
39     0 references
40     private void FixedUpdate()
41     {
42         Vector3 movement = new Vector3(movementX, 0.0f, movementY);
43         rb.AddForce(movement * speed);
44     }
45 }
```

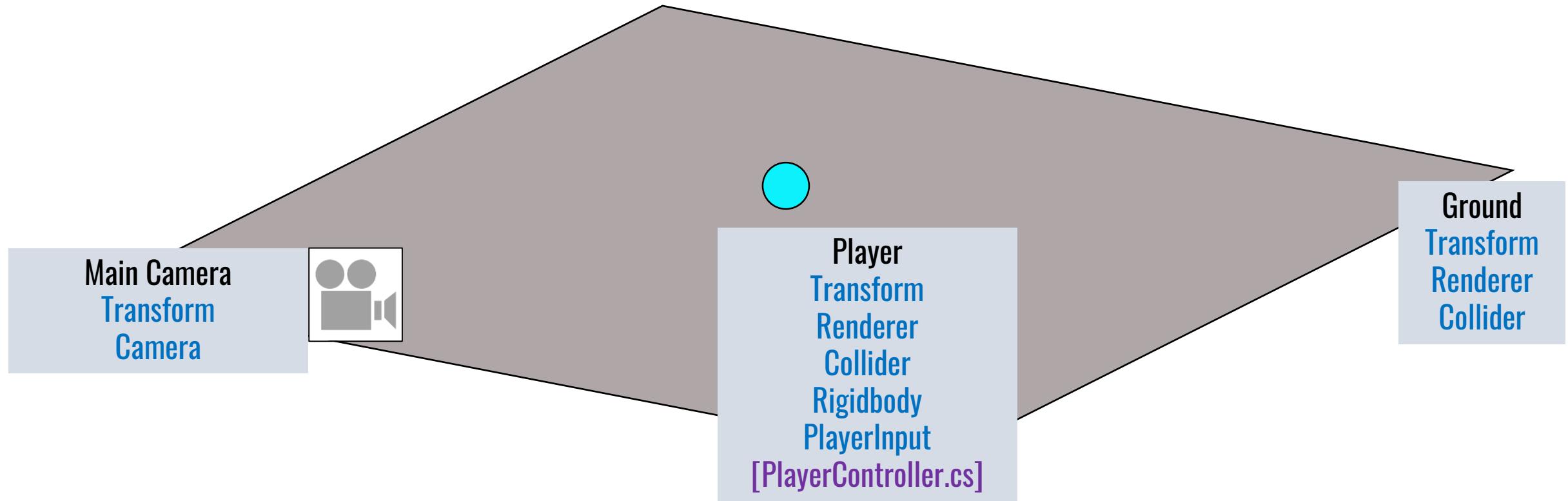
PlayerController.cs

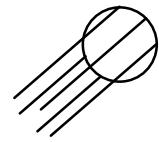
- FixedUpdate(): move the Player using .AddForce

```
24          0 references
25      void Update()
26      {
27
28      }
29
30          0 references
31      void OnMove(InputValue movementValue)
32      {
33          Vector2 movementVector = movementValue.Get<Vector2>();
34
35          movementX = movementVector.x;
36          movementY = movementVector.y;
37      }
38
39          0 references
40      private void FixedUpdate()
41      {
42          Vector3 movement = new Vector3(movementX, 0.0f, movementY);
43
44          rb.AddForce(movement * speed);
45      }
```

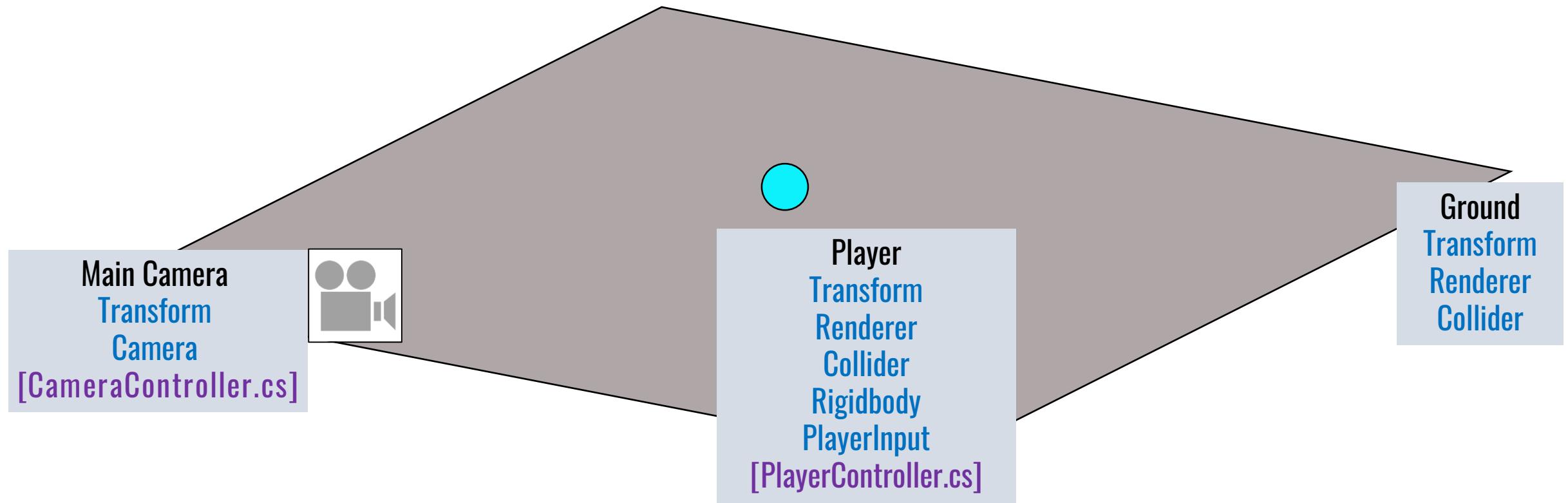


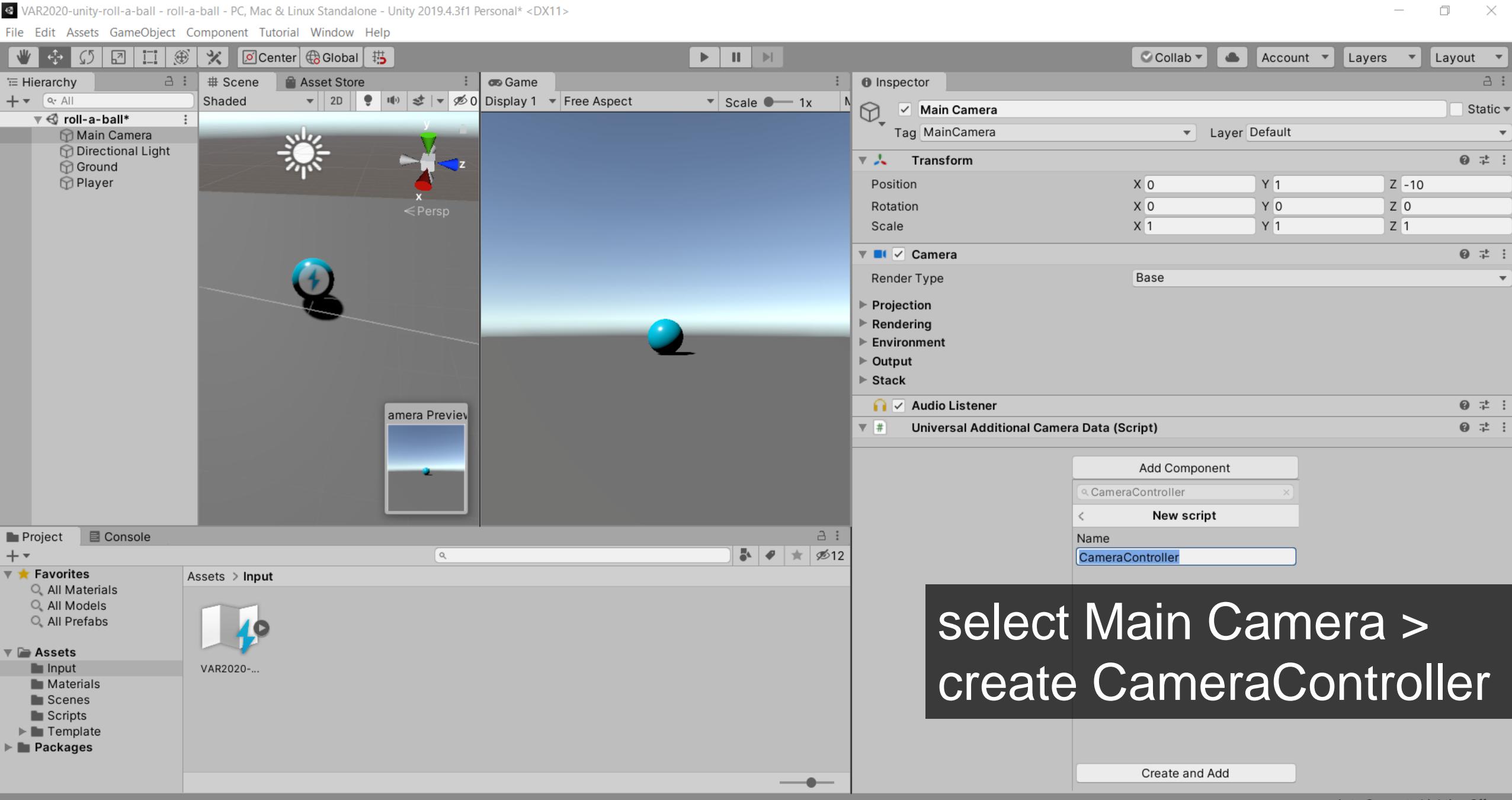
Directional Light
Transform
Light





Directional Light
Transform
Light





select Main Camera >
create CameraController

CameraController.cs

C# PlayerController.cs

C# CameraController.cs X

Release Notes: 1.49.1

Assets > Scripts > CameraController.cs > CameraController

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  0 references
6  public class CameraController : MonoBehaviour
7  {
8      2 references
9      public GameObject player;
10
11     2 references
12     private Vector3 offset;
13
14     // Start is called before the first frame update
15     0 references
16     void Start()
17     {
18         offset = transform.position - player.transform.position;
19     }
20
21     // Update is called once per frame
22     0 references
23     void LateUpdate()
24     {
25         transform.position = player.transform.position + offset;
26     }
27 }
```

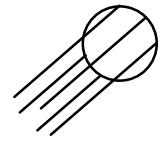
Inspector of Main Camera

1. Adjust the Transform for viewing angle of the game.
2. Drag the Player GameObject in the reference of CameraController.cs

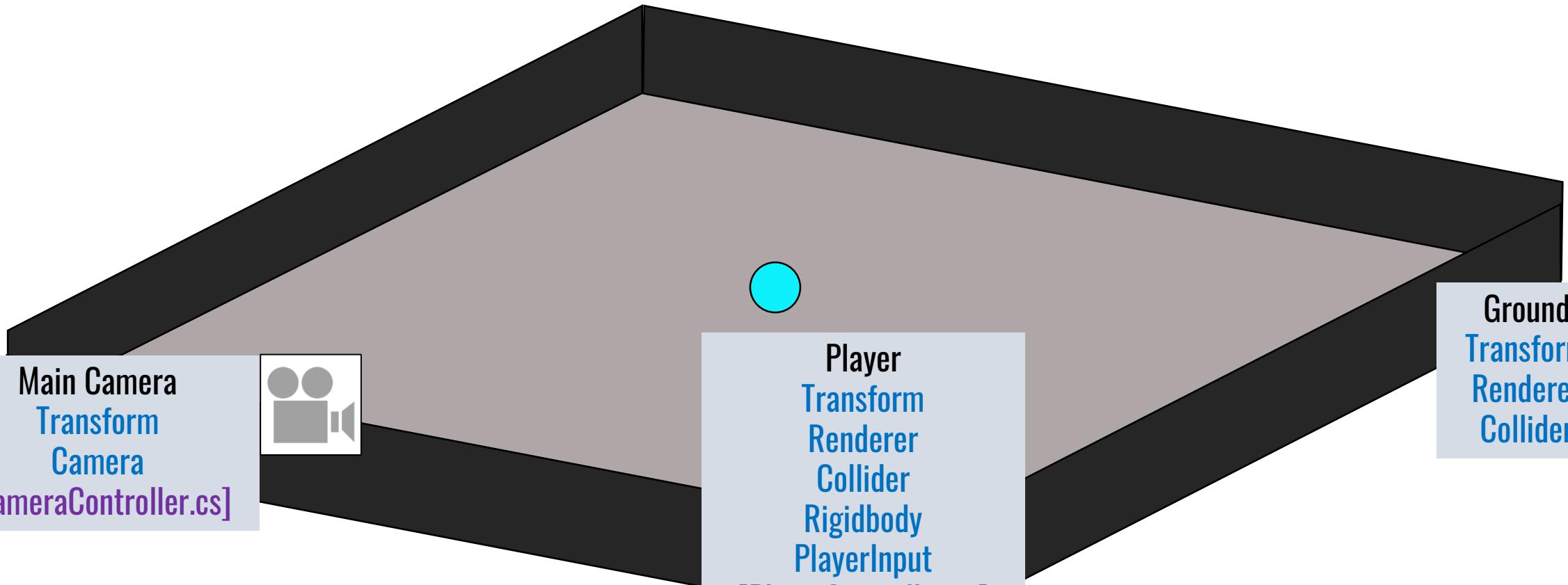




Wall
Transform
Child: Wall * 4

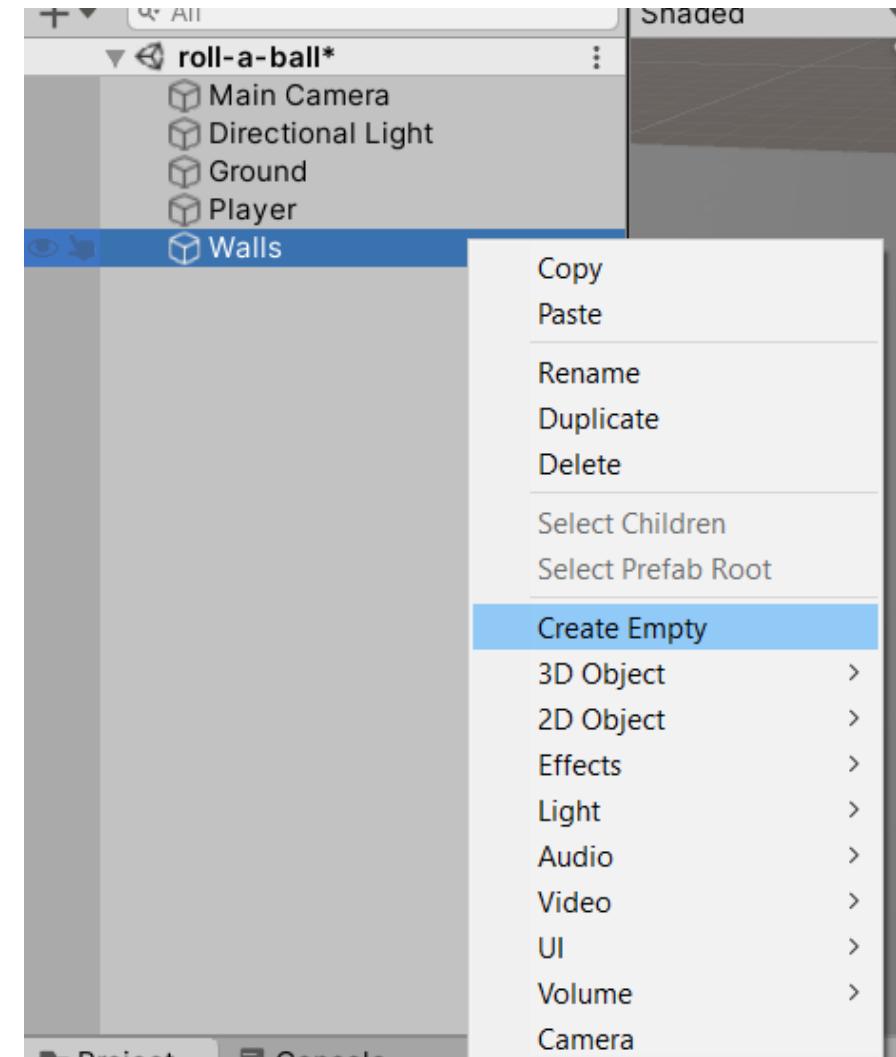


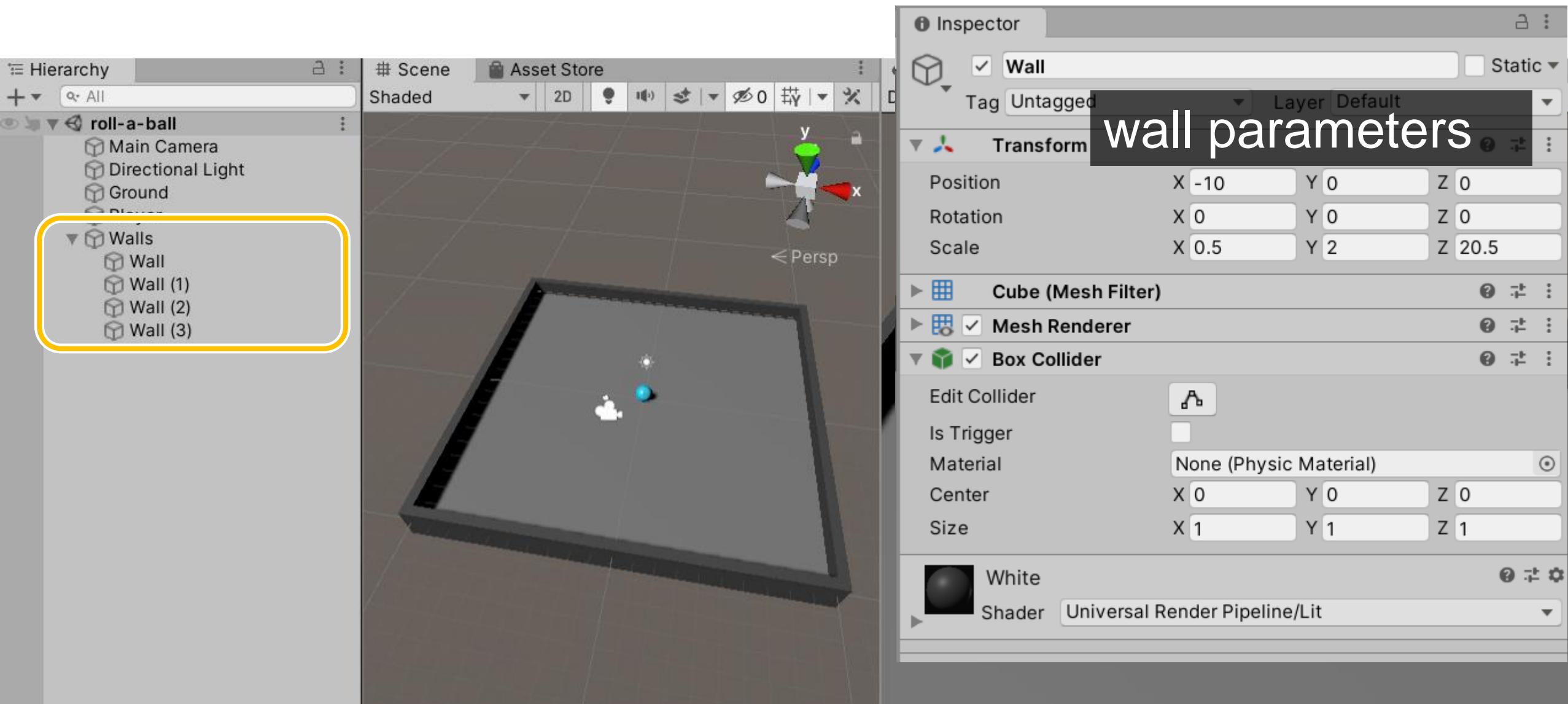
Directional Light
Transform
Light



create Empty GameObject

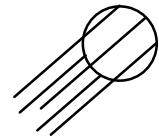
- We use empty GO to collect things together (e.g., walls, environment).
- Therefore, you can manipulate them all at the same time.



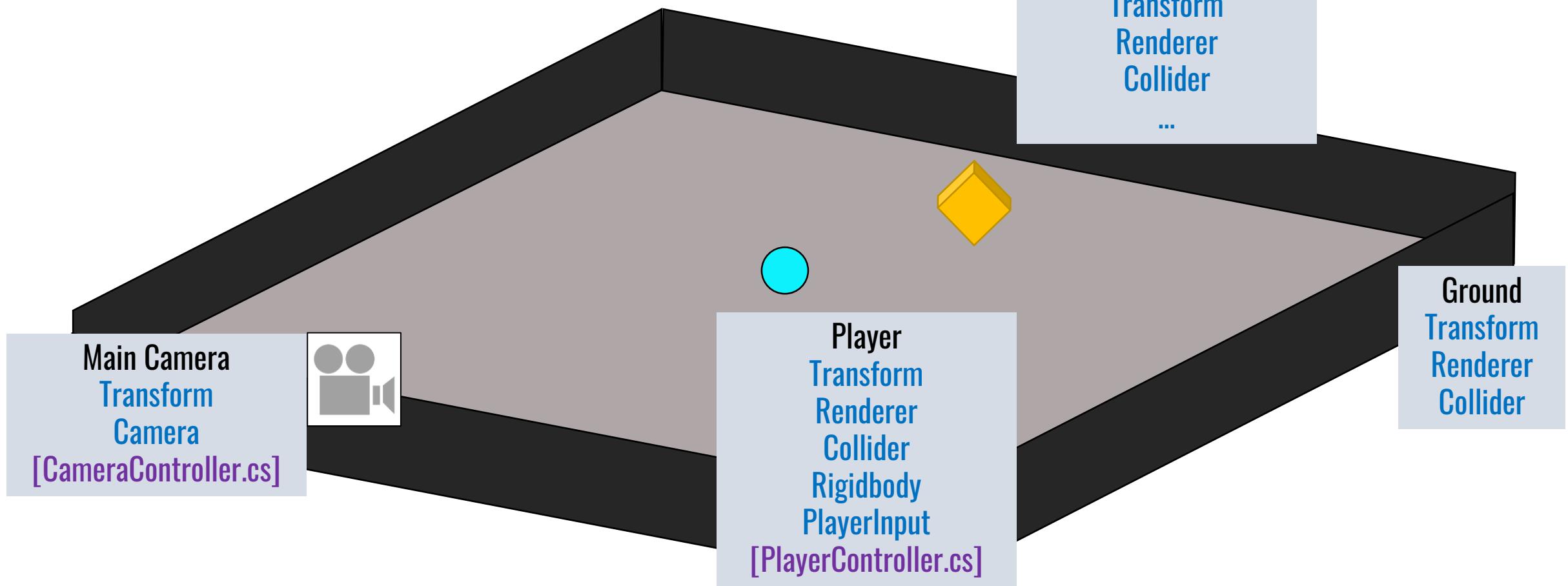




Wall
Transform
Child: Wall * 4

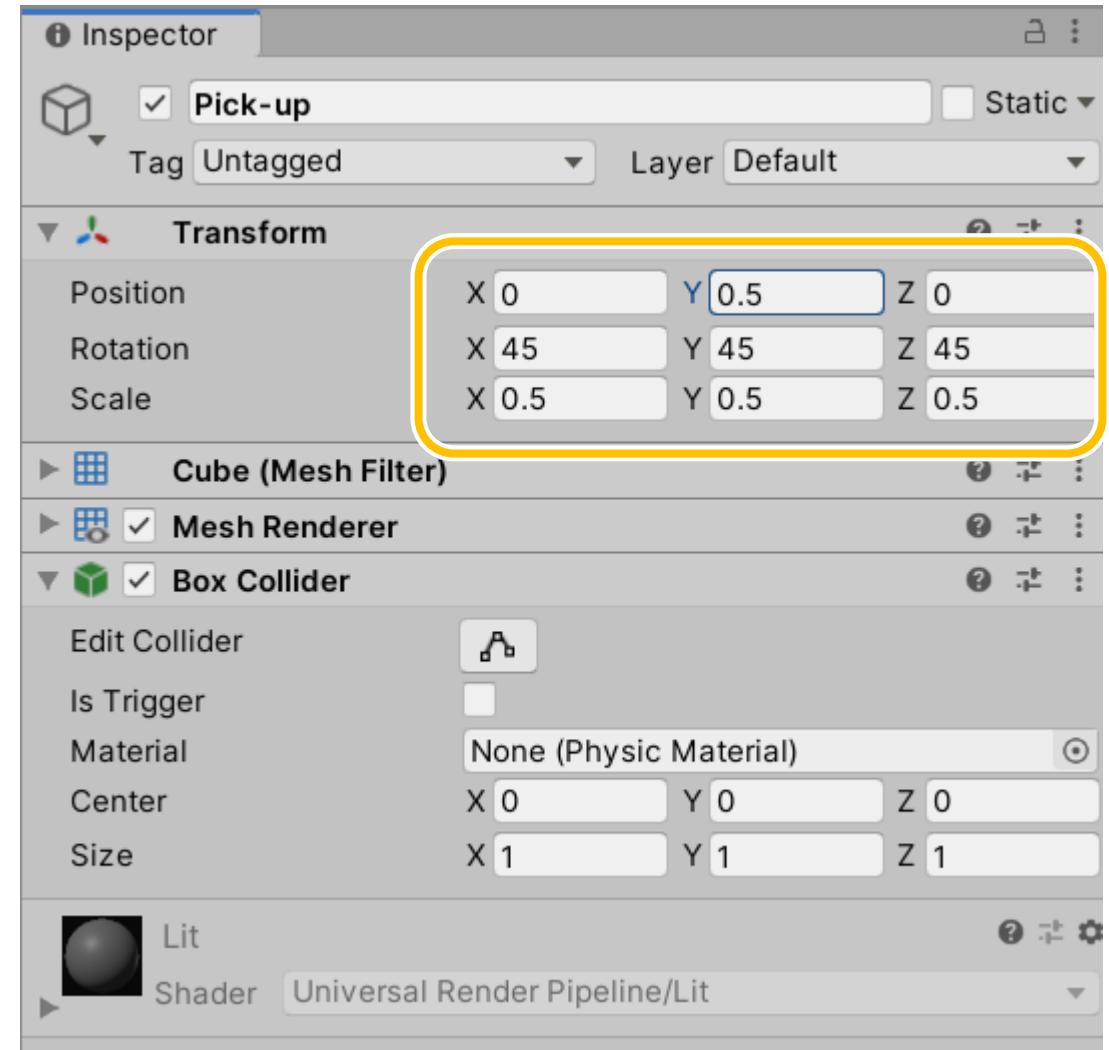


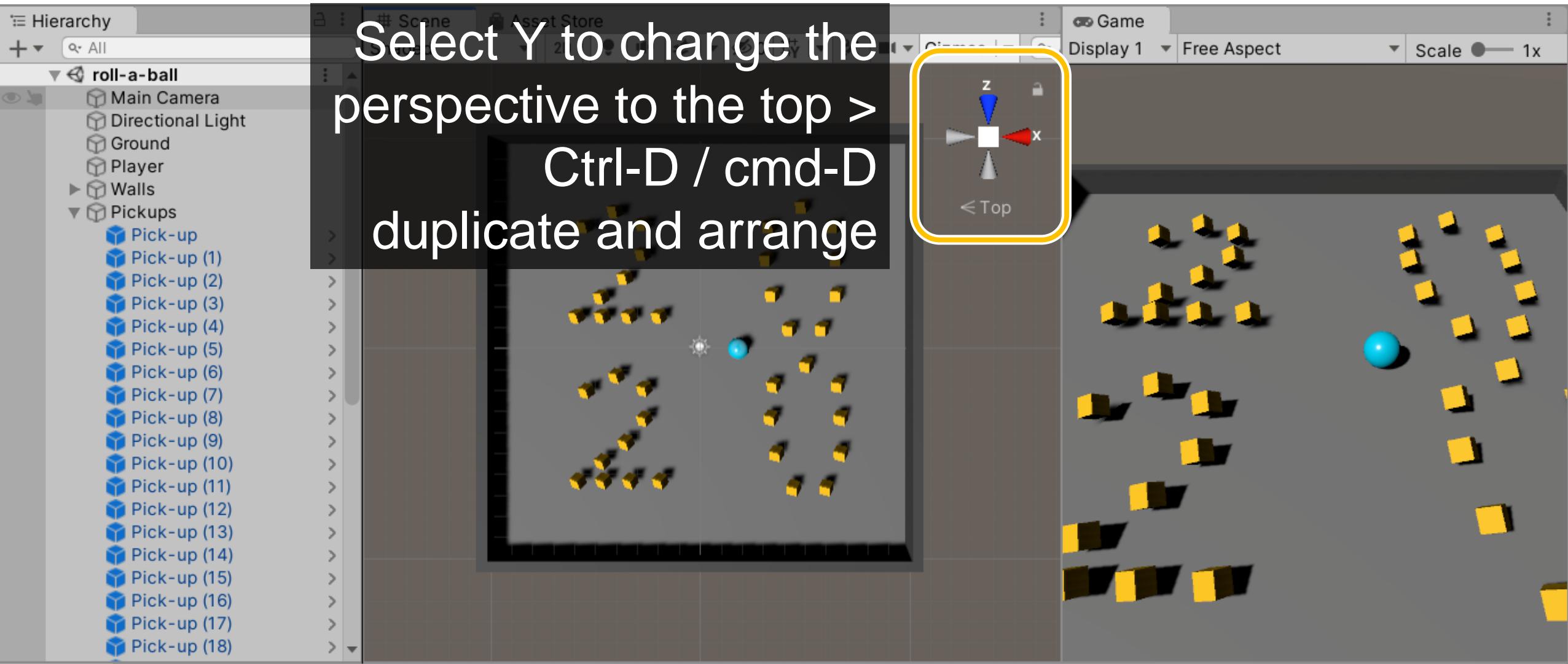
Directional Light
Transform
Light

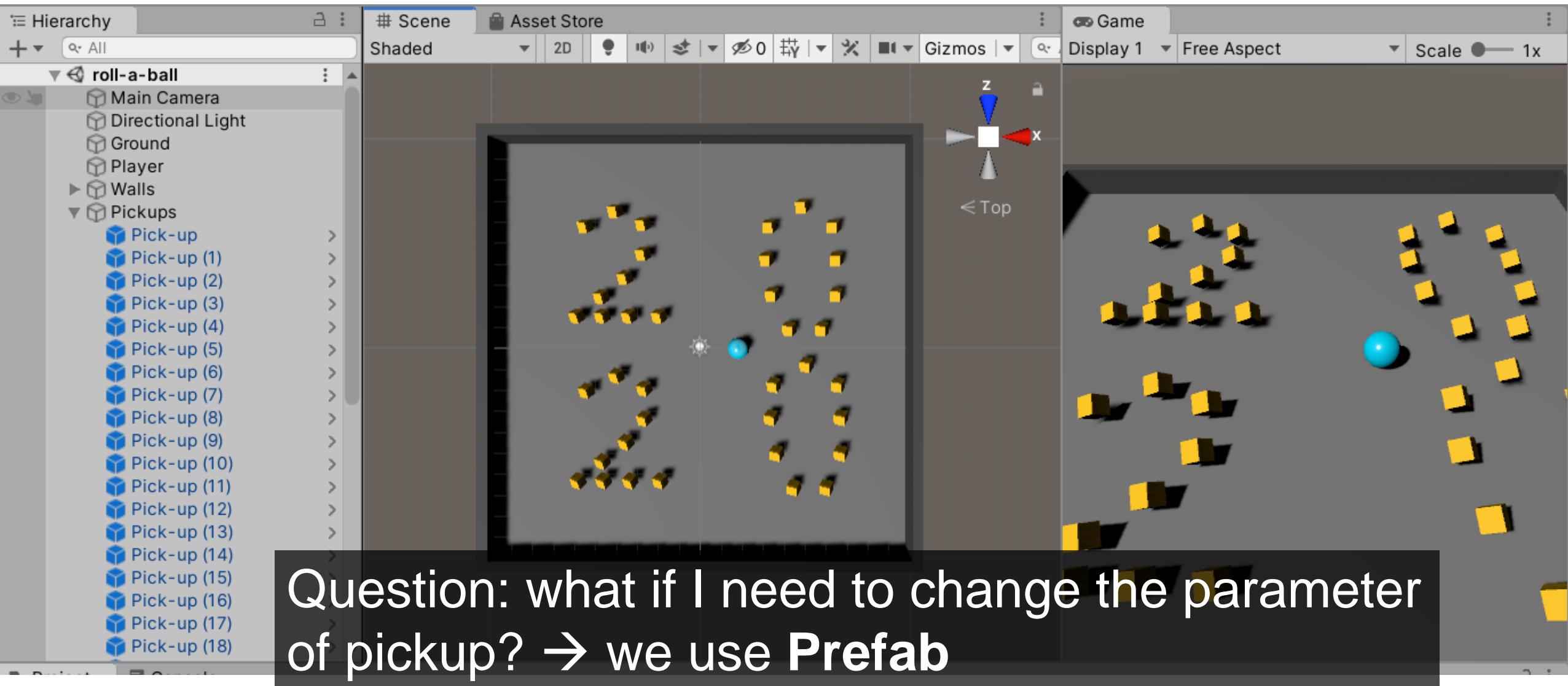


The pick-up

- Create a cube and name as pick-up
- Change Transform
- Create a material for the pick-up

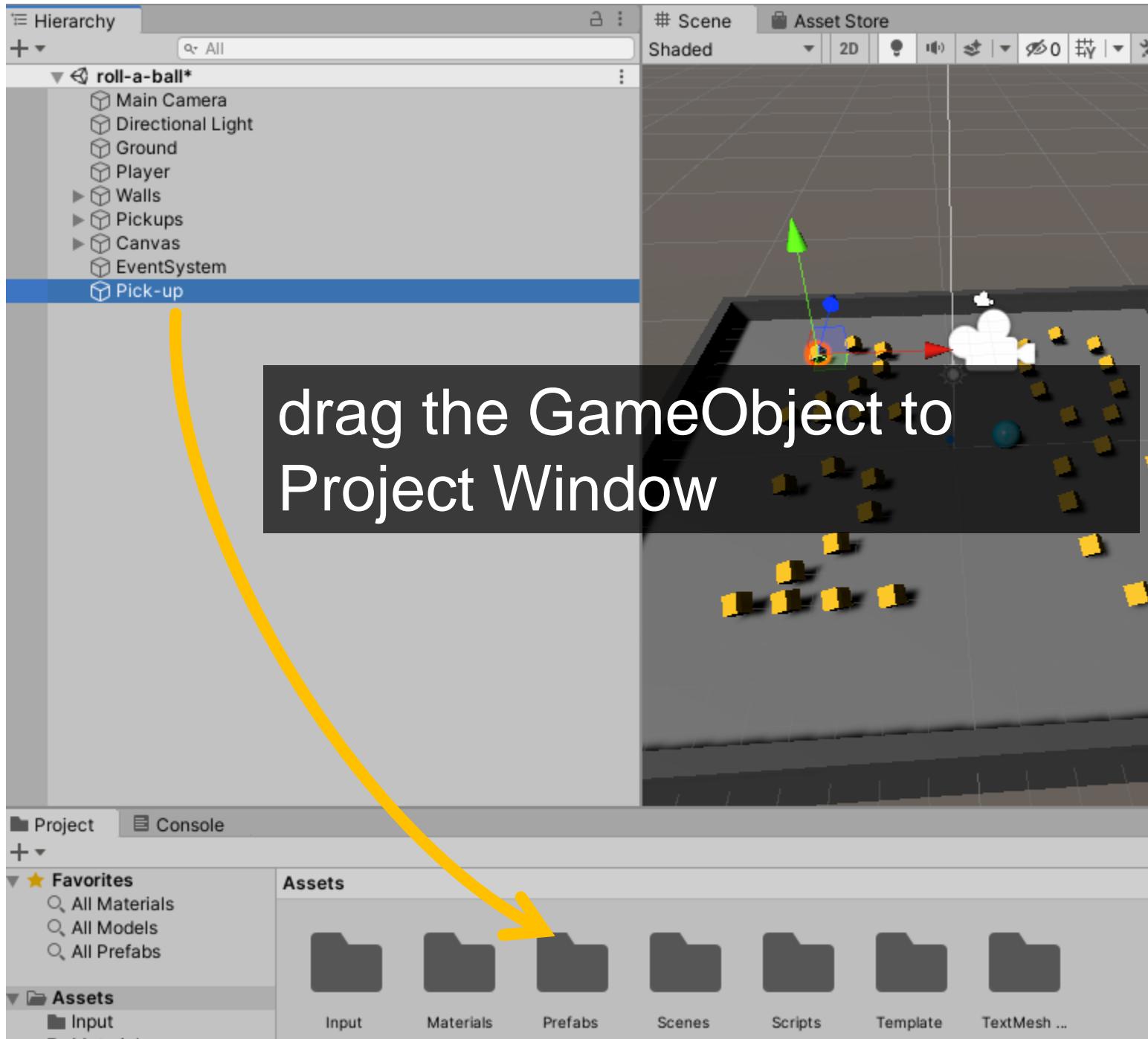






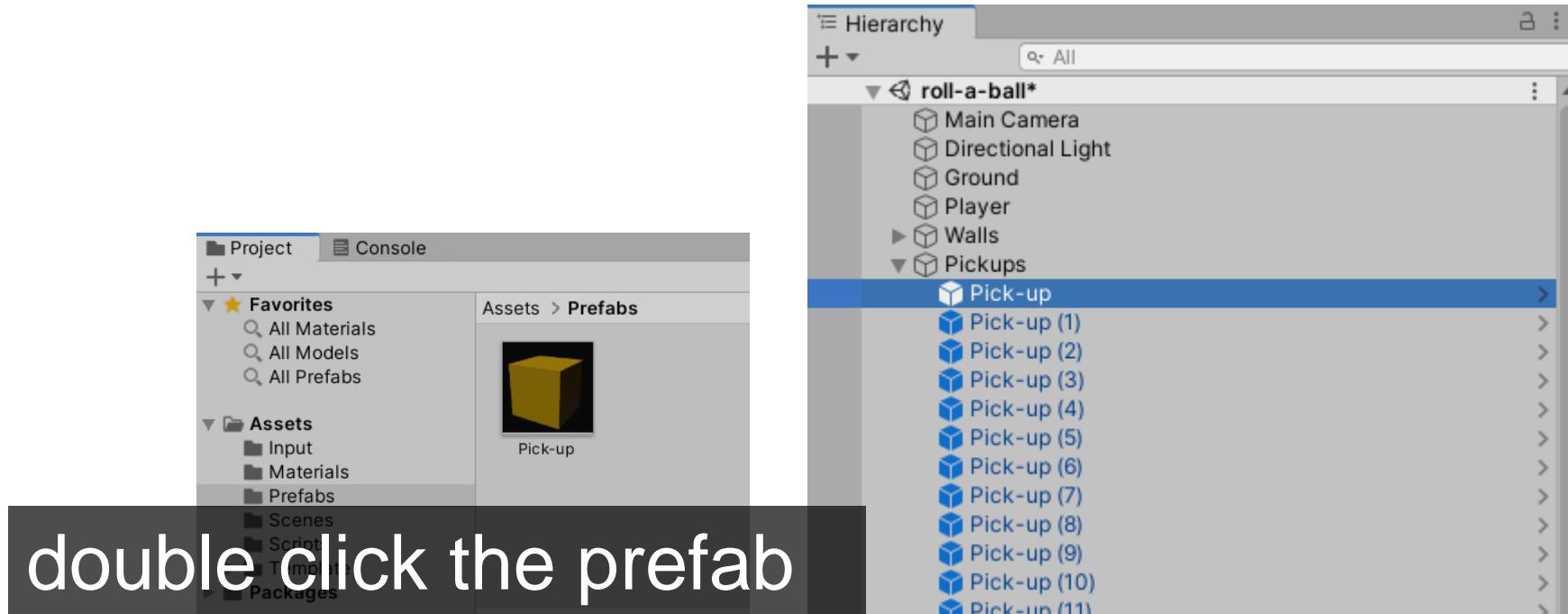
Question: what if I need to change the parameter of pickup? → we use **Prefab**

create Prefab



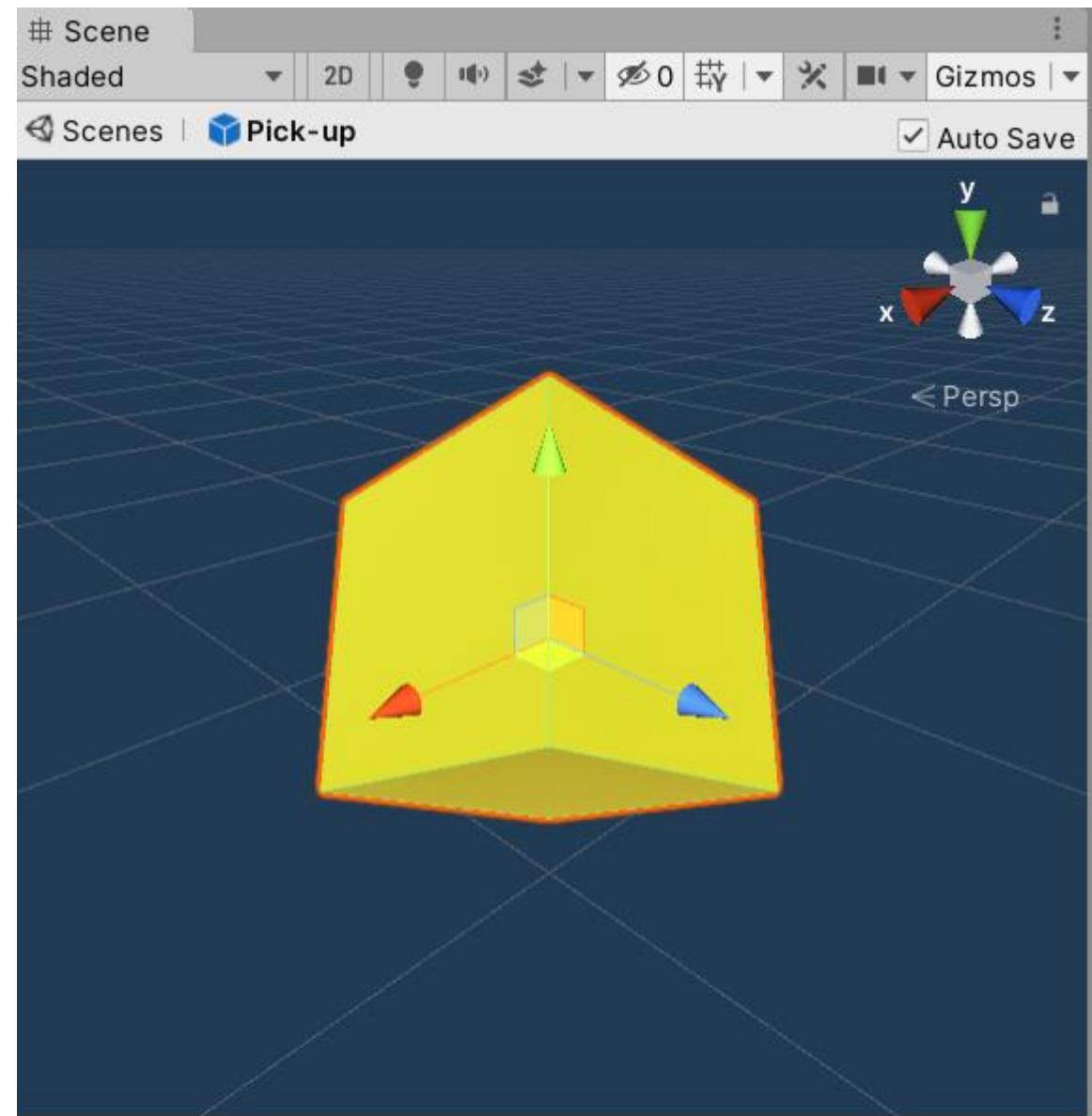
create Prefab

- The GameObject in hierarchy turns blue.



Prefab editing mode

- The changes you made in this mode will be passing to all the prefab gameobjects in the scene.

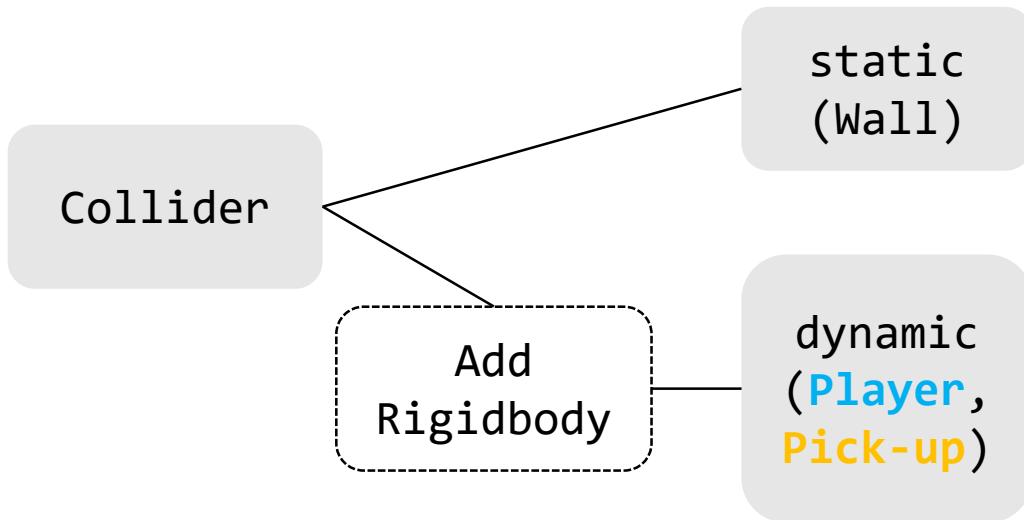


Our goal in this game:

- when **Player** hits **Pick-up**, the **Pick-up** disappears and increase the score.
- if score > X, win.

Let's have a look in our game

- Unity Colliders



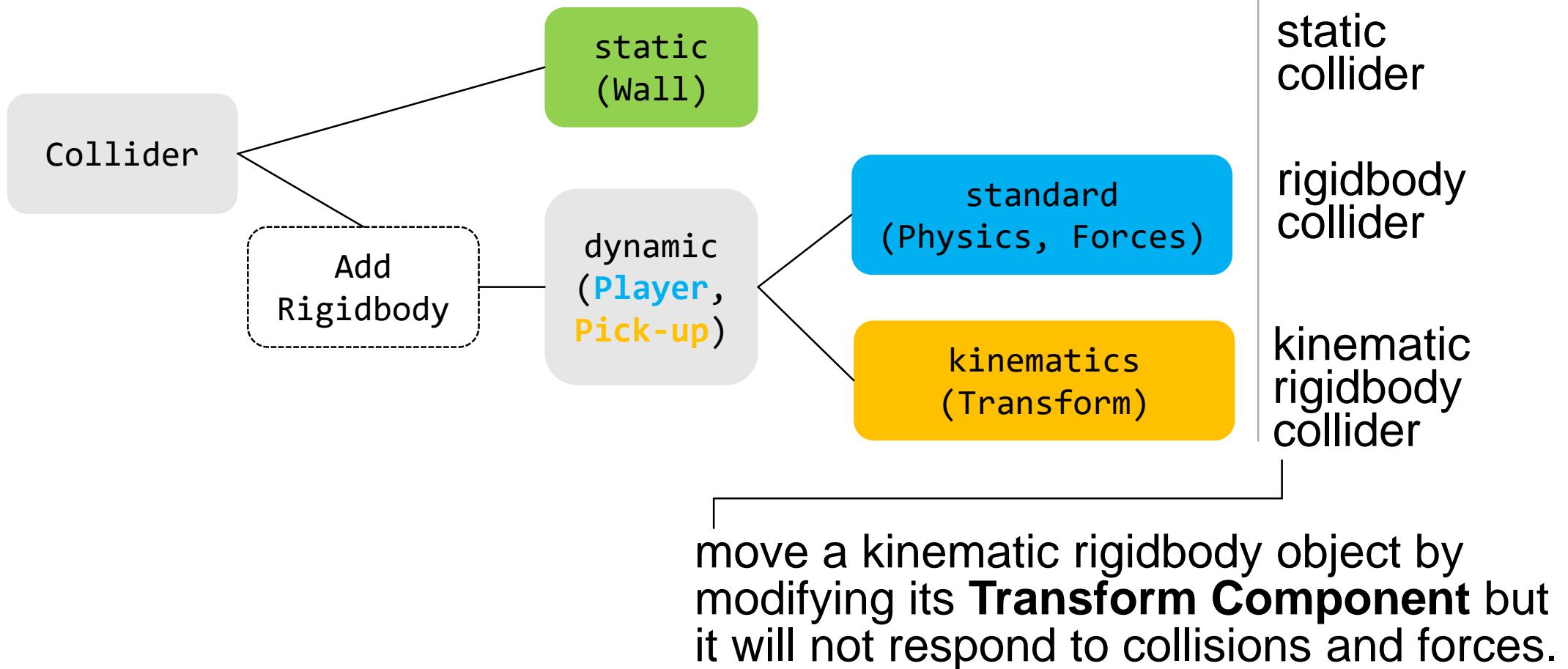
detect collision:

- OnCollisionEnter()
 - OnTriggerEnter()
- detect when one collider enters
the space of another without
creating a collision

In our example:
`Player` has `OnTriggerEnter`
`Pick-up` is triggered

Let's have a look in our game

- Unity Colliders



Unity Colliders

Collision detection occurs and messages are sent upon collision

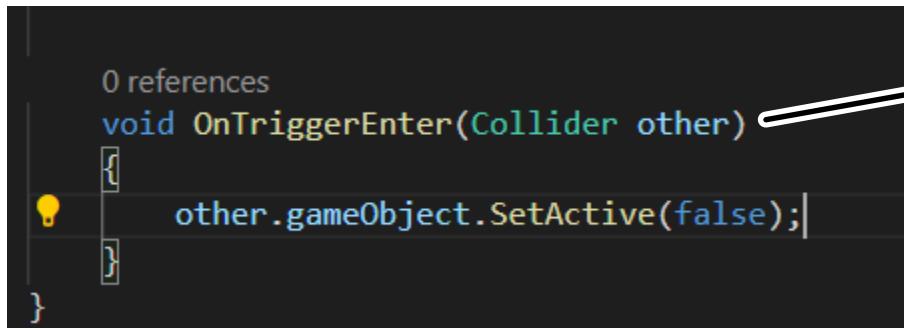
	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		Y				
Rigidbody Collider	Y	Y	Y			
Kinematic Rigidbody Collider		Y				
Static Trigger Collider						
Rigidbody Trigger Collider						
Kinematic Rigidbody Trigger Collider						

Trigger messages are sent upon collision

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider					Y	Y
Rigidbody Collider				Y	Y	Y
Kinematic Rigidbody Collider				Y	Y	Y
Static Trigger Collider		Y	Y		Y	Y
Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y

back to PlayerController.cs

- Unity Colliders



```
0 references
void OnTriggerEnter(Collider other)
{
    other.gameObject.SetActive(false);
}
```

A screenshot of a Unity code editor showing a C# script. The script contains a single method: void OnTriggerEnter(Collider other). Inside the method, there is a line of code: other.gameObject.SetActive(false);. A yellow lightbulb icon is visible on the left side of the code editor.

means Pick-up

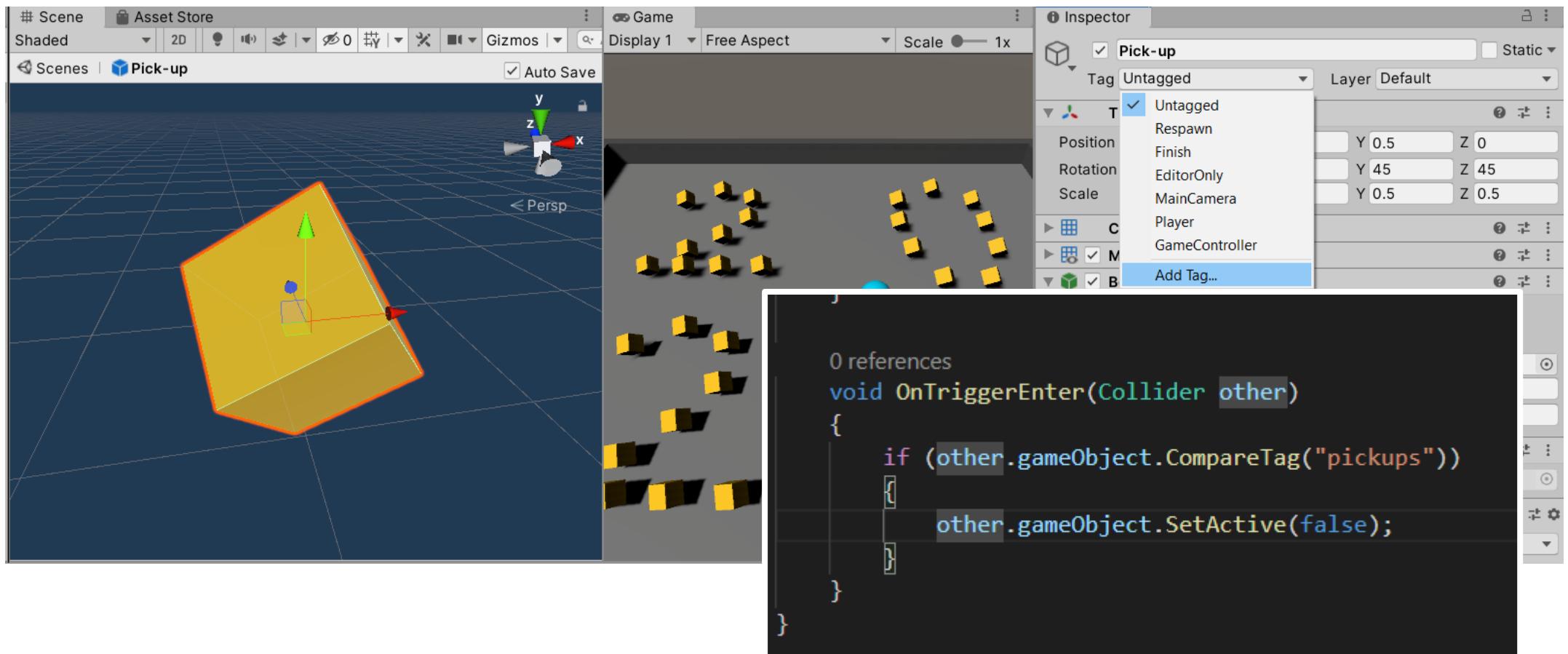
Trigger messages are sent upon collision						
	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider					Y	Y
Rigidbody Collider				Y	Y	Y
Kinematic Rigidbody Collider				Y	Y	Y
Static Trigger Collider		Y	Y		Y	Y
Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider	Y		Y	Y	Y	Y

Player

Pick-up

Add tag

- In the inspector of pick-up prefab, select Add tag and add “pickups”.



Rotator.cs

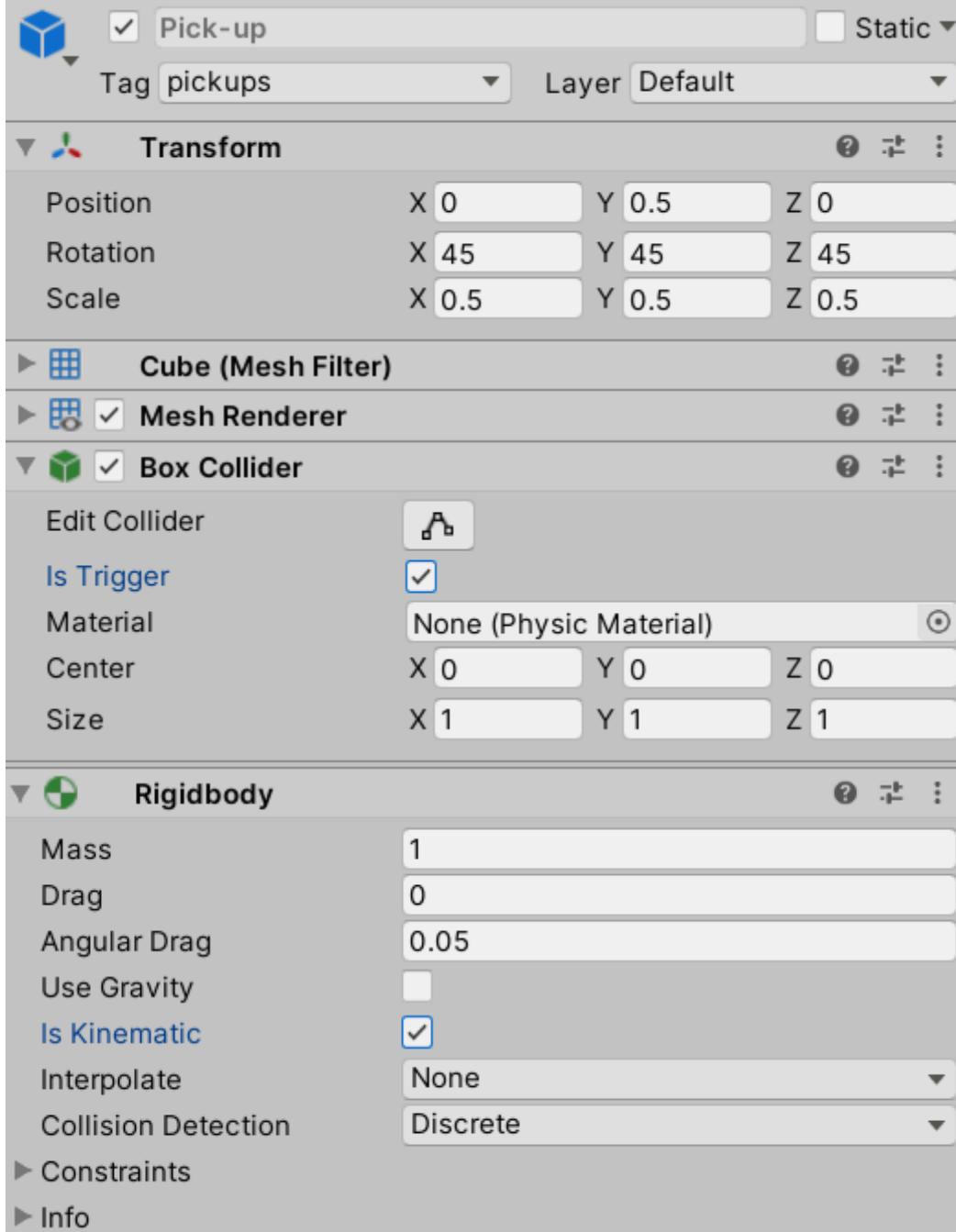
```
0 references
public class Rotator : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {

    }

    // Update is called once per frame
    0 references
    void Update()
    {
        // Rotate the game object that this script is attached to by 15 in the X axis,
        // 30 in the Y axis and 45 in the Z axis, multiplied by deltaTime in order to make it per second
        // rather than per frame.
        transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime);
    }
}
```

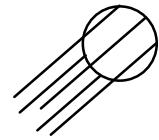
edit Pick-up prefab

- add Rotator.cs
- In Box Collider component
 - check **Is Trigger**
- In Rigidbody component
 - uncheck **Use Gravity**
 - check **Is Kinematic**

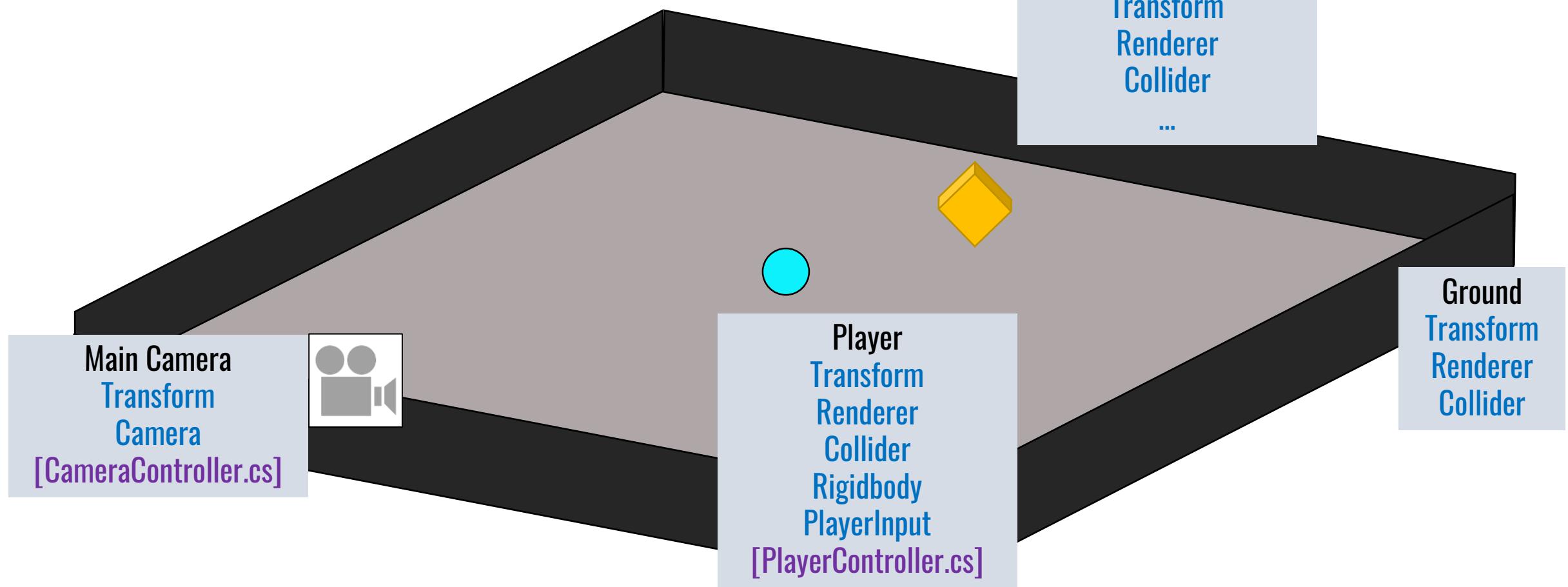




Wall
Transform
Child: Wall * 4

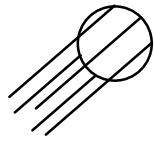


Directional Light
Transform
Light

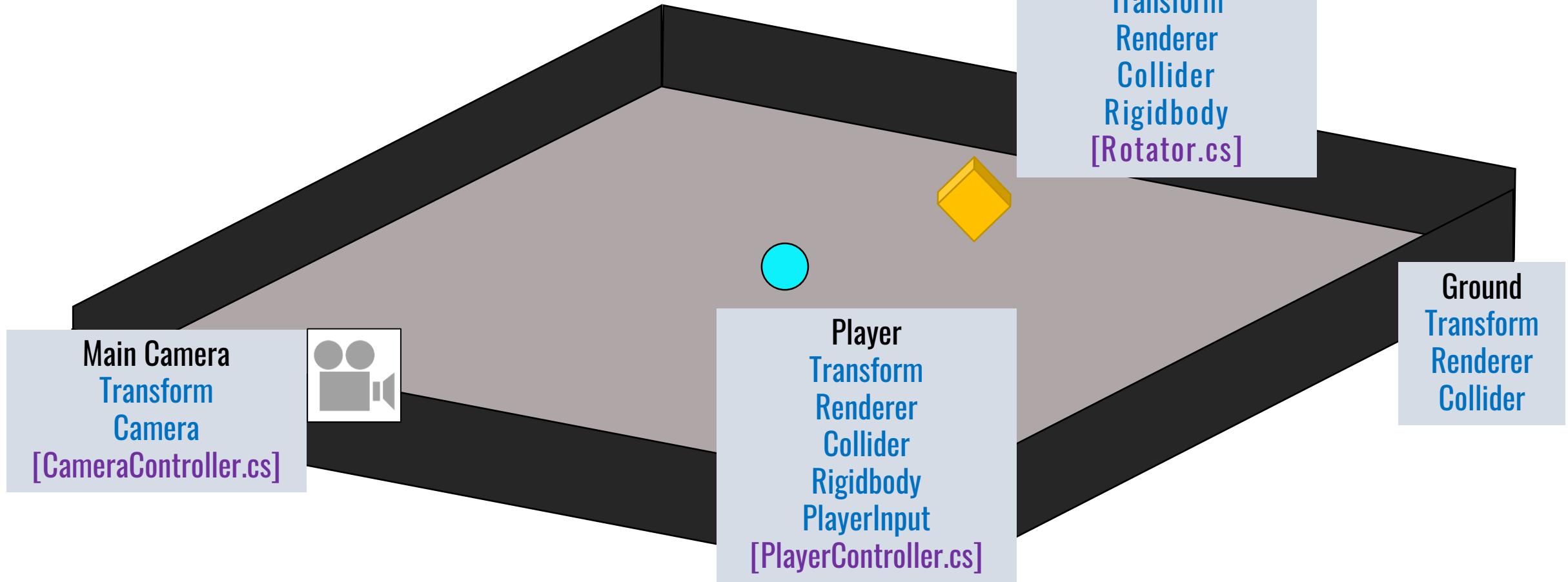




Wall
Transform
Child: Wall * 4



Directional Light
Transform
Light

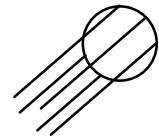




Wall

Transform

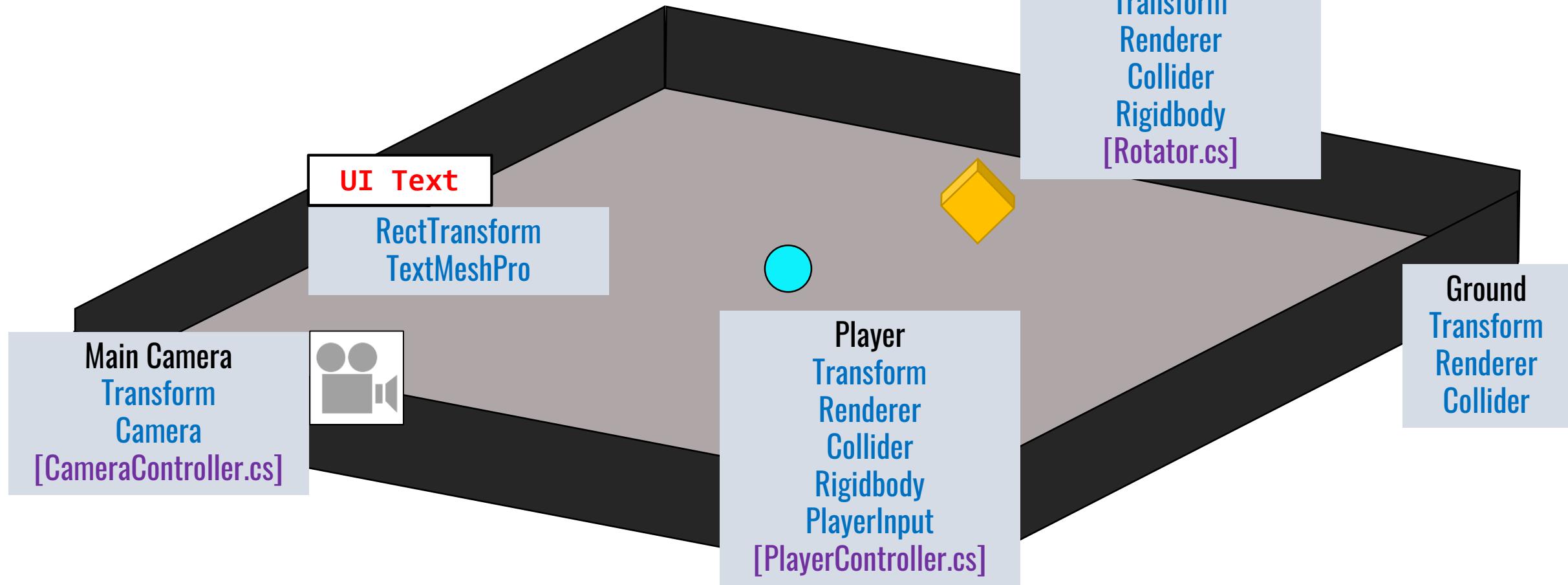
Child: Wall * 4



Directional Light

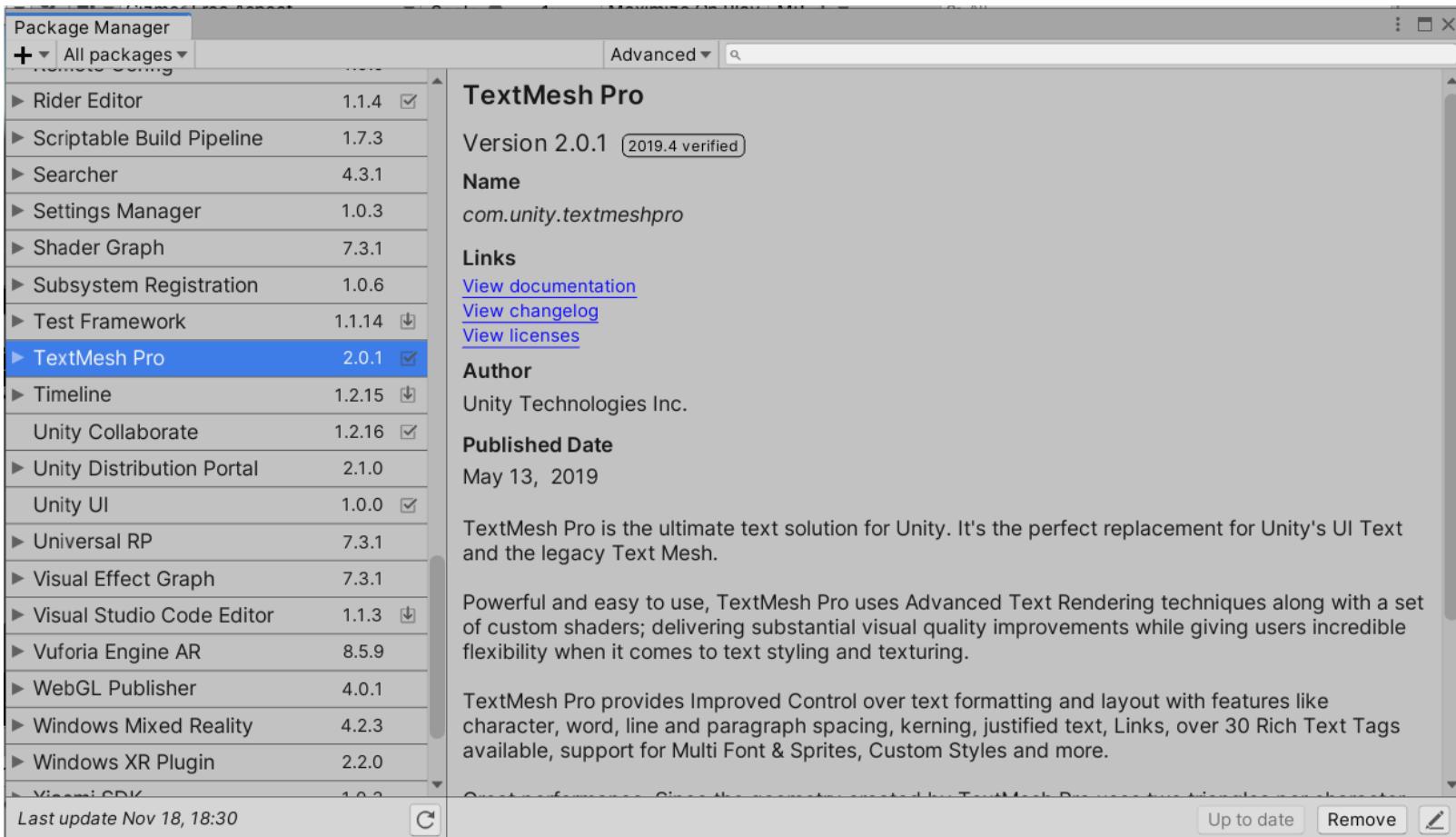
Transform

Light



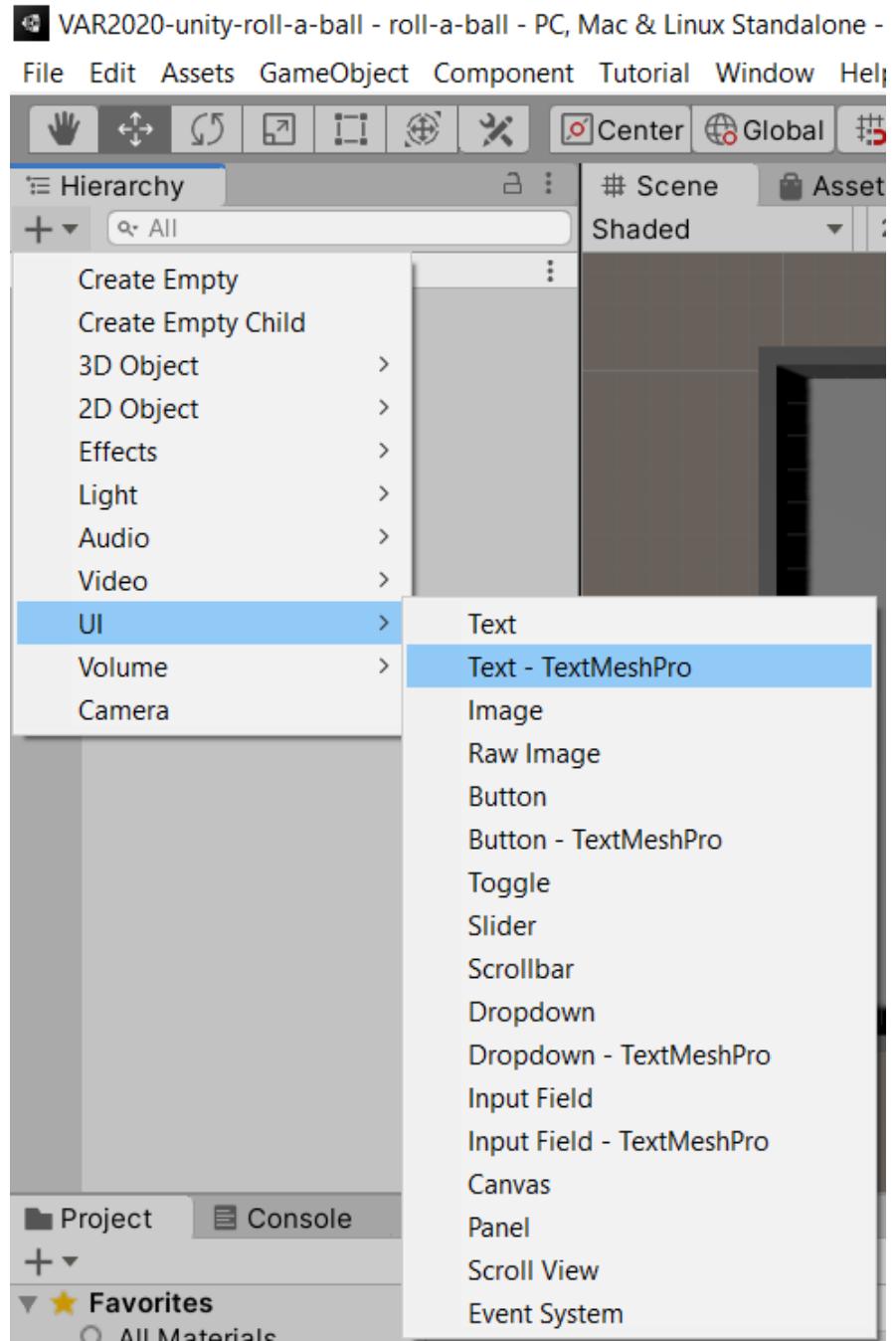
In Package Manager

- Add TextMesh Pro



UI: Count text

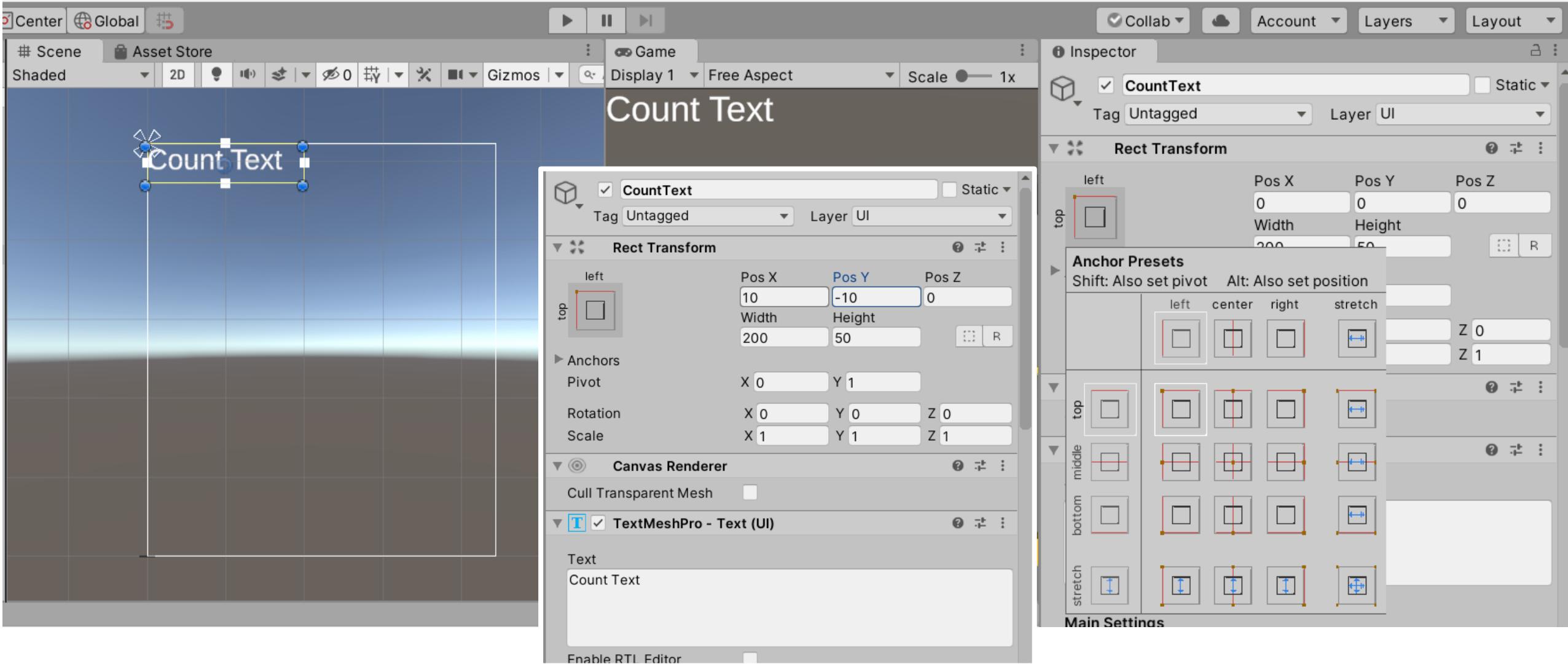
- Create UI
- Select Text - TextMeshPro



Shift + Alt and select upper left > change the anchor of UI

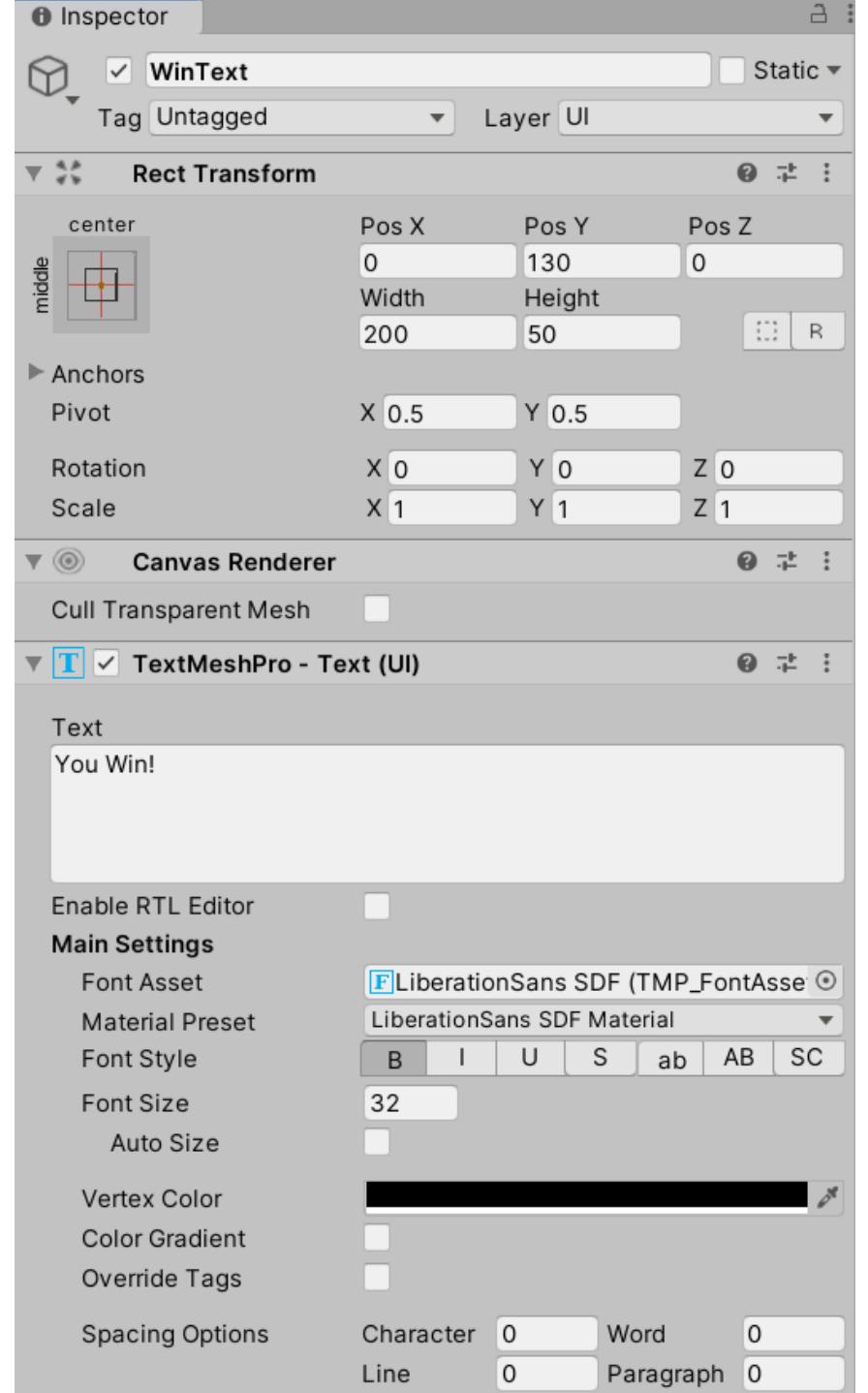
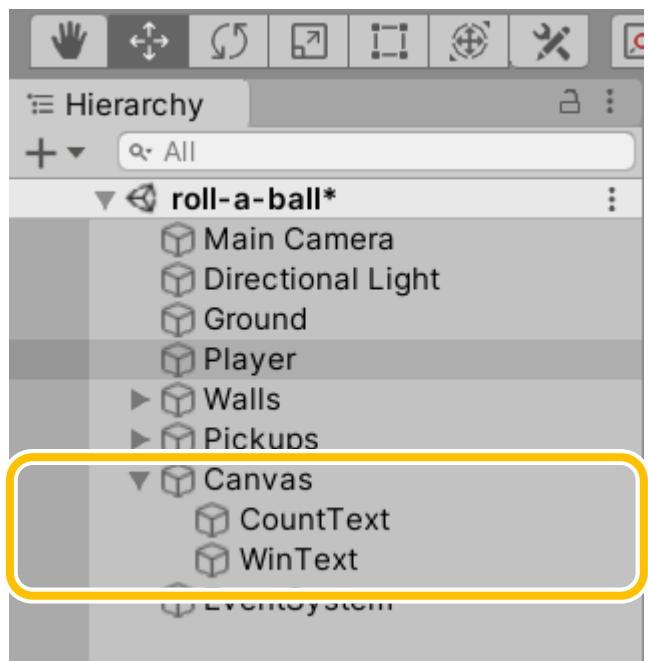
Mac & Linux Standalone - Unity 2019.4.3f1 Personal* <DX11>

Tutorial Window Help



UI: Win text

- Create another TMP text
- Anchor to the center
- Change text



back to PlayerController.cs

1. namespace: using TMPro; and global variables
2. Reset count and disable WinText in the beginning.
3. A function the update the CountText.

1

```
1 reference
public TextMeshProUGUI countText;
2 references
public GameObject winTextObject;
```

2

```
0 references
void Start()
{
    rb = this.GetComponent<Rigidbody>();
    count = 0;
    SetCountText();
    winTextObject.SetActive(false);
}
```

3

```
2 references
void SetCountText()
{
    countText.text = "Count: " + count.ToString();
    if (count >= 30)
    {
        winTextObject.SetActive(true);
    }
}
```

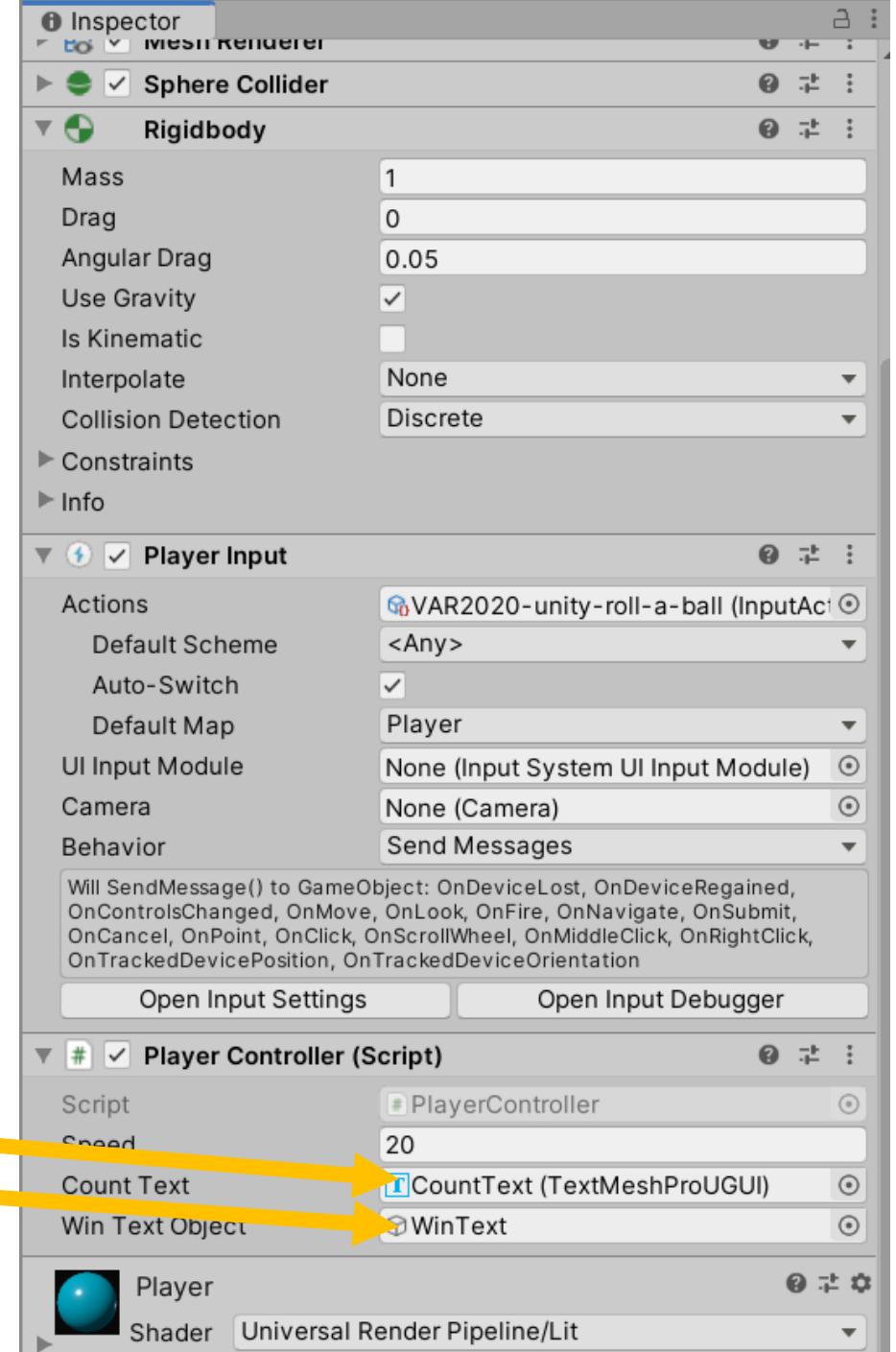
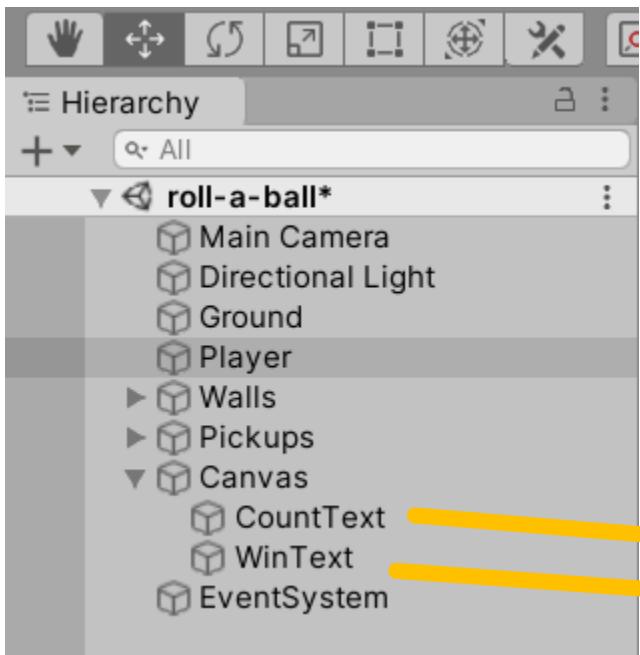
back to PlayerController.cs

4. When Player hits a pick-up,
increase the count and update
the CountText

```
0 references
17 > void Start() ...
24
0 references
25 > void Update() ...
29
0 references
30 > void OnMove(InputValue movementValue) ...
37
0 references
38 > private void FixedUpdate() ...
44
0 references
45 void OnTriggerEnter(Collider other)
46 {
47     if (other.gameObject.CompareTag("pickups"))
48     {
49         other.gameObject.SetActive(false);
50         4
51         count += 1;
52         SetCountText();
53     }
54
2 references
55 > void SetCountText() ...
63 }
64
```

In the scene

- Drag texts to the references of PlayerController

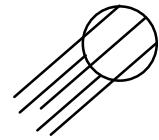




Wall

Transform

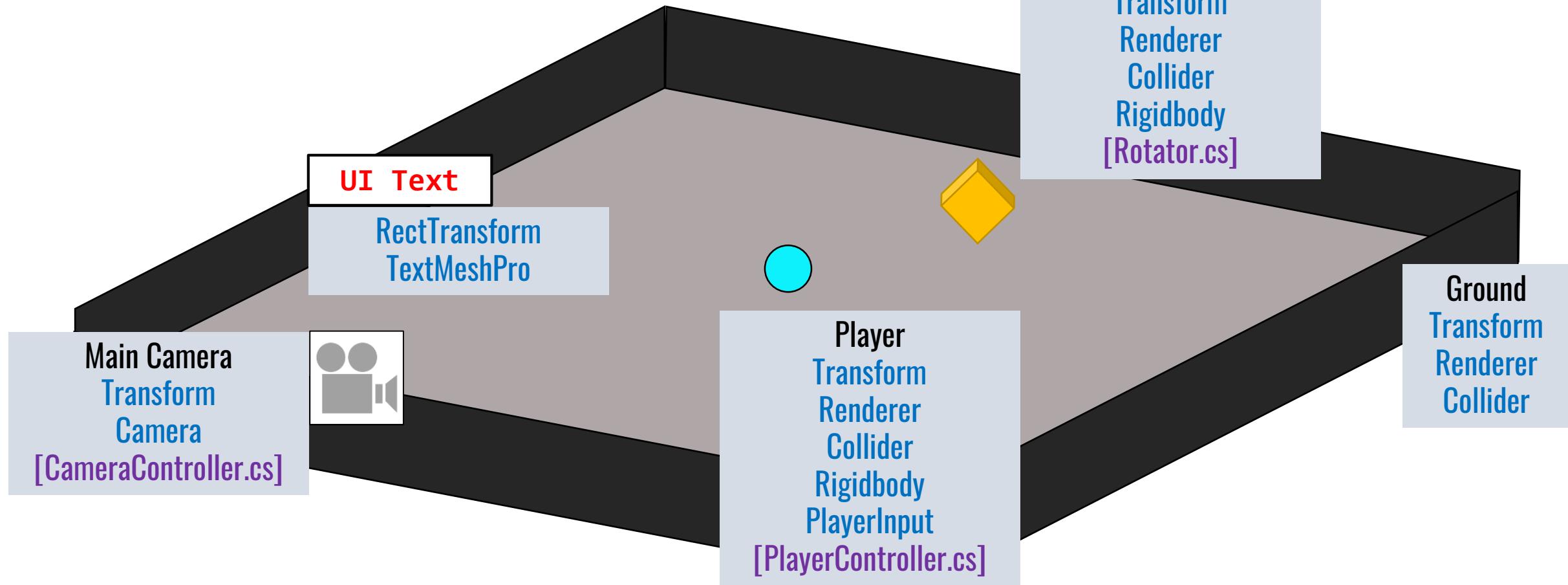
Child: Wall * 4



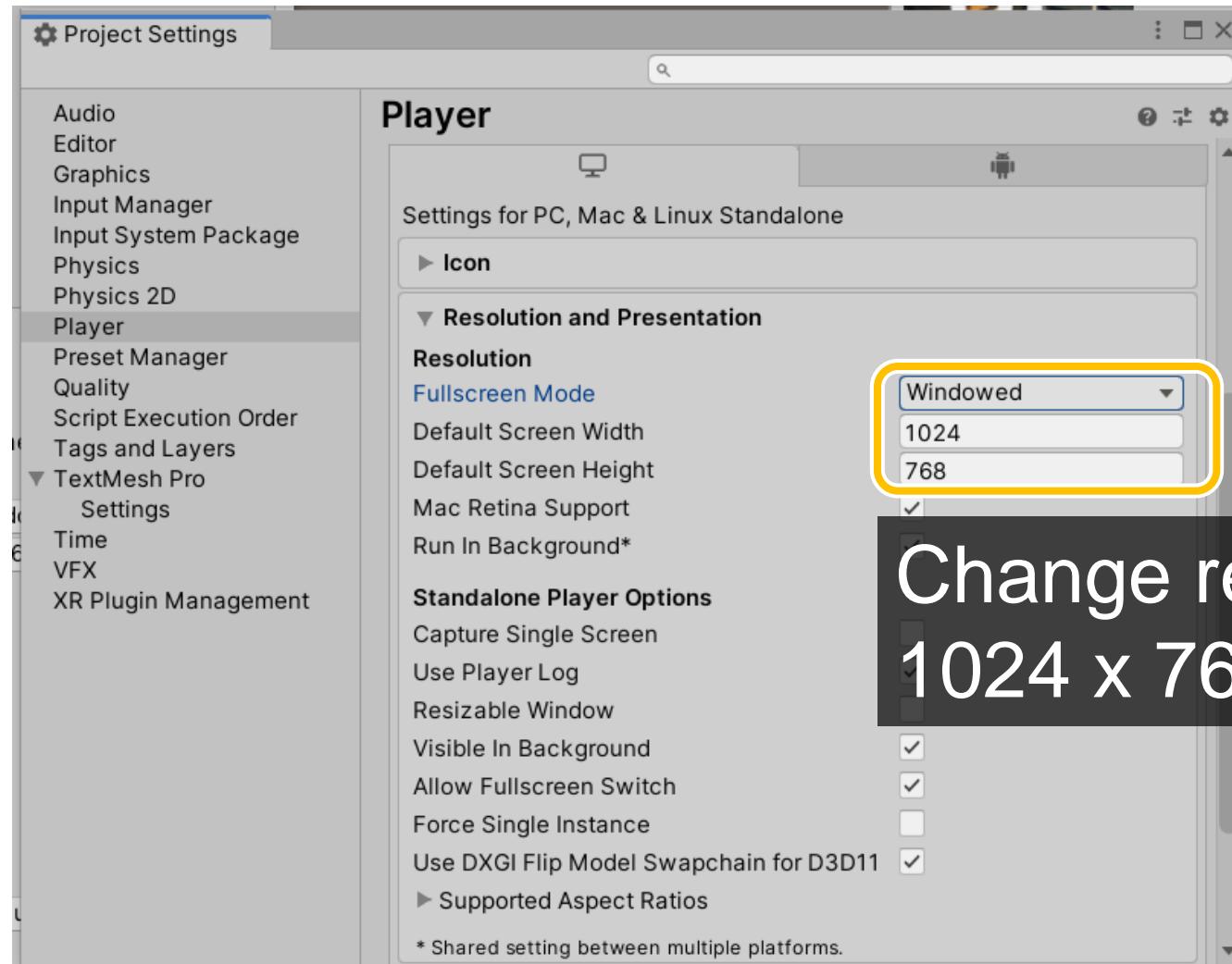
Directional Light

Transform

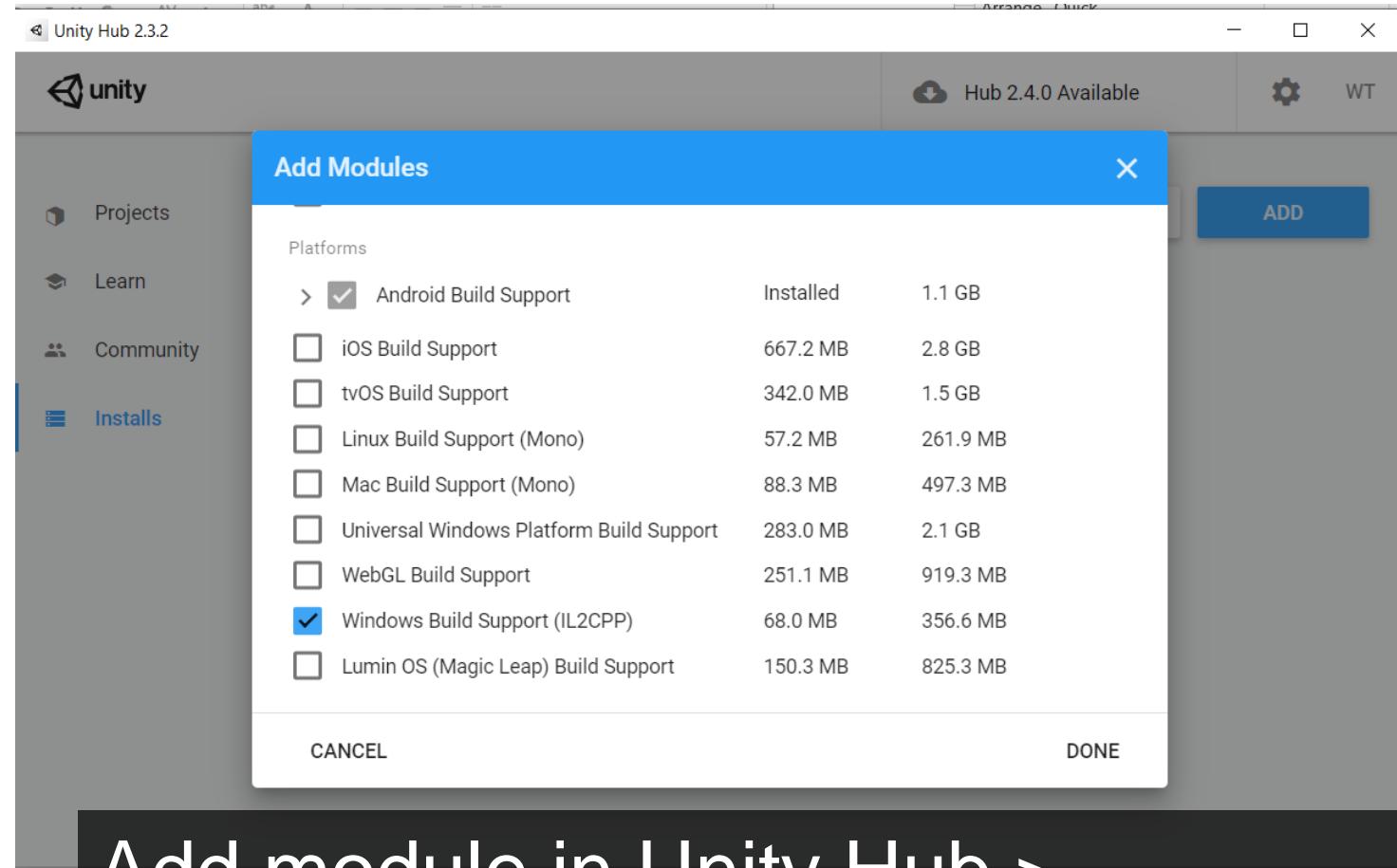
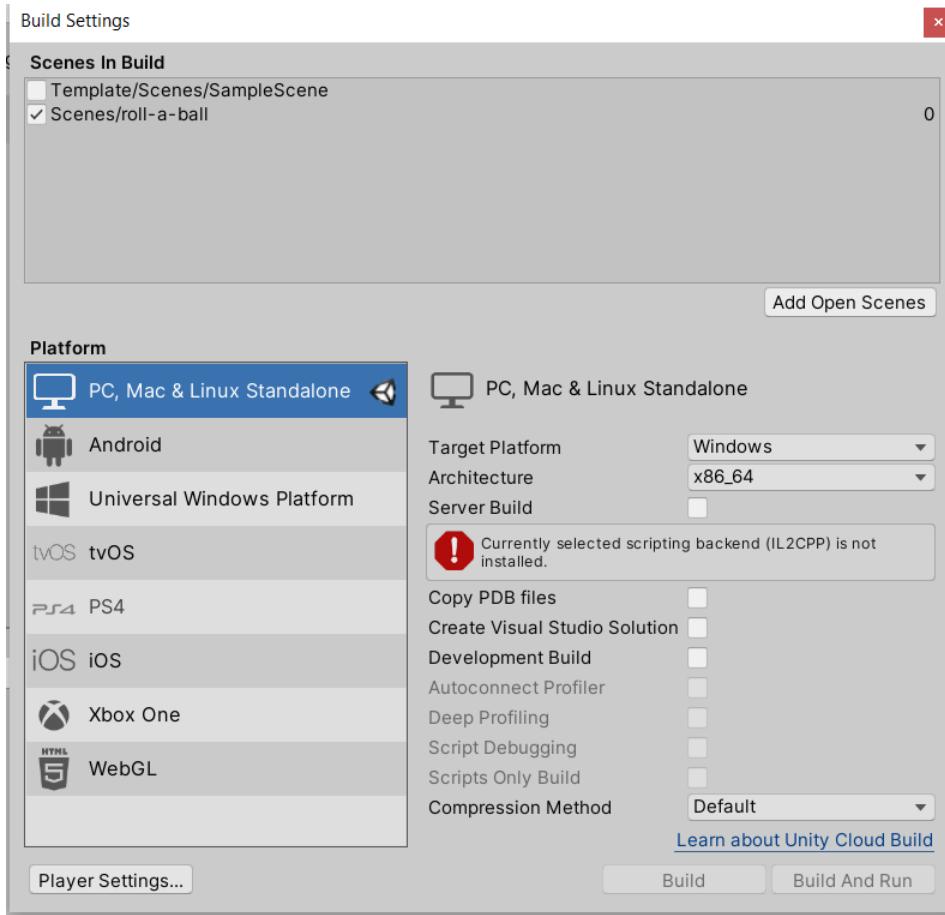
Light



Edit > Project Setting > Player



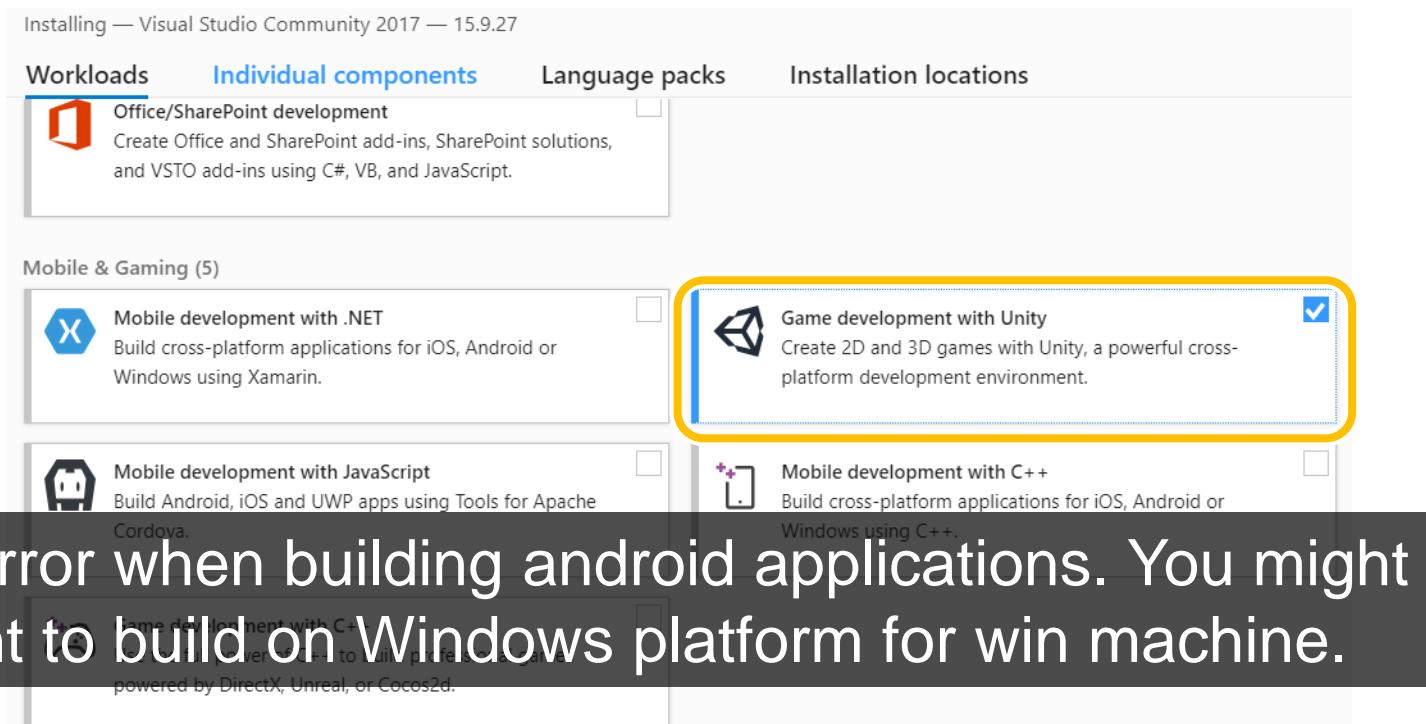
Build error on Windows?



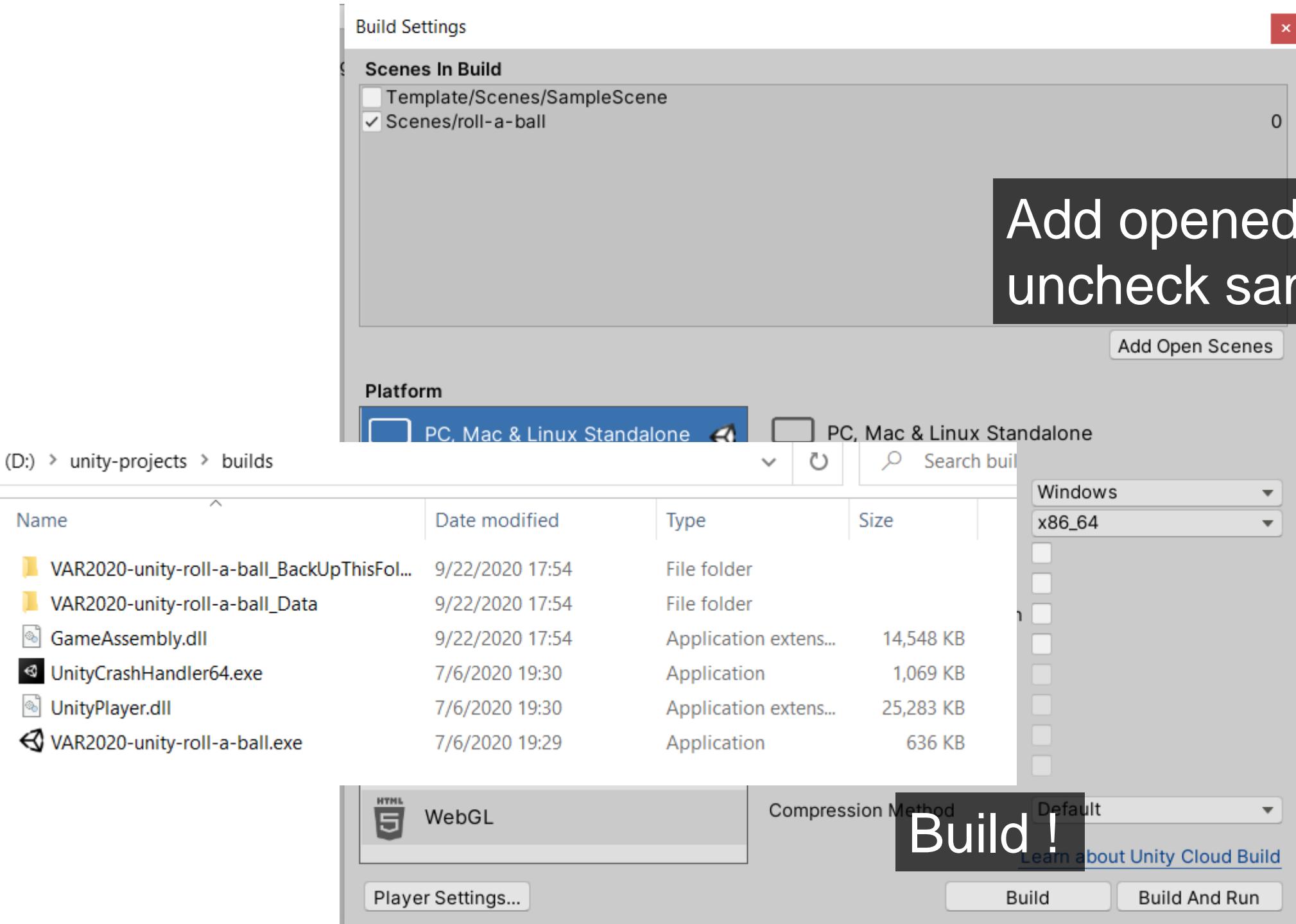
Add module in Unity Hub >
Windows Build Support (IL2CPP)

Build error on Windows

- Unity needs **Visual Studio 2017** and **Win 10 SDK** for building applications on Windows platform.
- <https://my.visualstudio.com/Downloads?q=visual%20studio%202017>



I did not have this error when building android applications. You might need this if you want to build on Windows platform for win machine.



Add opened scenes,
uncheck sample scene

Hands-on

- Start hacking the roll-a-ball
- Don't forget to document the progress and outcome on your website

If you want more?

- Some **optional** examples:
 - add sound/visual effect when picking up (e.g., explode)
 - add jump to collect pickups above the ground
 - choose another project on Unity Learn/Tutorial to follow

unity asset store

- <https://assetstore.unity.com/>

https://assetstore.unity.com/

Google 學術搜尋 The New York Times - ... MIT Technology Review wenjetseng (wjtse... DeepL Translator My Library | Zotero Google Calendar research PhD at Telecom Paris Interesting Unity/Graphics CV/ML/DL

New to Unity? Checkout these top picks for all experience levels

unity Asset Store Assets Tools Services By Unity Industries

Search for assets

Over 11,000 5 star assets Rated By: 85000+ customers Supported by over 100,000 forum members

Refine by clear filters

Hide Purchased Assets

All Categories

3D (35145) 2D (8137) Add-Ons (43) Audio (6118) Essentials (50) Templates (2910) Tools (8741) VFX (2518)

The screenshot shows the Unity Asset Store homepage. At the top, there's a navigation bar with links for 'Assets', 'Tools', 'Services', 'By Unity', and 'Industries'. Below the navigation is a search bar with the placeholder 'Search for assets'. Three promotional banners are displayed: 'Over 11,000 5 star assets', 'Rated By: 85000+ customers', and 'Supported by over 100,000 forum members'. The main content area features two large asset cards: 'genvid' (Revolutionize the way games are watched with Genvid) and 'playMaker' (represented by a flowchart icon). To the right, there's a sidebar titled 'Refine by' with a 'clear filters' link, containing checkboxes for 'Hide Purchased Assets' and a list of categories: 3D (35145), 2D (8137), Add-Ons (43), Audio (6118), Essentials (50), Templates (2910), Tools (8741), and VFX (2518).

I usually look for free assets here...

Refine by

[clear filters](#)

Hide Purchased Assets

All Categories

3D (2454)

2D (694)

Add-Ons (18)

Audio (539)

Essentials (43)

Templates (323)

Tools (2029)

VFX (227)

Pricing

Free Assets (2454)

furniture



Purchased

VERTEX STUDIO
Big Furniture Pack
★★★★★ (94)
FREE



Purchased

GEST
Pack Gesta Furniture #1
★★★★★ (53)
FREE



ELCANETAY

Toon Furniture
★★★★★ (16)
FREE



TRIDIFY

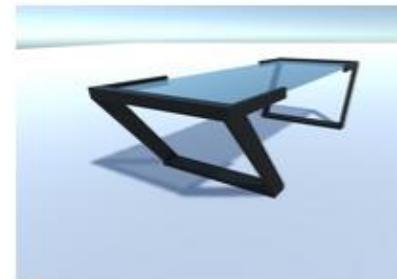
HDRP Furniture Pack
★★★★★ (5)
FREE



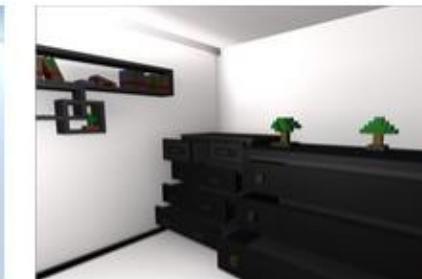
AMBIMESH
Raw Wooden Furniture Free
(not enough ratings)
FREE



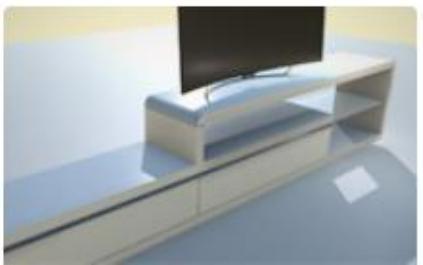
ELCANETAY
Office Room Furniture
(not enough ratings)
FREE



REYNARD DROSTE
Modern Furniture Pieces P...
★★★★★ (5)
FREE



KELHEM STORE
Voxel Functional Furniture ...
★★★★★ (5)
FREE



animal



Purchased

VERTEX CAT
Farm Animals Set
★★★★★ (30)
FREE



JKTIMMONS
Lovely Animals PACK
★★★★★ (27)
FREE



VOXELGUY
5 animated Voxel animals
★★★★★ (6)
FREE



TOTAL GAME ASSETS
Voxel Animals Pack
★★★★★ (5)
FREE



Purchased

MESHTINT STUDIO
Meshtint Free Chicken Me...
★★★★★ (17)
FREE



GLOOMY STUDIO
Free Low Polygon_Animal
★★★★★ (13)
FREE



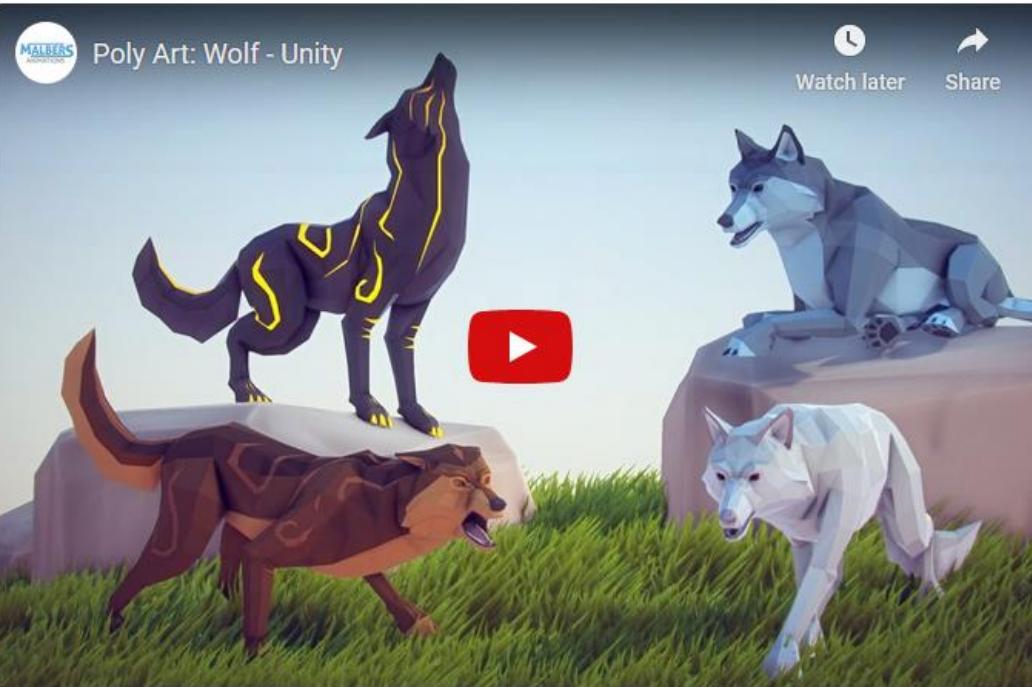
ZEALOUS INTERACTIVE
Golden Tiger
★★★★★ (62)
FREE



DUNGEON MASON
Dog Knight PBR Polyart
★★★★★ (21)
FREE



animal low poly



RECOMMENDED BY OUR USERS

MALBERS ANIMATIONS

Poly Art: Wolf

★★★★★ (61)

\$19.99



Purchased

VERTEX CAT

Farm Animals Set

★★★★★ (30)

FREE

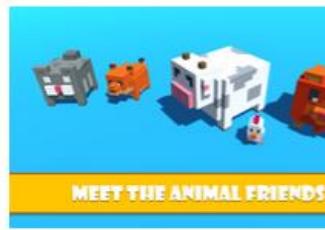


ACORN BRINGER

Simplistic Low Poly Nature

★★★★★ (15)

FREE



MEET THE ANIMAL FRIENDS!

TOTAL GAME ASSETS

Voxel Animals Pack

★★★★★ (5)

FREE

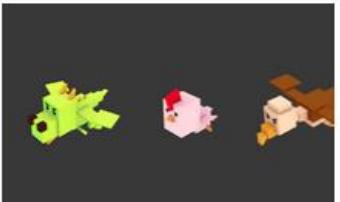


CATTLEYA

Animated low poly spider

★★★★☆ (6)

FREE



Low Poly Christmas
add-on



Low poly is a polygon mesh in 3D computer graphics that has a relatively small number of polygons. [\[wiki\]](#)



Questions?