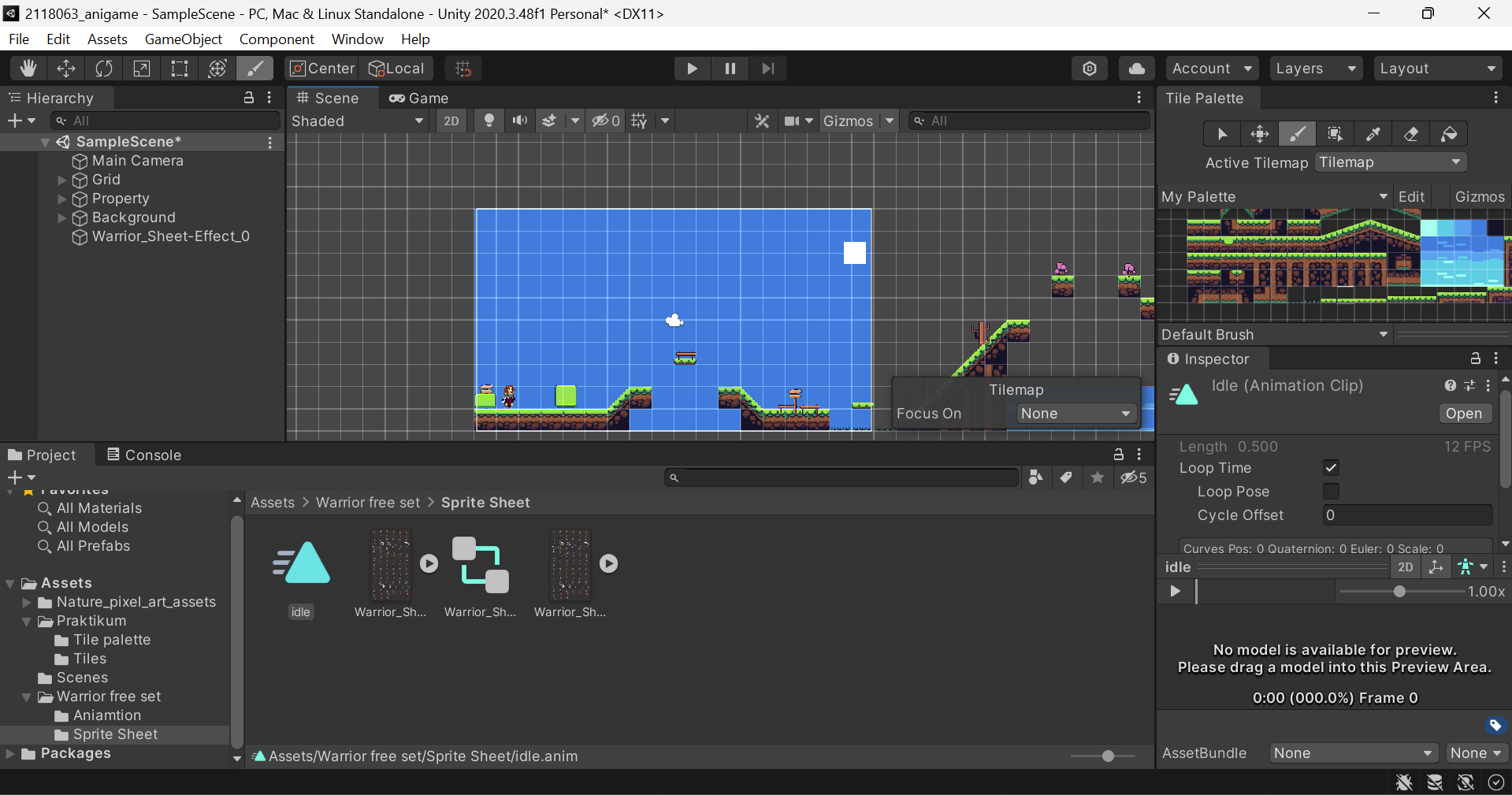
**TUGAS PERTEMUAN: 8 CAMERA & CHARACTER MOVEMENT**

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118063 |
| **Nama** | : | Ahmad Faruq Nuaval Ananda |
| **Kelas** | : | B |
| **Asisten Lab** | : | Bagas Anardi Surya W (2118004) |

8.1 Tugas 1 :

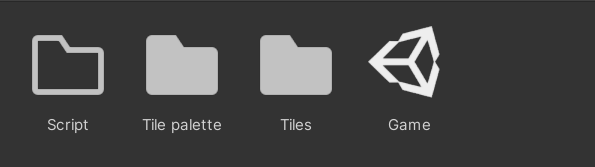
A. Pergerakan Player

* 1. Buka File projek unity sebelumnya pada bab 7 dan gunakan Kembali seperti pada gambar dibawah ini :



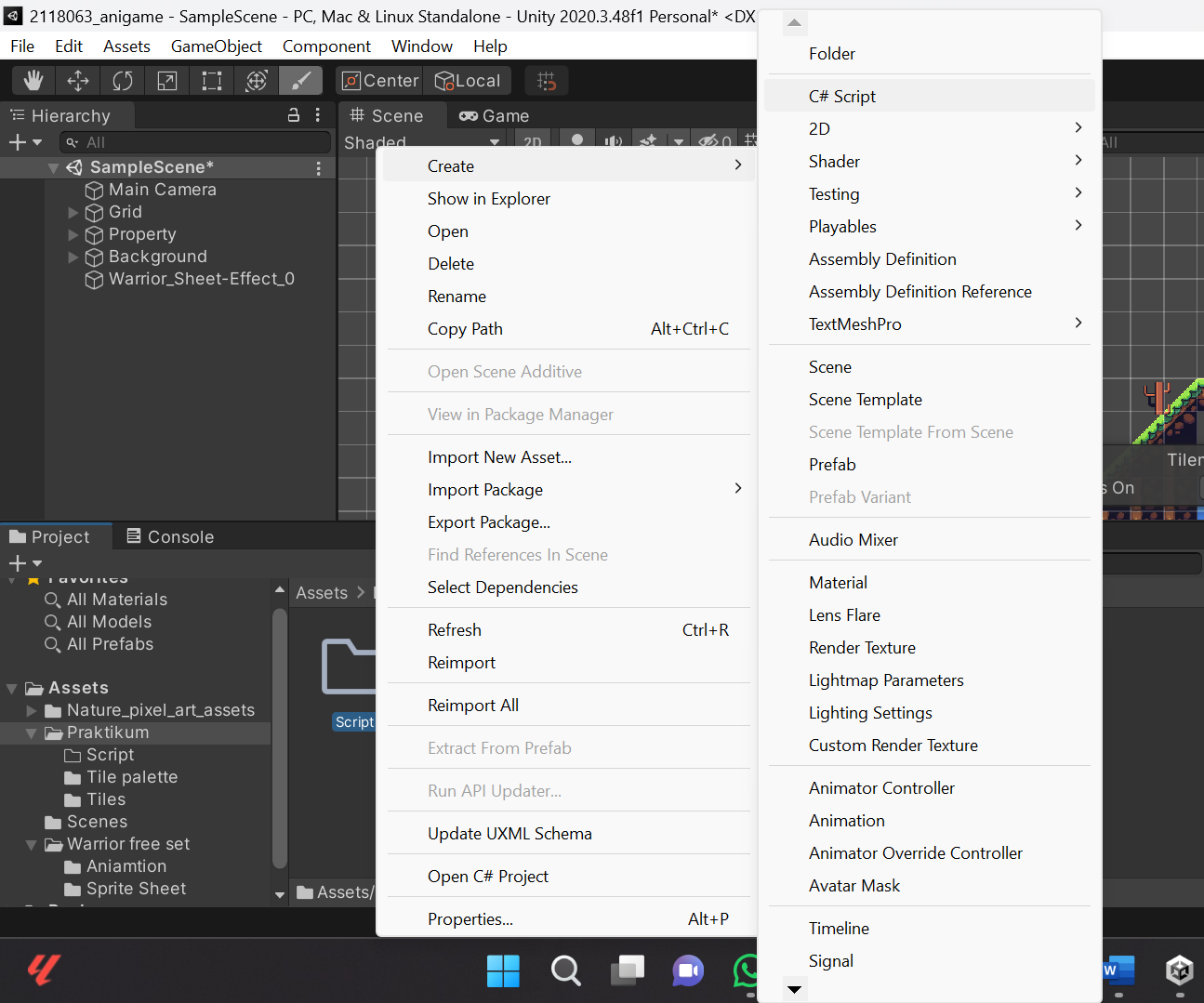
Gambar 8.1 Membuka Project Sebelumnya

* 1. Buat folder baru bernama “Script” dalam folder Tugas\_Praktikum



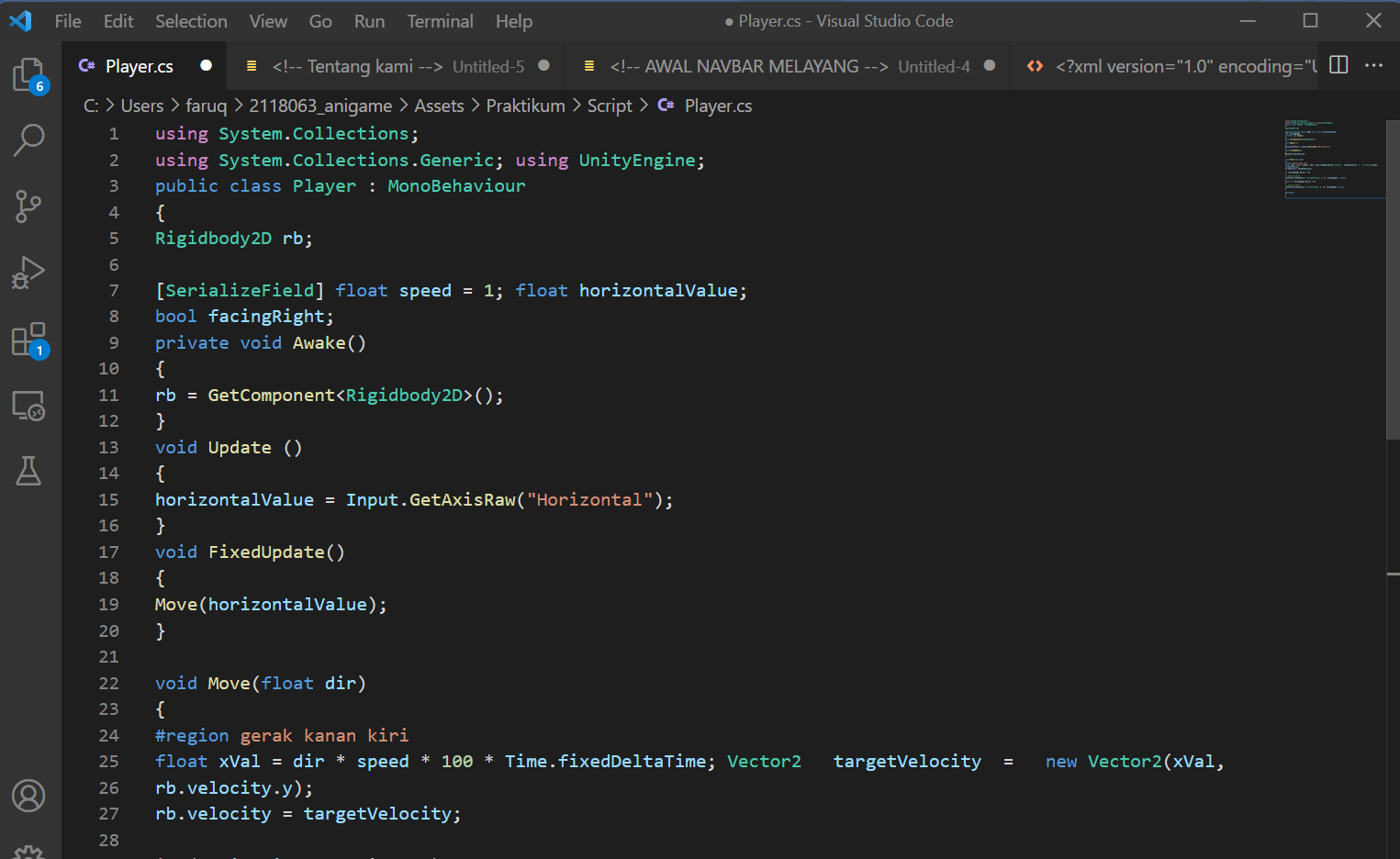
Gambar 8.2 Membuat Folder Baru

* 1. Buat file Script baru, caranya klik kanan folder Script, pilih Create>C# Script dan beri nama file script tersebut dengan “Player”



Gambar 8.3 Membuat File Script baru

* 1. Drag file script tersebut kedalam GameObject “Player” Klik 2x pada File script player tersebut maka akan di alihkan ke VS Code



Gambar 8.4 Tampilan Membuka File Script Hero

* 1. Masukan Source Code

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

}

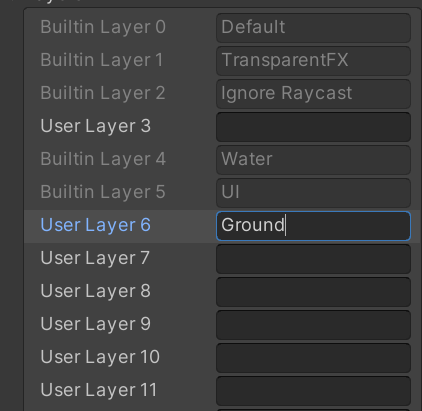
}

* 1. Jika di Play karakter sudah bisa bergerak ke kanan dan ke kiri dengan menekan arah kanan kiri di keyboard



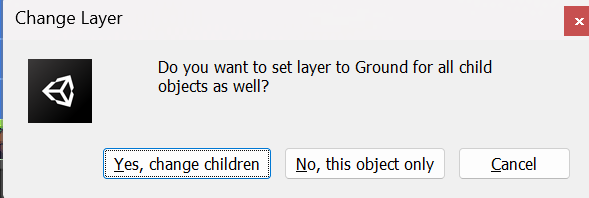
Gambar 8.5 Tampilan Bergerak Karakter

* 1. Untuk membuat Ground Check, klik Grid pada Hierarchy, pergi ke Inspector, pilih Layer, klik Add Layer. Pada User layer 6 isikan dengan nama Ground seperti pada gambar dibawah ini



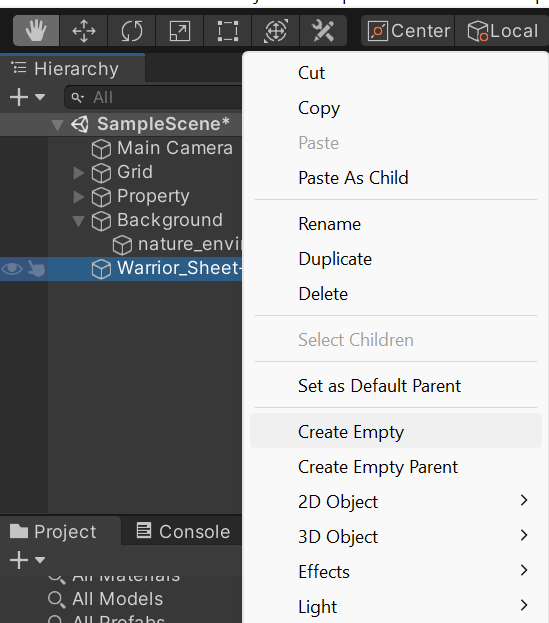
Gambar 8.6 Tampilan Memberi nama layer

* 1. Ganti Layer Default menjadi Ground, Jika muncul notifikasi lakukan klik Yes, change children



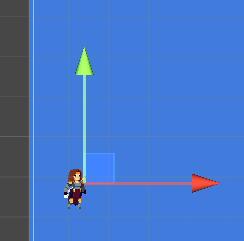
Gambar 8.7 Tampilan mengganti layer

* 1. Pada Hierarchy klik kanan pada Player kemudian pilih Create Empty dan ubah namanya menjadi GroundCheck



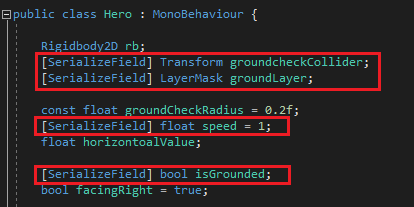
Gambar 8.8 Tampilan Create Empty pada Hierarchy

* 1. Klik GroundCheck, kemudian gunakan Move Tool untuk untuk menggeser panah berwarna hijau-merah pada objek karakter seperti gambar dibawah ini



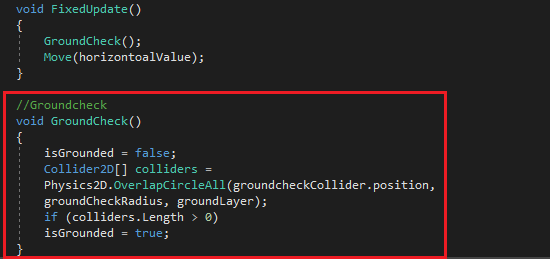
Gambar 8.9 Tampilan Move Tool pada GroundCheck

* 1. Kembali ke visual studio, tambahkan Source Code seperti gambar dibawah ini



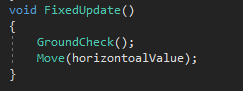
Gambar 8.10 Tampilan Menambahkan Source Code

* 1. Tambahkan Void baru yaitu GroundCheck() dibawah void FixedUpdate



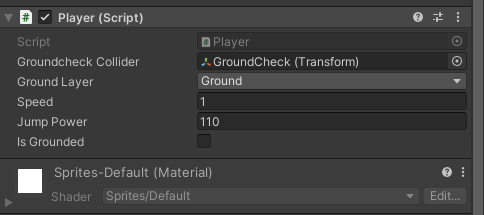
Gambar 8.11 Tampilan Menambahkan Source code

* 1. Tambahkan GrondCheck(); dalam void FixedUpdate



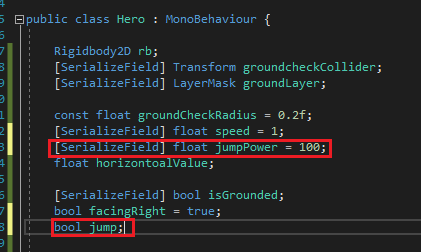
Gambar 8.12 Tampilan Menambahkan Source Code

* 1. Pada Hierarchy Unity Editor, klik Hero, kemudian klik dan drag GroundCheck kedalam Groundcheck Collider yang ada pada Inspector



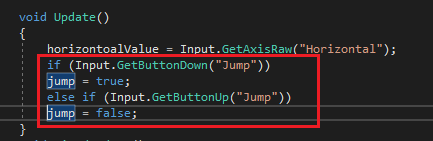
Gambar 8.13 Tampilan Drag GroundCheck

* 1. Untuk membuat sebuah mekanisme melompat pada karakter, tambahkan source code seperti pada gambar dibawah ini



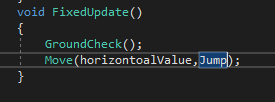
Gambar 8.14 Tampilan Menambahkan Source Code

* 1. Tambahkan Source code dibawah ini di dalam void update



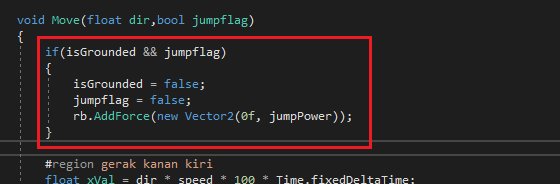
Gambar 8.15 Tampilan Menambahkan Source Code

* 1. Pada void FixedUpdate() tambahkan Kode “Jump” seperti pada gambar dibawah ini



Gambar 8.16 Tampilan Menambahkan Source Code

* 1. Tambahkan Source Code dibawah ini, letakkan dalam void Move



Gambar 8.17 Tampilan Menambahkan Source Code

* 1. Buat folder baru di dalam Tugas\_Pratikum dengan nama Physics



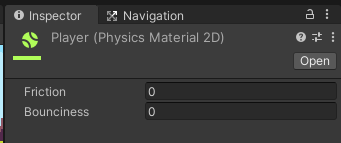
Gambar 8.18 Tampilan Membuat Folder Baru

* 1. Klik kanan folder “Physics” pilih Create>Physics Material 2D dan beri nama “Player”



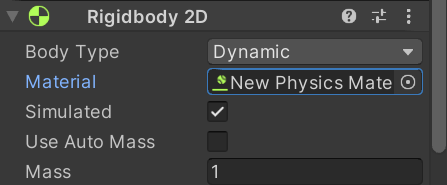
Gambar 8.19 Tampilan Create Physics Material 2D

* 1. Klik file tersebut, dan pergi ke Inspector, ubah nilai Friction dan Bouncies menjadi 0



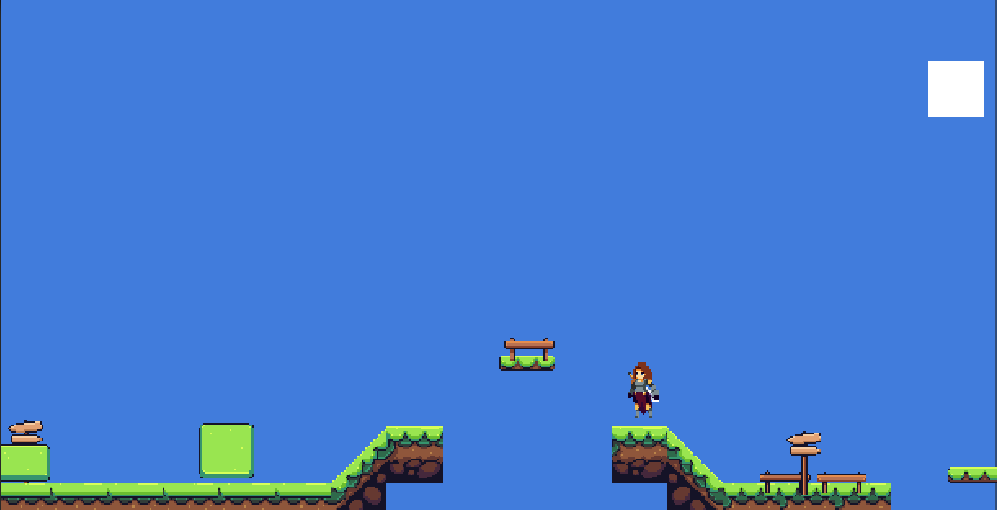
Gambar 8.20 Tampilan Konfigurasi Physics Material 2D

* 1. Klik objek Karakter, kemudian drag file Physics 2D tadi kedalam Material yang ada pada Rigidbody 2D



Gambar 8.21 Tampilan Drag File Physics 2D

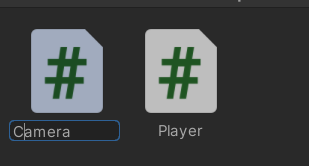
* 1. Tekan play untuk mengetahui hasil dari script yang telah di masukan maka karakter sekarang bisa bergerak ke kiri , ke kanan dan melompat



Gambar 8.22 Tampilan Running Aplikasi

B. Camera Movement

* 1. Tambahkan Script baru di dalam folder Tugas\_Praktikum dan berikan nama CameraFollow.cs



Gambar 8.23 Tampilan Menambahkan File script baru

* 1. Masukan Source Code pada CameraFollow.cs

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, maxXAndY.x); targetY =

Mathf.Clamp(targetY, maxXAndY.y); transform.position = new

Vector3(targetX, transform.position.z);

}

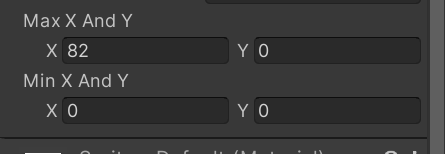
}

minXAndY.x,

minXAndY.y,

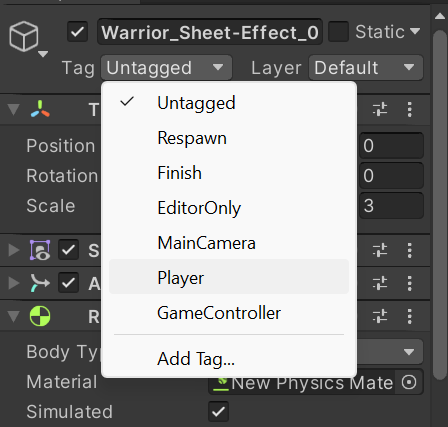
targetY,

* 1. Simpan file script tersebut, kemudian drag dan masukkan ke dalam Game Object Main Camera. Pada Inspector Main Camera lakukan setting Camera Follow seperti gambar berikut



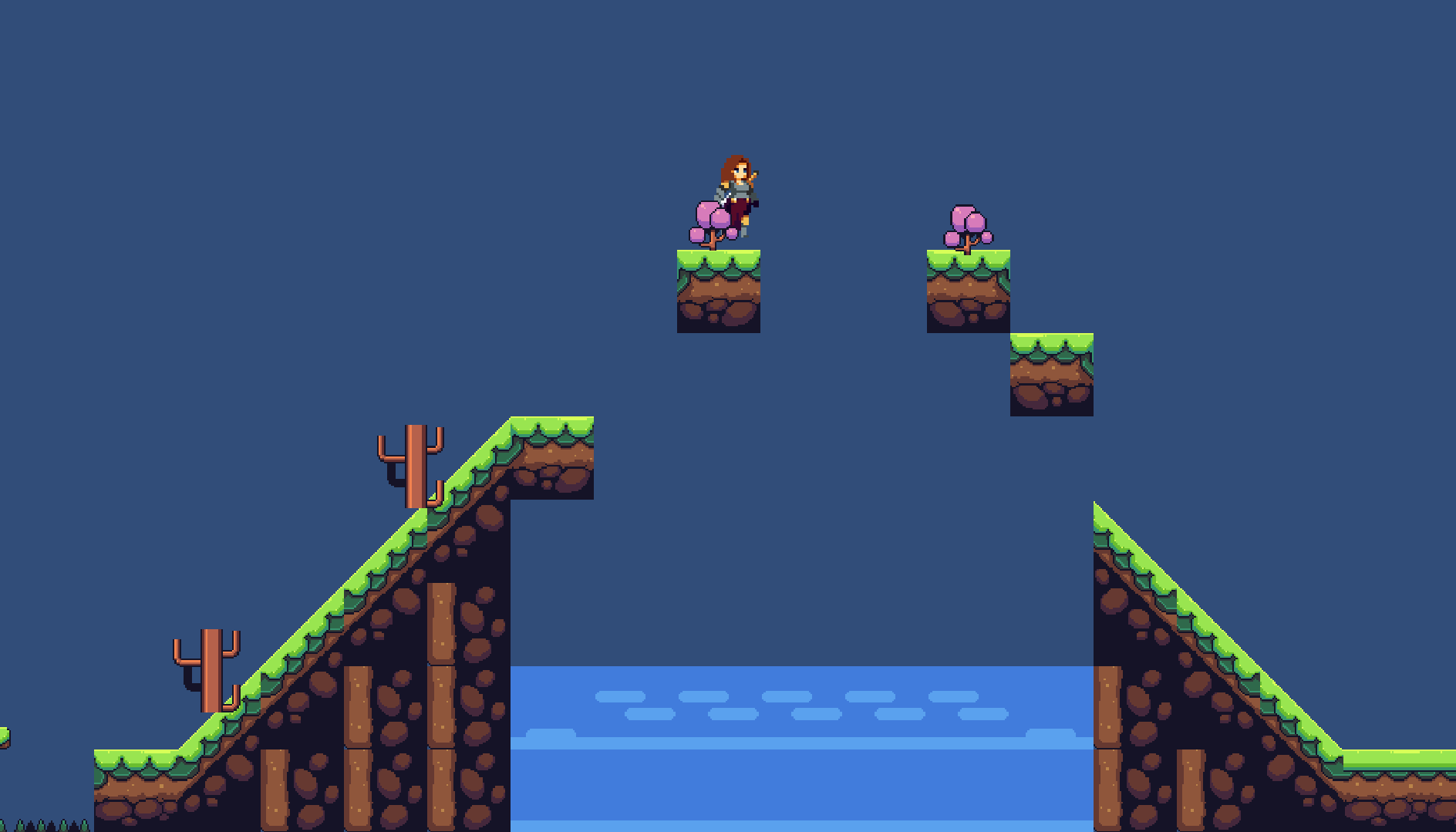
Gambar 8.24 Tampilan Setting CameraFollow

* 1. Klik game object Player, pergi ke Inspector dan ubah Tag menjadi Player



Gambar 8.25 Tampilan merubah Tag pada Inspector

* 1. Tampilan Hasil Running



Gambar 8.26 Tampilan Hasil Running

**Kuis CameraFollow**

**KUIS**

using System.Collections;

using System.Collections.Generic; using UnityEngine; public class Camera Follow: MonoBehaviour

{

Public class CameraFollow: MonoBehaviour

{

[SerializeField] private Transform player; void Update () {

transform.position= new Vector3 (player. position.x, transform.position.y, transform.position.z);

}

Penjelasan :

Di atas ialah coding Unity yang digunakan untuk membuat kamera mengikuti pergerakan *player*. Skrip ini mendeklarasikan kelas *CameraFollow`* yang diwarisi dari *MonoBehaviour*, yang merupakan kelas dasar untuk semua skrip yang terpasang pada objek di Unity. Pada skrip ini, terdapat variabel *player* bertipe *Transform* yang diberi atribut *SerializeField* agar bisa diatur melalui inspector di Unity editor. Di dalam metode *Updat*`, yang dipanggil sekali per frame, posisi kamera diatur untuk mengikuti posisi pemain pada sumbu x, sementara sumbu y dan z tetap sama dengan posisi kamera sebelumnya.