

ACTIVITY PERTEMUAN 5

NAMA : MUHAMMAD TARMIDZI BARIQ

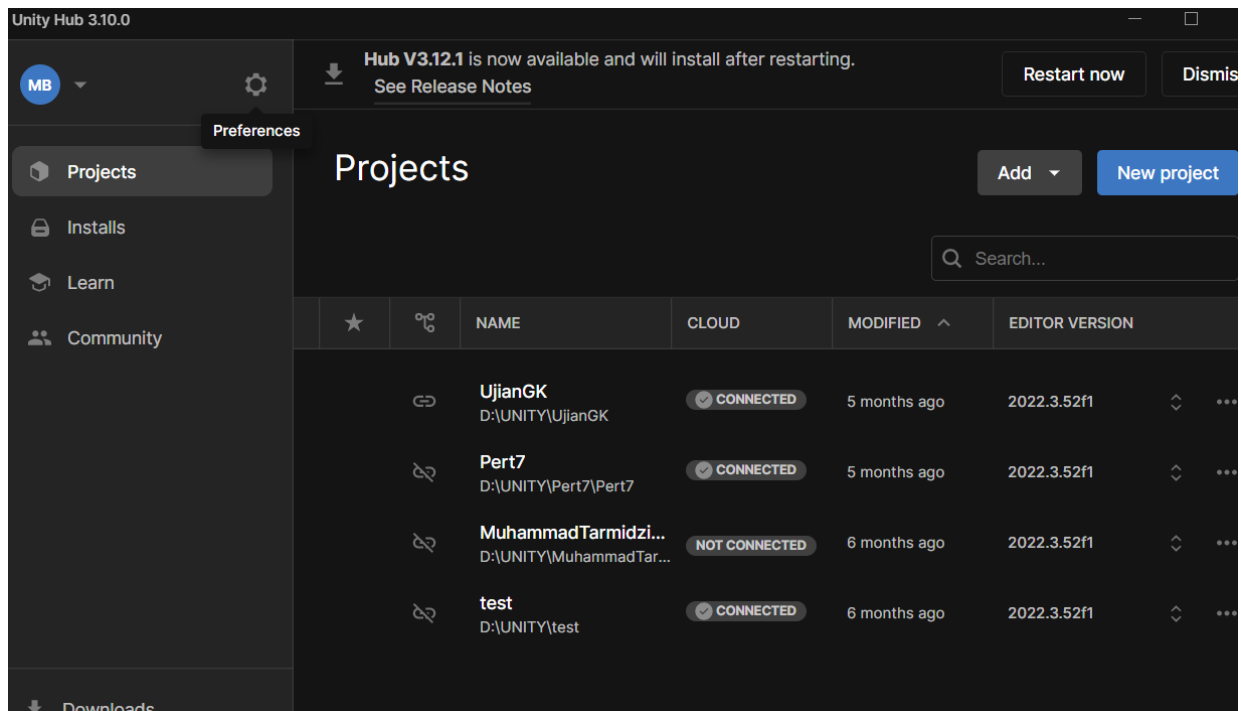
NPM : 51422161

KELAS : 3IA11

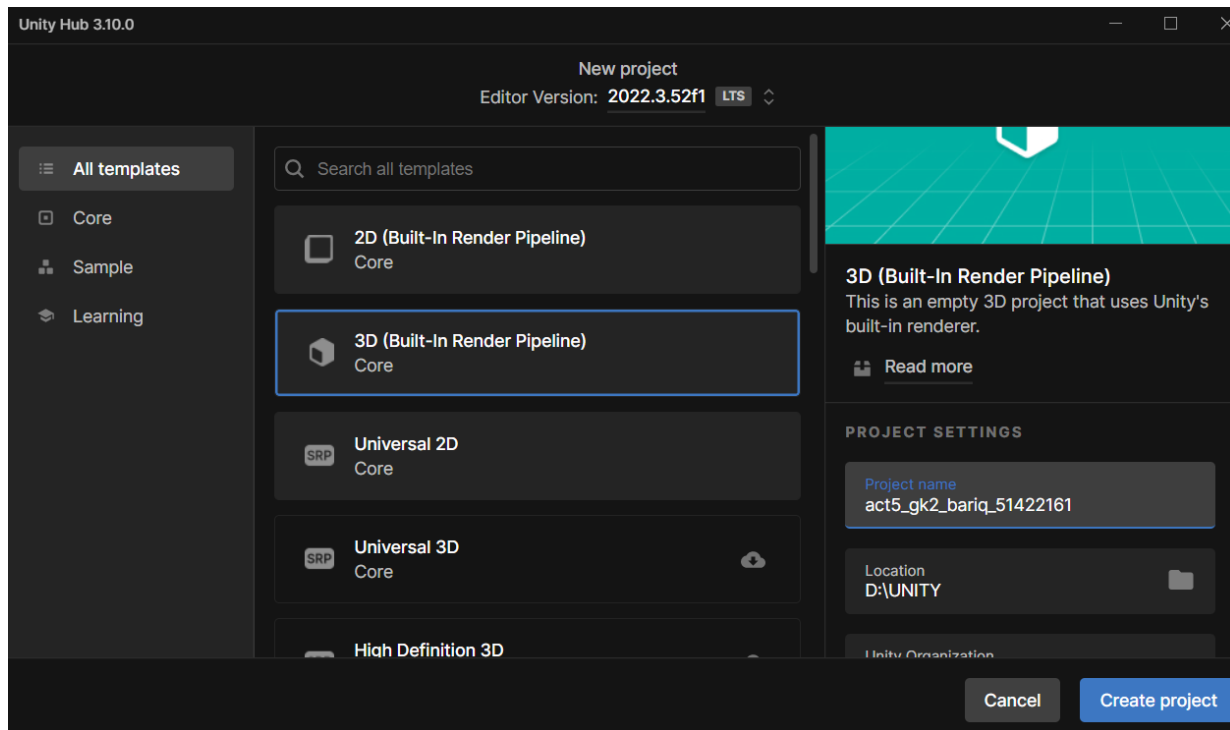
MATERI: Logika Game

MATA PRAKTIKUM : GRAFIK KOMPUTER 2

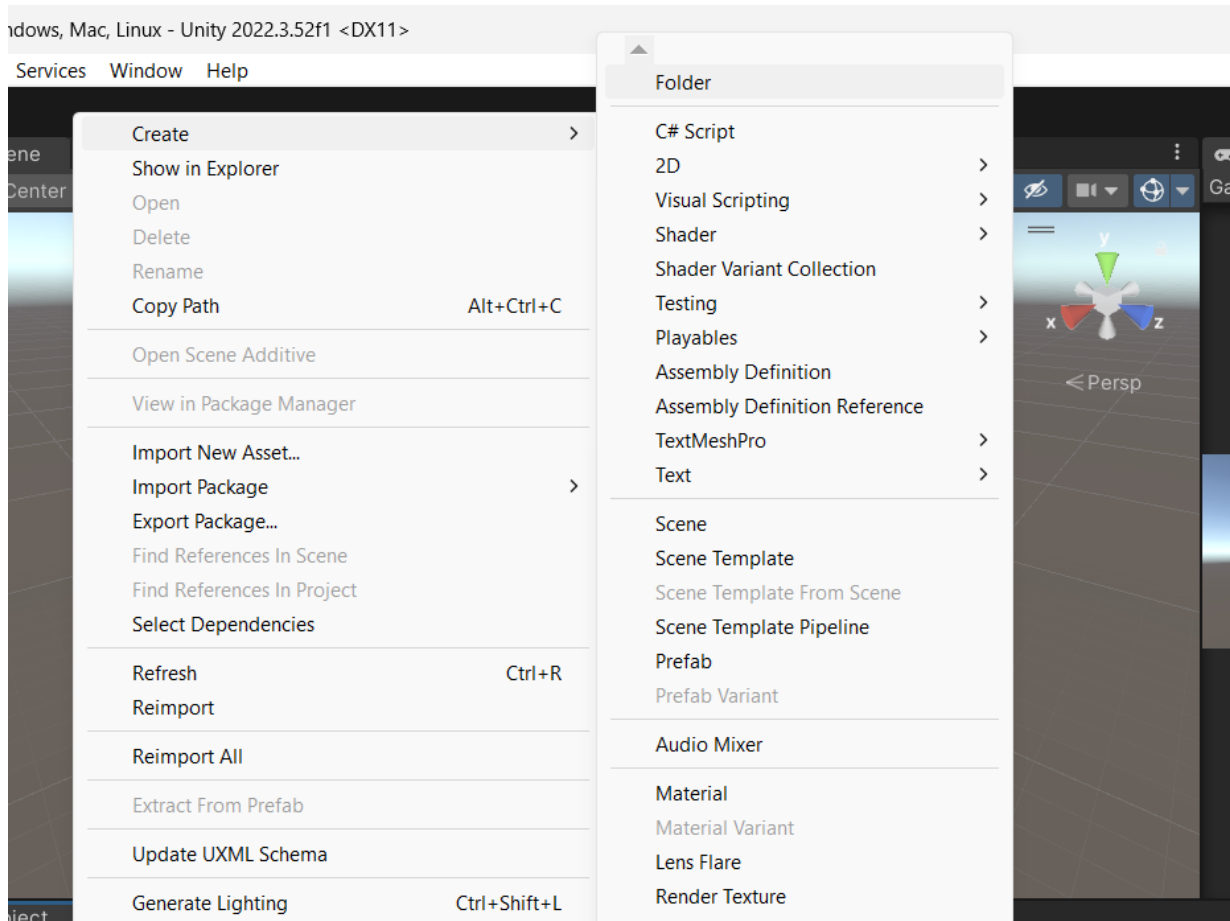
NEW PROJECT



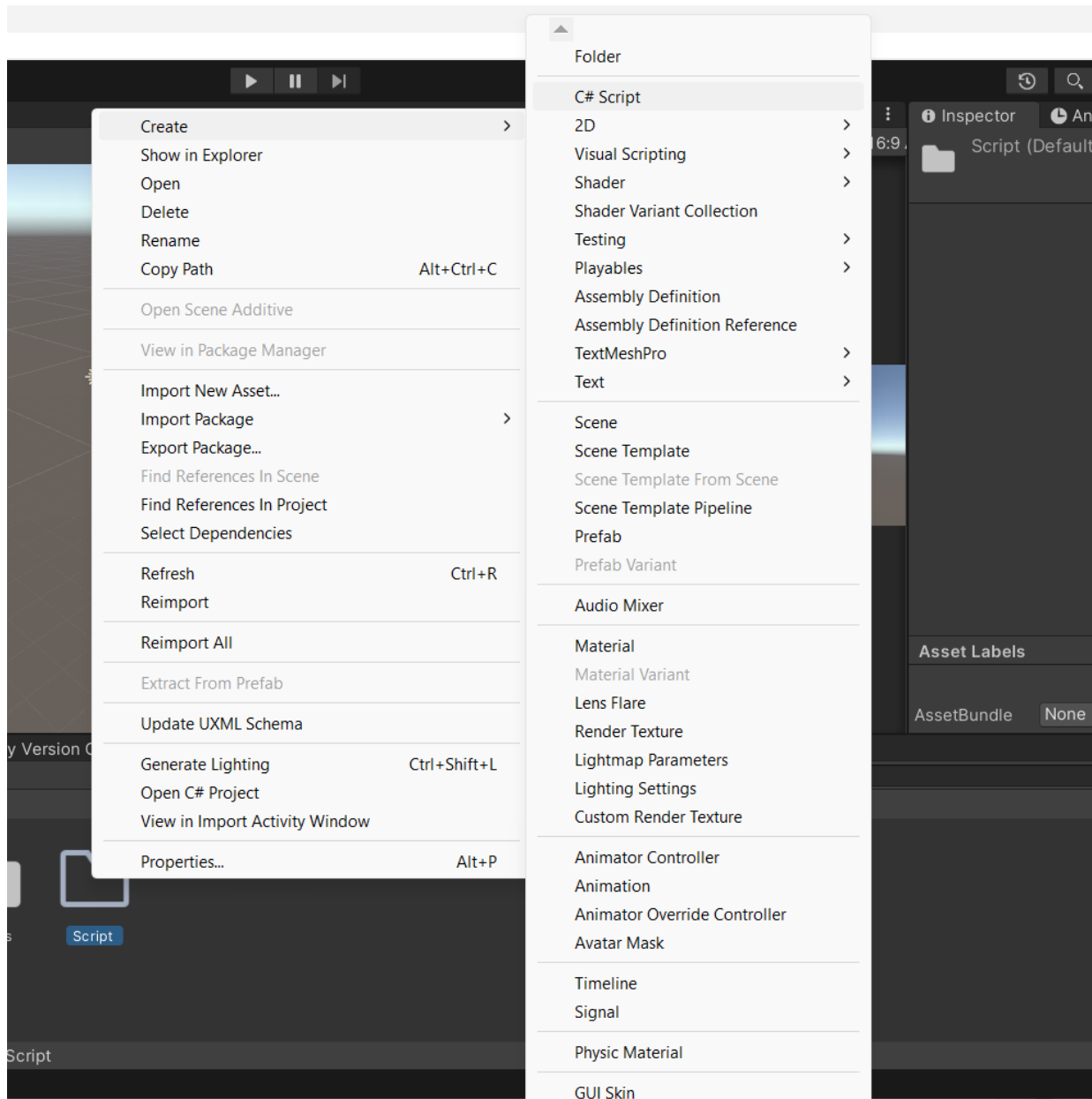
3D(built-in render Pipeline) -> berikan nama project -> create project



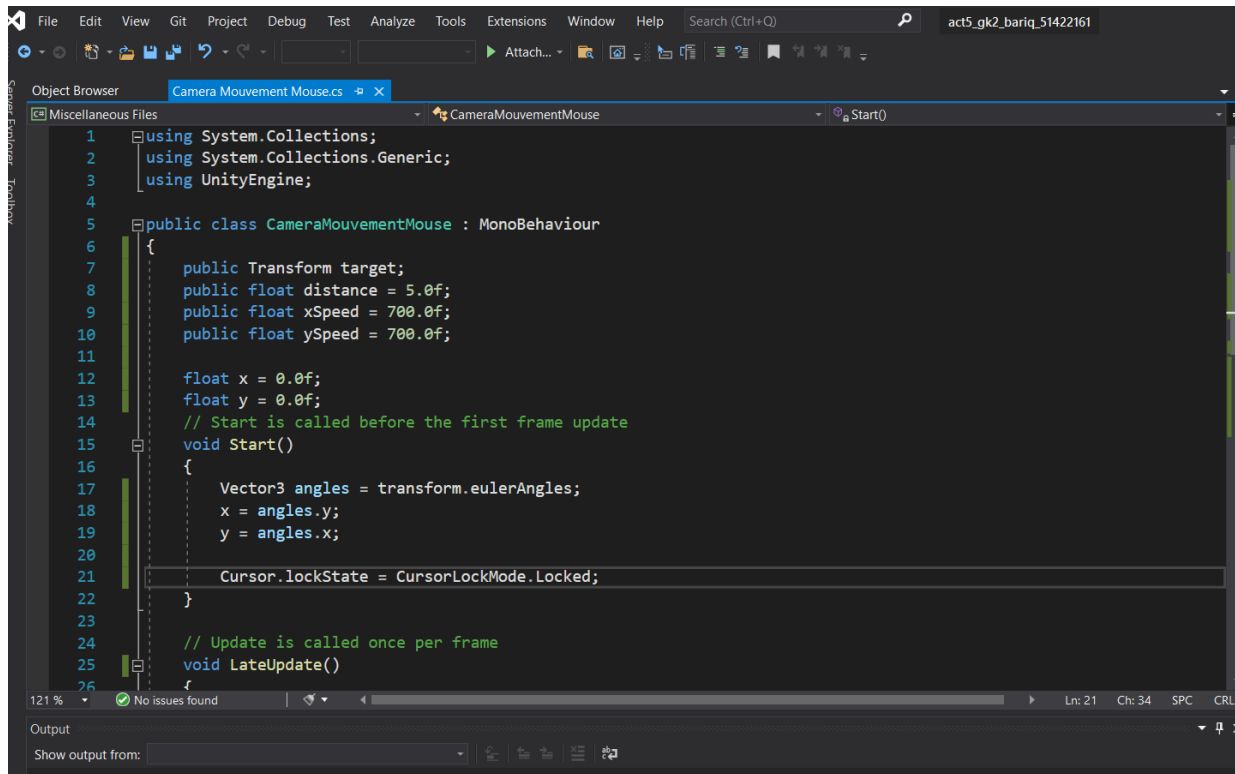
Klik kanan pada asset -> Create -> Folder -> kasih nama Scripting



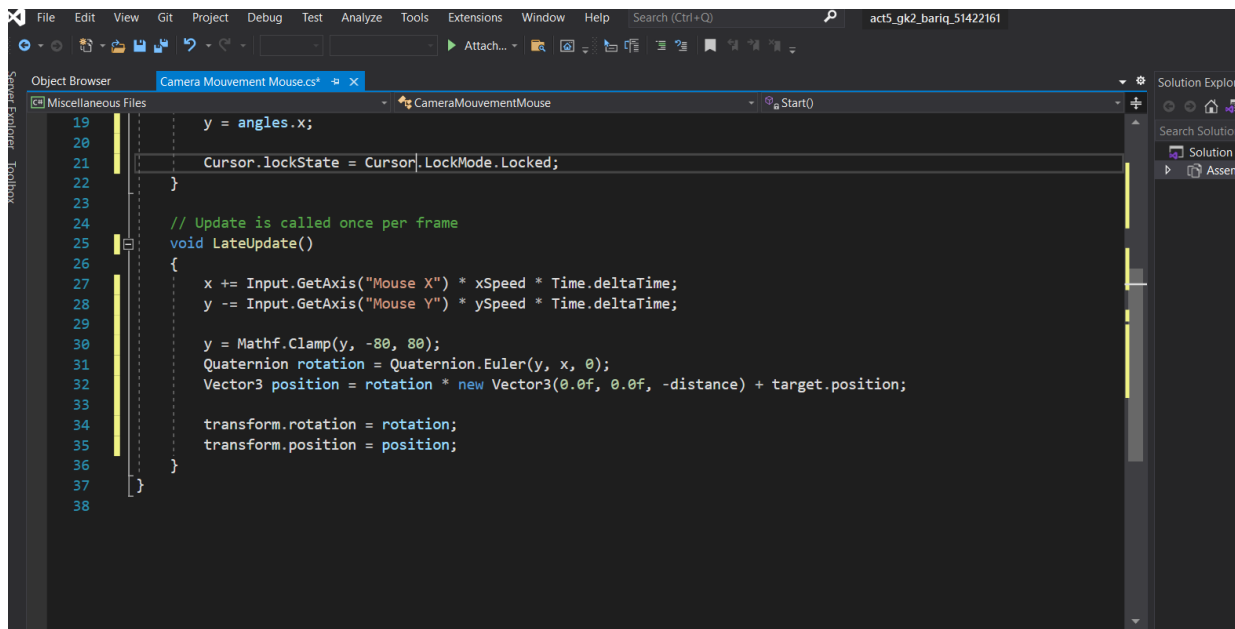
Pilih create -> C# Script -> kasih nama file Camera Mouvement Mouse



Ketik kode berikut dan jangan lupa disave

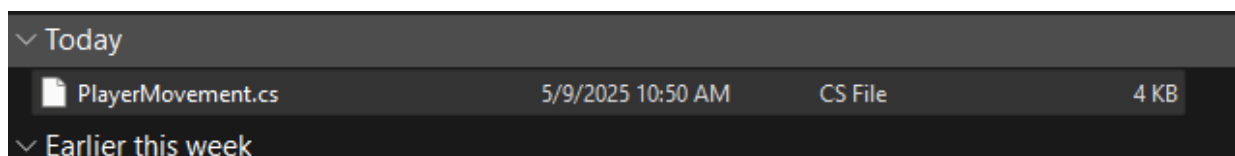


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class CameraMouvementMouse : MonoBehaviour
6 {
7     public Transform target;
8     public float distance = 5.0f;
9     public float xSpeed = 700.0f;
10    public float ySpeed = 700.0f;
11
12    float x = 0.0f;
13    float y = 0.0f;
14    // Start is called before the first frame update
15    void Start()
16    {
17        Vector3 angles = transform.eulerAngles;
18        x = angles.y;
19        y = angles.x;
20
21        Cursor.lockState = CursorLockMode.Locked;
22    }
23
24    // Update is called once per frame
25    void LateUpdate()
26    {
```

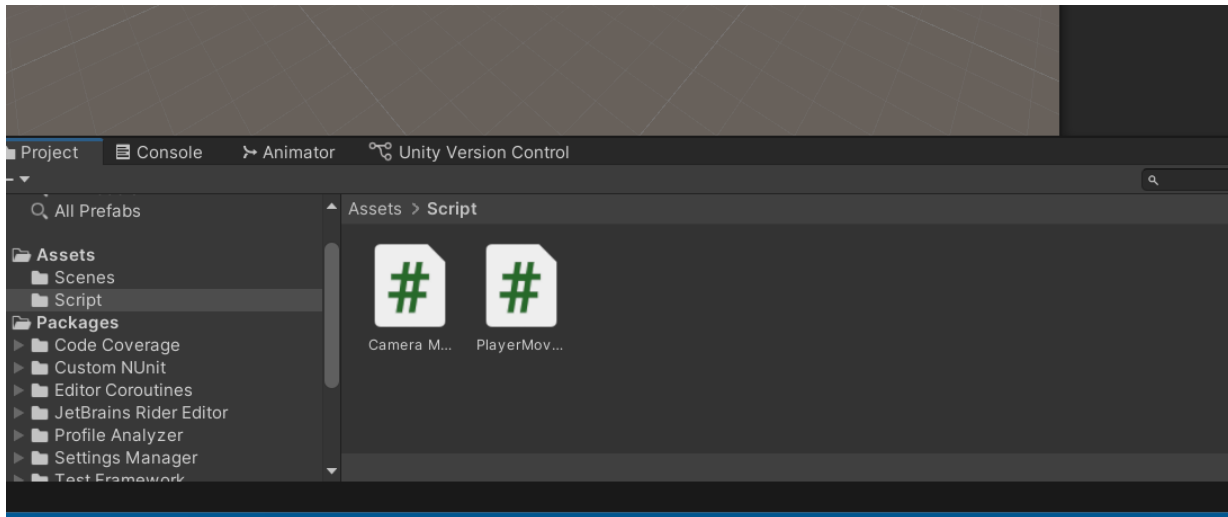


```
19     y = angles.x;
20
21     Cursor.lockState = CursorLockMode.Locked;
22 }
23
24 // Update is called once per frame
25 void LateUpdate()
26 {
27     x += Input.GetAxis("Mouse X") * xSpeed * Time.deltaTime;
28     y -= Input.GetAxis("Mouse Y") * ySpeed * Time.deltaTime;
29
30     y = Mathf.Clamp(y, -80, 80);
31     Quaternion rotation = Quaternion.Euler(y, x, 0);
32     Vector3 position = rotation * new Vector3(0.0f, 0.0f, -distance) + target.position;
33
34     transform.rotation = rotation;
35     transform.position = position;
36 }
37
38 }
```

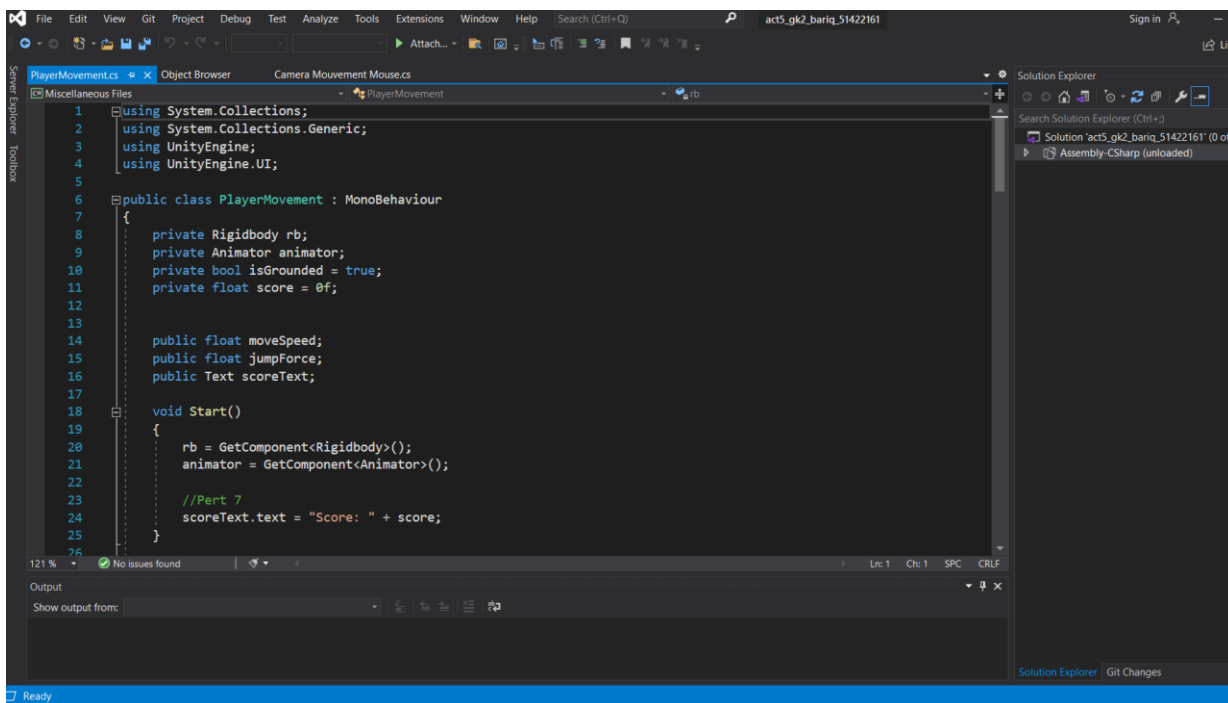
Drag n drop file tersebut ke unity asset

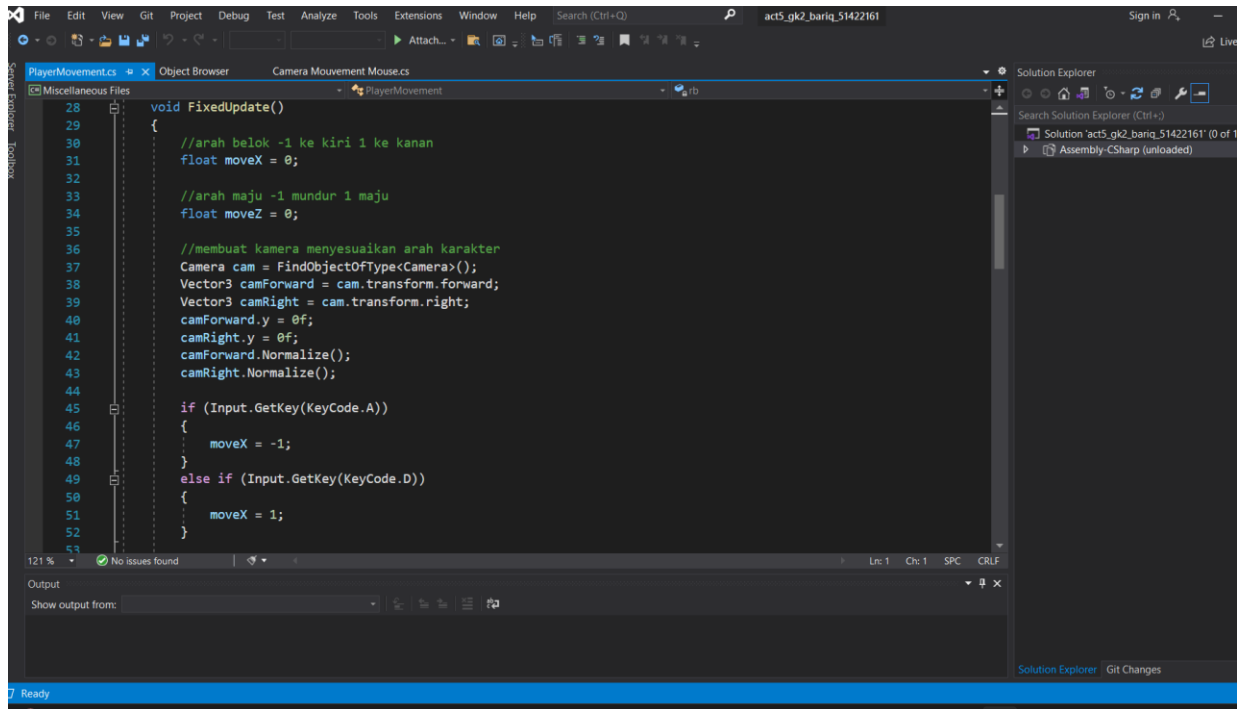


Klik 2x playerMovement akan terbuka code editor



Code playerMovement.cs





```
28 void FixedUpdate()
29 {
30     //arah belok -1 ke kiri 1 ke kanan
31     float moveX = 0;
32
33     //arah maju -1 mundur 1 maju
34     float moveZ = 0;
35
36     //membuat kamera menyesuaikan arah karakter
37     Camera cam = FindObjectOfType<Camera>();
38     Vector3 camForward = cam.transform.forward;
39     Vector3 camRight = cam.transform.right;
40     camForward.y = 0f;
41     camRight.y = 0f;
42     camForward.Normalize();
43     camRight.Normalize();
44
45     if (Input.GetKey(KeyCode.A))
46     {
47         moveX = -1;
48     }
49     else if (Input.GetKey(KeyCode.D))
50     {
51         moveX = 1;
52     }
53 }
```

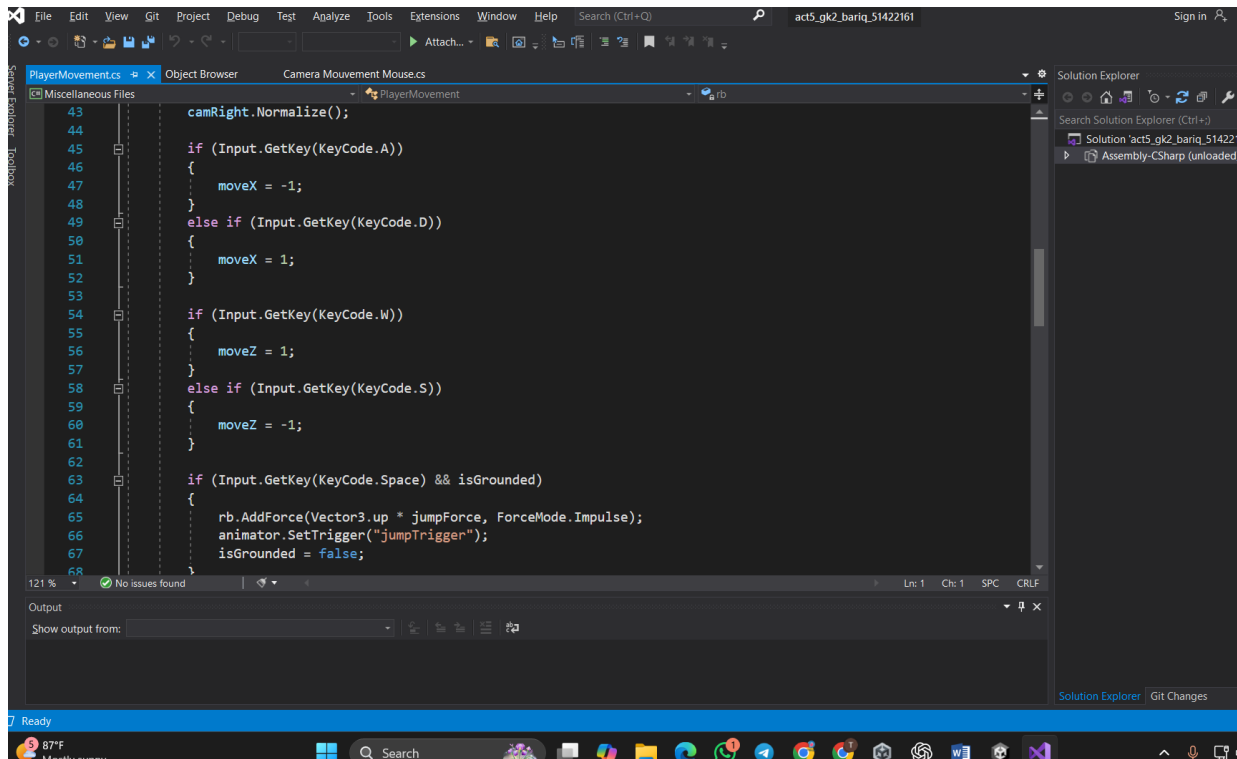
121 % No issues found

Output

Show output from:

Solution Explorer

Git Changes



```
43     camRight.Normalize();
44
45     if (Input.GetKey(KeyCode.A))
46     {
47         moveX = -1;
48     }
49     else if (Input.GetKey(KeyCode.D))
50     {
51         moveX = 1;
52     }
53
54     if (Input.GetKey(KeyCode.W))
55     {
56         moveZ = 1;
57     }
58     else if (Input.GetKey(KeyCode.S))
59     {
60         moveZ = -1;
61     }
62
63     if (Input.GetKey(KeyCode.Space) && isGrounded)
64     {
65         rb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
66         animator.SetTrigger("jumpTrigger");
67         isGrounded = false;
68     }
69 }
```

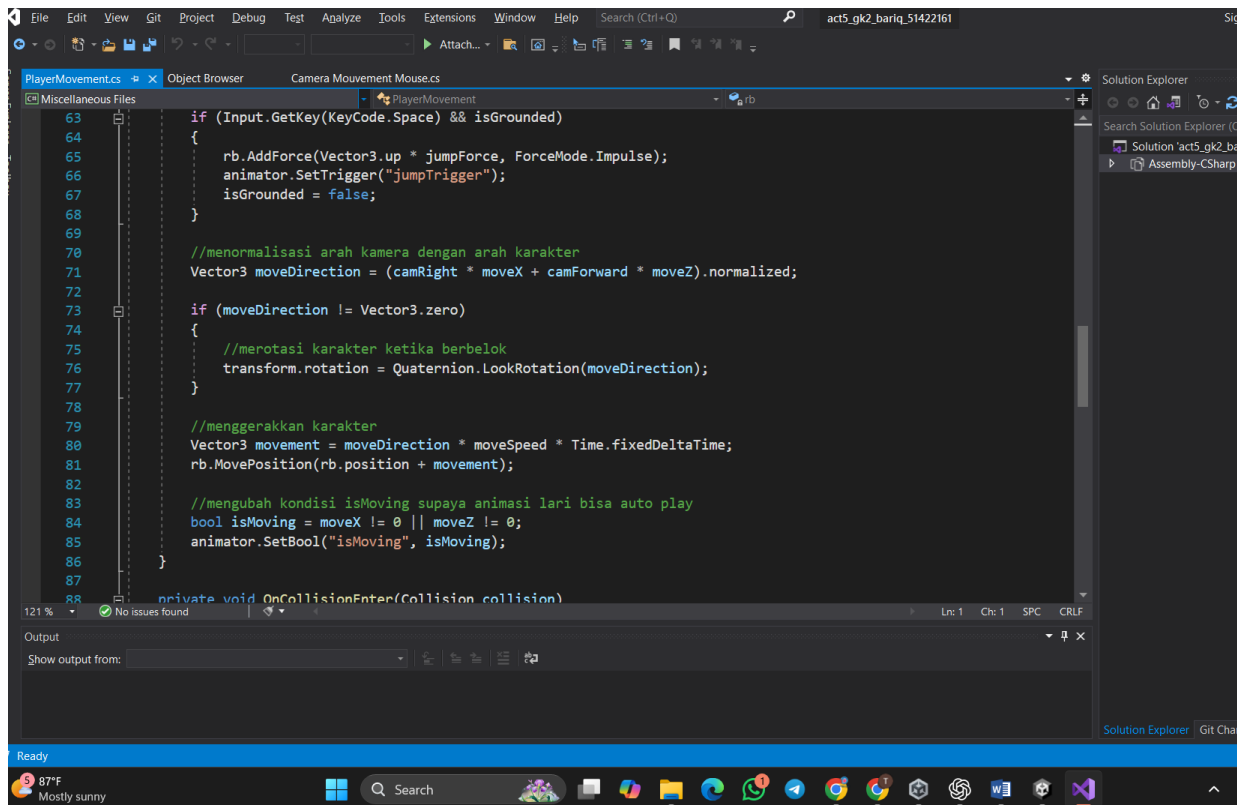
121 % No issues found

Output

Show output from:

Solution Explorer

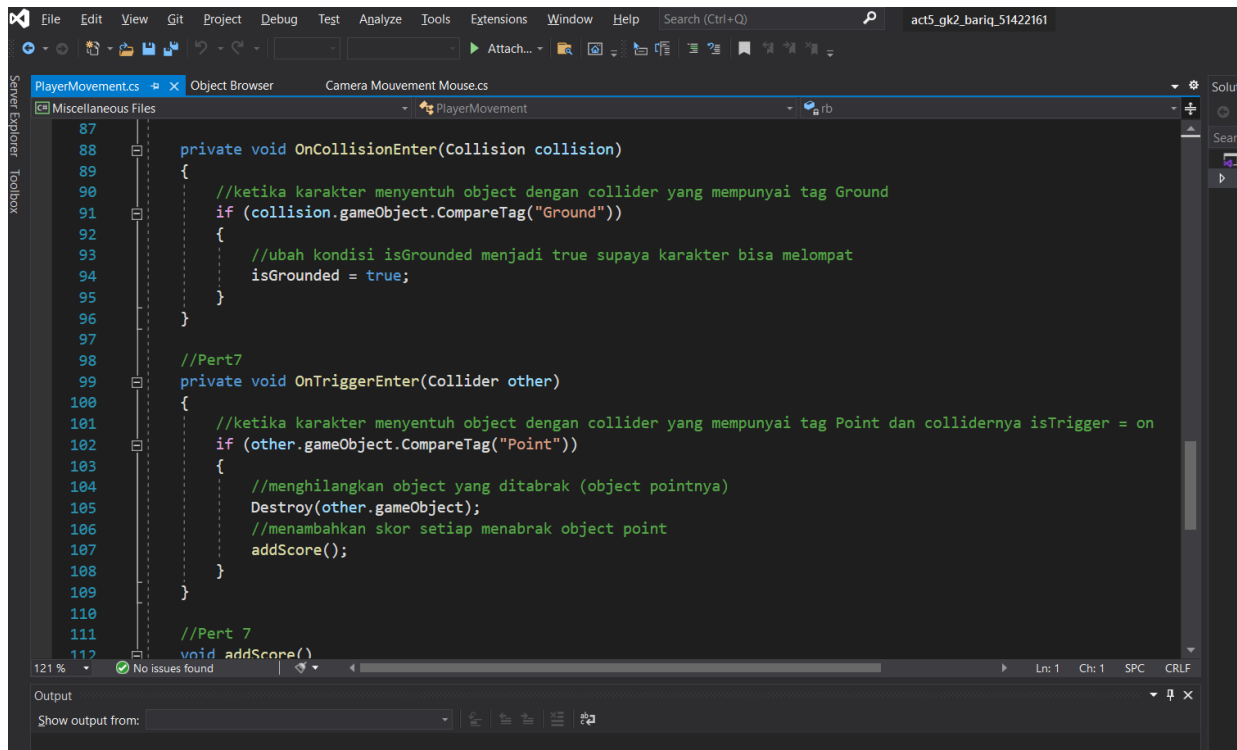
Git Changes



This screenshot shows the Visual Studio IDE with the `PlayerMovement.cs` file open. The code implements logic for a character's movement, including jumping and directional movement. The `Jump` method is triggered by the Space key, adding an upward force and setting a trigger. The `Move` method calculates a normalized movement direction from camera input and rotates the character accordingly. The `OnCollisionEnter` method is partially visible at the bottom.

```
63     if (Input.GetKey(KeyCode.Space) && isGrounded)
64     {
65         rb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
66         animator.SetTrigger("jumpTrigger");
67         isGrounded = false;
68     }
69
70     //menormalisasi arah kamera dengan arah karakter
71     Vector3 moveDirection = (camRight * moveX + camForward * moveZ).normalized;
72
73     if (moveDirection != Vector3.zero)
74     {
75         //merotasi karakter ketika berbelok
76         transform.rotation = Quaternion.LookRotation(moveDirection);
77     }
78
79     //menggerakkan karakter
80     Vector3 movement = moveDirection * moveSpeed * Time.fixedDeltaTime;
81     rb.MovePosition(rb.position + movement);
82
83     //mengubah kondisi isMoving supaya animasi lari bisa auto play
84     bool isMoving = moveX != 0 || moveZ != 0;
85     animator.SetBool("isMoving", isMoving);
86 }
87
88 private void OnCollisionEnter(Collision collision)
```

The bottom of the image shows the Windows taskbar with a weather widget indicating 87°F and 'Mostly sunny'.



This screenshot shows the continuation of the `PlayerMovement.cs` file. It details the `OnCollisionEnter` and `OnTriggerEnter` methods. The collision logic checks for a 'Ground' tag to set `isGrounded` to true. The trigger logic checks for a 'Point' tag, destroys the trigger object, and increments a score.

```
87
88 private void OnCollisionEnter(Collision collision)
89 {
90     //ketika karakter menyentuh object dengan collider yang mempunyai tag Ground
91     if (collision.gameObject.CompareTag("Ground"))
92     {
93         //ubah kondisi isGrounded menjadi true supaya karakter bisa melompat
94         isGrounded = true;
95     }
96 }
97
98 //Pert7
99 private void OnTriggerEnter(Collider other)
100 {
101     //ketika karakter menyentuh object dengan collider yang mempunyai tag Point dan collidernya isTrigger = on
102     if (other.gameObject.CompareTag("Point"))
103     {
104         //menghilangkan object yang ditabrak (object pointnya)
105         Destroy(other.gameObject);
106         //menambahkan skor setiap menabrak object point
107         addScore();
108     }
109 }
110
111 //Pert 7
112 void addScore()
```

```
109     }
110
111     //Pert 7
112     void addScore()
113     {
114         score++;
115         scoreText.text = "Score: " + score;
116     }
117 }
118
```

121 %  No issues found

Output