



TUGAS PERTEMUAN: 8

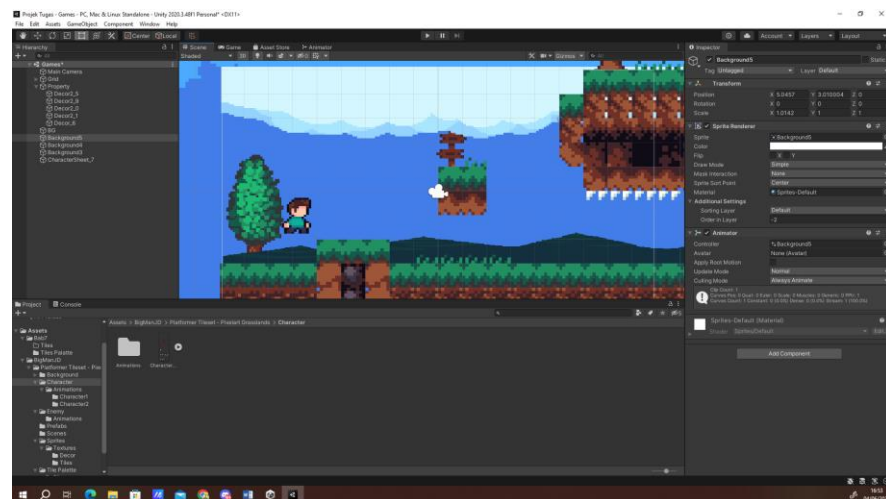
CAMERA & CHARACTER MOVEMENT

NIM	:	2118043
Nama	:	Kurniawan Yoga Pratama
Kelas	:	C
Asisten Lab	:	Difa Fisabililah (2118052)

8.1 Tugas 1: Character Movement

A. Buat Pergerakan Karakter

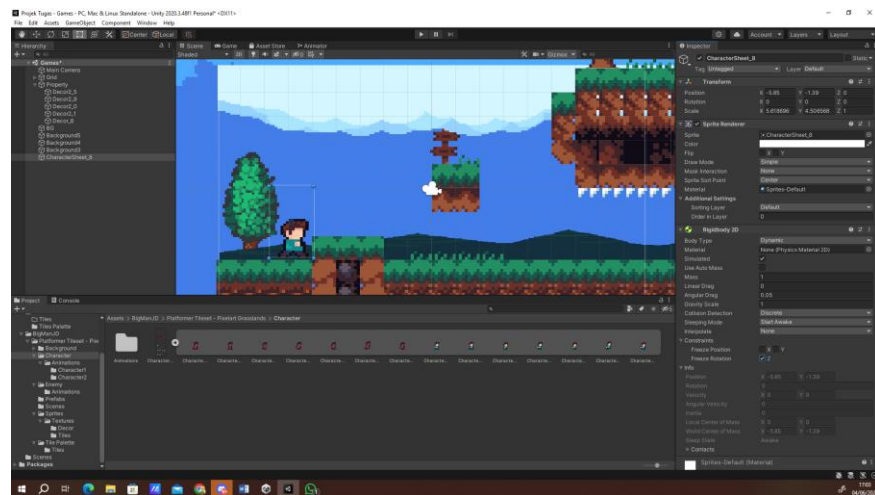
1. Buka kembali *project* yang kemarin kita buat melalui *Unity HUB*



8.1 Tampilan Membuka File Unity

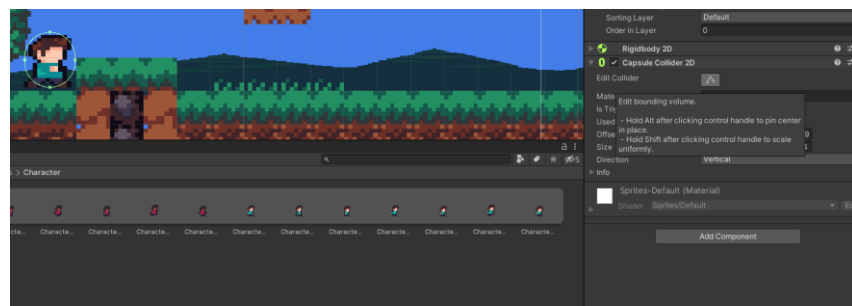


2. Tambahkan player idle 1 lalu tambahkan *component rigidbody 2d*, lalu pada *constraints* klik *freeze rotation Z*.



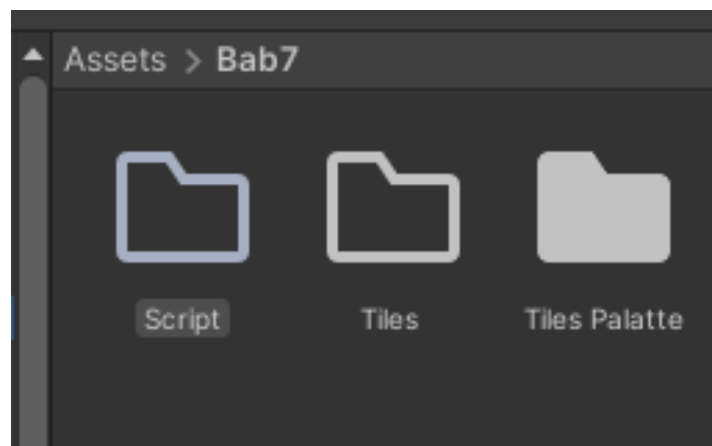
8.2 Tampilan Menambahkan Player Idle

3. Tambahkan juga *capsule collider 2d*. lalu letakan capsule agar pas pada player idle nya



Gambar 8.3 Tampilan Menambahkan *Componen Capsule Collider 2d*

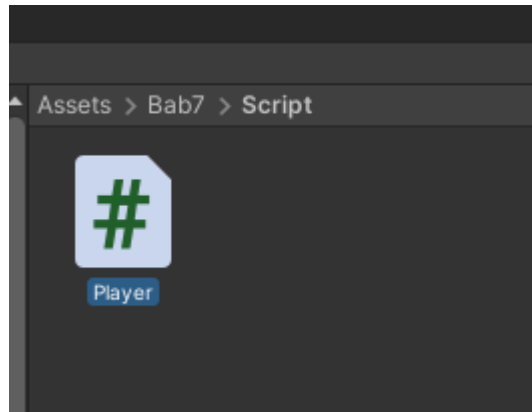
4. Buat folder baru pada bab 7 bernama *Script*.



Gambar 8.4 Tampilan Create Folder Script



5. Didalam Folder Script buat *C#Script* yang bernama *Player*.



Gambar 8. 5 Tampilan Createa Scene

6. *Drag and drop* *C#script* tadi pada *player*, lalu klik 2x *script* untuk membuka *code editor vscode*. Lalu tambahkan *source code* seperti berikut

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

public class Player : MonoBehaviour
{
    Rigidbody2D rb;

    [SerializeField] float speed = 1;
    float horizontalValue;
    bool facingRight;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update ()
    {
        horizontalValue = Input.GetAxisRaw("Horizontal");
    }

    void FixedUpdate()
    {
        Move(horizontalValue);
    }

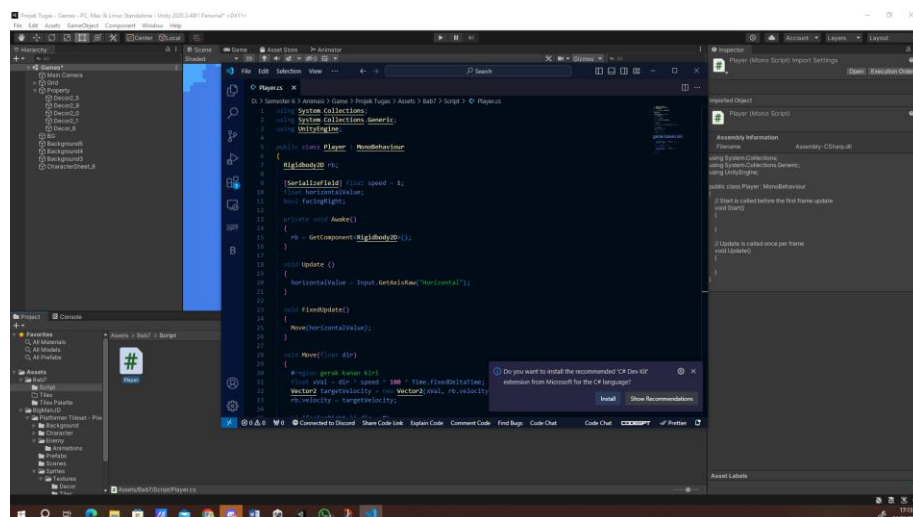
    void Move(float dir)
    {
        #region gerak kanan kiri
        float xVal = dir * speed * 100 *
Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
        rb.velocity = targetVelocity;
```



```
if (facingRight && dir < 0)
{
    // ukuran player
    transform.localScale = new Vector3(-1, 1, 1);
    facingRight = false;
}

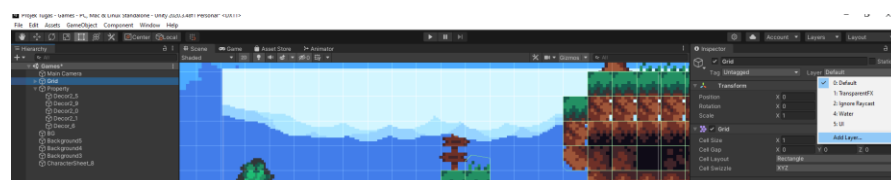
else if (!facingRight && dir > 0)
{
    // ukuran player
    transform.localScale = new Vector3(1, 1, 1);
    facingRight = true;
}

#endregion
}
```



Gambar 8. 6Tampilan Code Editor

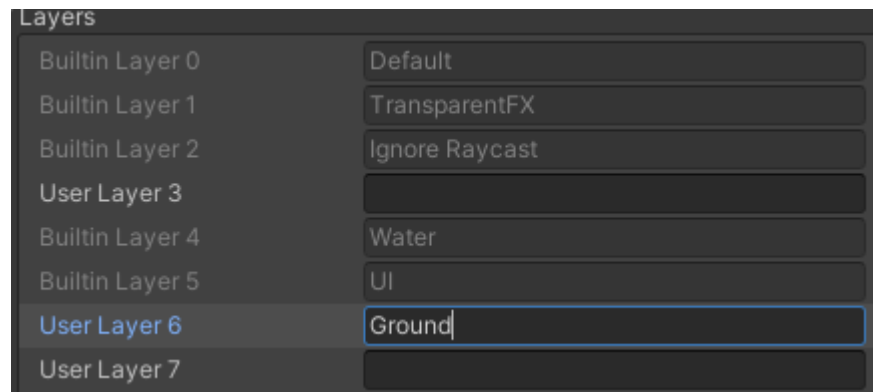
7. Untuk membuat player loncat menggunakan space, perlu dibuat groundcheck dengan klik grid pada pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer.



Gambar 8. 7 Tampilan Add Layer

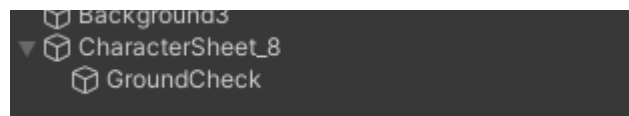


- Pilih Layer 6 untuk groundnya



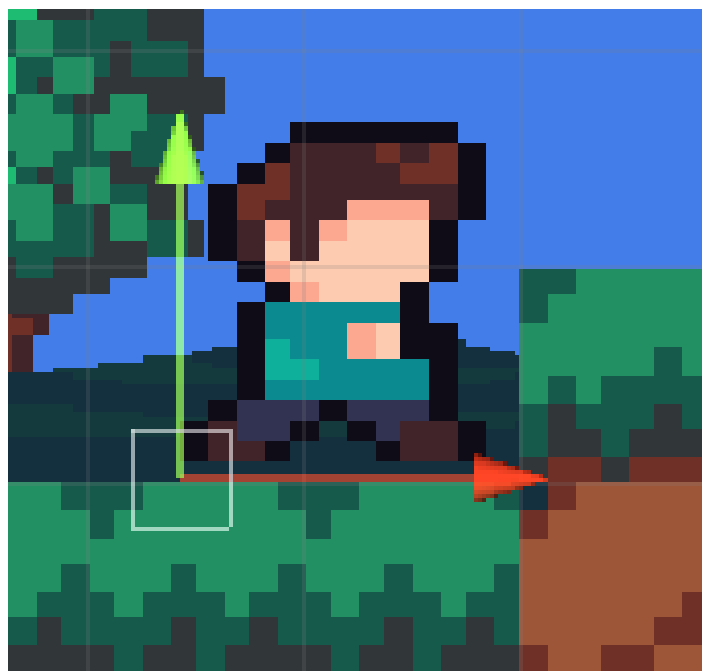
Gambar 8. 8Tampilan User Layer 6 Ground

- Klik *player* lalu *create empty* lalu beri nama *GroundCheck*



Gambar 8.9 Tampilan Create GroundCheck

- Klik groundcheck lalu gunakan move tool lalu posisikan seperti pada gambar dibawah ini.



Gambar 8.10 Tampilan Menempatkan *Move Tool*

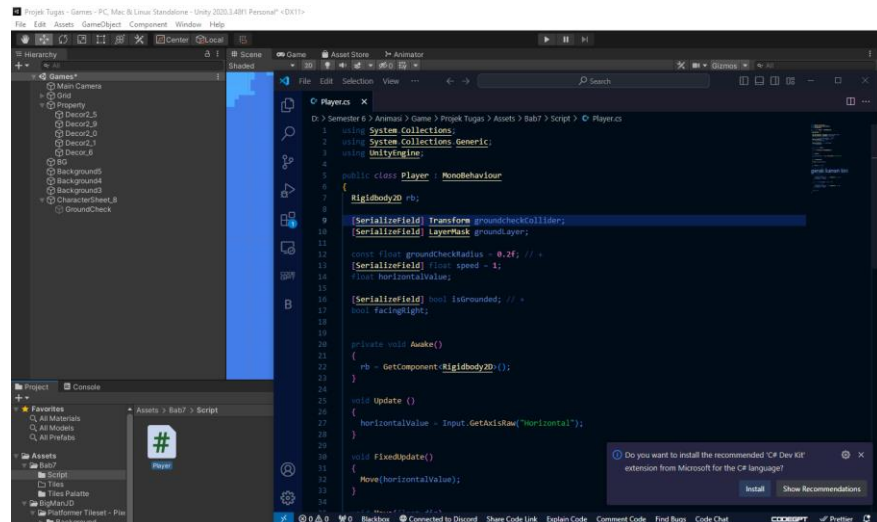


11. Buka kembali Script untuk untuk menambahkan Source Code seperti berikut

```
[SerializeField] Transform groundcheckCollider;
[SerializeField] LayerMask groundLayer;

const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
float horizontalValue;

[SerializeField] bool isGrounded; // +
bool facingRight;
```



Gambar 8. 11 Tampilan Menambahkan Script Part 1

12. Tambahkan juga source code seperti ini pada void fixedUpdate lalu buat fuction baru void groundcheck seperti berikut

```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue);
}

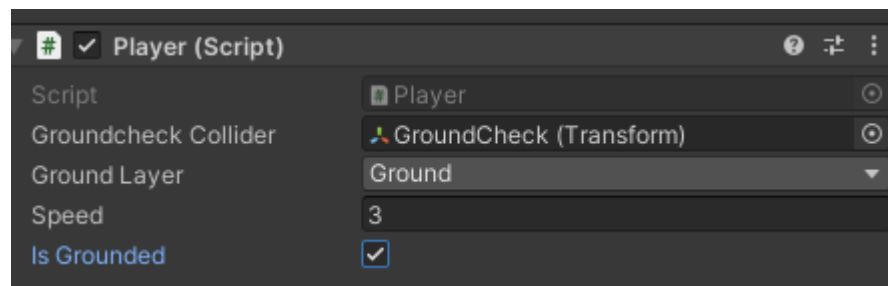
void GroundCheck()
{
    isGrounded = false;
    Collider2D[] colliders =
Physics2D.OverlapCircleAll(groundcheckCollider.position,
groundCheckRadius, groundLayer);
    if (colliders.Length > 0)
        isGrounded = true;
}
```



```
Player.cs
6 {
15
16 [SerializeField] bool isGrounded; // +
17 bool facingRight;
18
19
20 private void Awake()
21 {
22     rb = GetComponent<Rigidbody2D>();
23 }
24
25 void Update ()
26 {
27     horizontalValue = Input.GetAxisRaw("Horizontal");
28 }
29
30 void FixedUpdate()
31 {
32     GroundCheck();
33     Move(horizontalValue);
34 }
35
36 void GroundCheck()
37 {
38     isGrounded = false;
39     Collider2D[] colliders = Physics2D.OverlapCircleAll(groundCheckCollider.position, groundCheckRadius,
40     if (colliders.Length > 0)
41     isGrounded = true;
42 }
43 }
```

Gambar 8. 12Tampilan Menambahkan Script Part 2

13. Klik Player lalu pergi ke inspector lalu sesuaikan semua seperti pada gambar dibawah ini.



Gambar 8. 13 Tampilan Menambahkan Script Part 3

14. Tambahkan source code seprti ini untuk membuat player melompat.

```
[SerializeField] float jumpPower = 100;

bool jump;
```

```
const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
[SerializeField] float jumpPower = 100;
float horizontalValue;

[SerializeField] bool isGrounded; // +
bool facingRight;
bool jump;
```

Gambar 8. 14 Tampilan Menambahkan Script Part 4



15. Tambahkan source code seperti ini pada void update.

```
if (Input.GetButtonDown("Jump"))
    jump = true;
else if (Input.GetButtonUp("Jump"))
    jump = false;
```

```
void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
    if (Input.GetButtonDown("Jump"))
        jump = true;
    else if (Input.GetButtonUp("Jump"))
        jump = false;
}
```

Gambar 8. 15 Tampilan Menambahkan Script Part 5

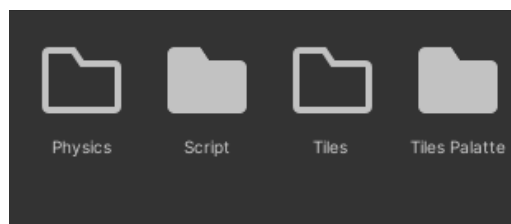
16. Tambahkan source code seperti ini pada void move

```
void Move(float dir, bool jumpflag)
{
    if(isGrounded && jumpflag)
    {
        isGrounded = false;
        jumpflag = false;
        rb.AddForce(new Vector2(0f, jumpPower));
    }
}
```

```
void Move(float dir, bool jumpflag)
{
    if(isGrounded && jumpflag)
    {
        isGrounded = false;
        jumpflag = false;
        rb.AddForce(new Vector2(0f, jumpPower));
    }
}
```

Gambar 8. 1 Tampilan Menambahkan Script Part 6

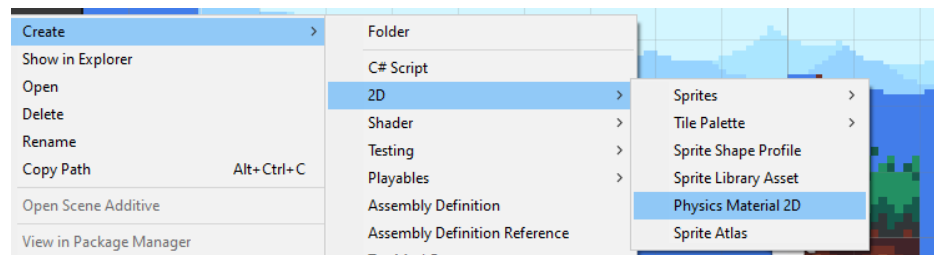
17. Pada Bab 7 Buat folder physics



Gambar 8. 2 Tampilan Memanbahkan folder Physics

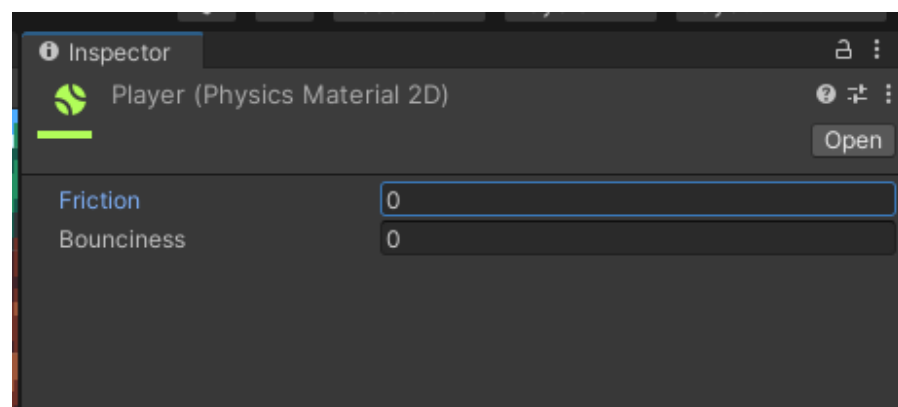


18. Didalam folder physics tersebut create > 2d> phycics material 2d, lalu beri nama Player.



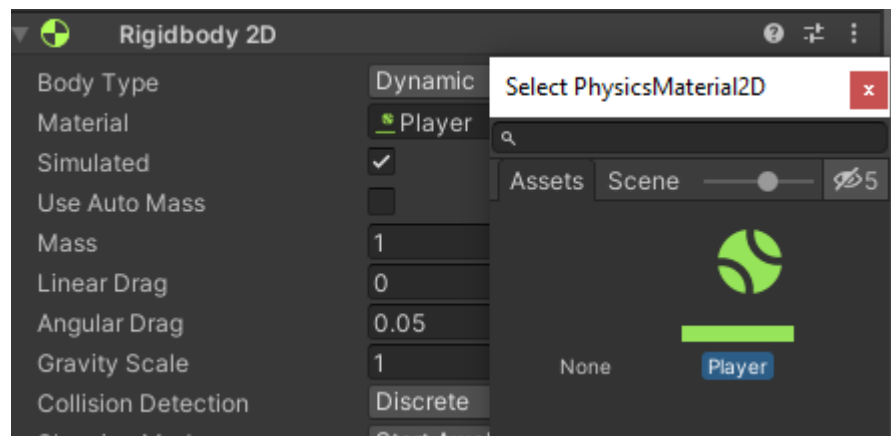
Gambar 8. 18 Tampilan Menambahkan Physics Material 2D

19. Klik tadi lalu ubah isi nya menjadi 0 semua.



Gambar 8. 19 Tampilan Ubah Friction

20. Klik palyer lalu pada component rigidbody 2D pilih player

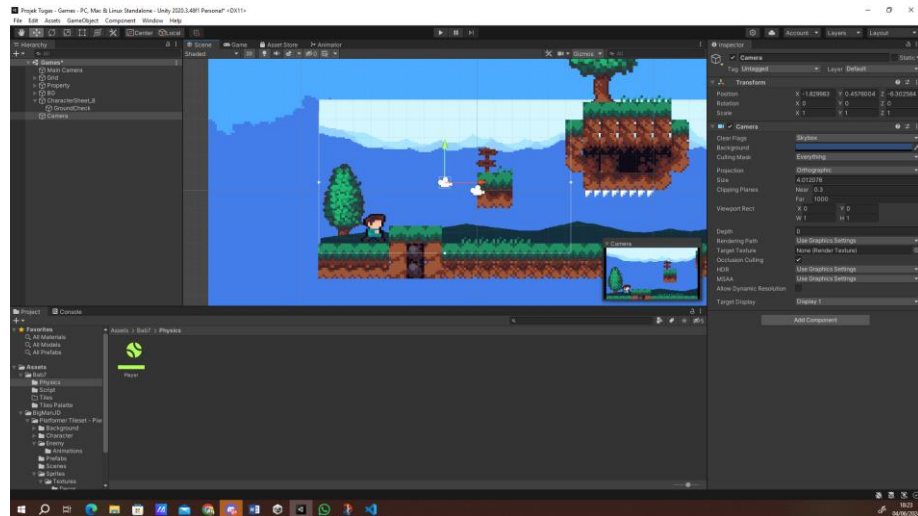


Gambar 8. 20 Tampilan Memilih Material



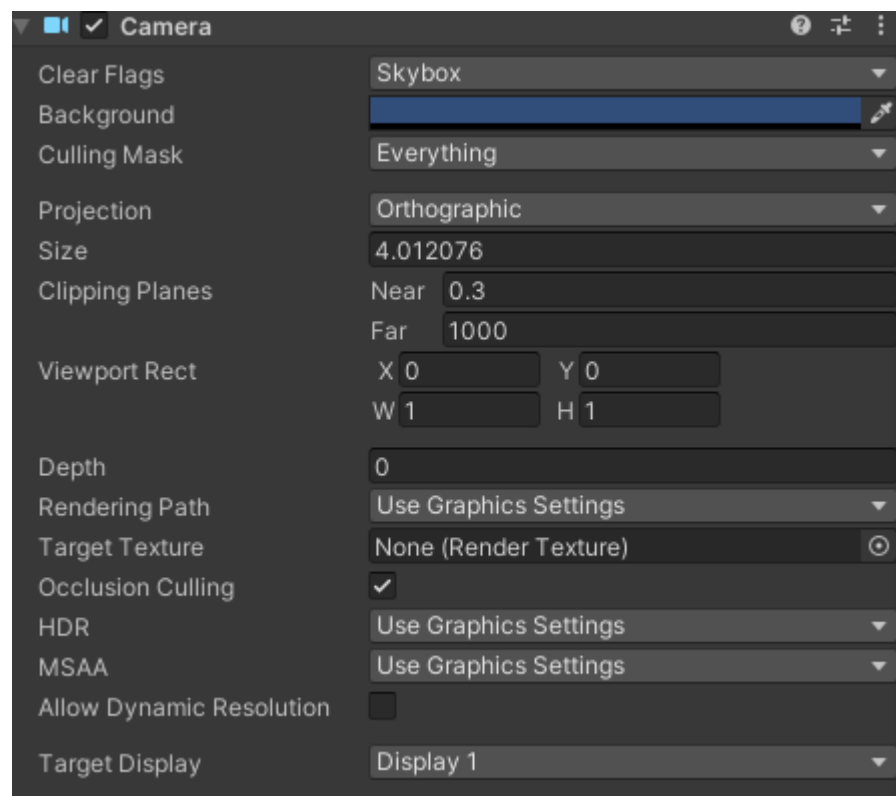
B. Camera Movement

21. Pada hierarchy create empty lalu beri nama camera, lalu pada inspector tambahkan component camera.



Gambar 8. 21 Tampilan Add Component Camera

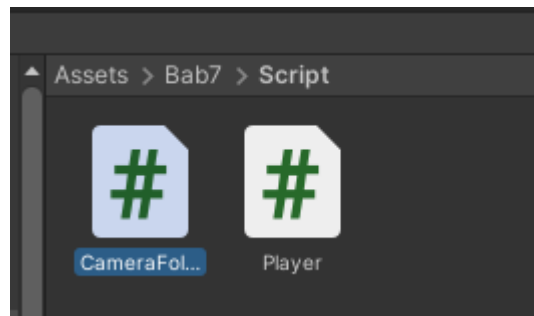
22. Lalu sama kan setingan dengan gambar dibawah ini.



Gambar 8. 22 Tampilan Isi dari comoponent camera

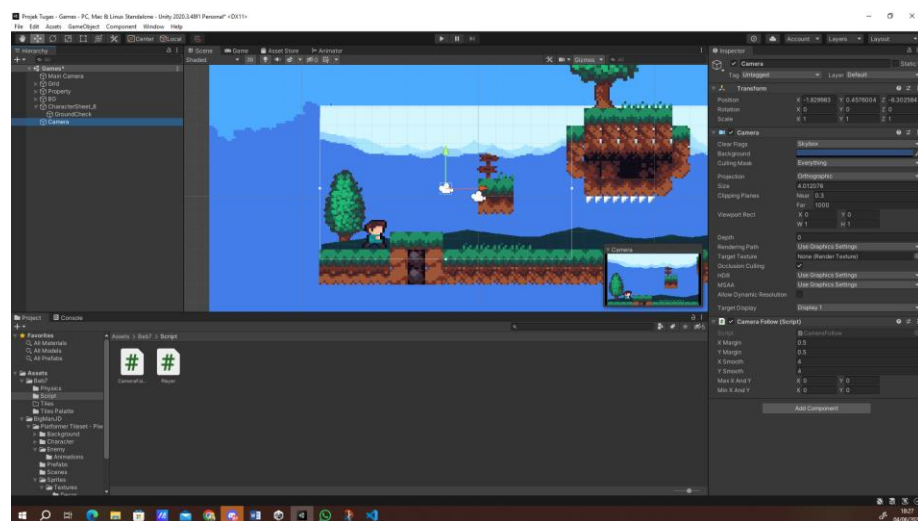


23. Buat script dengan nama CameraFollow.



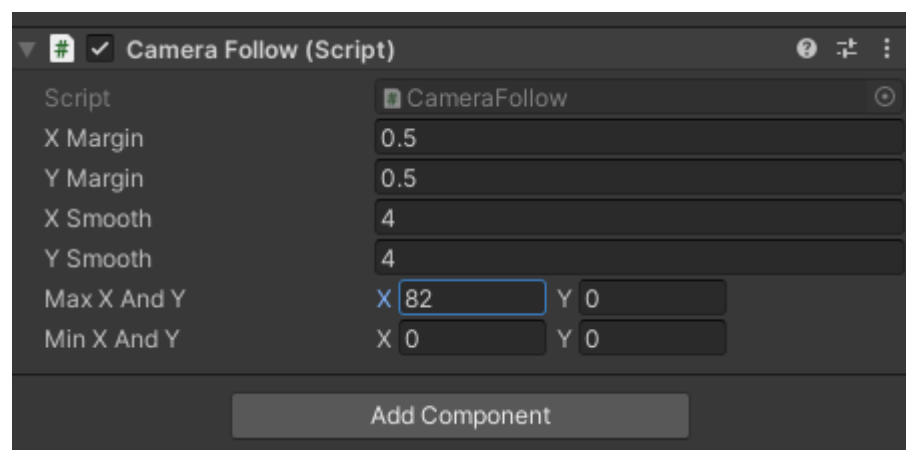
Gambar 8. 23 Tampilan Menambahkan Script CameraFollow

24. Drag and Drop script tadi pada camera.



Gambar 8. 24 Tampilan Drop Dawn Script Pada Camera

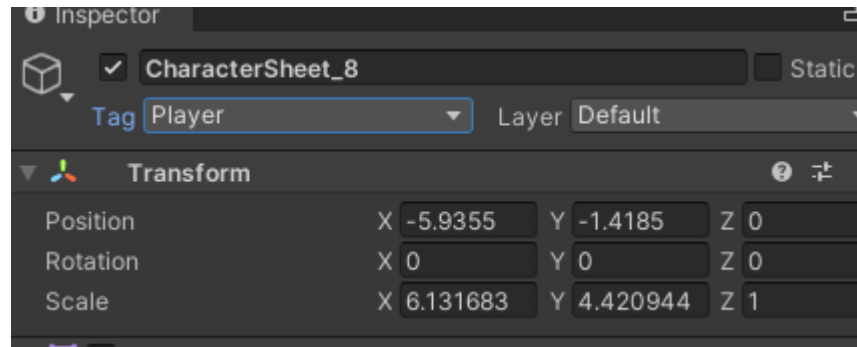
25. Klik pada camera lalu pergi ke inspector, ubah bagian max X and Y menjadi 82.



Gambar 8. 25 Tampilan ubah Max X dan Y

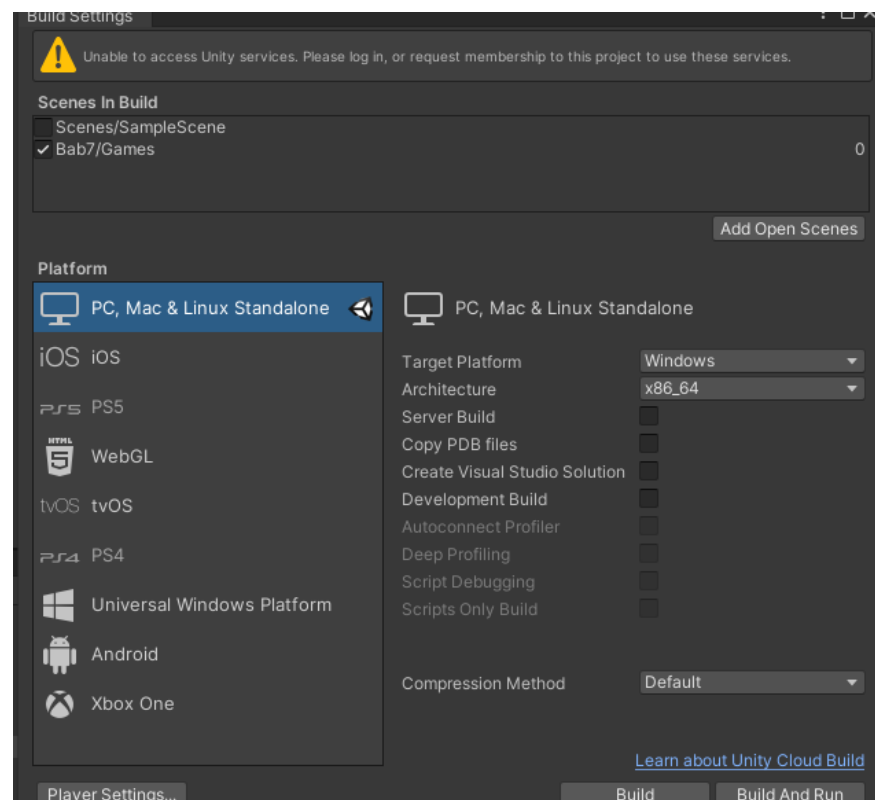


26. Klik Player ubag tag menjadi Player



Gambar 8 .28 Tampilan Merubah Tah Player

27. Langkah Terakhir tinggal render maka selesai.



Gambar 8. 28 Tampilan Render build Setting

C. Kuis: Camera Follow

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

public class CameraFollow : MonoBehaviour
{
    [SerializeField] private Transform player;

    void Update() {
```



```
transform.position = new Vector3 (player.  
position.x, transform.position.y, transform.position.z);  
}  
}
```

Penjelasan:

Sintaks dalam kode di atas adalah untuk mengikuti pergerakan pemain (player) dalam permainan. Baris `transform.position = new Vector3(player.position.x, transform.position.y, transform.position.z);` mengatur posisi kamera agar selalu sejajar dengan sumbu X pemain, tetapi tetap mempertahankan posisi Y dan Z kamera. Dengan demikian, kamera akan mengikuti pemain secara horizontal

Link Github Pengumpulan

https://github.com/Yktama18/2118043_PRAK_ANIGAME.git