



TUGAS PERTEMUAN: 8

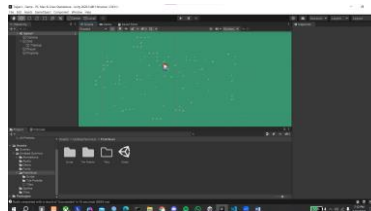
Camera & Character Movement

NIM	:	2118003
Nama	:	Kevin Majesta Ivano
Kelas	:	A
Asisten Lab	:	M. Rafi Faddilani (2118144)

1.1 Tugas 1 : Membuat Pergerakan Player

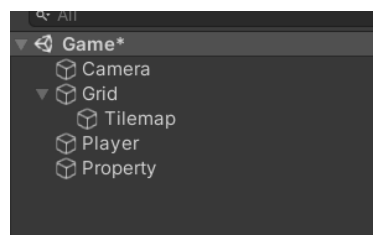
A. Pergerakan Player

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



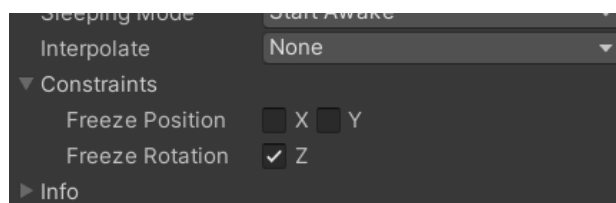
Gambar 1.1 Buka file

2. Tambahkan player bernama player-id, pilih yang idle, Import kedalam Hirarki



Gambar 1.2 Tambah player

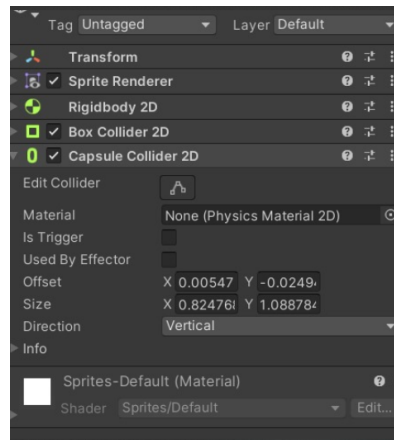
3. Klik player-idle-1 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



Gambar 1.3 Tambah komponen



4. Lalu tambahkan komponen Capsule Colider di player-idle-1, lalu klik icon sebelah kanan edit colider



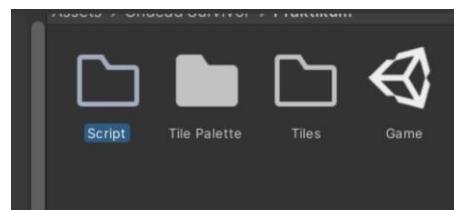
Gambar 1.4 Tambah capsule colider

5. Lalu cocokkan garis oval dengan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya



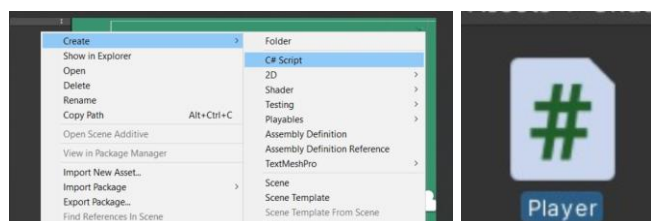
Gambar 1.5 Cocokkan garis oval

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



Gambar 1.6 Buat script

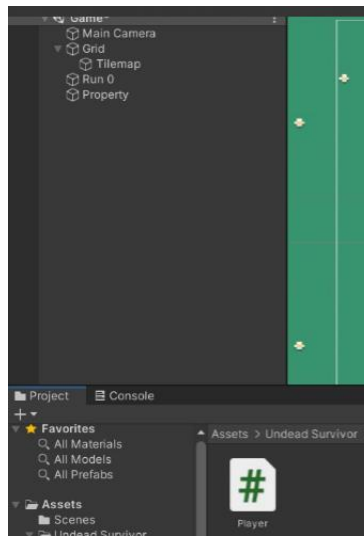
7. Masuk kedalam folder Script, lalu buat C# Script, beri nama Player



Gambar 1.7 Buat script



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



Gambar 1.8 Drop ke script

9. Masukkan source code dibawah ini, pastikan nama public class harus sama 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;
    float verticalValue;
    bool facingRight = true;

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

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

    void FixedUpdate()
    {
        Move(horizontalValue, verticalValue);
    }

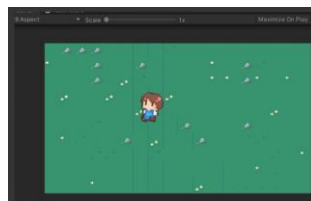
    void Move(float horizontalDir, float verticalDir)
```



```
{
    #region gerak kanan kiri dan atas bawah
    float xVal = horizontalDir * speed * 100 *
Time.fixedDeltaTime;
    float yVal = verticalDir * speed * 100 *
Time.fixedDeltaTime;
    Vector2 targetVelocity = new Vector2(xVal, yVal);
    rb.velocity = targetVelocity;

    if (facingRight && horizontalDir < 0)
    {
        transform.localScale = new Vector3(-2, 2,
1);
        facingRight = false;
    }
    else if (!facingRight && horizontalDir > 0)
    {
        transform.localScale = new Vector3(2, 2, 1);
        facingRight = true;
    }
    #endregion
}
}
```

10. Untuk mencoba Source code diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan

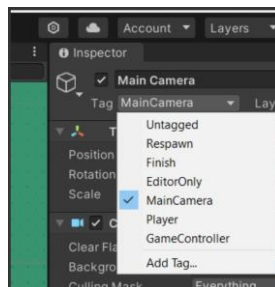


Gambar 1.9 Hasil

1.2 Tugas 2 : Pergerakan Camera

B. Pergerakan Camera

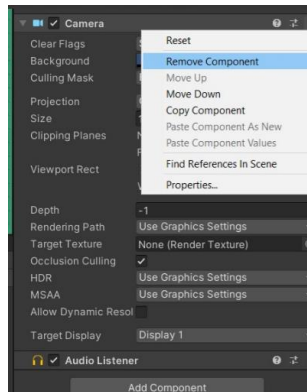
1. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untaged



Gambar 1.10 Ubah tag

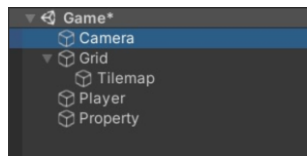


2. Pada Effect Camera pilih Remove Component



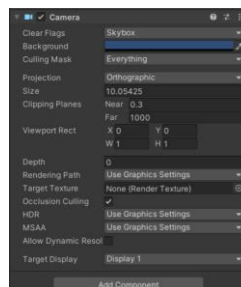
Gambar 1.11 Remove component

3. Create Empty pada Hirarki, dan Rename Menjadi Camera



Gambar 1.12 Create empty

4. Sesuaikan Setting Layer Camera seperti gambar dibawah ini



Gambar 1.13 Sesuaikan setting

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



Gambar 1.14 Buat script



6. 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;

        if (player == null)
        {
            Debug.LogError("Pemain tidak ditemukan!
            Pastikan objek pemain memiliki tag 'Player'.");
        }

        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()
        {
            if (player != null)
            {
                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);

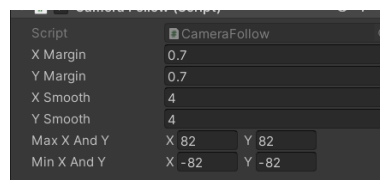
transform.position = new Vector3(targetX,
targetY, transform.position.z);
}
```

7. Drag & drop script CameraFollow Kedalam Layer Camera



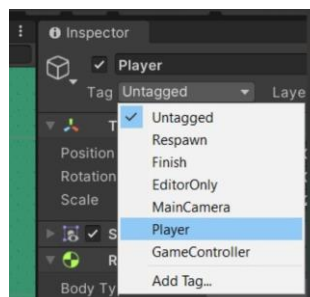
Gambar 1.16 Drop script

8. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya



Gambar 1.17 Ubah Max koordinat

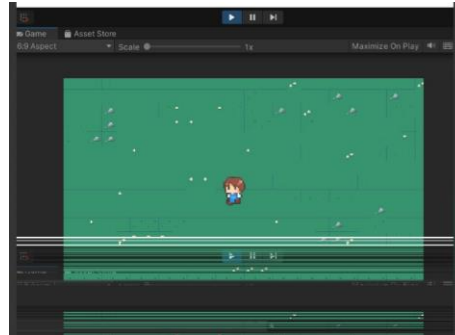
9. Ubah tag di player Untagged menjadi "Player"



Gambar 1.18 Ubah tag player



10. Hasil

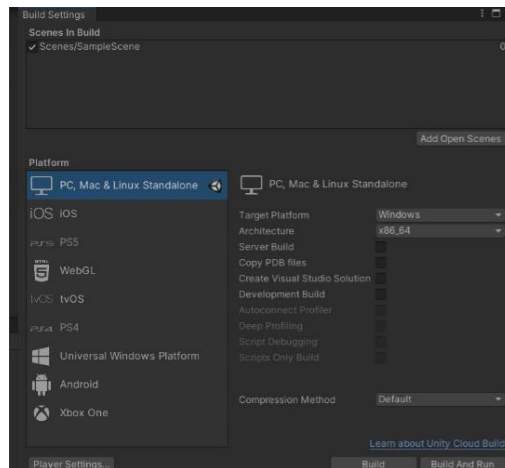


Gambar 1.19 Hasil

Tugas 3 : Render

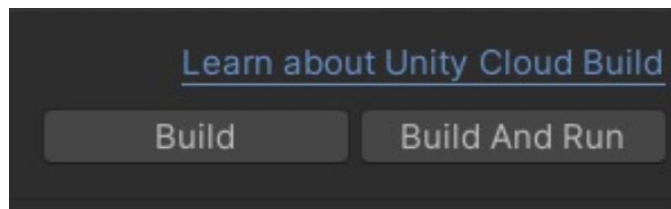
C. Render

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



Gambar 1.20 Tampilan Render

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



Gambar 1.21 Build



3. Hasil



Gambar 1.22 Hasil

D. Repository GitHub

Kevinmajesta/2118003_PRAK_ANIGAME

E. Kuis CameraFollow

```
using UnityEngine;

public class CameraFollow : MonoBehaviour
{
    [SerializeField] private Transform target; // Reference
    to the player object
    [SerializeField] private float smoothSpeed = 0.125f; //
    Speed of camera movement
    [SerializeField] private Vector3 offset = Vector3.zero;
    // Offset from the player object

    private void Update()
    {
        // Calculate the desired camera position
        Vector3 desiredPosition = target.position + offset;

        // Smoothly move the camera towards the desired
        position
        transform.position =
        Vector3.Lerp(transform.position, desiredPosition,
        smoothSpeed);
    }
}
```

Penjelasan :

Kode di atas adalah sebuah script dari Unity untuk membuat kamera mengikuti objek pemain dengan pergerakan yang smooth. Script ini memiliki tiga variabel yang dapat disesuaikan: target untuk referensi ke objek pemain, smoothSpeed untuk mengatur kecepatan pergerakan kamera, dan offset untuk menentukan jarak offset dari objek pemain. Pada setiap frame di metode Update, skrip menghitung posisi yang diinginkan untuk kamera dengan menambahkan offset ke posisi pemain, kemudian



menggunakan `Vector3.Lerp` untuk secara halus menggerakkan kamera dari posisi saat ini menuju posisi yang diinginkan berdasarkan kecepatan yang telah ditentukan. Hal ini menciptakan efek kamera yang mengikuti pemain dengan pergerakan yang smooth dan stabil.