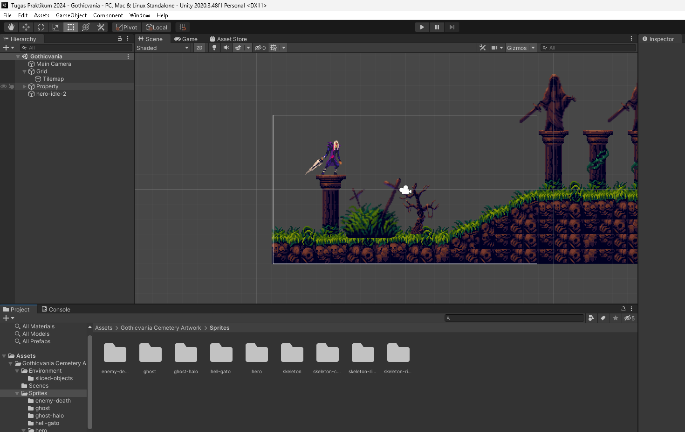
# 8 CAMERA & CHARACTER MOVEMENT

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118073 |
| **Nama** | : | Enrico Erdhani |
| **Kelas** | : | B |
| **Asisten Lab** | : | Maria Avriliana Surat Lelaona (2218096) |

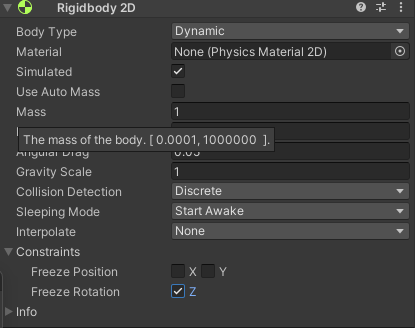
## Tugas 8 : Membuat Membuat Tilemap sesuai asset

1. **Pergerakan Player**
2. Pertama Buka file projek Unity sebelumnya pada bab 7 untuk digunakan kembali



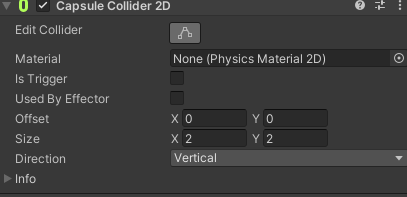
### 8.1 Open Project

1. Lalu Pilih Player Idle tambahkan Component *Rigidbody* 2D Dan Setting seperti gambar berikut, Centang pada*Freeze Rotation* Z



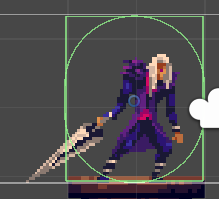
### 8.2 Rigidbody 2D

1. Kemudian tambahkan komponen*Capsule Colider*di Player Idle, lalu klik icon sebelah kanan *edit collider*.



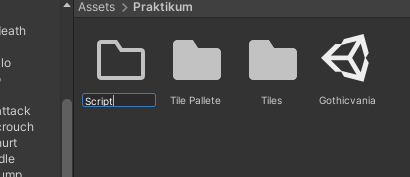
### 8.3 Setting Capsule Colider

1. Kita Paskan Player Dan garis oval dengan karakternya Seperti ini.



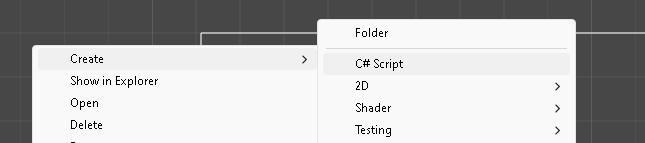
### 8.4 Capsule Colider

1. Jika Sudah, Kita Buat folder baru bernama Script Di Dalam Folder Pratikum.



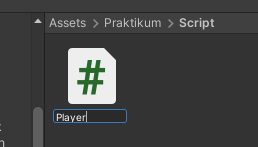
### 8.5 Membuat Folder Baru

1. Kemudian Ke folder *Script*, lalu buat C# *Script*, beri nama Player.



### 8.6 Buat C# Script

1. Lalu Drag & drop *script* player Fall, lalu klik 2x pada *script* player maka akan masuk kedalam *text editor* seperti ini

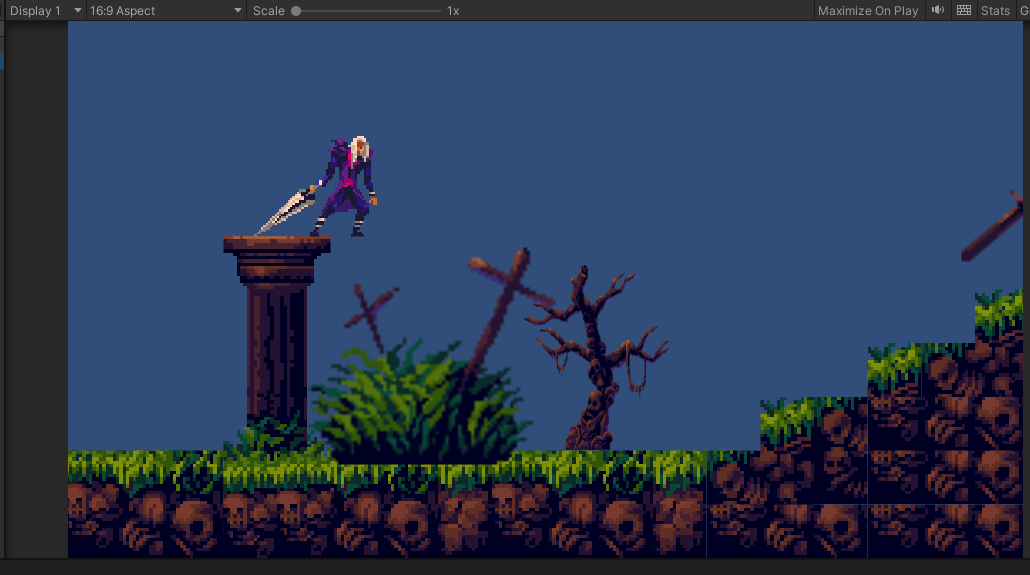


### 8.7 Setting Script

1. Selanjutnya *source code* dibawah ini, pastikan nama *public class* harus sama dengan nama file yang dibuat.

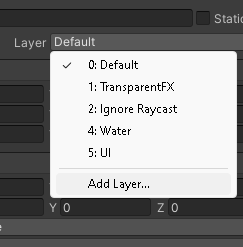
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {    Rigidbody2D rb;      [SerializeField] float speed = 1;    float horizontalValue;    bool facingRight;    private void Awake()    {      rb = GetComponent<Rigidbody2D>();    }    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    }  } |

1. Kemudian Dengan mencoba *Source code* diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan



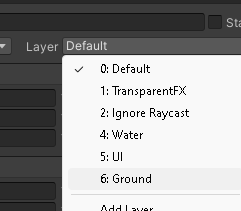
### 8.8 Character bergerak

1. Kita membuat player loncat menggunakan spasi, dengan cara, klik Grid pada *Hierarchy*, pergi ke inspector, pilih *Layer*, Klik *Add Layer*



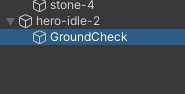
### 8.9 Membuat Layer pada Grid

1. Lalu *Layer* menjadi Ground, jika muncul *pop up Change Layer*, klik yes saja.



### 8.10 Menambahkan layer Ground

1. Klik kanan pada Player Idle, lalu *Create empty*, beri nama *Gorundcheck*



### 8.11 Membuat Groundcheck

1. Pada pada *Hirarki GorundCheck*, lalu gunakan *Move Tool* untuk memindahkan ke bagian bawah Player seperti gambar berikut.



### 8.12 Menggunakan Move Tool

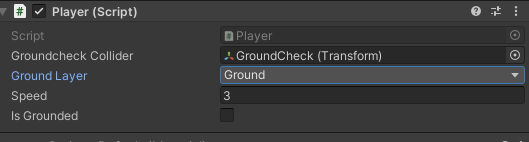
1. ke script Player tambahkan *source code* seperti ini.

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

1. Lalu Buat void *ground check* dibawah void f*ixedUpdate* & 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;  } |

1. Pencet player Fall, lalu ke inspector ke *Grouncheck* *collider* tekan icon lalu pilih yang *Grouncheck Transform*, dan pada *Ground Layer* pilih Groundcheck



### 8.13 Edit Player script

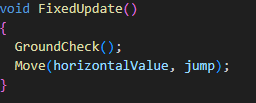
1. Lalu untuk membuat player melompat tambahkan ***script*** berikut

|  |
| --- |
| [SerializeField] float jumpPower = 100;  bool jump; |

1. Tambahkan Lagi *script* berikut di bagian *void* *update*.

|  |
| --- |
| if (Input.GetButtonDown("Jump"))  jump = true;  else if (Input.GetButtonUp("Jump"))  jump = false; |

1. Tambahkan Juga jump pada parameter Move

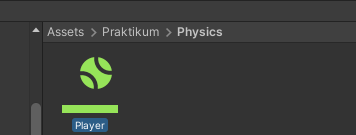


### 8.14 membuat parameter move

1. Tambahkan script berikut pada *void Move*.

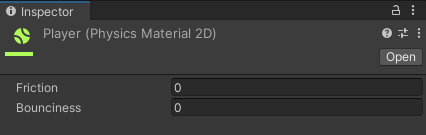
|  |
| --- |
| bool jumpflag  if(isGrounded && jumpflag)  {  isGrounded = false;  jumpflag = false;  rb.AddForce(new Vector2(0f, jumpPower));  } |

1. Pada folder Pyshics create > 2d > physical material 2d , berinama Player



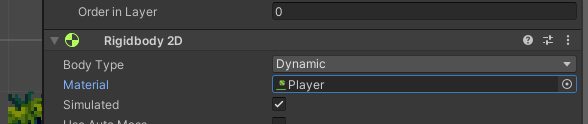
### 8.15 Membuat Physical material 2d

1. Pencet Player *Physics Material* 2D, dibagian menu *inspector*, *friction* & *bounces* ubah menjadi 0



### 8.16 Setting Physical material 2d

1. Menu *Hierarchy* pilih layer player Idle, pada Inspector Cari *Rigidbody* 2D lalu klik icon untuk membuka box select *physhics material* 2d , lalu pilih asset Player yang sudah kita buat tadi



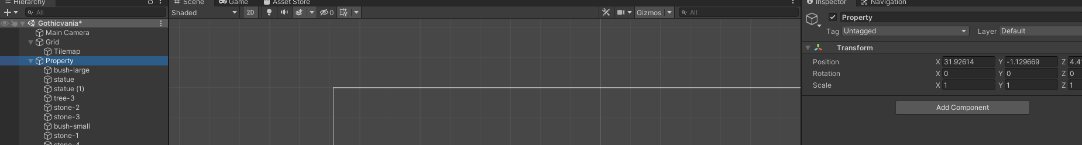
### 8.17 Edit Body Type Rigidbody 2D

1. Kemudian play, player melompat dengan Spasi.



### 8.18 Hasil akhir Gerakan Karakter

1. **Camera Movement**
2. Pertama, Pada *Property* Ubah *Inspector* pada tag Main camera Menjadi untaged

****

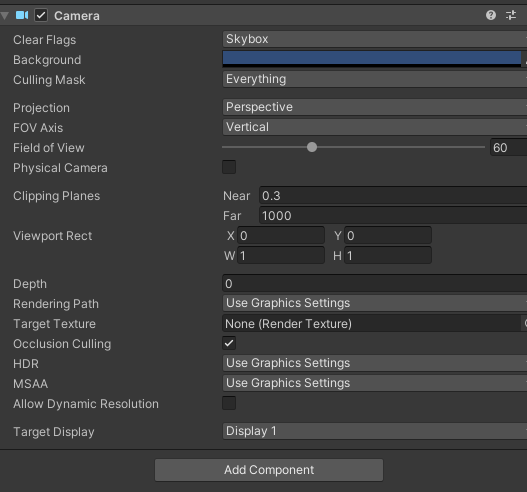
### 8.19 Ubah Property

1. Lalu *Create Empty* pada Hirarki, dan Rename Menjadi Camera

****

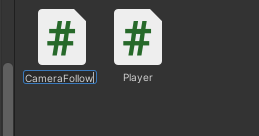
### 8.20 Membuat camera

1. Setting *Layer Camera* seperti gambar dibawah ini.

****

### 8.21 Setting layer camera

1. Kemudian Membuat File *script*baru di folder *Script*dengan nama CameraFollow.

****

### 8.22 File Camerafollow

1. 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);  }  } |

1. Setelah Iyu, Drag & drop script *CameraFollow* Kedalam *Layer Camera*.



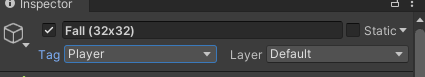
### 8.23 Drag And Drop Script Camerafollow

1. Kemudian Lalu klik pada *camera*, buka *inspector* setting menjadi berikut.

****

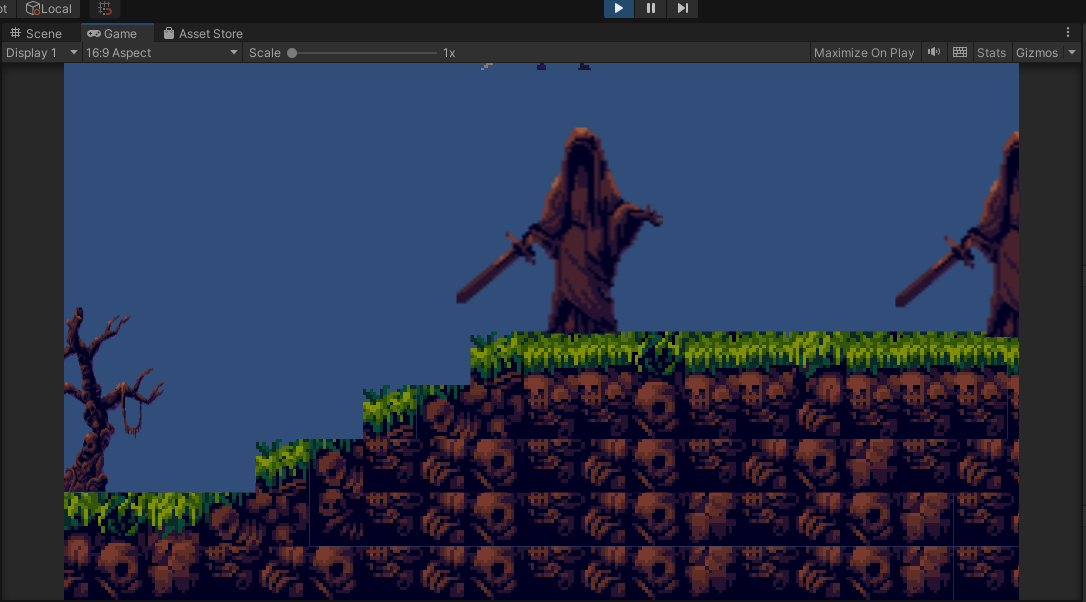
### 8.24 Inspector Camerafollow

1. Kita Ubah tag di player Idle Untagged menjadi Player.

****

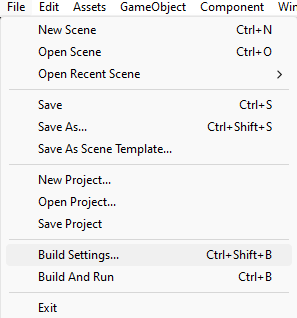
### 8.25 Mengubah Tag Player

1. Pencet play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter,

****

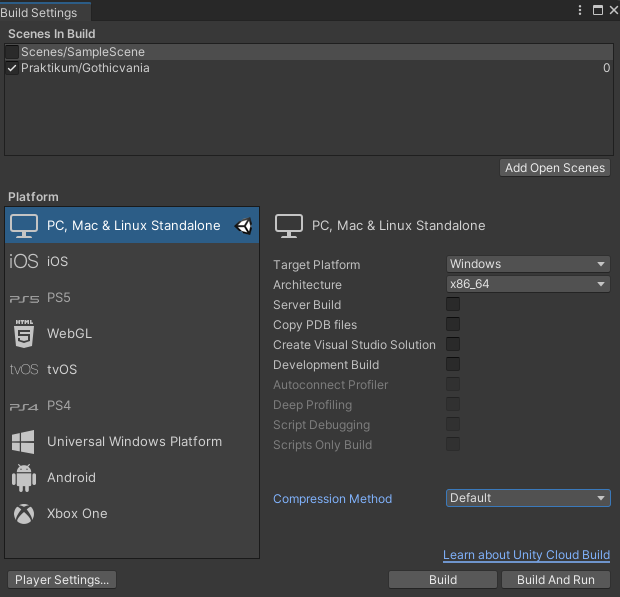
### 8.26 Hasil akhir Camera

1. **Render**
2. Pertama, Pergi Menu File kemudian pilih *Build Setting*.

****

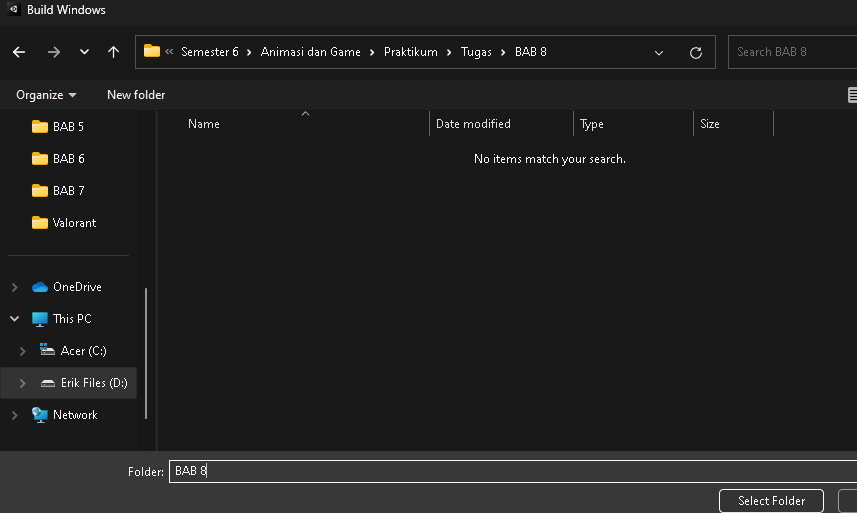
### 8.27 Menu Build Setting

1. Ke *Setting Build* pilih PC, Mac & Linux, Tekan Build, pada project Tugas Kalian.

****

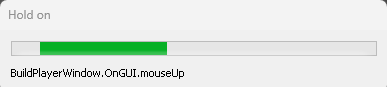
### 8.28 Pilih Project Tugas

1. Kemudian Kita Pilih dimana Project disimpan.



### 8.29 Menyimpan Project Game

1. Setelah Itu Kita Tunggu Hasil Render Dari Pembuatan Game Kita.

****

### 8.30 Hasil Akhir Render

1. **Kuis**

|  |
| --- |
| 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 :

*Source Code* Di Atas Adalah *Class* bernama *CameraFollow* yang diturunkan dari *MonoBehaviour*. Variabel Player bertipe *Transform* yang diberi atribut [*SerializeField*]. Atribut ini memungkinkan variabel player untuk disetel dari editor Unity tanpa menjadikannya *public*. posisi kamera (transform.position) diatur agar mengikuti posisi sumbu x dari objek player (pemain), sedangkan posisi y dan z kamera tetap sama dengan posisi y dan z sebelumnya.

1. **Link Github Pengumpulan**

<https://github.com/EnricoErdhani/2118073_Prak_AniGame>