



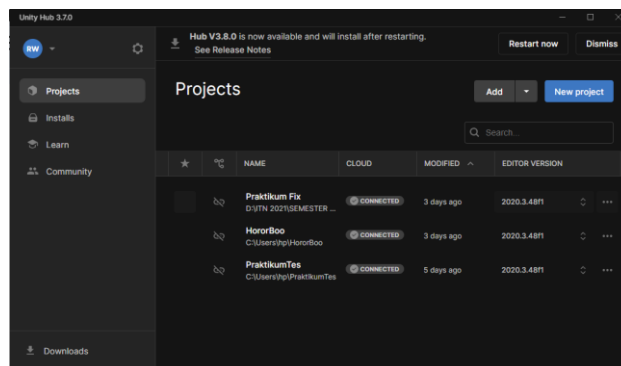
TUGAS PERTEMUAN: 8

Camera & Character Movement

NIM	:	2118055
Nama	:	Ridho Arif Wicaksono
Kelas	:	B
Asisten Lab	:	Difa Fisabililah (2118052)

1.1 Tugas 8: Camera & Character Movement

1. Buka unity hub untuk memulai pembuatan game, dan buka project sebelumnya yaitu “Horor Boo”



Gambar 8. 1 Unity Hub

2. Berikut merupakan project yang sebelumnya dibuat, setelah itu hapus bagian idle(1)



Gambar 8. 2 Project Sebelumnya

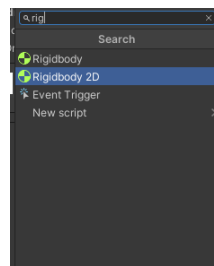


3. Tambahkan idle(1) kedalam bagian hierarki object, jaan lupa untuk melakukan resize



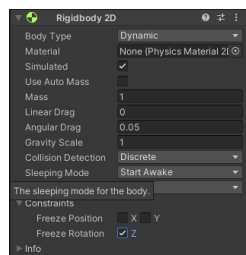
Gambar 8. 3 Menambahkan idle(1)

4. Setelah itu klik pada bagian karakter dan tambahnkan Rigidbody 2D pada karakter



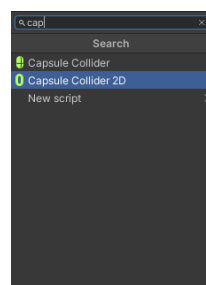
Gambar 8. 4 Riggid Body

5. Setelah itu pada bagian Riggid Body 2D centang pada bagian freeze rotation



Gambar 8. 5 Freeze Rotation

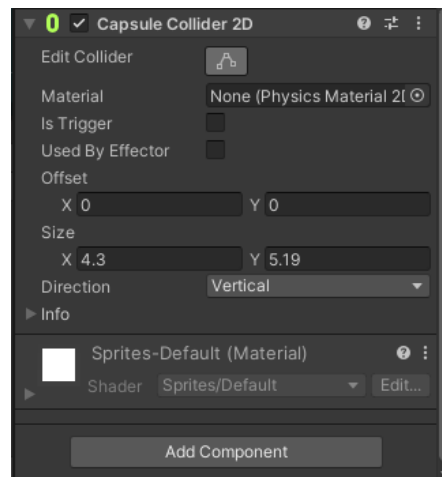
6. Di dalam karakter yang sama tambahkan capsule collider setelah riggid body 2D



Gambar 8. 6 Menambahkan Capsule Colider



- Setelah itu dibagian capsule collider 2D klik pada bagian Edit Colider



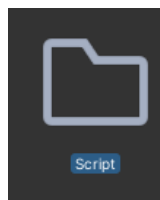
Gambar 8. 7 Edit Colider

- Pada bagian edit collider, paskan bagian collider (garis lonjong hijau) kedalam karakter pastikan bagian karakter tertutup oleh collider



Gambar 8. 8 Edit Colider

- Buat folder baru bernama “Script” di dalam folder “TugasPraktikum”



Gambar 8. 9 Folder Baru Bernama Script

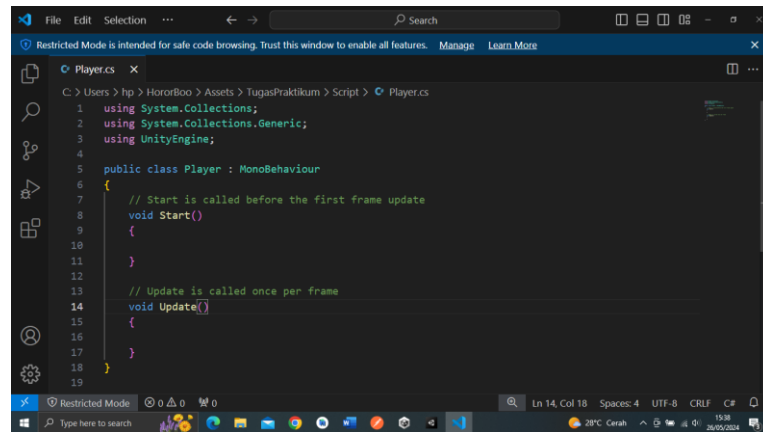
- Dalam folder script buat file script baru dengan nama “Player”



Gambar 8. 10 Membuat File Script



11. Klik 2 kali pada file “Player.cs” maka akan diarahkan kedalam aplikasi vscode, namun perlu di perhatikan pada saat instalasi untuk text editor jika mencentang visual studio maka nantinay untuk editing script menggunakan visual studio begitupun sebaliknya maka menggunakan vscode



Gambar 8. 11 Visual Studio Editor

12. Setelah itu tambahkan kode program berikut kedalam file “Player.cs”

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

public class Player : MonoBehaviour{

    Rigidbody2D rb;
    [SerializeField] float speed = 7;
    float horizontalValue;
    bool facingRight;
    bool jump;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }
    void Update ()
    {
        horizontalValue = Input.GetAxisRaw("Horizontal");
    }
    void FixedUpdate()
    {
        GroundCheck();
        Move(horizontalValue, jump);
    }
    void Move(float dir, bool jumpflag)
    {
        if(isGrounded && jumpflag)
        {
            isGrounded = false;
            jumpflag = false;
            rb.AddForce(new Vector2(0f, jumpPower));
        }
    }
}
```



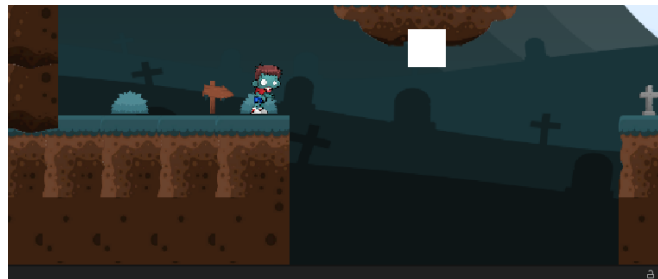
```
#region gerak kanan kiri
float xVal = dir * speed * 100 * Time.fixedDeltaTime;
Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
rb.velocity = targetVelocity;

if (facingRight && dir < 0)
{
    // ukuran player
    transform.localScale = new Vector3(-0.3f, 0.3f,
0.3f);
    facingRight = false;
}

else if (!facingRight && dir > 0)
{
    // ukuran player
    transform.localScale = new Vector3(0.3f, 0.3f,
0.3f);
    facingRight = true;
}

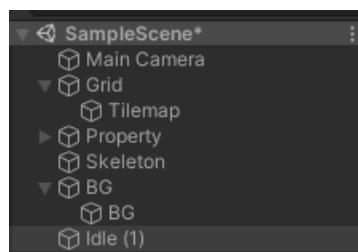
#endregion
}
```

13. Setelah save dan uji kode program tersebut dengan klik tombol “Play” maka karakter akan bergereak



Gambar 8. 12 Mode Play

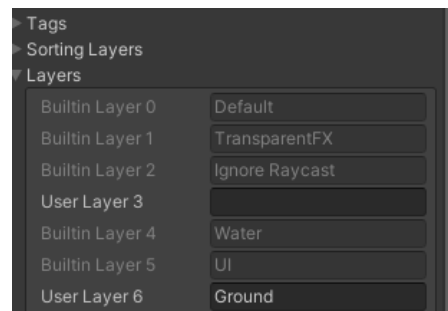
14. Pada hierarki klik pada bagian grid



Gambar 8. 13 Masuk Kedalam Grid

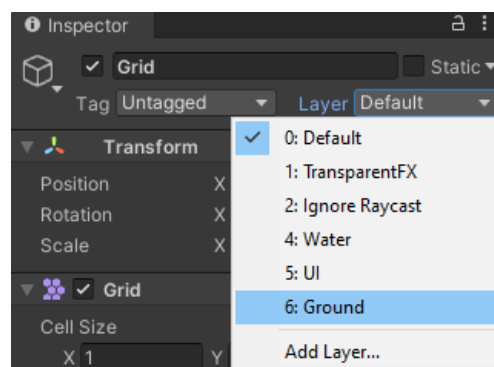


15. Pada grid pada bagian layer klik “Add Layer” pada layer 6 beri nama Ground



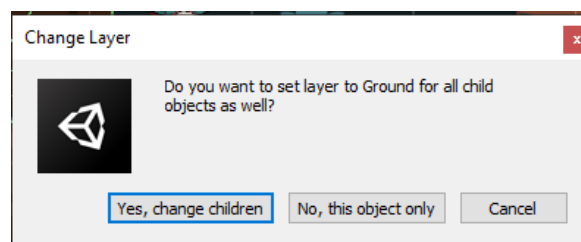
Gambar 8. 14 Ground Layer

16. Setelah itu masuk kedalam bagian layer pilih ground



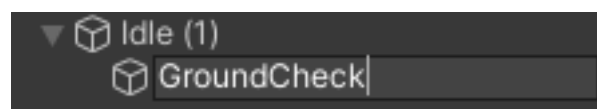
Gambar 8. 15 Layer Ground

17. Pada bagian change layer pilih “Yes, change children”



Gambar 8. 16 Change Layer

18. Pada hierarki, pada hierarki Idle(1) buatlah hierarki baru bernama GroundCheck



Gambar 8. 17 Hierarki GroundCheck



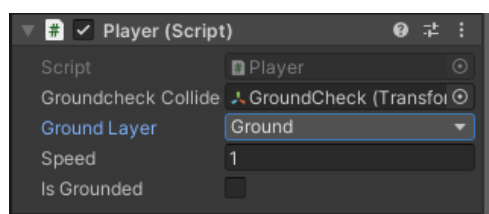
19. Setelah itu masuk kedalam source code “Player.cs” tambahkan kode berikut untuk pelengkap agar player bisa melompat

```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;  
[SerializeField] float jumpPower = 500;  
const float groundCheckRadius = 0.2f; // +  
[SerializeField] float speed = 7;  
float horizontalValue;  
[SerializeField] bool isGrounded; // +  
bool jump;
```

20. Tambahkan kode berikut pada bagian FixedUpdate() dan bagian GroundCheck()

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

21. Masuk kedalam hierarki Idle(1) pada bagian Component Script isi pada bagian GroundCheck Colider dengan GroundCheck(Transform) dan bagian Ground Layer pilih Ground



Gambar 8. 18 Ground Check

22. Tambahkan kode berikut dibagian atas di bawah nama class

```
[SerializeField] float jumpPower = 500;  
bool jump;
```

23. Tambahkan kode berikut pada bagian method update()

```
void Update ()  
{  
    horizontalValue = Input.GetAxisRaw("Horizontal");  
    if (Input.GetButtonDown("Jump"))
```



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

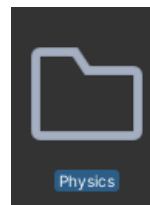
24. Setelah itu bagian method fixedUpdate() tambahkan script berikut

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

25. Setelah bagian method move(parameter1, dan paramter2) tambahkan kode program berikut

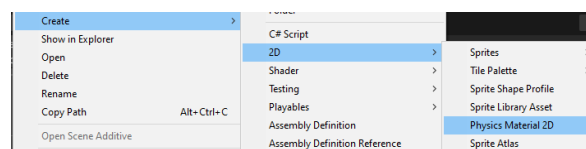
```
if (isGrounded && jumpflag)
{
    isGrounded = false;
    jumpflag = false;
    rb.AddForce(new Vector2(0f, jumpPower));
}
```

31. Setelah itu buat folder baru di dalam folder “TugasPraktikum” dengan nama “Physic”



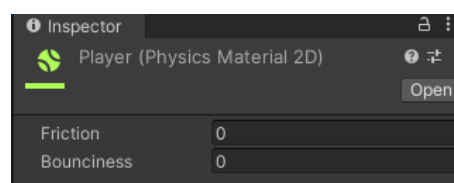
Gambar 8. 19 Membuat folder physic

32. Dalam folder physic tambahkan Physic Material 2D beri nama dengan nama Player



Gambar 8. 20 Folder Physic

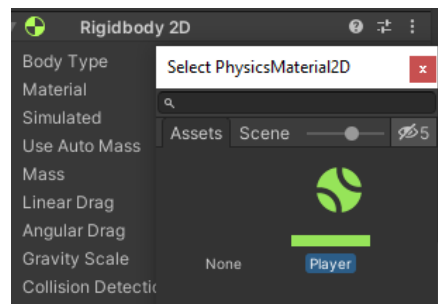
33. Pada physic bagian Friction dan Bounciness ubah menjadi 0



Gambar 8. 21 Inspetor Physic

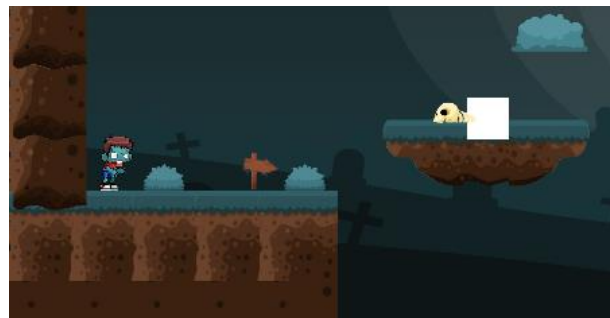


34. Pada hierarki Idle(1) pada bagian Rigid Body 2D pada bagian Physic Material 2D pilih Physic



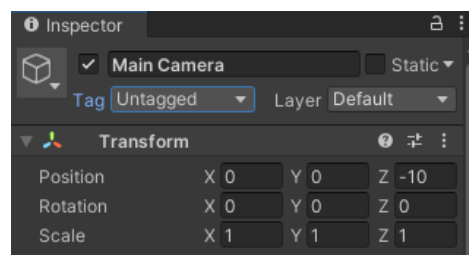
Gambar 8. 22 Material Physic

35. Mainkan game untuk menguji apakah karakter sudah dapat loncat



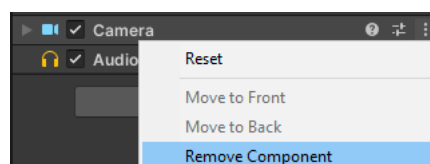
Gambar 8. 23 Mainkan Game

36. Klik pada bagian hierarki kamera lalu masuk kedalam bagian inspector pada bagian tag pilih “Untagged”



Gambar 8. 24 Tag Untagged Camera

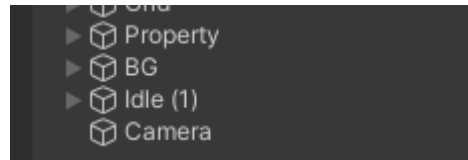
37. Pada bagian komponen “Camera” klik “Remove Component”



Gambar 8. 25 Remove Component Camera

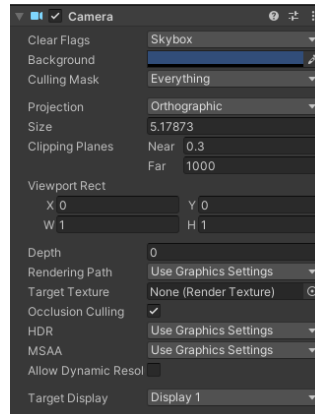


38. Buat hierarki baru dengan nama kamera



Gambar 8. 26 Hierarki Baru Kamera

39. Pada bagian projection inspector Camera pilih orthobliographic



Gambar 8. 27 Setting Orthobliographic

40. Pada folder TugasPraktikum buat file script baru bernama CameraFollow



Gambar 8. 28 File Camera Follow

41. Masukkan kode program berikut kedalam script CameraFollow

```
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);
}
```

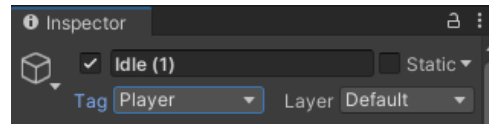
42. Setelah itu ubah settingan pada bagian Camera Follow



Gambar 8. 29 Camera Follow

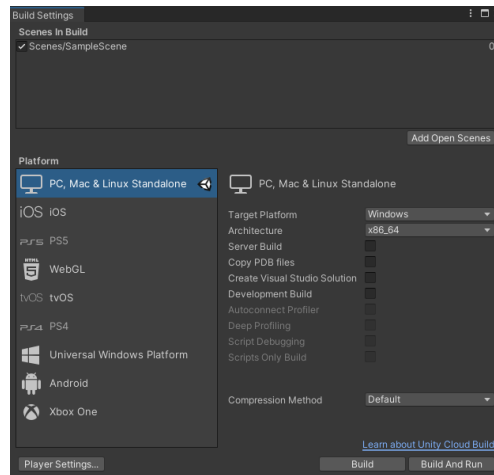


43. Pada idle(1) ubah pada bagian tag menjadi Player



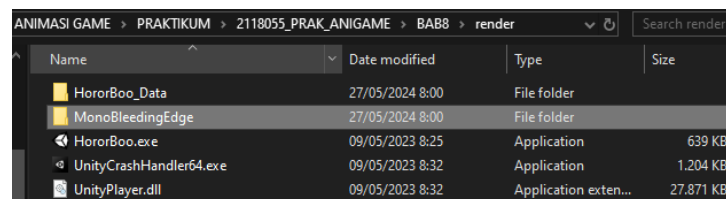
Gambar 8. 30 Player Idle

44. Untuk melakukan render klik File->Build Setting lalu klik Build And Run



Gambar 8. 31 Build Setting

45. Hasil akhir render



Gambar 8. 32 Hasil Akhir Render

Link Github Pengumpulan

Link : https://github.com/arifmalabar/2118055_PRAK_ANIGAME.git