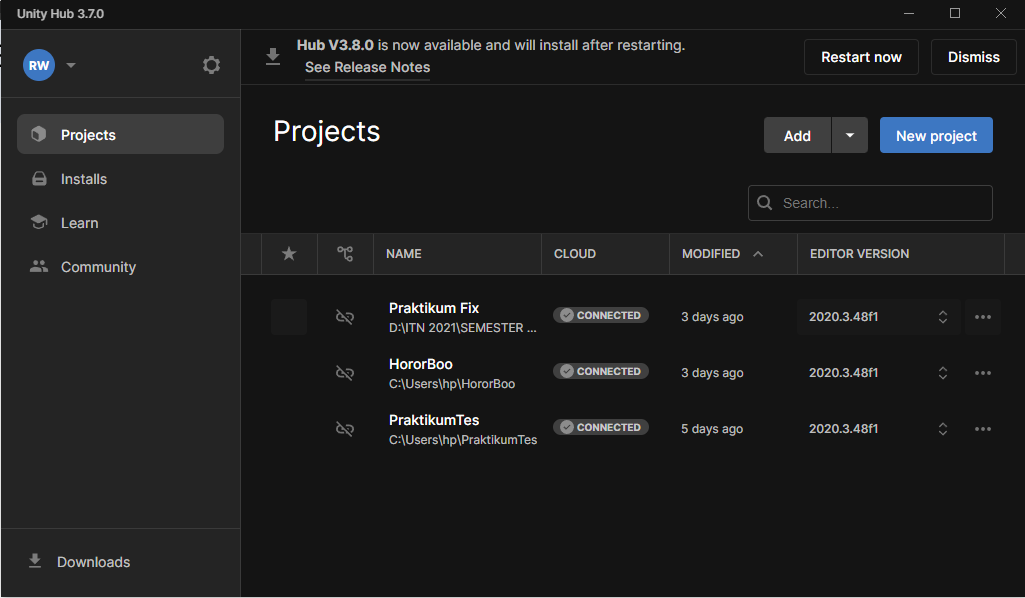
# 8 Camera & Character Movement

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

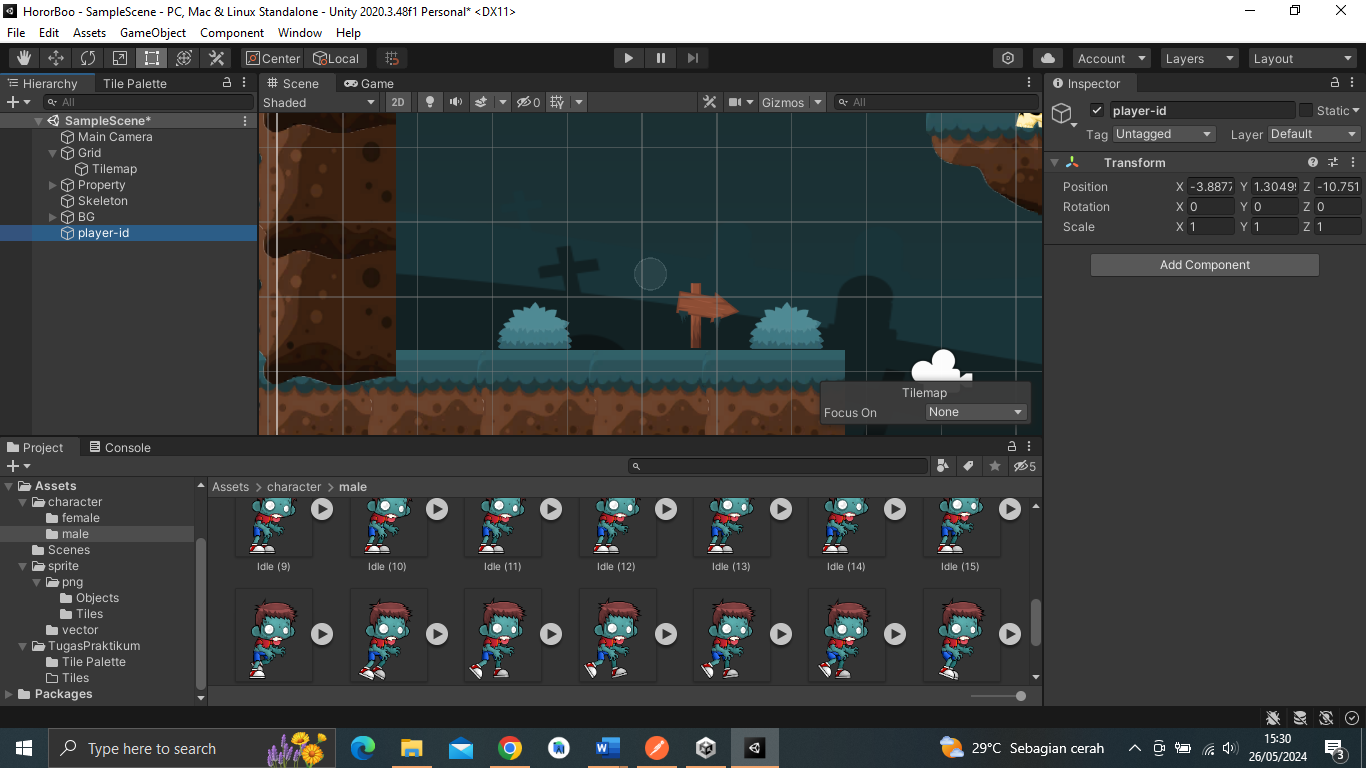
## 1.1 Tugas 8: Camerea & Character Movement

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



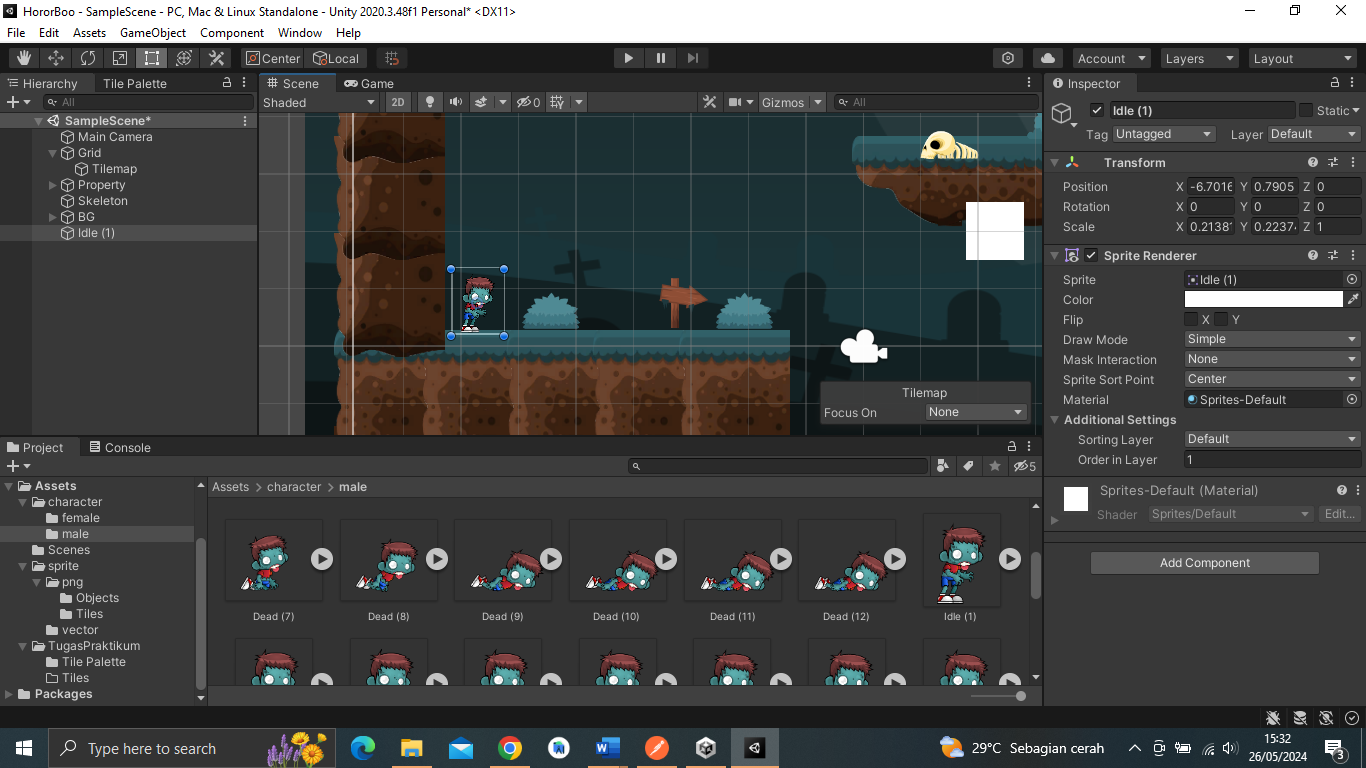
Gambar 8. 1 Unity Hub

1. Berikut merupaan project yang sebelemunnya dibuat, setelah itu hapus bagian idle(1)



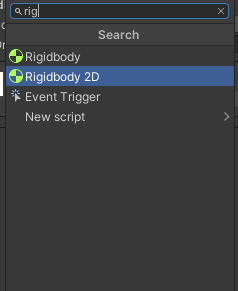
Gambar 8. 2 Project Sebelumnya

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



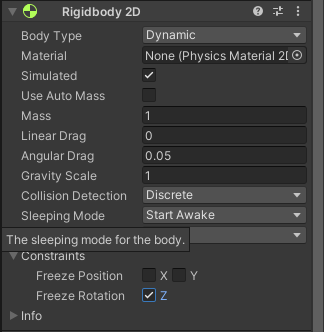
Gambar 8. 3 Menambahkan idle(1)

1. Setelah itu klik pada bagian karakter dan tambahnkan RigidBody 2D pada karakter



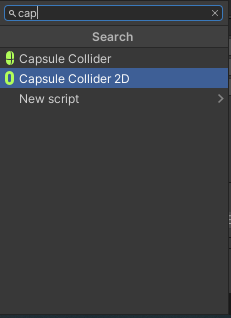
Gambar 8. 4 Riggid Body

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



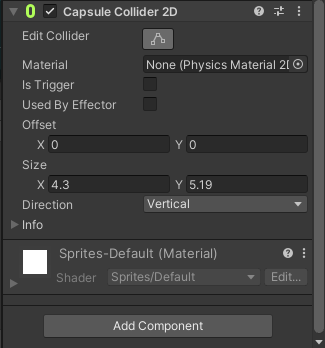
Gambar 8. 5 Freeze Rotation

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



Gambar 8. 6 Menambahkan Capsule Colider

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



Gambar 8. 7 Edit Colider

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



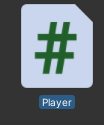
Gambar 8. 8 Edit Colider

1. Buat folder baru bernama “Script” di dalam folder “TugasPraktikum”



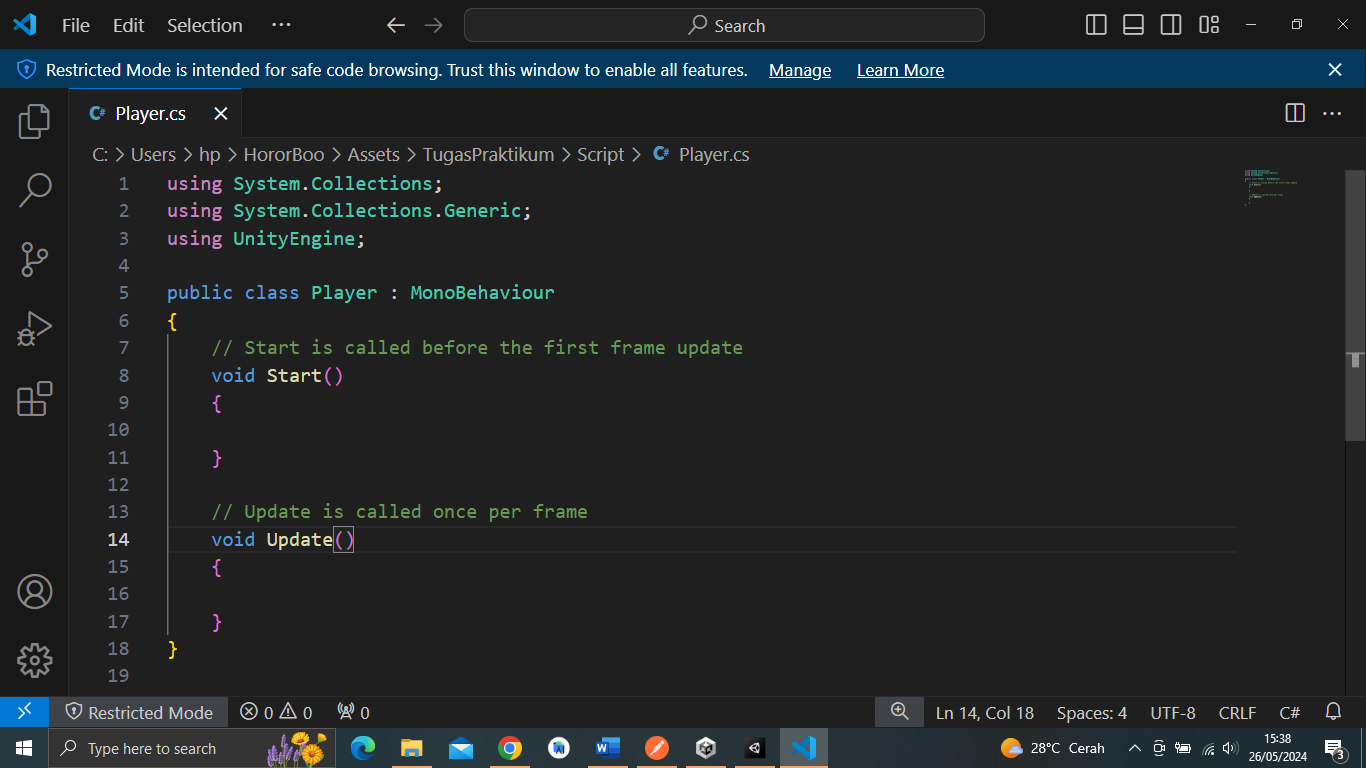
Gambar 8. 9 Folder Baru Bernama Script

1. Dalam folder script buat file script baru dengan nama “Player”



Gambar 8. 10 Membuat File Script

1. 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 menggunkana visual studio begitupun sebaliknya maka menggunaka vscode

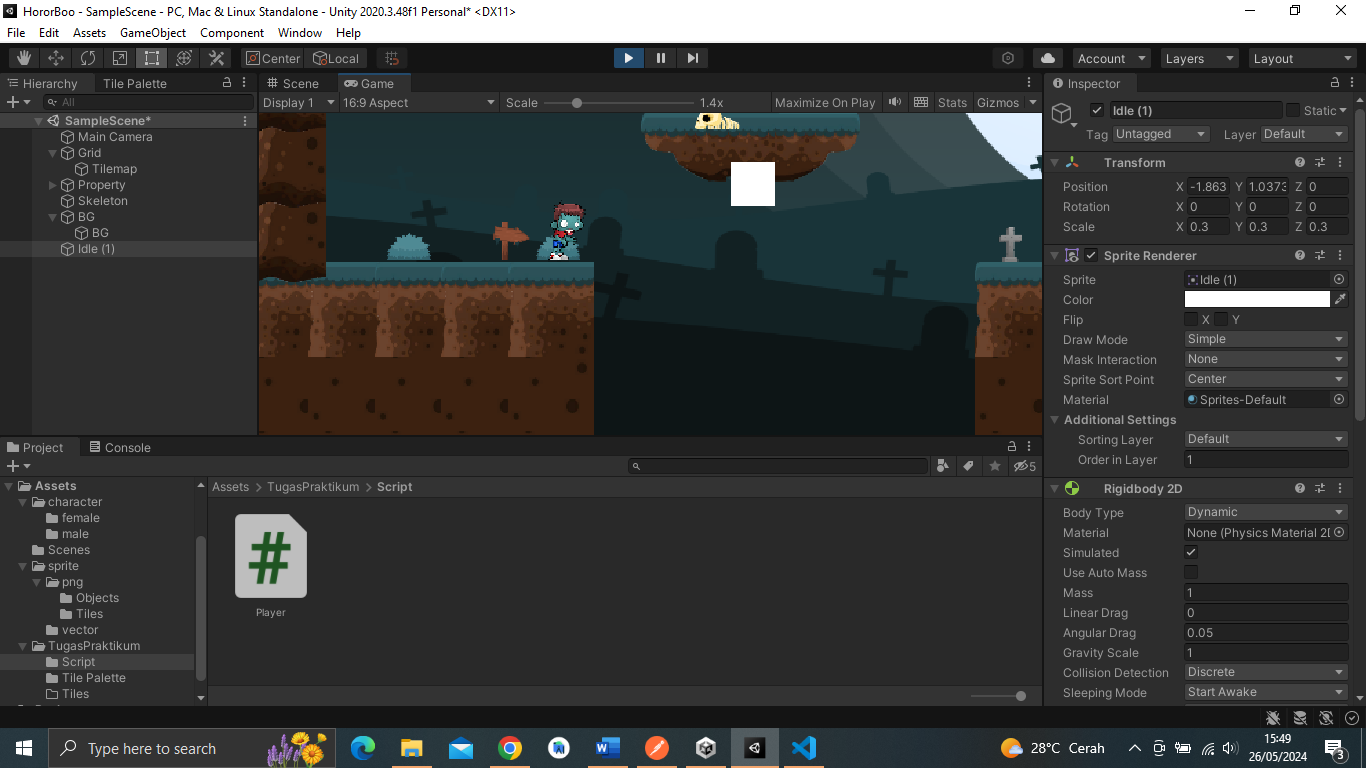


Gambar 8. 11 Visual Studio Editor

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

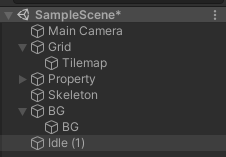
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour{  Rigidbody2D rb;  [SerializeField] float speed = 7;  float horizontalValue;  bool facingRight;  bool jump;  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  }  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  } |

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



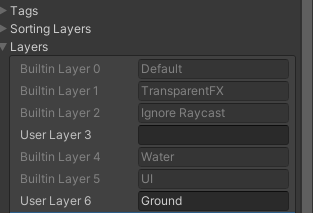
Gambar 8. 12 Mode Play

1. Pada hierarki klik pada bagian grid



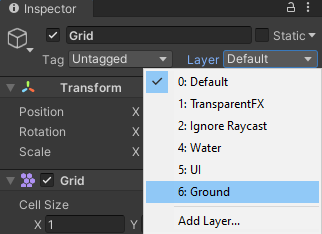
Gambar 8. 13 Masuk Kedalam Grid

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



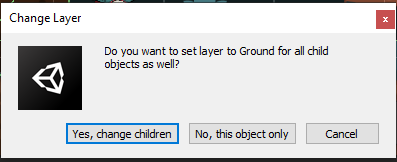
Gambar 8. 14 Ground Layer

1. Setelah itu masuk kedalam bagian layer pilih ground



Gambar 8. 15 Layer Ground

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



Gambar 8. 16 Change Layer

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



Gambar 8. 17 Hierarki GroundCheck

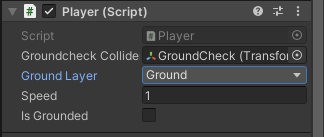
1. Setelah itu masuk kedalam source code “Player.cs” tambahkan kode berikut untuk pelengkap agar player bisa meloncat

|  |
| --- |
| [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; |

1. 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;  } |

1. 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

1. Tambahkan kode berikut dibagian atas di bawah nama class

|  |
| --- |
| [SerializeField] float jumpPower = 500;  bool jump; |

1. 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;  } |

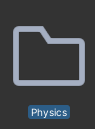
1. Setelah itu bagian method fixedUpdate() tambahkan script berikut

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

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

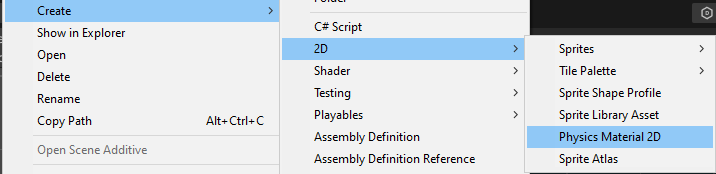
|  |
| --- |
| 1. if(isGrounded && jumpflag) 2. { 3. isGrounded = false; 4. jumpflag = false; 5. rb.AddForce(new Vector2(0f, jumpPower));   } |

1. Setelah itu buat folder beru di dalam folder “TugasPraktikum” dengan nama “Physic”



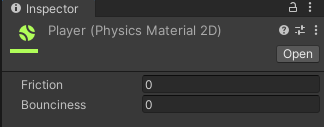
Gambar 8. 19 Membuat folder physic

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



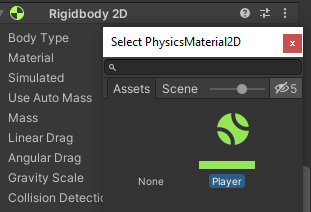
Gambar 8. 20 Folder Physic

1. Pada physic bagian Friction dan Bouches ubah menjadi 0



Gambar 8. 21 Inspetor Physic

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



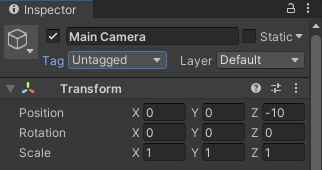
Gambar 8. 22 Material Physic

1. Mainkan game untuk menguji apalkah karakter sudah dapat loncat



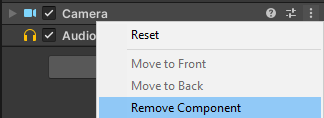
Gambar 8. 23 Mainkan Game

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



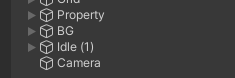
Gambar 8. 24 Tag Untagged Camera

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



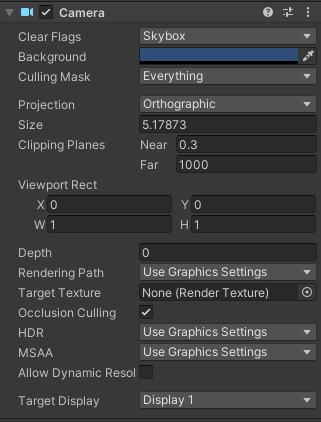
Gambar 8. 25 Remove Component Camera

1. Buat hierarki baru dengan nama kamera



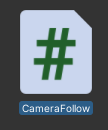
Gambar 8. 26 Hierarki Baru Kamera

1. Pada bagian projection inspector Camera pilih orthobliographic



Gambar 8. 27 Setting Orthobliograpic

1. Pada folder TugasPraktikum buat file script baru bernama CameraFollow



Gambar 8. 28 File Camera Follow

1. Masukan kode program berikut kedalam script CameraFollow

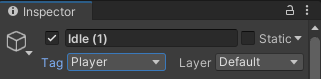
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraFollow : MonoBehaviour  {  public float xMargin = 0.5f;  public float yMargin = 0.5f;  public float xSmooth = 4f;  public float ySmooth = 4f;  public Vector2 maxXAndY;  public Vector2 minXAndY;  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);  }  } |

1. Setelah itu ubah settingan pada bagian Camera Follow



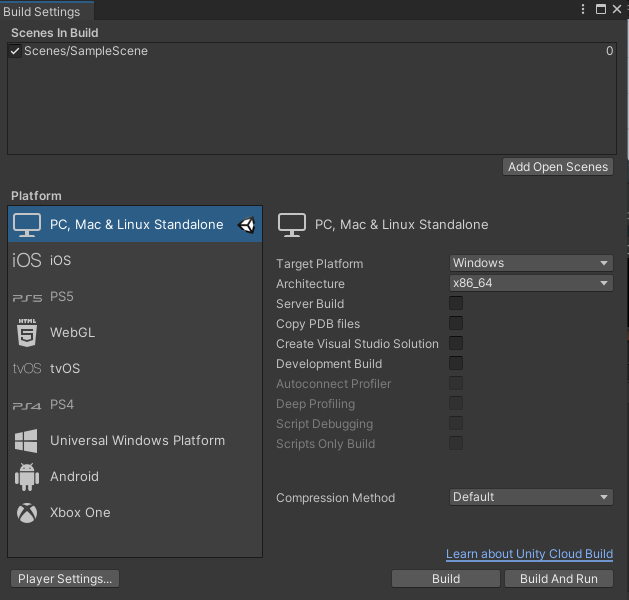
Gambar 8. 29 Camera Follow

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



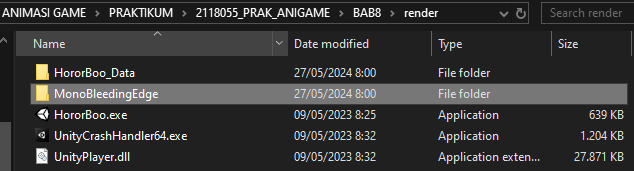
Gambar 8. 30 Player Idle

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



Gambar 8. 31 Build Setting

1. Hasil akhir render



Gambar 8. 32 Hasil Akhir Render

**Link Github Pengumpulan**

Link : https://github.com/arifmalabar/2118055\_PRAK\_ANIGAME.git