



TUGAS PERTEMUAN: 8

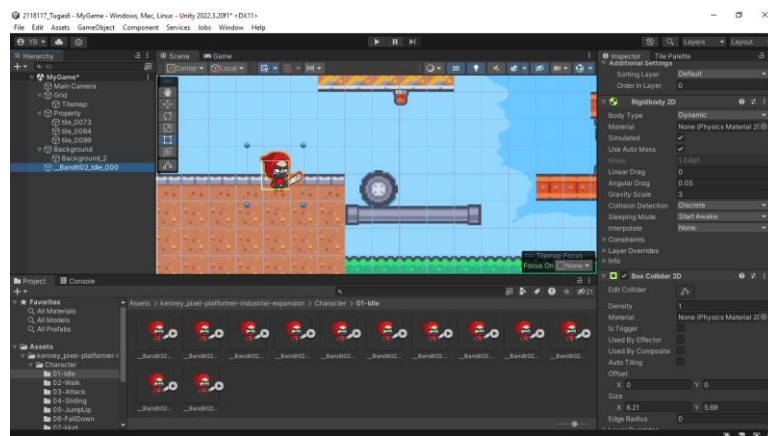
CAMERA & CHARACTER MOVEMENT

NIM	:	2118117
Nama	:	Yohanes Yudha Saputra Bangko
Kelas	:	C
Asisten Lab	:	Nayaka Apta N (2218102)

8.1 Tugas 8 : Camera & Character Movement

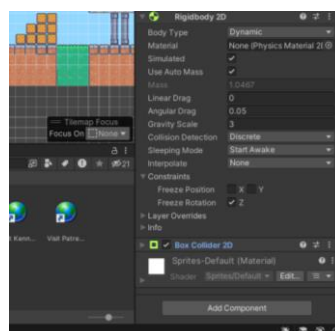
A. Membuat Camera dan Character Movement

1. Buka project sebelumnya untuk melanjutkan ke tahap character.



Gambar 8.1 Membuka Project sebelumnya

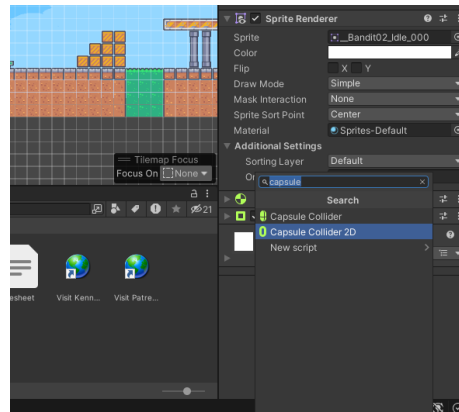
2. Setelah itu pada klik player bandit idle 1 di hierarki lalu menuju inspector dan tambahkan component rigidbody 2D dan centang freeze position z.



Gambar 8.2 Memilih freeze position rigidbody 2d

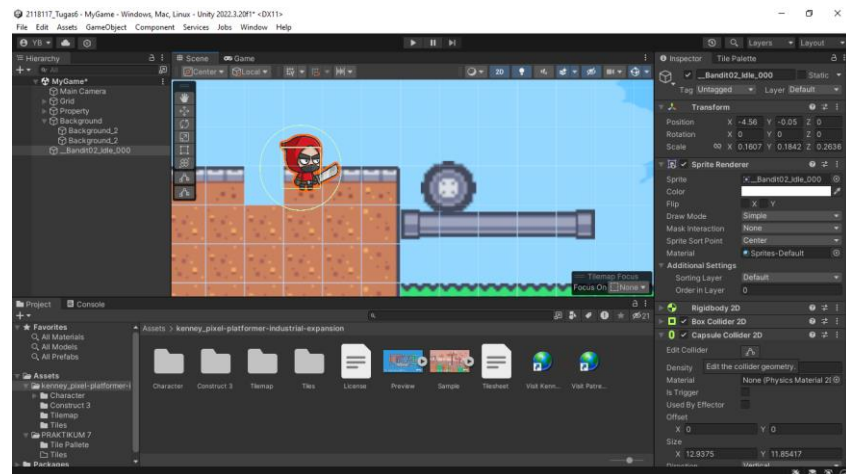


3. Lalu tambahkan komponen Capsule Collider di player bandit idle 1.



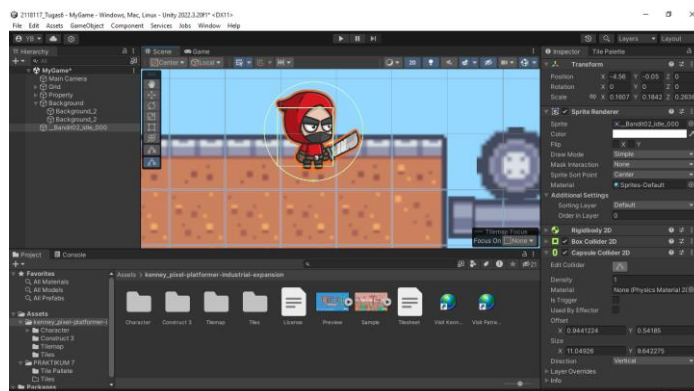
Gambar 8.3 Capsule collider 2D

4. Pada capsule collider terdapat menu edit collider, pilih icon edit dengan nama Edit the collider geometry.



Gambar 8.4 Edit the collider geometry

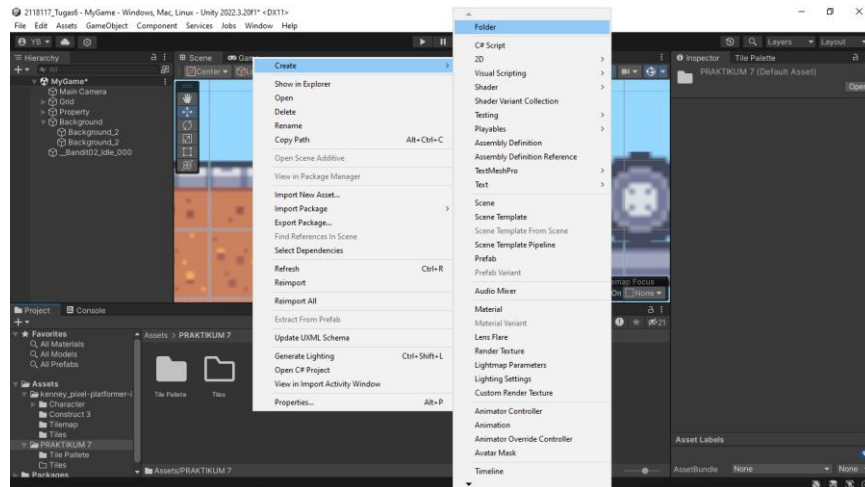
5. Lalu cocokkan garis oval collider dengan karakternya.



Gambar 8.5 Mencocokkan garis collider

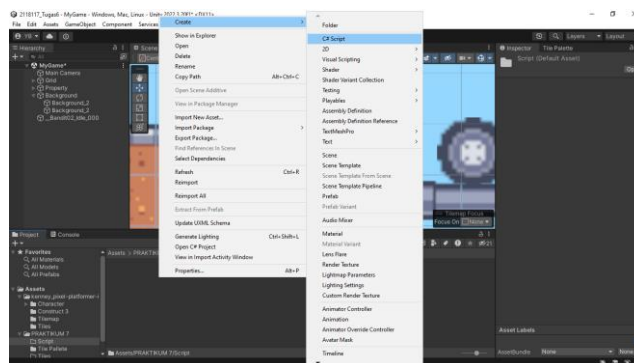


6. Masuk pada folder praktikum lalu buat folder baru dengan nama script.



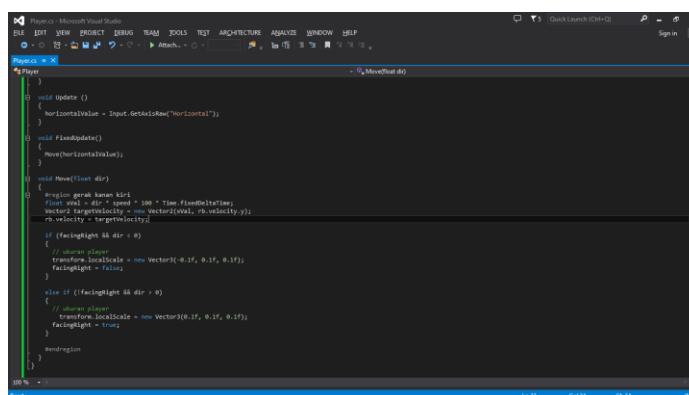
Gambar 8. 6 Create Folder

7. Pada folder script buat C# script dengan cara klik kanan dan pilih create C# script lalu beri nama Player.



Gambar 8. 7 Membuat C# script

8. Drag and drop file script tersebut ke player bandit idle 1 lalu klik dua kali maka akan masuk ke dalam visual studio.



Gambar 8. 8 Masuk ke Visual Studio



9. Masukkan script 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;

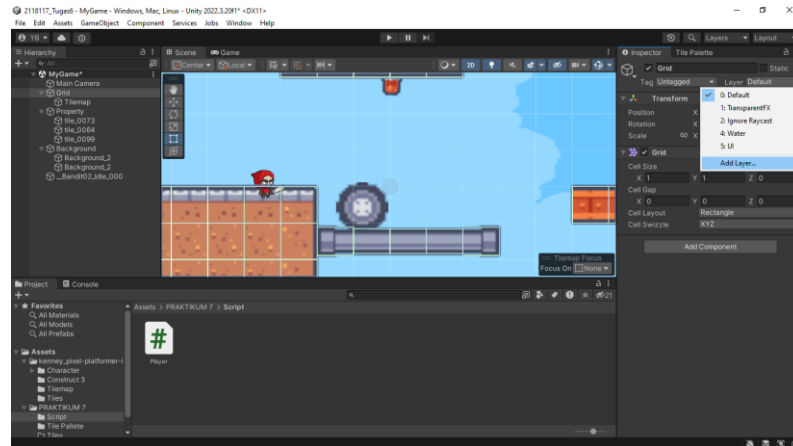
        if (facingRight && dir < 0)
        {
            // ukuran player
            transform.localScale = new Vector3(-1, 1,
        1);
            facingRight = false;
        }

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

        #endregion
    }
}
```

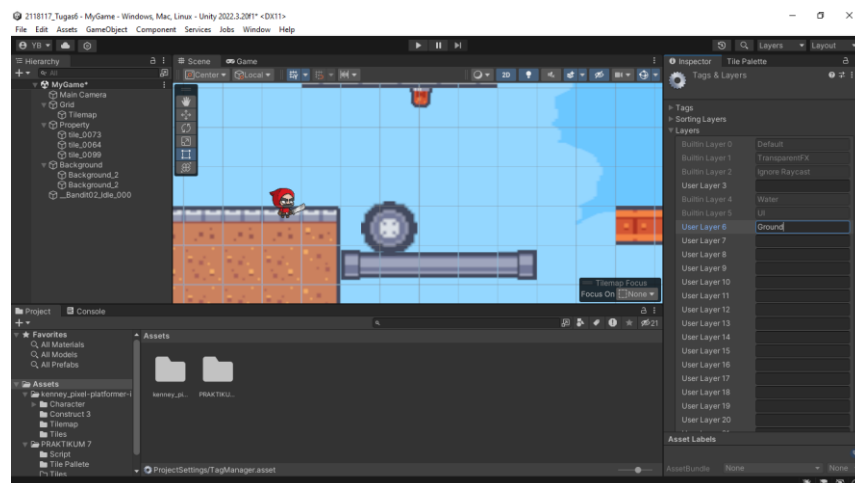


10. Untuk membuat player dapat lompat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, lalu klik Add Layer.



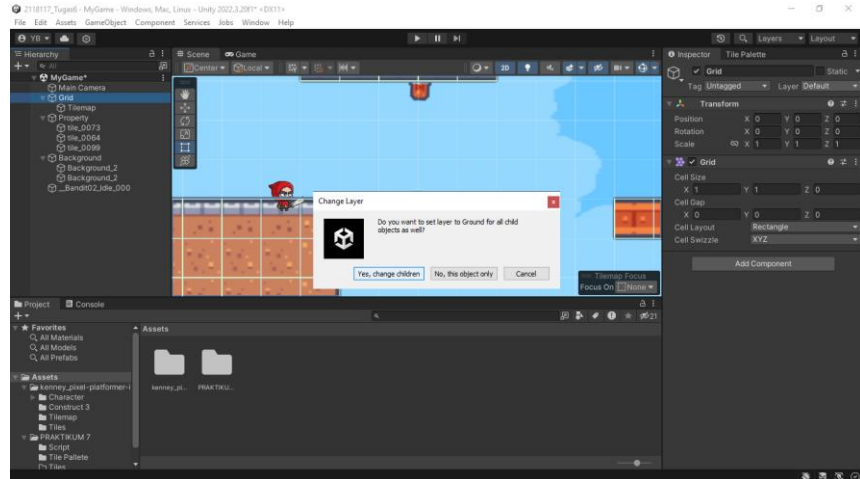
Gambar 8. 9 Add Layer Grid

11. Isi User Layer 6 dengan nama Ground.



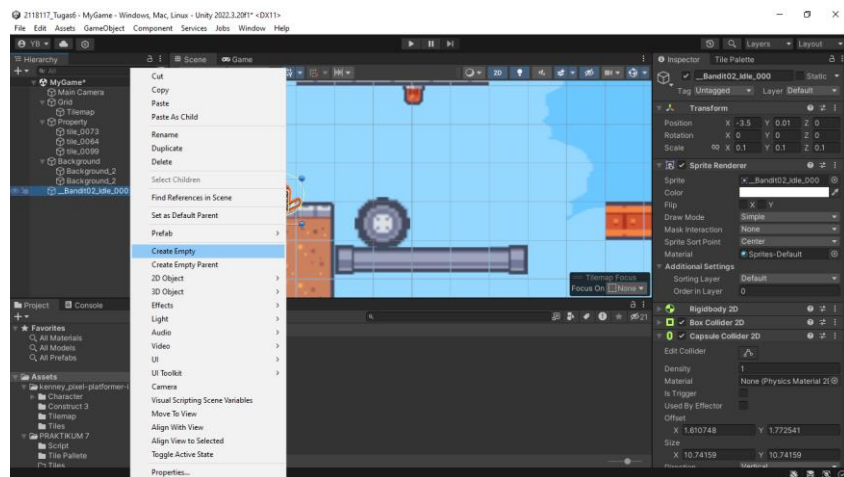
Gambar 8. 10 User Layer 6 Ground

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



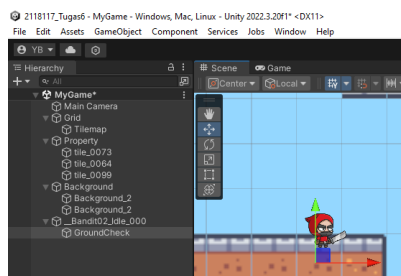
Gambar 8. 11 Ubah layer ground

13. Klik kanan pada player bandit idle 1, lalu pilih Create empty dan beri nama GroundCheck.



Gambar 8. 12 Create Empty

14. Klik pada Hirarki GroundCheck, lalu gunakan “Move Tools” untuk memindahkan ke bagian bawah Player seperti gambar berikut.



Gambar 8. 13 Memindahkan bagian bawah player



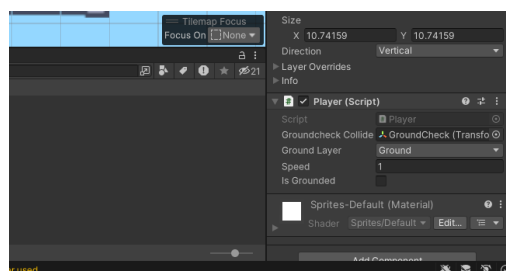
15. Kembali ke script Player.cs dan tambahkan script berikut lalu sesuaikan seperti di bawah ini.

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

16. Buat function void groundcheck dibawah void fixedUpdate & tambahkan GroundCheck() pada void fixedUpdate seperti gambar di bawah ini.

```
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 bandit idle 1, lalu pada bagian inspector pilih Player (script) dan di bagian “GroundCheck Collider” pilih yang GroundCheck (Transform), kemudian pada Ground Layer pilih Ground.



Gambar 8. 14 GroundCheck Collider

18. Tambahkan script [SerializeField] float jumpPower = 100 dan bool jump untuk membuat player melompat di bagian Public Class Player.

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



19. Tambahkan script seperti berikut pada bagian function void update.

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

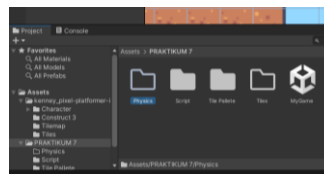
20. Tambahkan jump pada parameter Move.

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

21. Tambahkan script berikut pada void Move

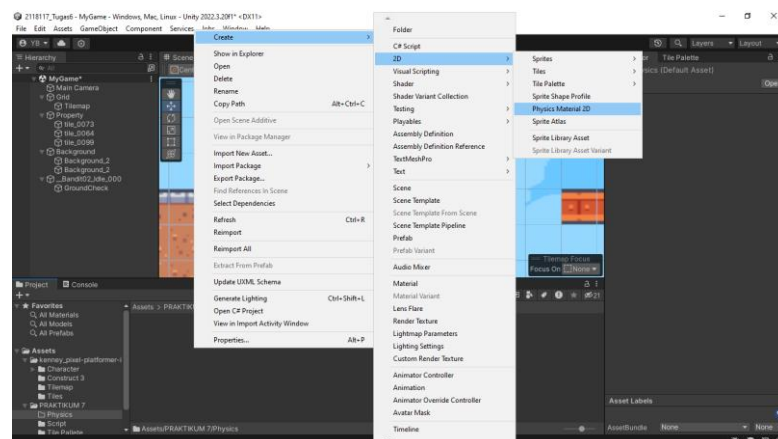
```
void Move(float dir, bool jumpflag)
{
    if(isGrounded && jumpflag)
    {
        isGrounded = false;
        jumpflag = false;
        rb.AddForce(new Vector2(0f, jumpPower));
    }
}
```

22. Buatlah folder baru di Praktikum dengan nama Physics



Gambar 8. 15 Folder Physics

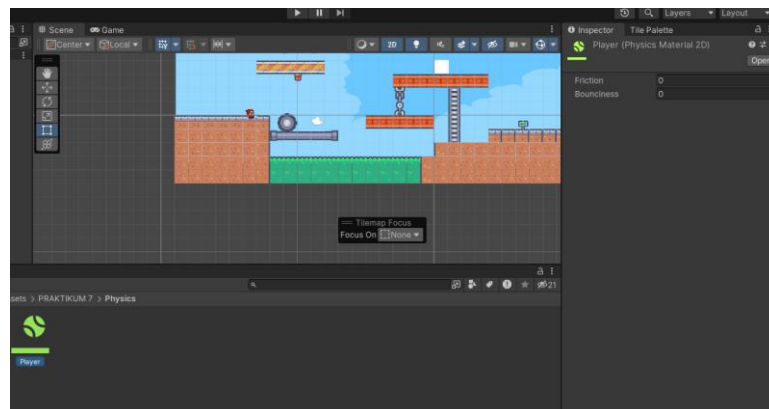
23. Didalam folder Physics klik kanan dan pilih create lalu 2d kemudian physical material 2d , berinama sebagai Player.



Gambar 8. 16 Physics Material 2D

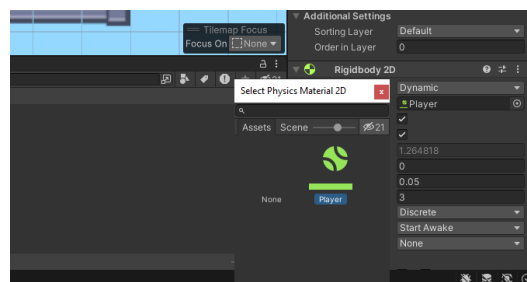


24. Klik Player pada Physics Material 2D yang telah dibuat, dibagian menu inspector, friction & bounces ubah menjadi 0



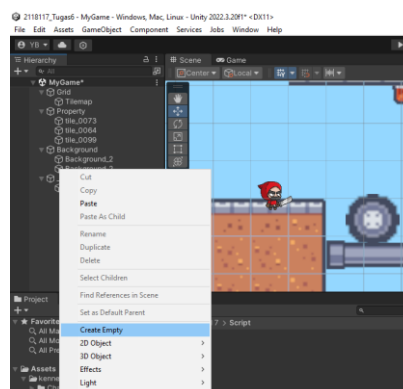
Gambar 8. 17 Bouncess and Friction

25. Pilih layer player bandit idle 1, pada Inspector Cari Rigidbody 2D lalu klik icon untuk membuka box select physics material 2d , lalu pilih asset Player yang sudah kita buat tadi.



Gambar 8. 18 Memiilih physics material 2D

26. Untuk camera movement bisa dengan menghapus main camera yang telah ada lalu klik kanan pada hierarki dan pilih create empty.



Gambar 8. 19 Create empty

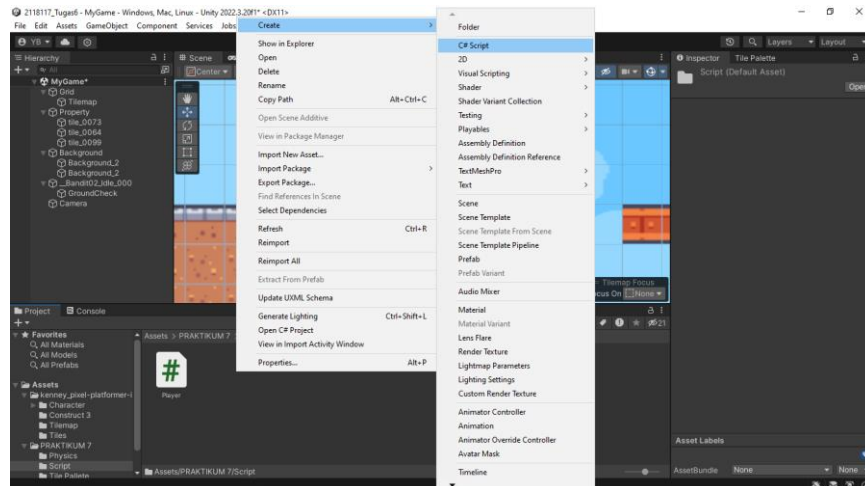


27. Tambahkan component camera di dalam inspector layer Camera tersebut dan atur sesuai dengan gambar di bawah ini.



Gambar 8. 20 Menyesuaikan camera

28. Masuk ke dalam Folder Scripts tadi yang berada di dalam folder Praktikum. Klik kanan pilih create lalu pilih C# script dan beri nama CameraFollow.



Gambar 8. 21 Membuat script

29. 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();
    }
}
```

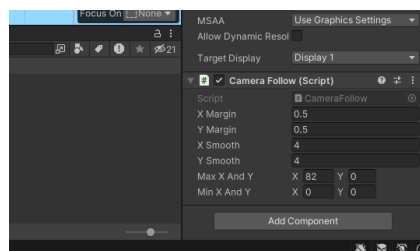


```

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);
}
}

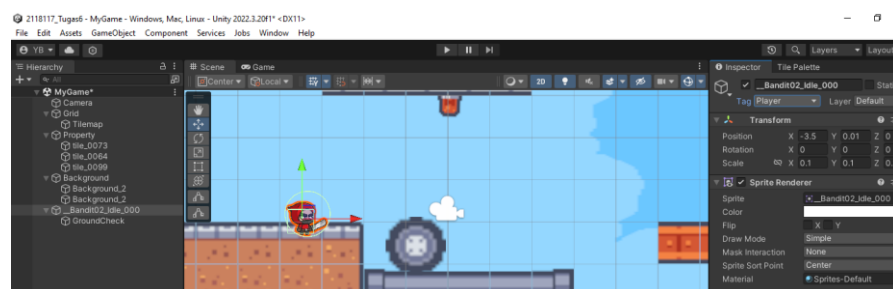
```

30. Drag & drop script CameraFollow ke dalam Layer Camera lalu atur Max X dan Max Y pada Camera follow (script).



Gambar 8. 22 Mengatur Max X dan Y

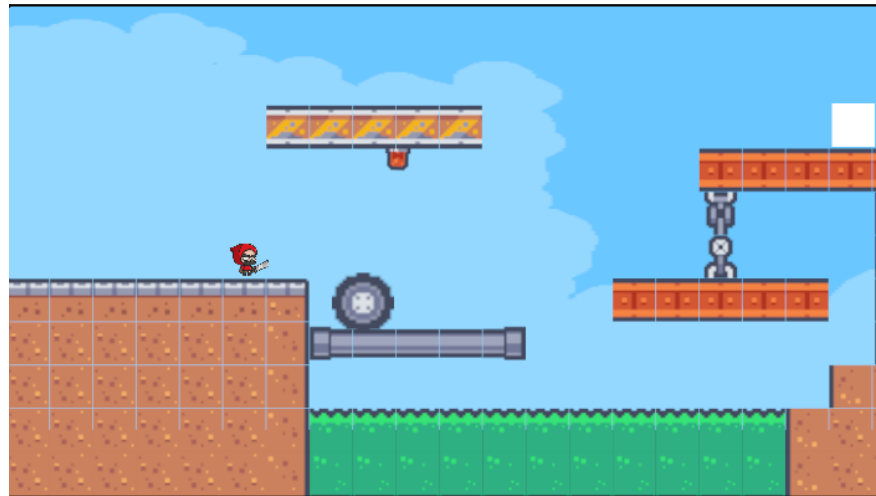
31. Klik layer Bandit idle 1 dan ubah tag menjadi Player.



Gambar 8. 23 Mengubah tag menjadi player

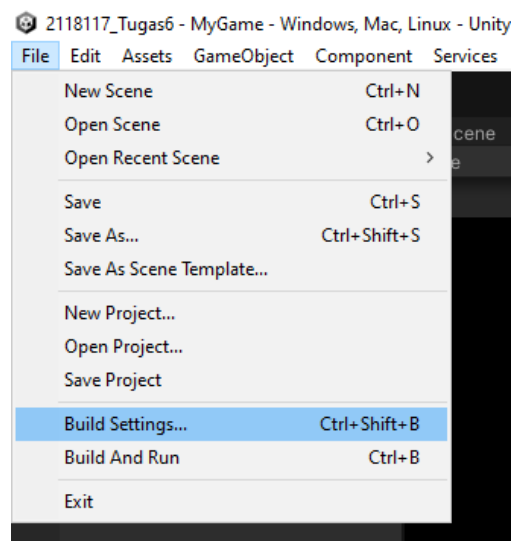


32. Running game dan karakter sekarang bisa bergerak, lompat dan kamera juga akan mengikuti pergerakan karakter.



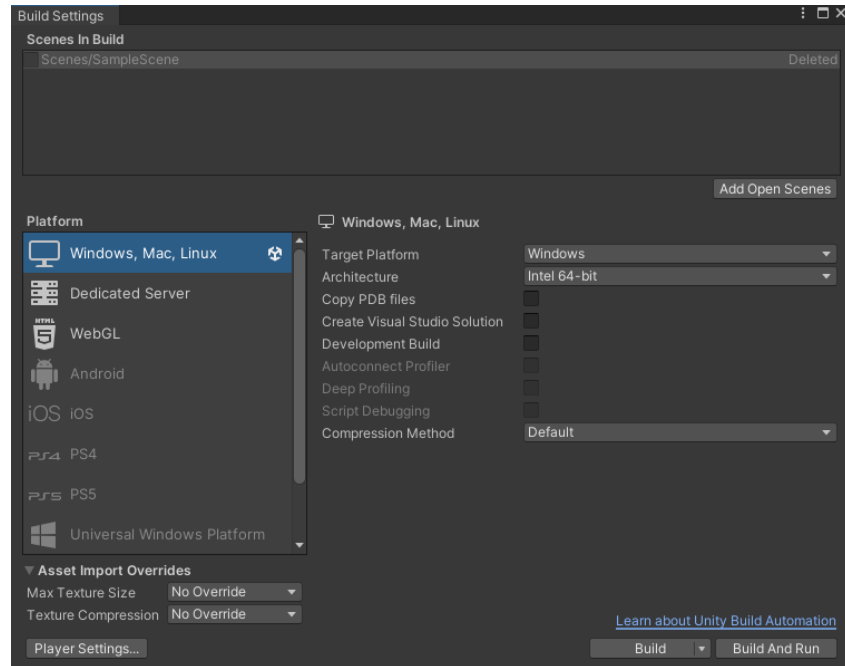
Gambar 8. 24 Hasil Running

33. Setelah itu lakukan render dengan cara pergi ke menu File dan pilih Build Setting.



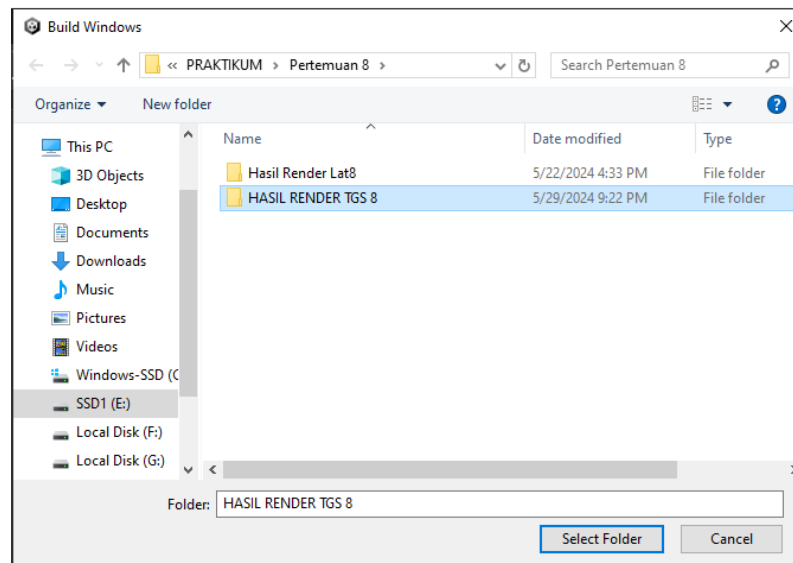
Gambar 8. 25 Build Settings

34. Pada platform Build ini pilih PC, Mac & Linux, Tekan Build.



Gambar 8. 26 Build

35. Pilih direktori penyimpanan dan tunggu hingga selesai.



Gambar 8. 27 Memilih folder penyimpanan

B. Link Github Pengumpulan

<https://github.com/Yohanes-Yudha/2118117-PRAK-ANIGAME>