



TUGAS PERTEMUAN: 8

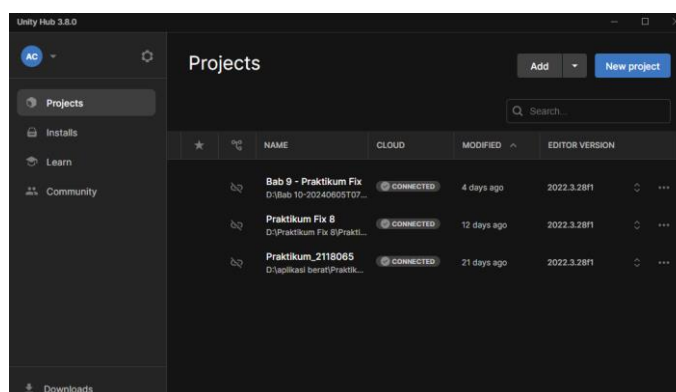
MOVEMENT

NIM	:	2118122
Nama	:	Egia Suranta Perangin-Angin
Kelas	:	D
Asisten Lab	:	Berchmans Bayu Bin Jaya (2218034)
Referensi	:	https://assetstore.unity.com/packages/2d/characters/pixel-adventure-1-155360

8.1 Tugas 1 : MOVEMENT

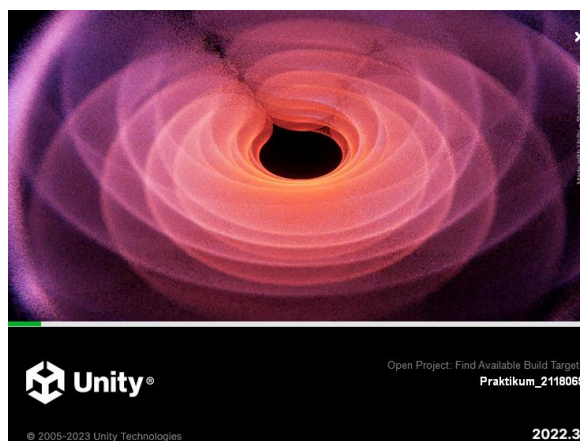
A. Movement

1. Masuk ke dalam aplikasi unity



Gambar 8.1 Membuka Unity

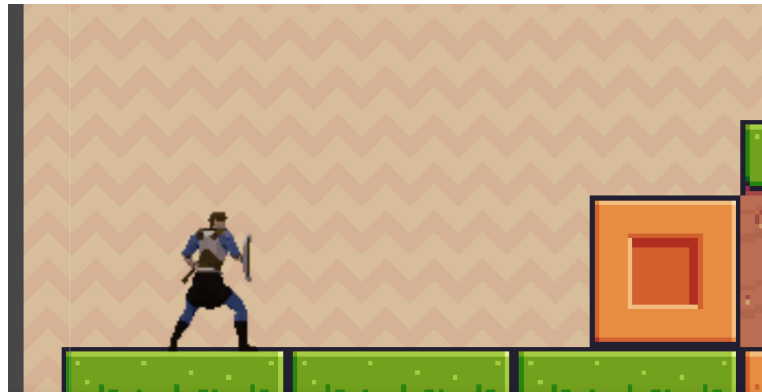
2. Pilih folder yang ingin dibuka atau yang sudah diunduh property



Gambar 8.2 membuat akun unity

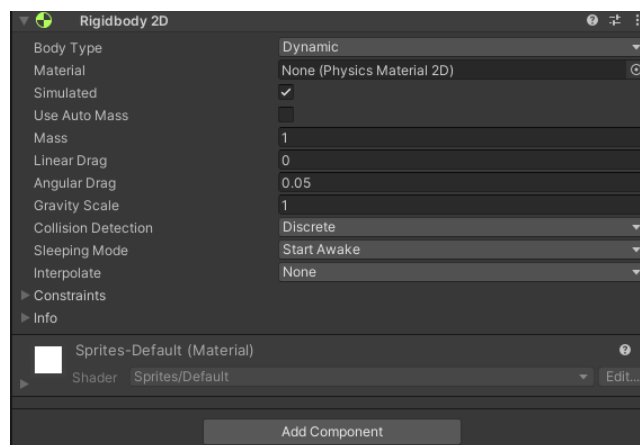


3. Masukkan karakter idle pada hirarki



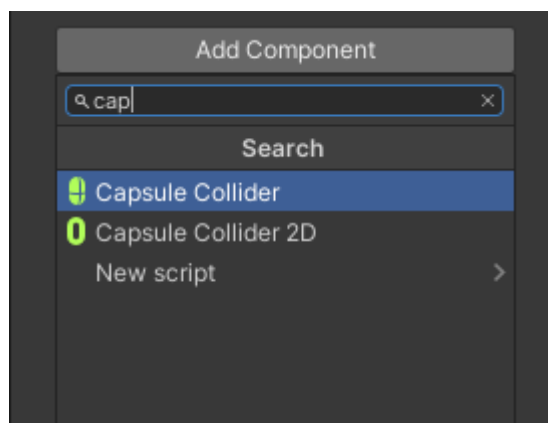
Gambar 8.3 Menambahkan karakter idle

4. Klik karakter tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



Gambar 8.4 menambahkan komponen pada karakter

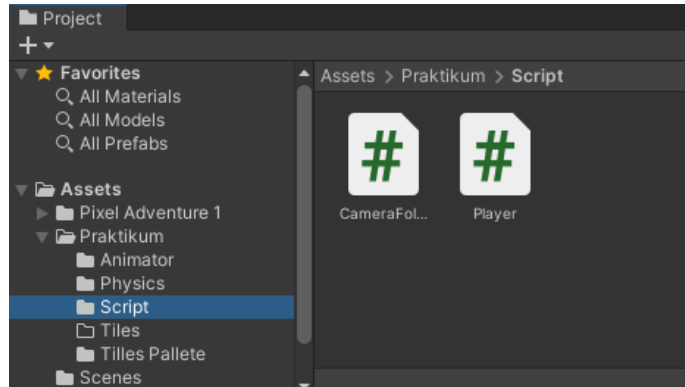
5. Lalu tambahkan komponen Capsule Colider di player-idle-1, lalu klik icon sebelah kanan edit colideresuaikan kapsul collidernya dengan karakter





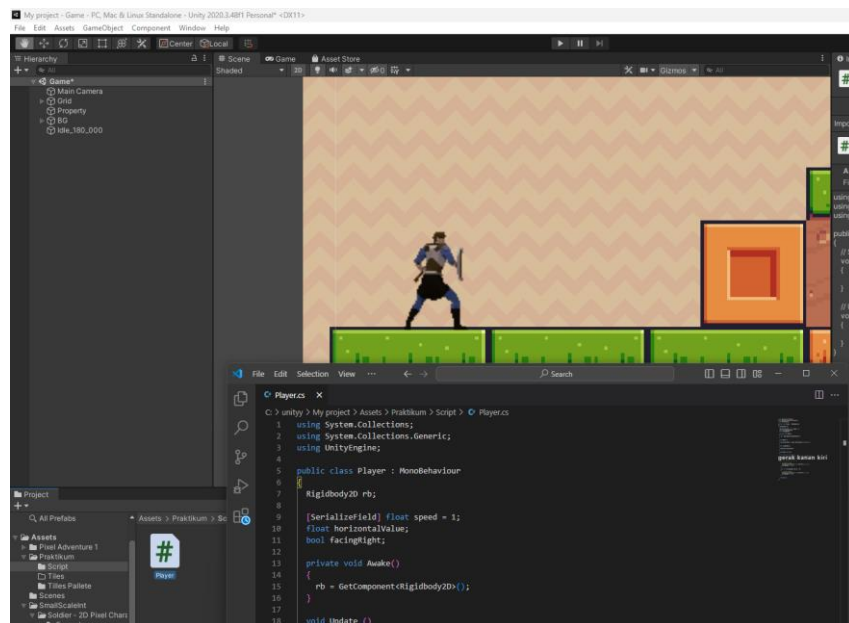
Gambar 8.5 menyesuaikan kapsul collider

6. Buka Folder praktikum, lalu bikin folder baru bernama Script



Gambar 8.6 menambahkan script pada folder

7. Drag & drop script player kedalam Hirarki player-idle-1, lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini



Gambar 8.7 menambahkan script pada player

8. Tambahkan kode pada script player

```
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;

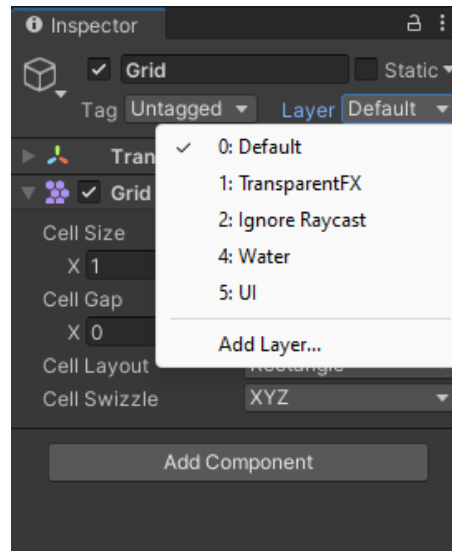
    if (facingRight && dir < 0)
    {
        // ukuran player
        transform.localScale = new Vector3(-1, 1, 1);
        facingRight = false;
    }

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

    #endregion
}
}
```



9. Untuk membuat player loncat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer



Gambar 8.8 Membuat GroundCheck

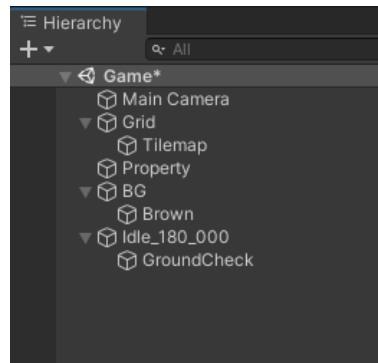
10. Lalu isi “Ground” pada User Layer 6



Gambar 8.9 Menambahkan ground layer

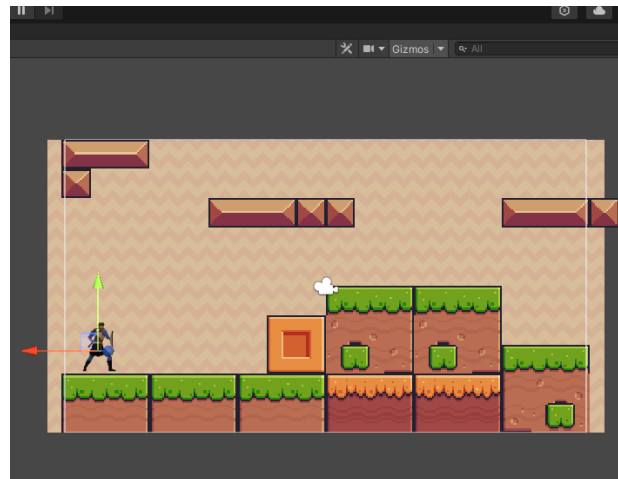


11. Ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja empty dan rename menjadi groundcheck pada player



Gambar 8.10 Membuat groundcheck pada layer

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



Gambar 8.11 Memindahkan groundcheck

13. 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;
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue);
}

void GroundCheck()
{

```



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

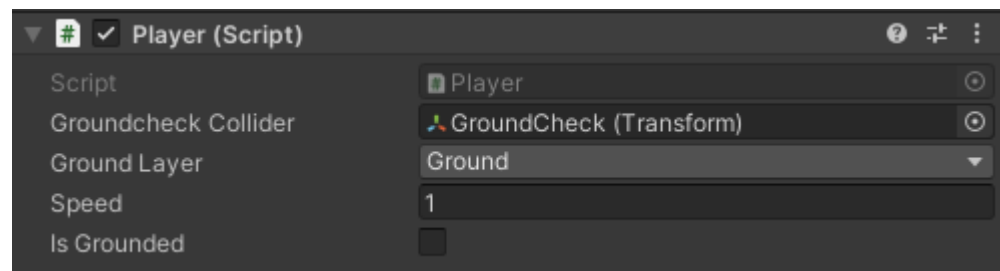
[SerializeField] float jumpPower = 100;

bool jump;

bool jumpflag

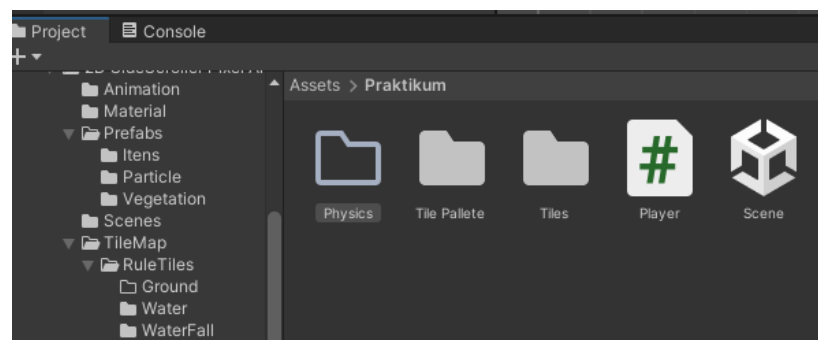
if(isGrounded && jumpflag)
{
isGrounded = false;
jumpflag = false;
rb.AddForce(new Vector2(0f, jumpPower));
}
```

14. Klik karakter, 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 8.13 Ubah inspector

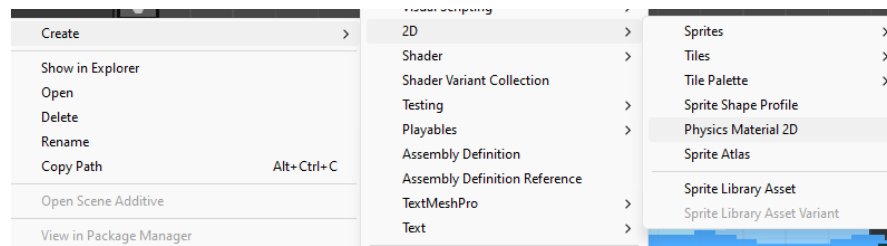
15. Buat folder baru di praktikum bernama “Physics”.



Gambar 8.14 Membuat folder physics

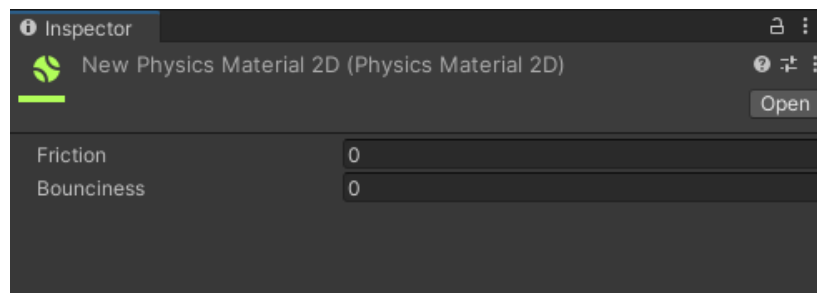


16. Pada folder physics buat 2d physics material



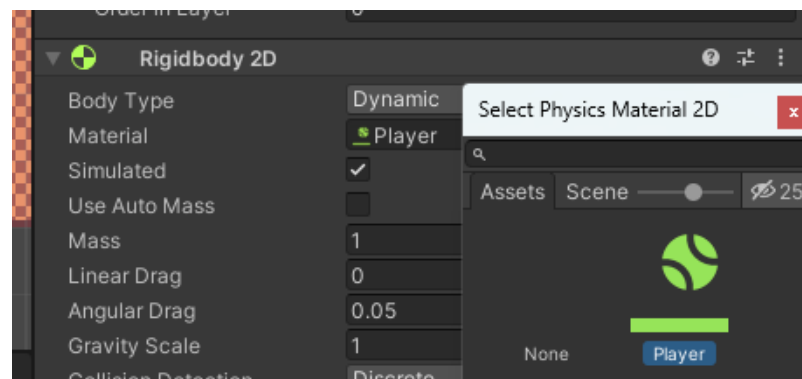
Gambar 8.15 membuat physics material 2d

17. Rename menjadi player dan ubah friction menjadi 0 pada inspector



Gambar 8.16 Mengubah friction

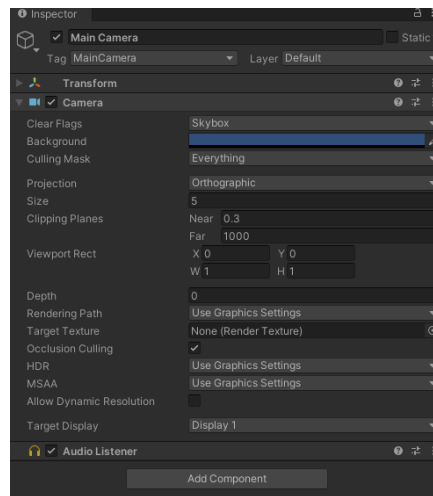
18. Pada bagian hirarki player inspector rigidbody, ubah material ke material physics tadi



Gambar 8.17 menambahkan physics material 2d



19. Pada bagian hirarki main camera, bagian inspector ubah setting seperlunya agar kamera dapat bekerja optimal



Gambar 8.18 mengubah setting inspector camera

20. Masukkan background sesuai asset yang sudah didownload kemudian ubak layer agar background tidak bertumpuk

```
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;
    }

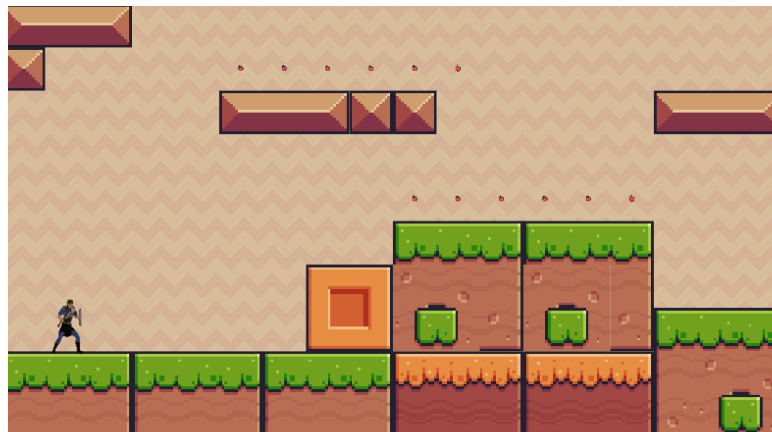
    void FixedUpdate()
```



```
{
    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);}}
```

21. Menambahkan script camera follow ke dalam komponen main kamera pada hirarki, sehingga kamera akan mengikuti player, kemudian render



Gambar 8.19 Hasil camera dan character movement

B. Kuis

X. Kuis 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

3 baris utama digunakan untuk menyambungkan dengan library unity agar komponen dan kode bisa digunakan, pada class terdapat inisialisasi variable



yang digunakan berupa player, kemudian akan diupdate pada void sebagai movement posisi kamera koordinat x, y dan z kamera agar mengikuti target variable yaitu player