Game : Lost in Woods

Name: Punit Game Development 3D (Unity)

Main Menu:

Step 1 : Make a scene call it as menu

Step 2 : Now create a canvas by right clicking > UI > canvas

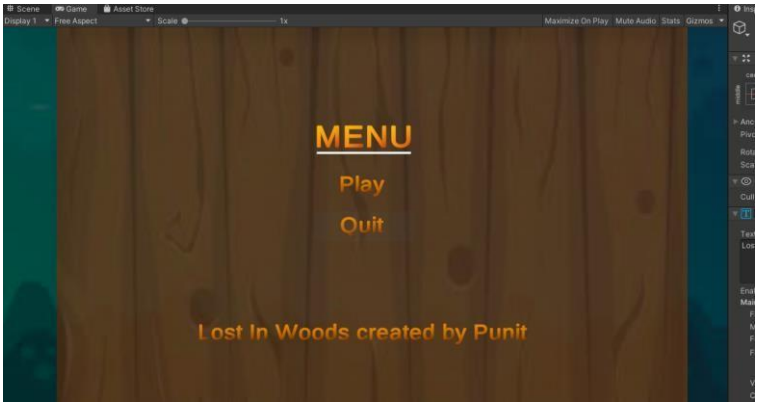
Step 3 : Now add another canvas inside canvas for Background.

Step 4 : Now for background add new component “image” and browse image for your background.

Step 5 : now inside this add two button (Right click > UI > Button) for Play and quite.

Step 6 : Also add text for game name and labels.

Step 7 : Now add scripts for particular buttons.

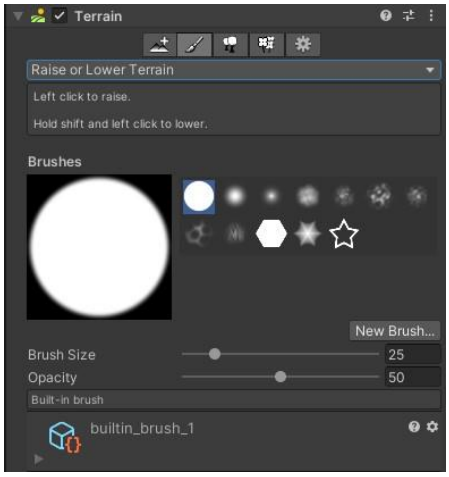


Terrain and Level 1:

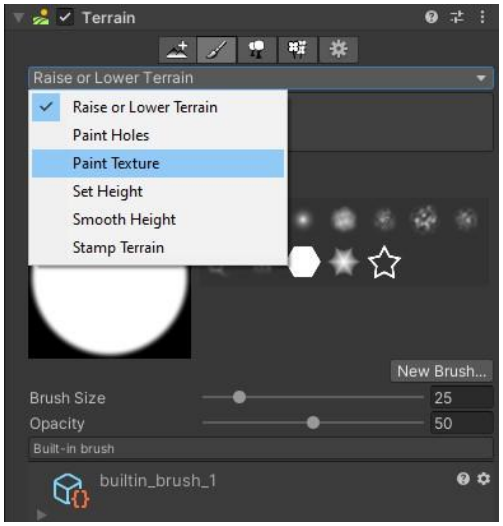
Step 1 : Make Scene call it as level_1.

Step 2 : Now add terrain (Right click > 3D object > Terrain) , You’ll see a large rectangular on the scene.

Step 3 : Now go to inspector and choose brushed in Raise and lower Terrain option, Set size and opacity as per your requirements and start raising the terrain.

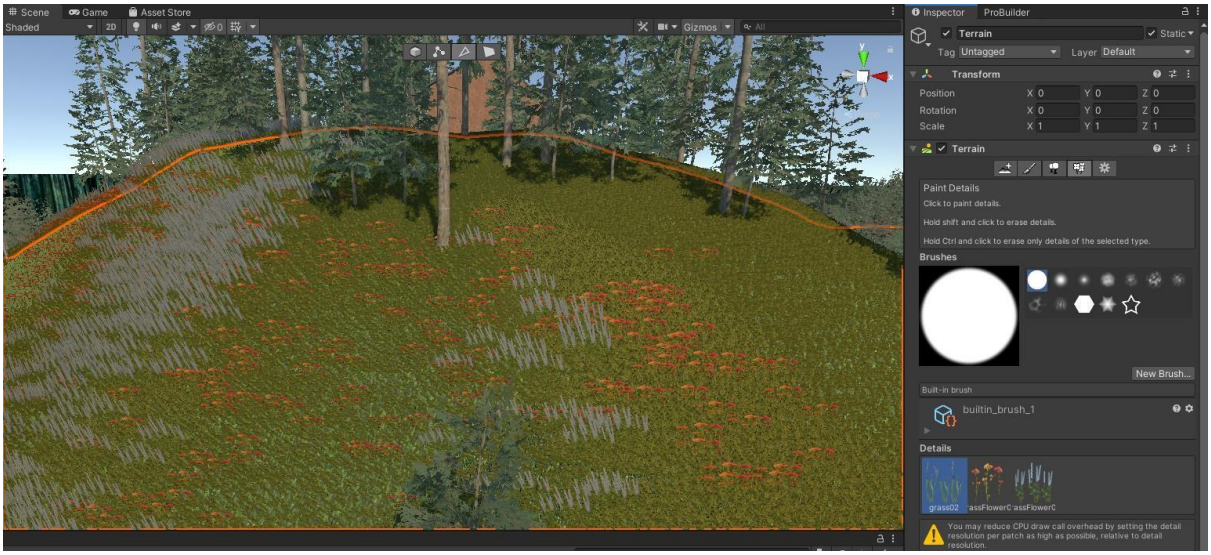


Step 4 : Now choose Pain texture to pain texture on your terrain.



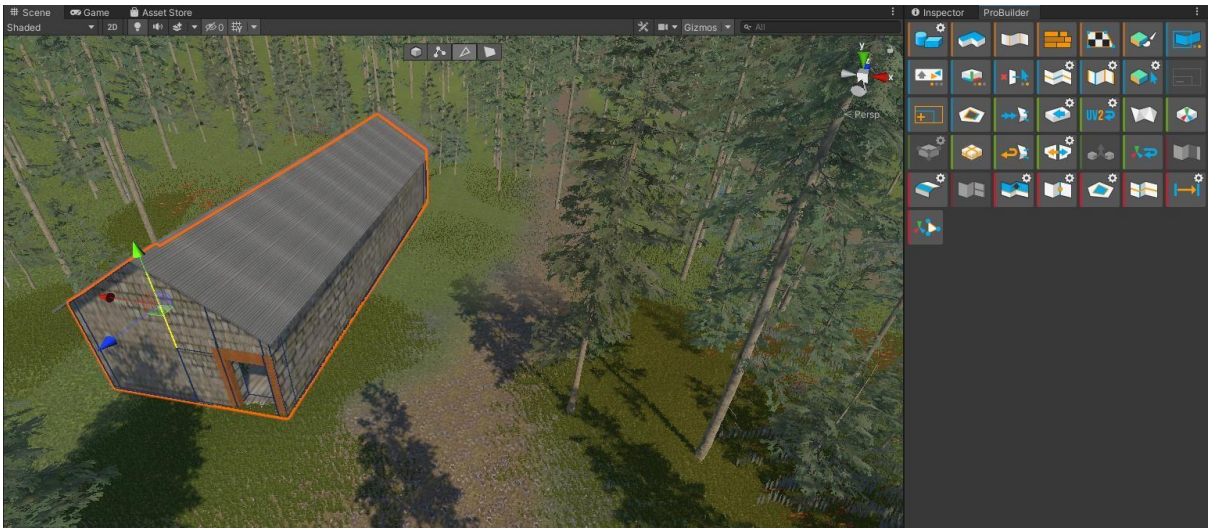
Step 5 : Now you can add the layers and paint the textures.

Step 6 : Now add trees and grass using pain tress and paint details options:

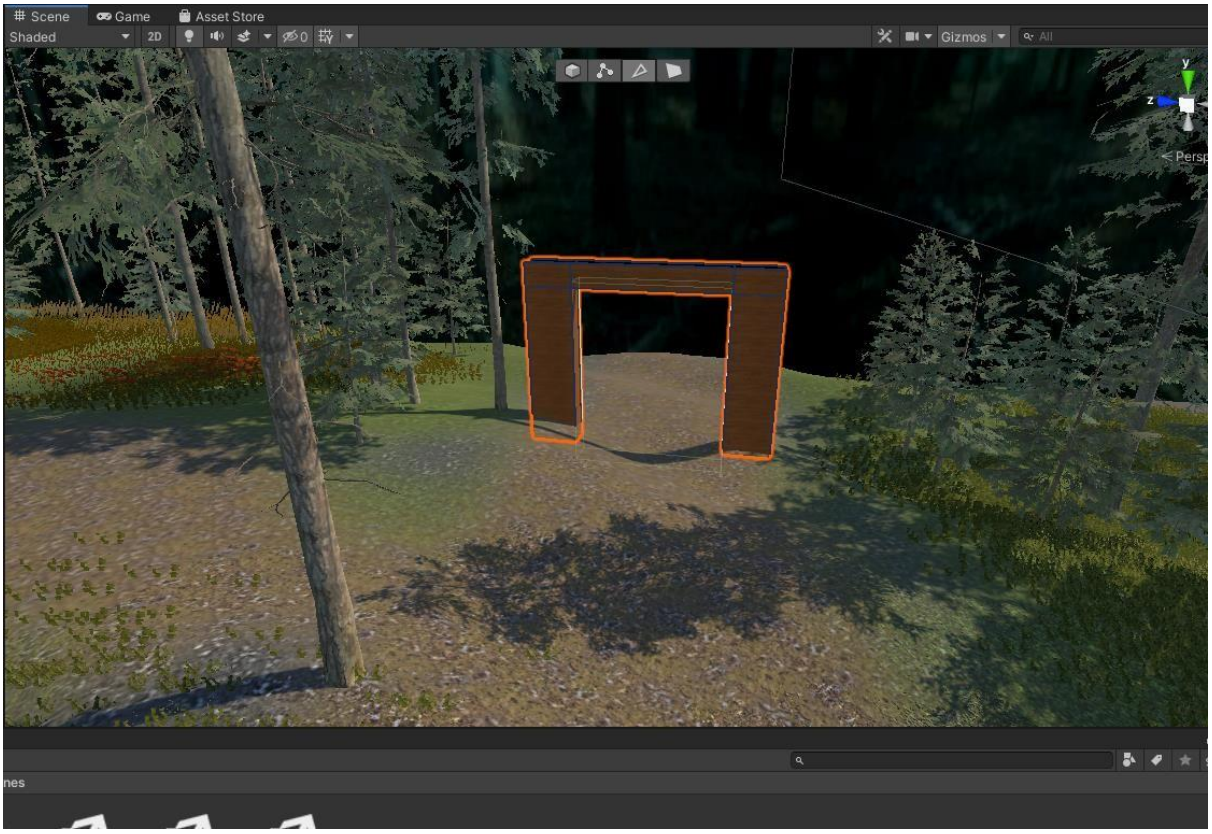


Step 7 : Now we can build more assets using Probuilder , like House , door and more.

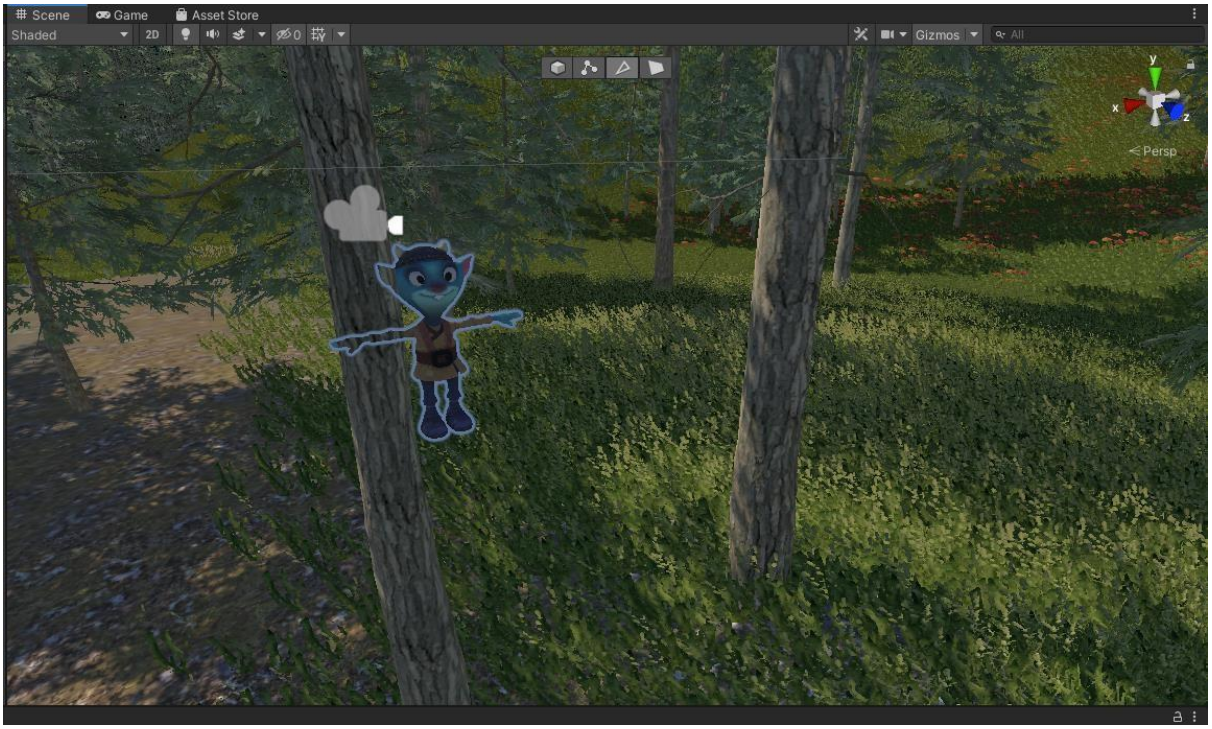
Step 8 : I've build some home and a Door for achieving completion of the game.



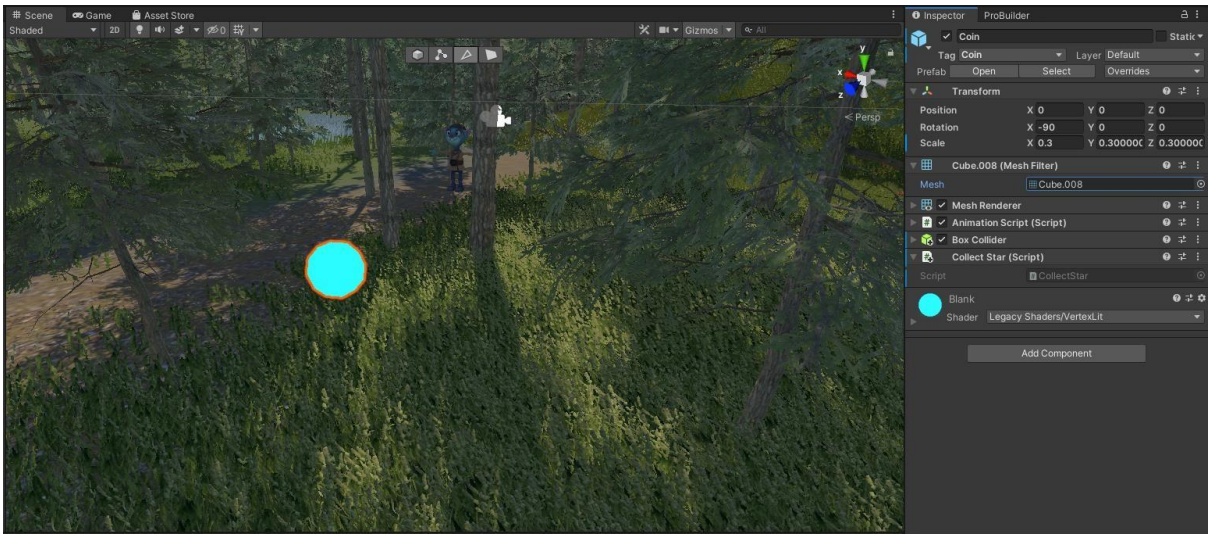
Step 9 : I've add a door for completion of the game.



Step 10 : now add a character to your game, I have download this character from internet with playercontroller.



Step 11 : Now add I have added some orbs with, which will increment our score. Those are just simple modals which I've downloaded.



Step 12 : create in game UI for score , Right click > UI > Canvas , it will create a canvas .

Step 13 : Now add some labels for level and score.

Step 14 : Now add particular scripts to particular components.

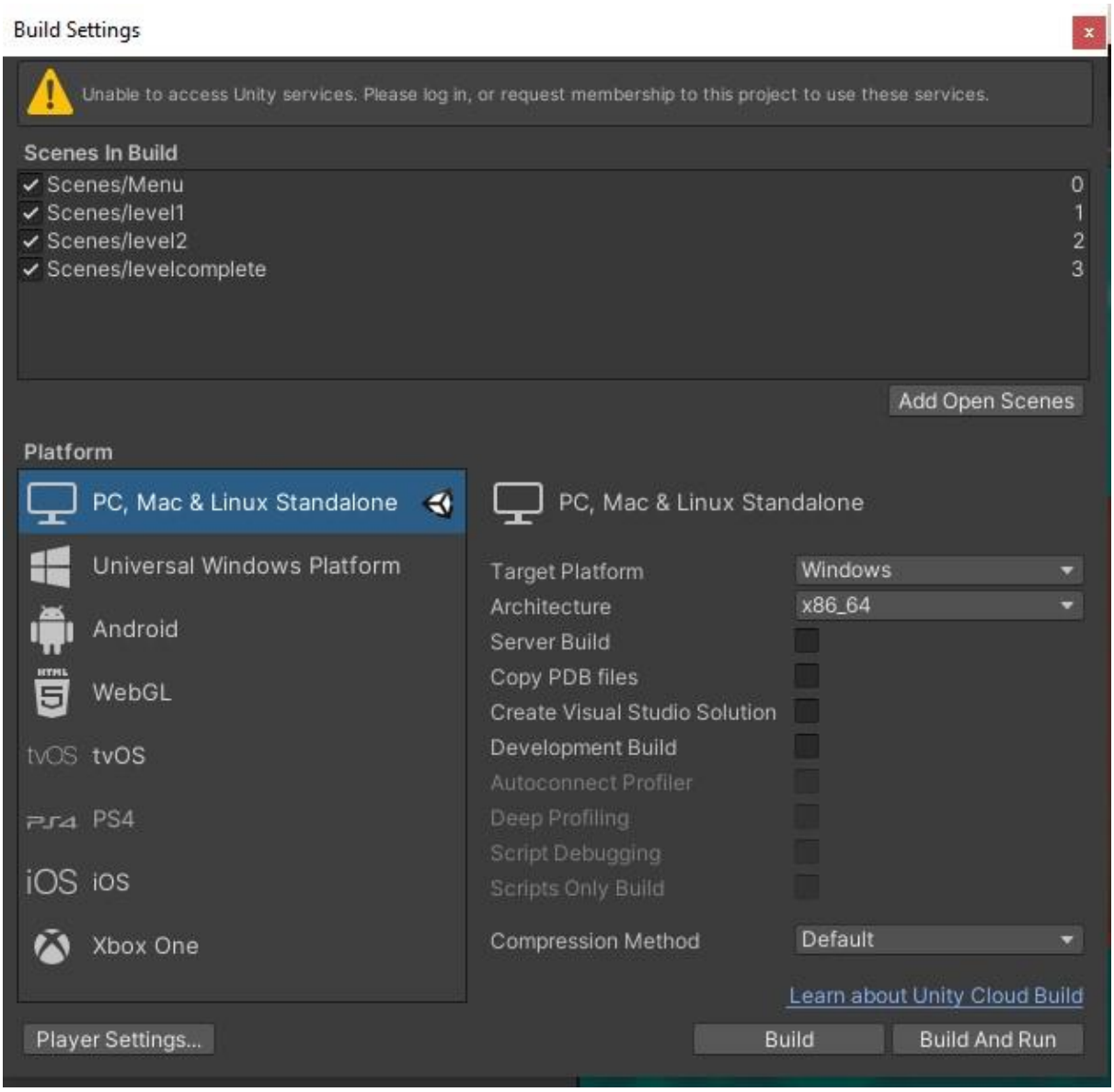
Creating Finish Menu:

Step 1 : Create a new scene and call it finish.

Step 2 : Right > UI > Canvas , it will create a new canvas

Step 3 : and repeat the same step we did while creating main menu , but this time putting label as level completed and a button to Next level.

Step 4 : now adding all scenes as indexing in build settings :



Scripts Used in the game :

CollectCoins.cs

```
using System.Collections; using
System.Collections.Generic;
using UnityEngine;
 public class CollectStar :
MonoBehaviour{


    void OnTriggerEnter(Collider other){
        ScoringSystem.theScore += 50;
        Destroy(gameObject);
    }


}
```

FinishGame.cs

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



public class levelup_script : MonoBehaviour
{     void OnTriggerEnter(Collider
other){         if
(other.CompareTag("Player")){


            SceneManager.LoadScene(3);
        }
    }
}
```

MainMenu.cs

```
using System.Collections; using
System.Collections.Generic; using
UnityEngine; using
UnityEngine.SceneManagement;
 public class menu_script :
MonoBehaviour
{
    // Start is called before the first frame update
public void Quit()
    {
      Debug.Log("quit pressed");
     // Application.Quit();
     UnityEditor.EditorApplication.isPlaying = false;
    }
     public void PlayGame(){
       SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
     }


}
```

Playermoment.cs

```
using System.Collections; using
System.Collections.Generic;
using UnityEngine;
 public class playermovement :
MonoBehaviour
{
    public CharacterController controller;
public Animator anim;


    public float speed = 0;
public float movespeed = 12f;
public float gravity = -9.81f;
public float jumpHeight = 3f;


    public Transform groundcheck;
public float groundDistance = 0.4f;
public LayerMask groundMask;
     public Vector3
velocity;     public
Vector3 move;
```

```
    bool isGrounded;     void Start(){
controller = GetComponent<CharacterController>();
anim = GetComponent<Animator>();
    }     void
Update()
    {                 isGrounded  =  Physics.CheckSphere(groundcheck.position,
groundDistance,  groundMask);
         if(isGrounded && velocity.y <
0){             velocity.y = -2f;
        }
         float x =
Input.GetAxis("Horizontal");
                 float z =
Input.GetAxis("Vertical");
         move = transform.right * x + transform.forward
*z;
                 if(move ==
Vector3.zero){
idel();
        }
else{
run();
        }                  move *= speed;
controller.Move(move * Time.deltaTime);

if(Input.GetButtonDown("Jump")){
jump();
        }


        velocity.y += gravity * Time.deltaTime;
         controller.Move(velocity *
Time.deltaTime);


    }
    private void run(){
speed = movespeed;
        anim.SetFloat("speed", 1f);
    }     private void idel(){
anim.SetFloat("speed", 0f);
    }     private void
jump(){
```

```
        velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity);
    }
}
```

Scoring.cs

```
using System.Collections; using
System.Collections.Generic;
using UnityEngine; using
UnityEngine.UI;
 public class ScoringSystem :
MonoBehaviour{    public GameObject
scoreText;    public static int theScore;
    void
Update()
   {     scoreText.GetComponent<Text>().text = "Score: " +
theScore;    }
}
```

SeeingMouse.cs

```
using System.Collections; using
System.Collections.Generic; using
UnityEngine;
 public class seeing :
MonoBehaviour
{
    public float mouseSenstivity = 100f;


    public Transform playerBody;
     float xRotation =
0f;


    void Start()
    {


    }
```

```
    // Update is called once per frame
    void Update()
    {
        float mouseX = Input.GetAxis("Mouse X") * mouseSenstivity *
Time.delta Time;
        float mouseY = Input.GetAxis("Mouse Y") * mouseSenstivity * Time.delta
Time;                   xRotation -= mouseY;
xRotation = Mathf.Clamp(xRotation, -90f, 90f);


        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
playerBody.Rotate(Vector3.up * mouseX);


   }
}
```