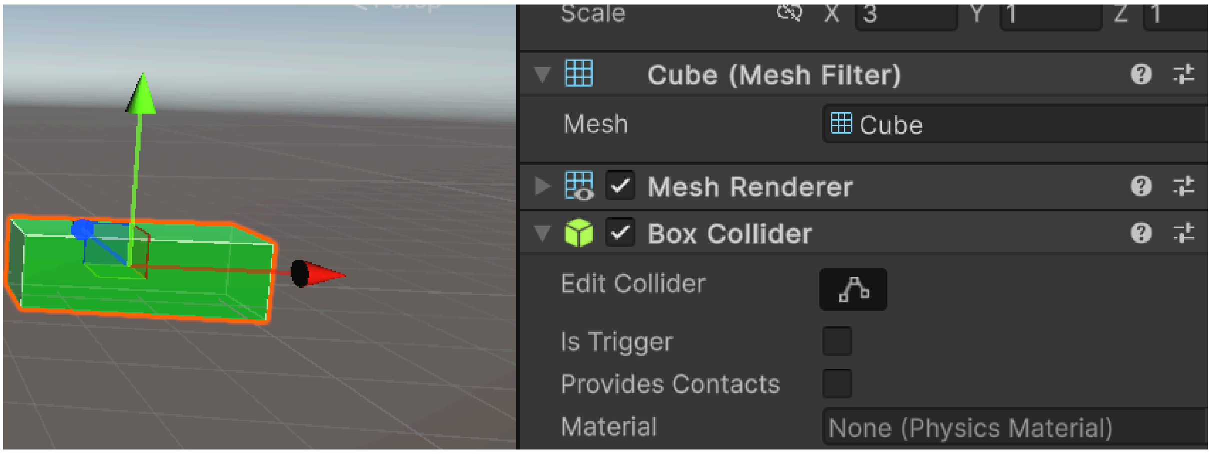
Aror University of Art, Architecture, Design and Heritage
SUKKUR,Sindh
Department of Multimedia and Gaming
Course: Game Programming CSC-221 (Practical)
Instructor: Engr. Fatima Jaffar

Lab No. 05

Objective: Collision of GameObjects, Levels in unity

Collider in Unity

Collider is a property of a game object that enables the effects of when a gameobject collides with an other.



Scale X 3 Y 1 Z 1

Cube (Mesh Filter)

Mesh: Cube

✓ Mesh Renderer

✓ Box Collider

Edit Collider

Is Trigger □

Provides Contacts □

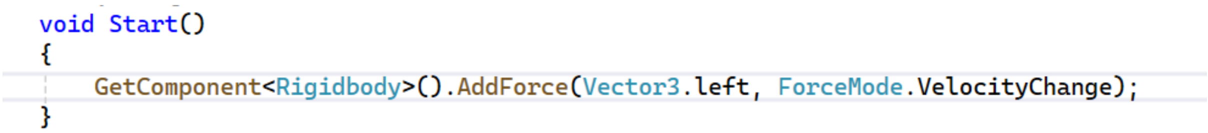
Material: None (Physics Material)

- The green border shows the collider is applied.
- You can check or uncheck the colliding property from the inspector window

onCollision() function

it is a function used to specify a user defined action that will happen if the gameobject collides with another.

Moving the platform towards left when game starts



```
void Start()
{
    GetComponent<Rigidbody>().AddForce(Vector3.left, ForceMode.VelocityChange);
}
```

Tags in game

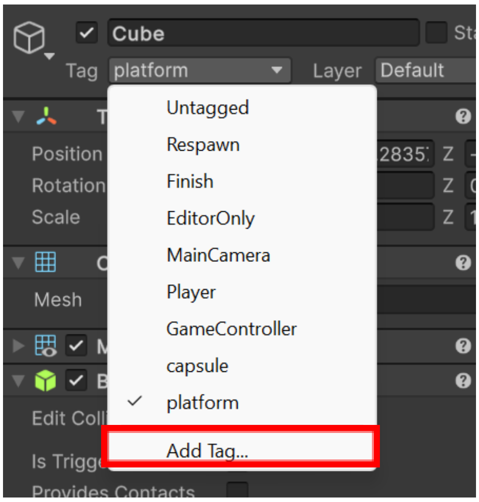
- In Unity, tags are words or phrases that you can use to identify GameObjects.
- Tags are useful for organizing and labeling GameObjects, which can be characters, props, scenery, and more.

How to use tags

- You can use tags for scripting purposes.
- You can use the GameObject.FindWithTag() function to find a GameObject by its tag.
- You can use tags to identify objects in your scripts, which can make your development more efficient.
- You can use tags to categorize your game objects. For example, you can tag a floor tile as "stone", "wood", or "both

How to add and remove tags

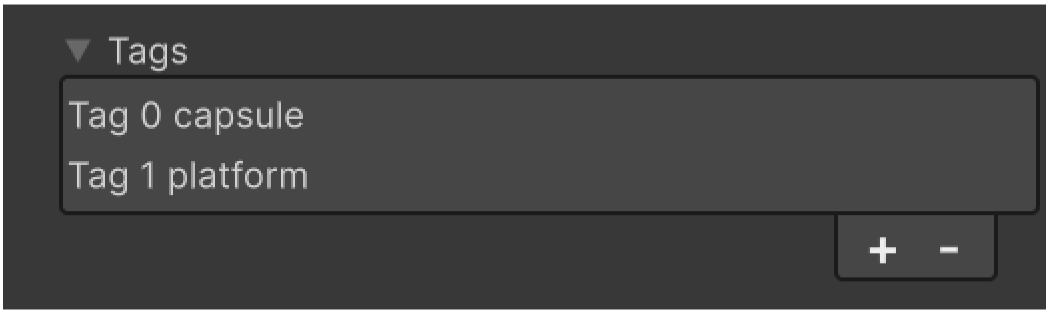
- To add a new tag, click the plus button (+) at the bottom-right of the list, and name your new tag.



☑ Cube

Tag platform

Layer Default



Tags

Tag 0 capsule

Tag 1 platform

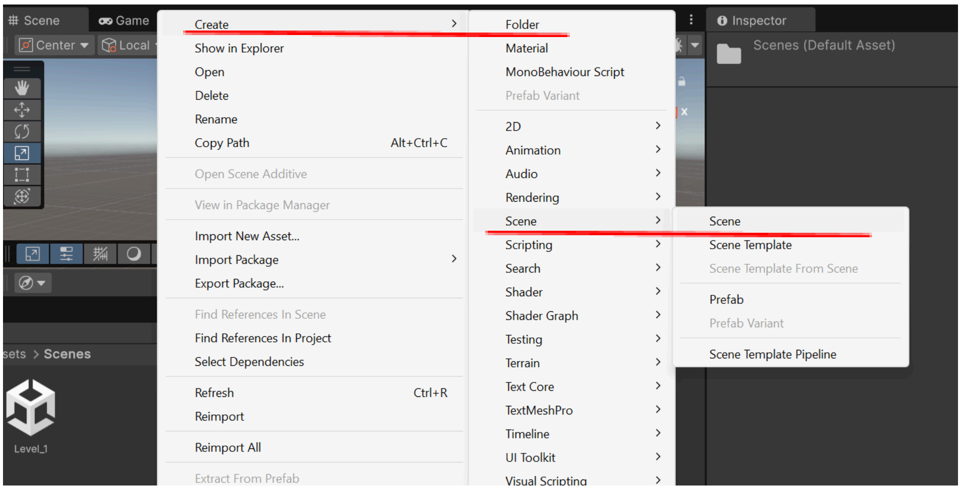
- To remove a tag, click on it and then click the minus (-) button at the bottom-right of the list.

Conditions on Tags

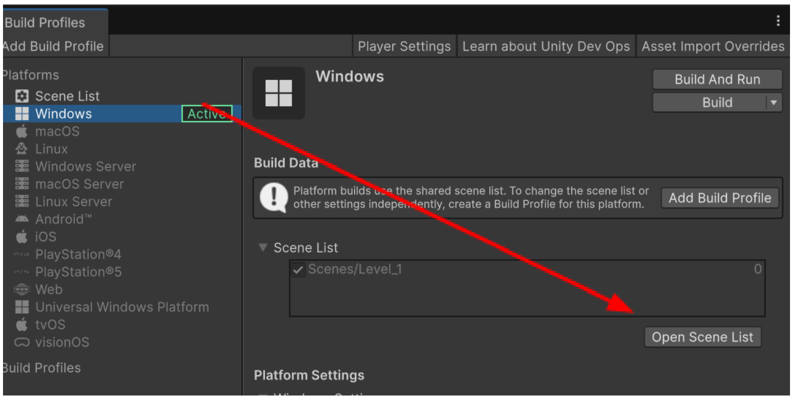
```
if (collision.gameObject.CompareTag("capsule"))
{
    Debug.Log("Congratulations!!  you have reached the end of Level 1");
}
```

The Next Level

- You can go to assets and rename the sample scene as Level_1
- Now you can just copy and paste the same scene,it will create a new but duplicate scene
- Or you can goto create and create a new scene, which will be empty



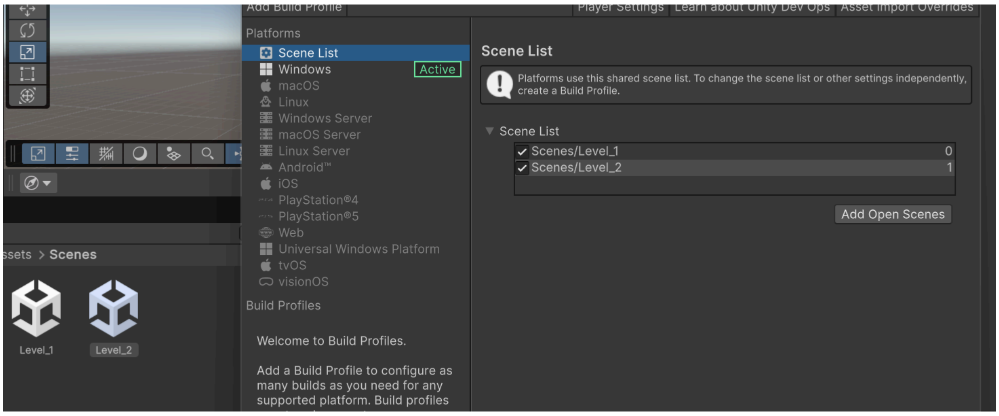
- It will create a new scene, you can rename it accordingly
- Now we have to tell the etidor which level comes first and all that
- To do so, goto file □ build Profile □ click on open Scenes list



Platform builds use the shared scene list. To change the scene list or other settings independently, create a Build Profile for this platform.

✓ Scenes/Level_1 0

You can drag and drop the scenes from Assets to Scenes List



Scene List

Platforms use this shared scene list. To change the scene list or other settings independently, create a Build Profile.

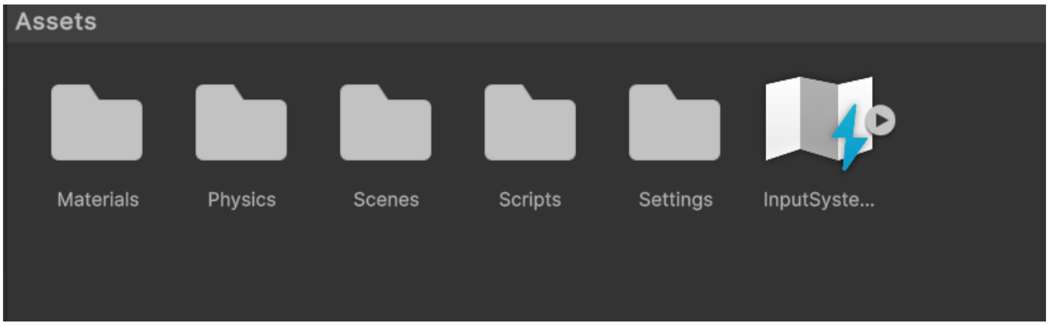
Scene List

✓ Scenes/Level_1 0
✓ Scenes/Level_2 1

Welcome to Build Profiles.

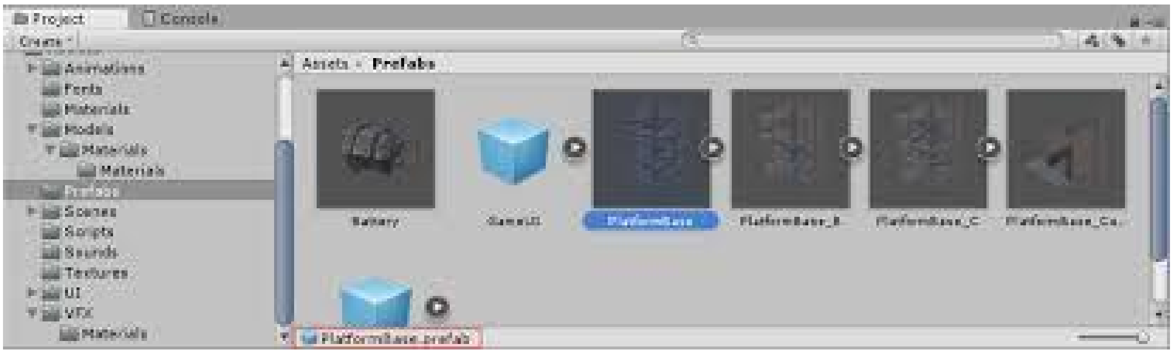
Add a Build Profile to configure as many builds as you need for any supported platform. Build profiles

- First, organize the assets into corresponding folders

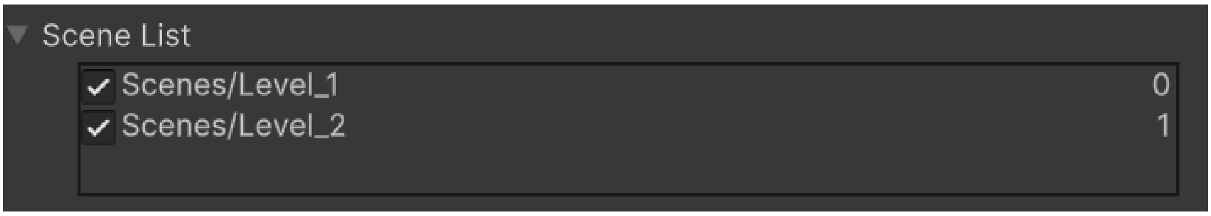


Prefabs

Prefab is a reusable template that stores a fully configured GameObject, including all its components and properties, allowing you to easily create multiple identical instances of that object within your scene, essentially acting as a blueprint for creating similar objects in your game, where any changes made to the prefab are automatically reflected in all its instances.



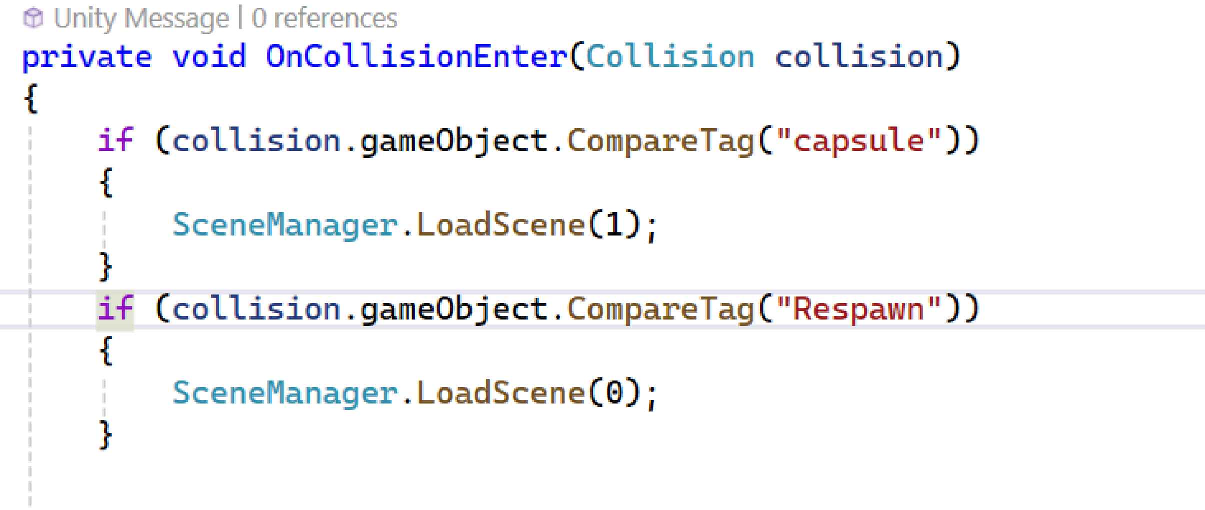
How to move from one level to another level based on condition?



Scene List

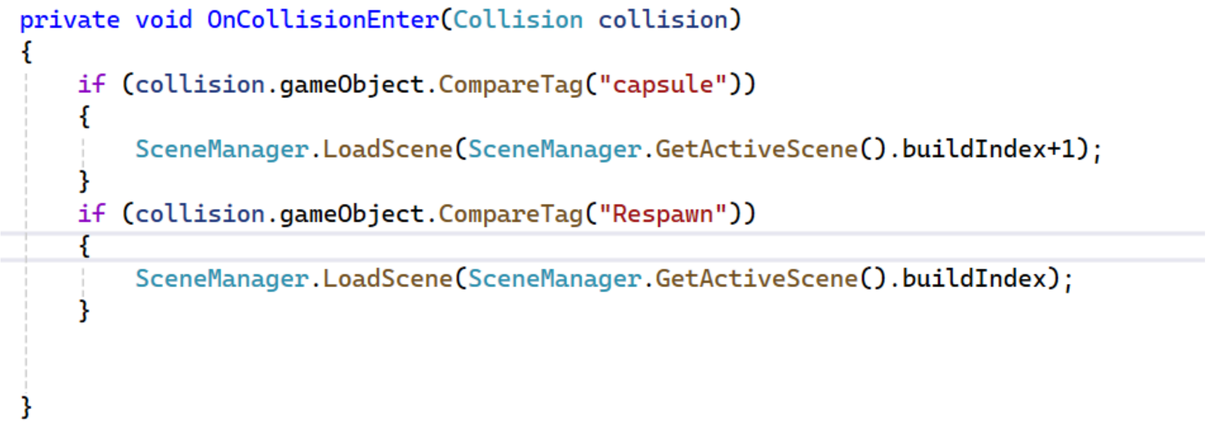
☑ Scenes/Level_1	0
☑ Scenes/Level_2	1

- By default scene has indexes which start from 0.
- You can load the scene by passing the index value to method LoadScene() in the following way



```
Unity Message | 0 references
private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("capsule"))
    {
        SceneManager.LoadScene(1);
    }
    if (collision.gameObject.CompareTag("Respawn"))
    {
        SceneManager.LoadScene(0);
    }
```

- But the problem with the above code is that if it runs it will always load the scenes whose index is mentioned.
- So to avoid that you need to get the current scene's index and code accordingly



```
private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("capsule"))
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex+1);
    }
    if (collision.gameObject.CompareTag("Respawn"))
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }


}
```

- And So on.

Lab Tasks

- Create atleast 5 levels, adding other objects as well.
    - Apply your logic to do actions when particular gameobject collides with another
    -