



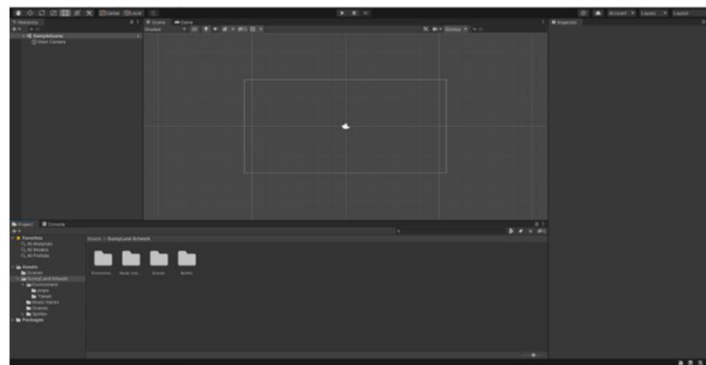
TUGAS PERTEMUAN: 8

Camera & Character Movement

NIM	:	2118019
Nama	:	Januarizky Bintang Astira
Kelas	:	B
Asisten Lab	:	Natasya Octavia (2118034)

1.1 Tugas 1 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

1. Buka file proyek Unity sebelumnya pada bab 7 untuk digunakan kembali



Gambar 8.1 Membuka Project Unity

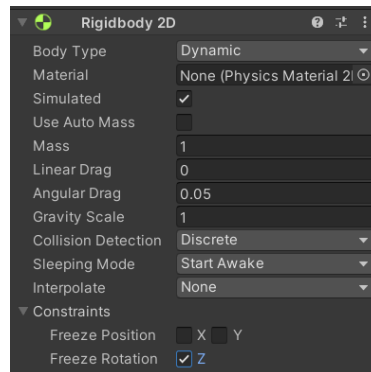
2. Tambahkan pemain bernama character, impor ke dalam Hirarki



Gambar 8.2 Import Character



3. Klik character, tambahkan Komponen Rigidbody 2D, sesuaikan pengaturannya seperti gambar berikut, dan centang opsi Freeze Rotation Z.



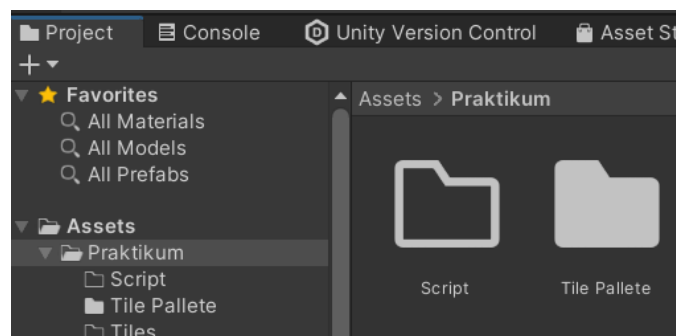
Gambar 8.3 Menambahkan Komponen Rigidbody 2D

4. Kemudian sesuaikan bentuk oval dengan karakter tersebut atau masukkan nilai Offset X, Y serta Size X, Y sesuai kebutuhan



Gambar 8.4 Menambahkan Capsule Collider

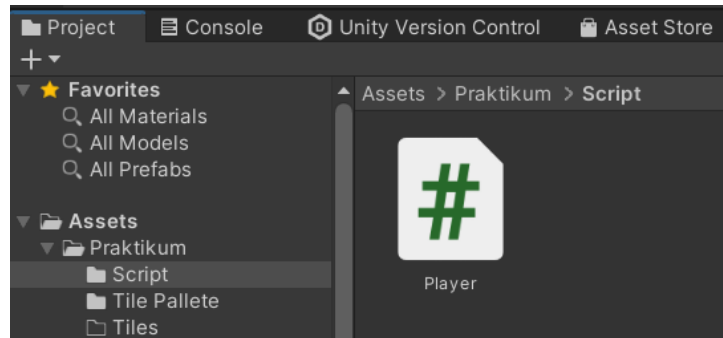
5. Buka Folder praktikum, kemudian buat folder baru dengan nama Script.



Gambar 8.5 Membuat Folder Script

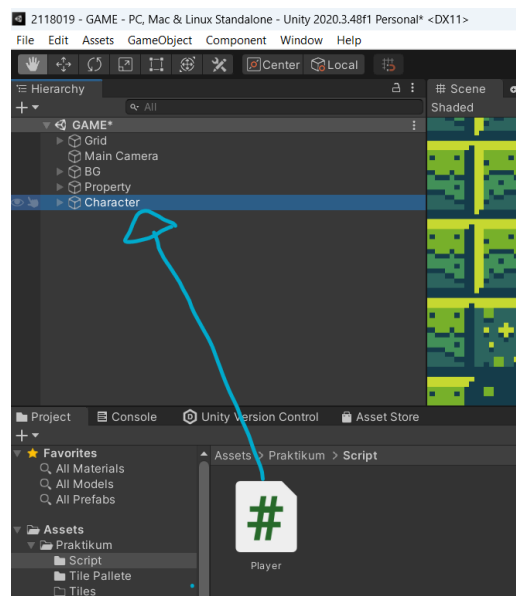


6. Masuk ke dalam folder Script, lalu buat skrip C# dan beri nama Player.



Gambar 8.6 Membuat file skrip C#

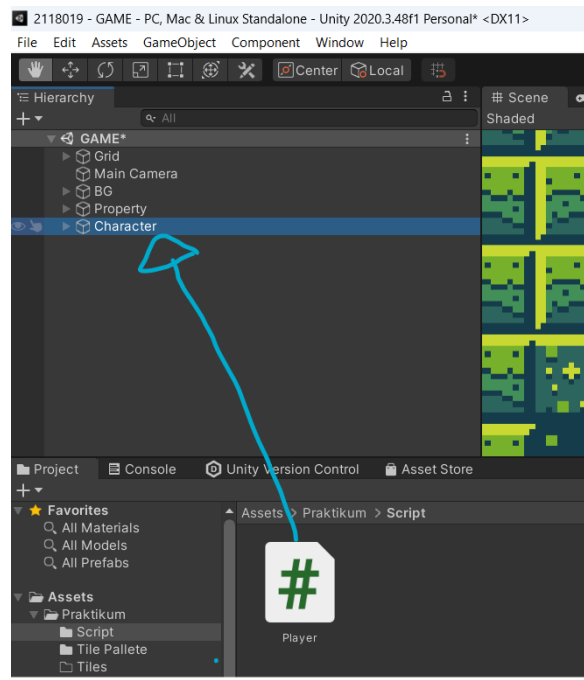
7. Seret dan lepas skrip Player ke dalam Hirarki pada Character, lalu klik dua kali pada skrip Player untuk membukanya di text editor seperti ini.



Gambar 8.7 Memasukkan file skrip ke caharcter



8. Seret dan lepas skrip Player ke dalam Hirarki pada Character, lalu klik dua kali pada skrip Player untuk membukanya di text editor seperti ini.



Gambar 8.7 Memasukkan file skrip ke caharcter

8. Masukkan source code berikut, dan pastikan nama public class sesuai dengan nama file yang dibuat.

```
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;

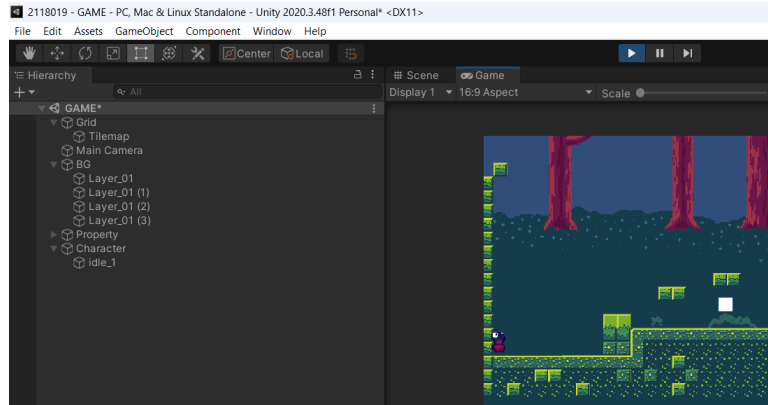
    if (facingRight && dir < 0)
    {
        // ukuran player
        transform.localScale = new Vector3(-3, 3, 3);
        facingRight = false;
    }

    else if (!facingRight && dir > 0)
    {
        // ukuran player
        transform.localScale = new Vector3(3, 3, 3);
        facingRight = true;
    }

    #endregion
}
}
```

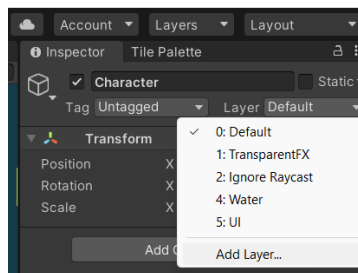


9. Untuk mencoba apakah source code di atas berhasil, tekan "a" atau "left arrow" pada keyboard untuk bergerak ke kiri, dan tekan "d" atau "right arrow" untuk bergerak ke kanan



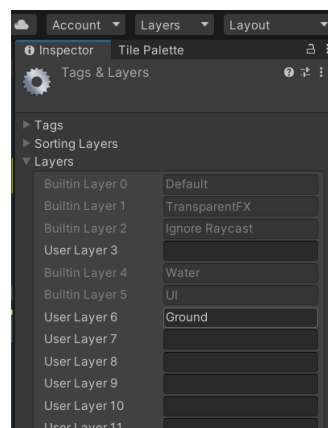
Gambar 8.8 Simpan Folder

10. Untuk membuat pemain dapat melompat menggunakan tombol spasi, kita perlu membuat GroundCheck. Caranya adalah dengan mengklik Grid pada Hirarki, lalu pergi ke inspektor, pilih Layer, dan klik Tambahkan Layer.



Gambar 8.9 Buat New Layer

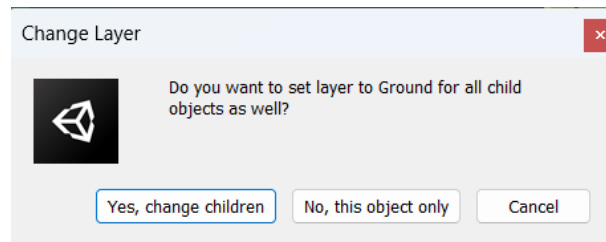
11. Lalu isi "Ground" pada User Layer 6



Gambar 8.10 Mengisi layer 6 menjadi ground

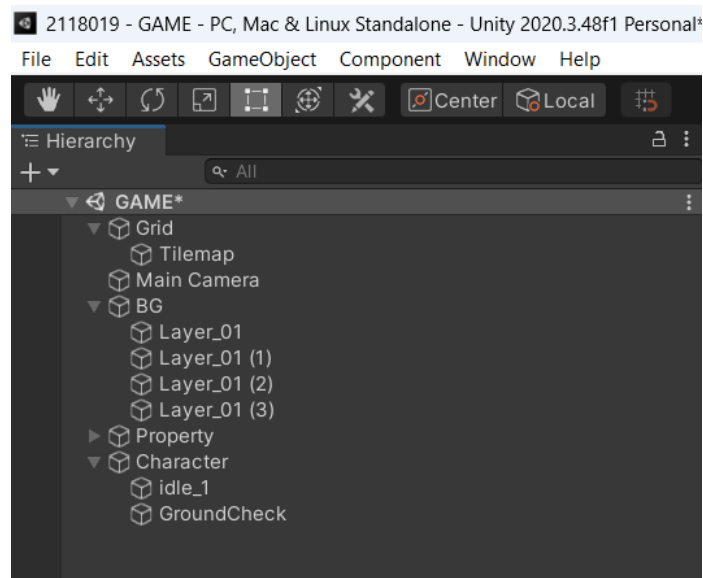


12. Ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja



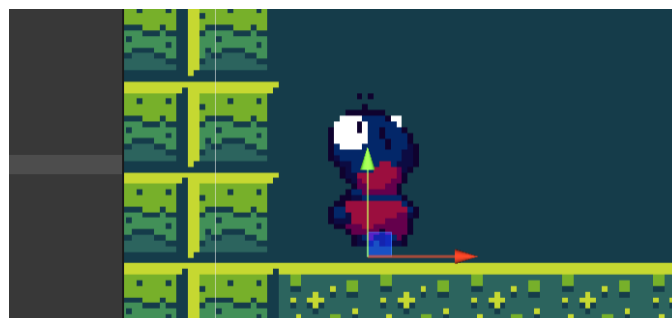
Gambar 8.11 Mengubah Layer Menjadi Ground

13. Klik kanan pada character, lalu Create empty, beri nama GorundCheck.



Gambar 8.12 Membuat GroundCheck pada hierarchy

14. Klik pada GroundCheck di Hirarki, kemudian gunakan "Move Tools" untuk memindahkannya ke bagian bawah Pemain seperti yang ditunjukkan dalam gambar berikut.



Gambar 8.13 Memindahkan GroundCheck Charackter



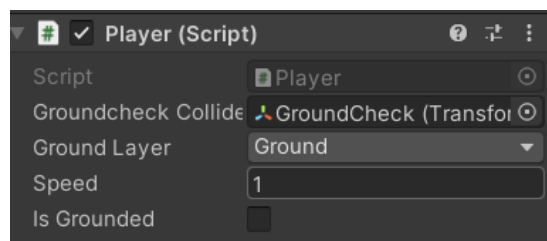
15. Kembali ke script Player tambahkan source code seperti ini

```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;  
  
const float groundCheckRadius = 0.2f; // +  
[SerializeField] float speed = 1;  
float horizontalValue;  
  
[SerializeField] bool isGrounded; // +  
bool facingRight;
```

16. Buat void ground check dibawah void fixedUpdate & tambahkan GorunCheck(); pada void fixedUpdate

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

17. Klik player-idle-1, lalu ke inspector ke effect Player script di bagian “Goruncheck collider” tekan icon lalu pilih yang GorundCheck Transform, dan pada Ground Layer pilih Ground



Gambar 8.14 Mengubah GroundCheck Collider

18. Lalu untuk membuat player melompat tambahkan script berikut

```
[SerializeField] float jumpPower = 100;  
bool jump;
```




19. Tambahkan juga script berikut di bagian void update

```
horizontalValue = Input.GetAxisRaw("Horizontal");  
// Set jump to true only  
if the player is grounded and the jump button is pressed  
if (Input.GetButtonDown("Jump") && isGrounded) { jump =  
true; }
```

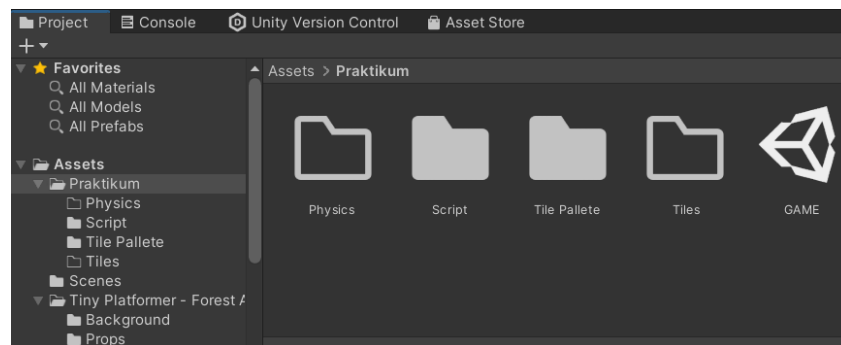
20. Tambahkan juga jump pada parameter Move

```
void FixedUpdate() {  
    GroundCheck(); Move(horizontalValue, jump); jump = false;  
    // Reset jump after applying it in Move  
}
```

21. Tambahkan script berikut pada void Move

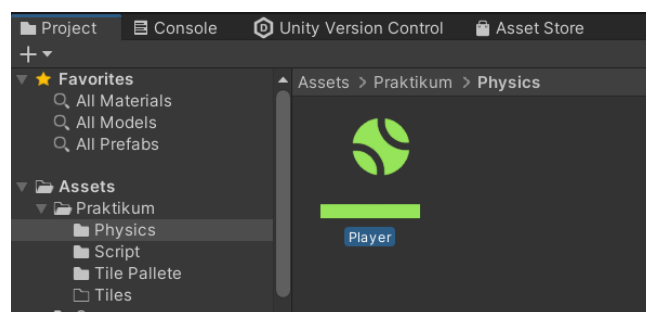
```
bool jumpflag  
  
    if(isGrounded && jumpflag)  
    {  
        isGrounded = false;  
        jumpflag = false;  
        rb.AddForce(new Vector2(0f, jumpPower));  
    }
```

22. Buat folder baru di praktikum bernama “Physics”



Gambar 8.15 Membuat Folder Baru Physic

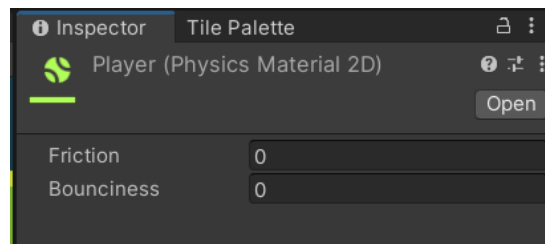
23. Didalam folder Pyshics create > 2d > physical material 2d , bernama “Player”



Gambar 8.16 Membuat Material Fisik 2D



24. Klik Player (Physics Material 2D), di bagian inspektor, atur gesekan (friction) dan pantulan (bounces) menjadi 0.



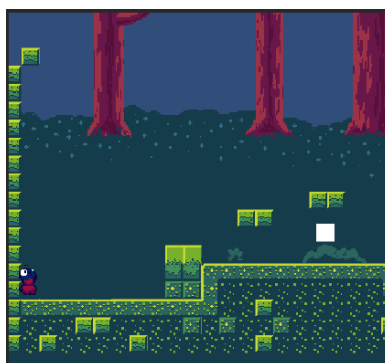
Gambar 8.17 Mengubah friction dan bounces

25. Klik Hierarchy, pilih layer player idle 1, di Inspector cari Rigidbody 2D, kemudian klik ikon untuk membuka pilihan material fisika 2D, dan pilih asset Player yang sudah dibuat sebelumnya.



Gambar 8.18 Membuka Material Fisika 2D

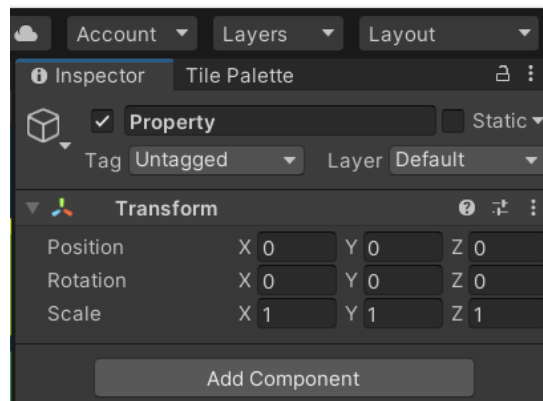
26. Tekan play, maka player bisa melompat dengan menekan spasi



Gambar 8.19 Tampilan Player Melompat

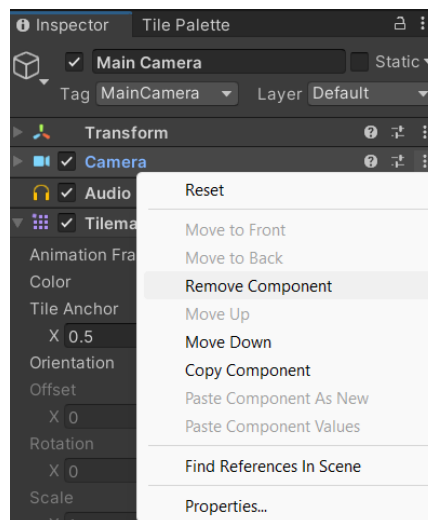


27. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untaged



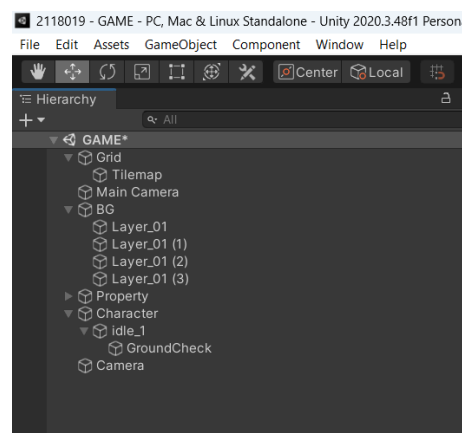
Gambar 8.20 Mengubah Tag Untagged

28. Pada Effect Camera pilih Remove Component



Gambar 8.21 Remove Component Camera

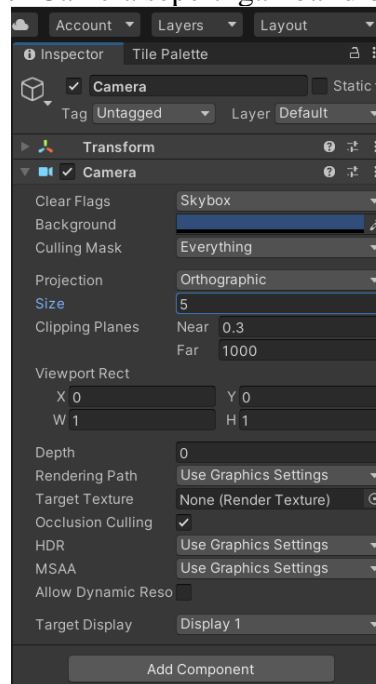
29. Create Empty pada Hirarki, dan Rename Menjadi Camera



Gambar 8.22 Mengubah object hirarki menjadi Camera

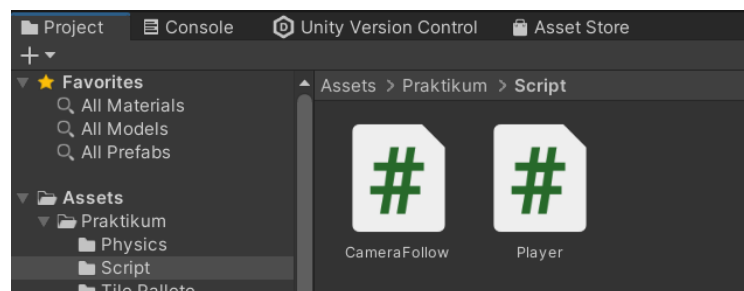


30. Sesuaikan Setting Layer Camera seperti gambar dibawah ini



Gambar 8.23 Setting Layer Camera

31. Buat file script baru di folder Script dengan nama "CameraFollow"



Gambar 8.24 Buat File Script "Camera Follow"

32. Lalu tuliskan script berikut ini

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

public class CameraFollow : MonoBehaviour
{
    public float xMargin = 0.5f;
    public float yMargin = 0.5f;
    public float xSmooth = 4f;
    public float ySmooth = 4f;
    public Vector2 maxXAndY;
    public Vector2 minXAndY;
    private Transform player;

    void Awake()
    {

```



```
player =
    GameObject.FindGameObjectWithTag("Player").transform
;

bool CheckXMargin()
{
    return Mathf.Abs(transform.position.x -
        player.position.x) > xMargin;
}

bool CheckYMargin()
{
    return Mathf.Abs(transform.position.y -
        player.position.y) > yMargin;
}

void FixedUpdate()
{
    TrackPlayer();
}

void TrackPlayer()
{
    float targetX = transform.position.x;
    float targetY = transform.position.y;
    if (CheckXMargin())
        targetX = Mathf.Lerp(transform.position.x,
            player.position.x,
            xSmooth * Time.deltaTime);
    if (CheckYMargin())
        targetY = Mathf.Lerp(transform.position.y,
            player.position.y,
            ySmooth * Time.deltaTime);
    targetX = Mathf.Clamp(targetX, minXAndY.x,
        maxXAndY.x); targetY =
        Mathf.Clamp(targetY, minXAndY.y, maxXAndY.y);
    transform.position = new
        Vector3(targetX, targetY,
            transform.position.z);
}
```

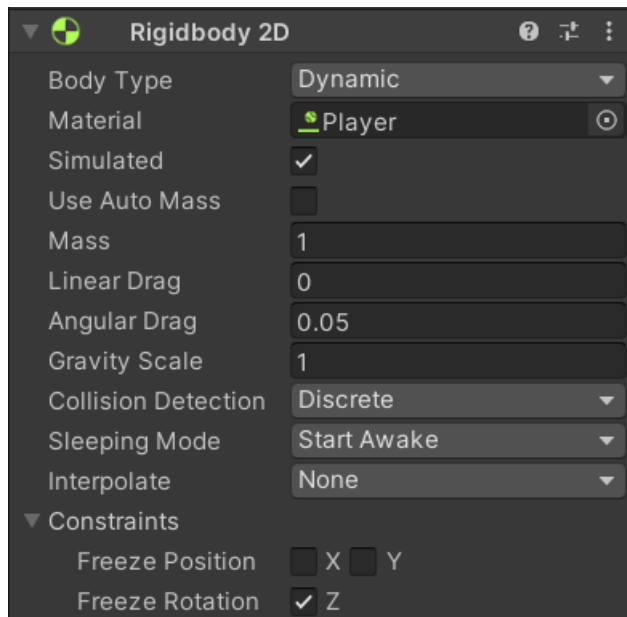
33. Setelah itu, klik pada kamera, buka inspector, dan di bagian "Camera Follow (Script)", ubah nilai Max X dan Max Y-nya.



Gambar 8.25 Mengubah nilai max x dan y

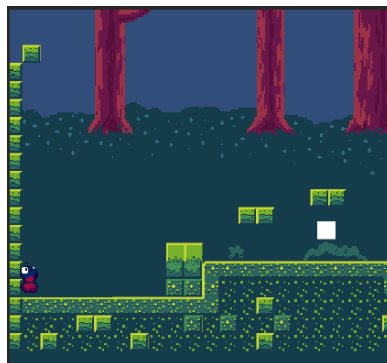


34. Ubah tag di character Untagged menjadi "Player"



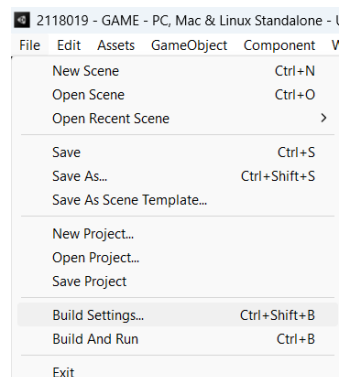
Gambar 8.26 Ganti Tag Jadi Player

35. Tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter.



Gambar 8.27 Tampilan kamera akan mengikuti pergerakan karakter

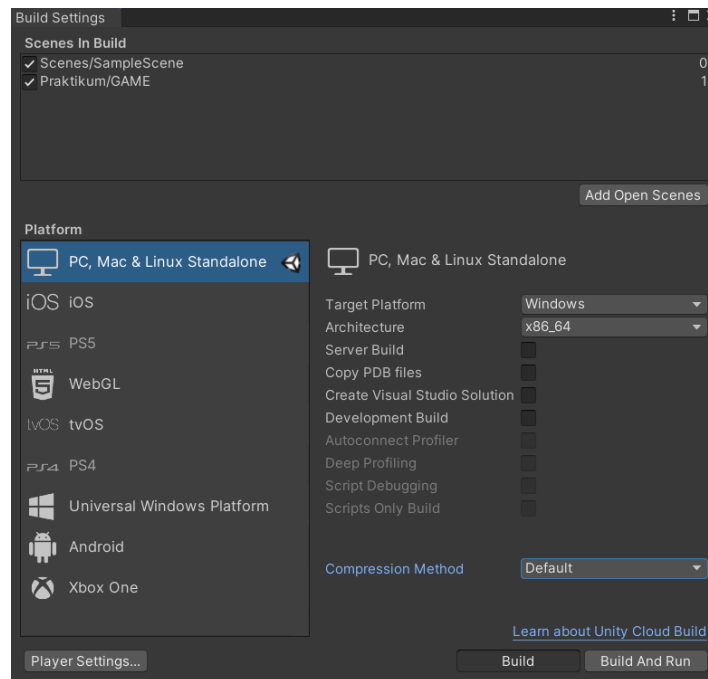
36. Pergi ke menu File kemudian pilih Build Setting (Ctrl + Shift + B)



Gambar 8.28 Build & Run

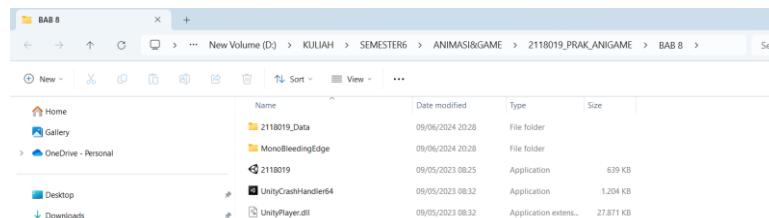


37. Pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian



Gambar 8.29 Mensetting Build

38. Pilih dimana Project disimpan, dan tunggu hasilnya



Gambar 8.30 Menyimpan Project

B. Kuis : Menjelaskan Source Code CameraFollow

```
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 :

Program di atas adalah skrip untuk membuat kamera mengikuti pergerakan pemain dalam permainan. Dengan menggunakan Unity dan bahasa pemrograman C#, skrip ini mengatur agar kamera bergerak horizontal mengikuti posisi pemain. Setiap frame yang diperbarui, posisi kamera diatur untuk memiliki nilai X yang sama dengan posisi X pemain, tetapi nilai Y dan Z kamera tetap tidak berubah. Ini membuat kamera hanya mengikuti gerakan pemain secara horizontal tanpa bergerak vertikal atau kedalam. Dengan ini, kamera akan terus mengikuti pemain saat bergerak ke kanan atau kiri dalam permainan, memberikan tampilan yang konsisten dan menangkap seluruh aksi horizontal dalam permainan.