

Pick up and Drop Weapon

1. Create a new Scene

Create a scene called Pick up weapon.

Create a 3D Cube called “Floor” and elongate it to become the plane where the player will walk through.

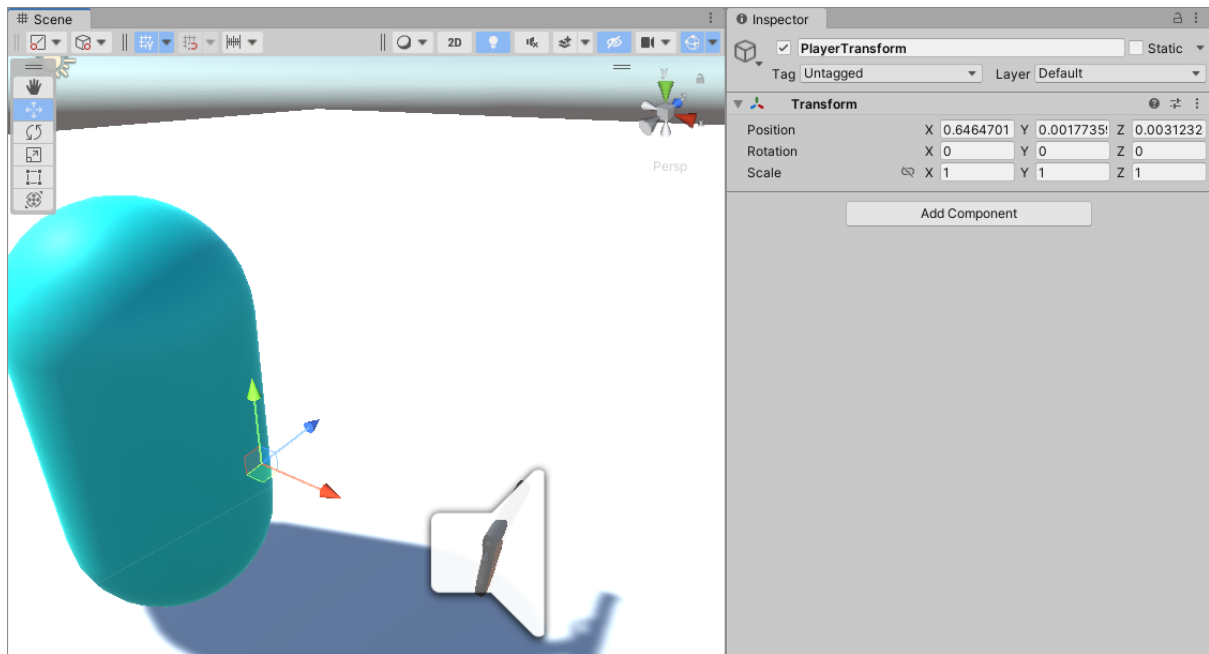
Create a capsule and call it “Player”, also add a Player tag into the capsule.

Add following components to the “Player”, “Rigidbody”, “Character Controller” and “Player Controller” script.

Within the Rigidbody we will untick the “Use Gravity” box and we will open the constraints tab inside the Rigidbody and we will tick all of the boxes to prevent the player from launching to the sky as soon as you test it.



Next inside the player we will create an empty “Game Object” and we will name it “PlayerTransform”, roughly relocate this “Game Object” to the side of the character where the weapon will be located. The “PlayerTransform” will help the script deduce where the weapon will be attached to.

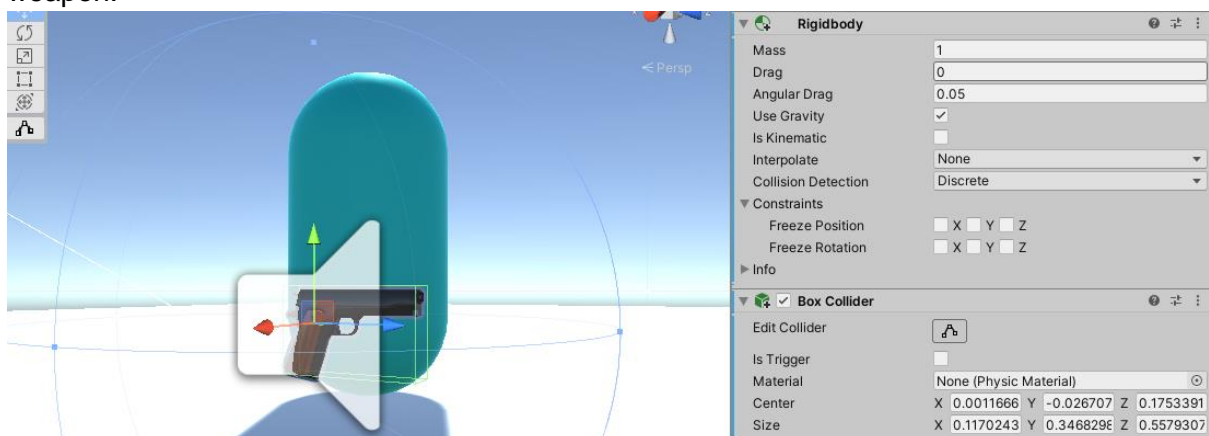


2. Add your weapon to the scene

Drag and drop a weapon either made by you or downloaded from the asset store.

Inside the weapon we will add two components, one being a “Rigidbody” and the other a “Box Collider”.

Within the “Rigidbody” we will make sure to leave the “Use Gravity” ticked so the weapon will act with gravity. And we will edit the “Box Collider” to be roughly the same size as the weapon.



3. Create a new script

Create a new script called “Target” in the assets tab.

This script will allow us to know when the player is close to the weapon by sending a message to the console in unity, to do this we will use this code:

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine;
5
6  public class target1 : MonoBehaviour
7  {
8      public float drop = 2f;
9      public equipScript eq;
10
11     0 references
12     public void Commence (float amount)
13     {
14         drop -= amount;
15         if (drop <= 0f)
16         {
17             Go();
18         }
19     }
20
21     1 reference
22     void Go()
23     {
24         Destroy(gameObject);
25     }
26
27     Unity Message | 0 references
28     private void OnCollisionEnter(Collision other)
29     {
30         Debug.Log("Touching");
31         if (other.gameObject.CompareTag("Player"))
32         {
33             eq.nearGun = true;
34         }
35     }
36 }

```

4. Creart another script.

We will create a new script in the asset tab called "equipScript".

This script will do most of the work, as it will allow us to show whether we are within range to grab a weapon and also show if we are holding the weapon in order to determine if we need to equip or unequip the weapon. In order to do this we will create various variables in order to detect the "Weapon" and the "Player Transform". We will use the following variables to do this:

```

6      {
7          public Transform PlayerTransform;
8          public GameObject Gun;
9          public bool nearGun = false;
10         private bool holdGun = false;
11     }

```

To finish this script we will code the action button and the functions that will be called when the action button is pressed, in order to do this we will follow the following code:

```

12     void Start()
13     {
14         Gun.GetComponent<Rigidbody>().isKinematic = true;
15     }
16
17     void Update()
18     {
19         if (Input.GetKeyDown("f") && nearGun && !holdGun)
20         {
21             holdGun = true;
22             Shoot();
23         }
24         else if (Input.GetKeyDown("f") && holdGun)
25         {
26             holdGun = false;
27             UnequipObject();
28         }
29     }
30
31     void UnequipObject()
32     {
33         PlayerTransform.DetachChildren();
34         Gun.transform.eulerAngles = new Vector3(Gun.transform.eulerAngles.x, Gun.transform.eulerAngles.y, Gun.transform.eulerAngles.z -45);
35         Gun.GetComponent<Rigidbody>().isKinematic = false;
36     }
37
38     void Shoot()
39     {
40         EquipObject();
41     }
42     void EquipObject()
43     {
44         Gun.GetComponent<Rigidbody>().isKinematic = true;
45         Gun.transform.position = PlayerTransform.transform.position;
46         Gun.transform.rotation = PlayerTransform.transform.rotation;
47         Gun.transform.SetParent(PlayerTransform);
48     }
49
50

```

5. Assing the scripts and Test

Finally, we will assign the scripts to it's respective "GameObjects". The "equipScript" will be assigned to the Player were we will drag the "Weapon" and the "PlayerTransform" into the script slots; the "Target" will be assigned to the "Weapon", once done that we will drag the "Player" into the Eq slot, this is done so the "Target" script can read from the "equipScript". After doing this we will Test and see if the script work.