TUGAS PERTEMUAN: 8

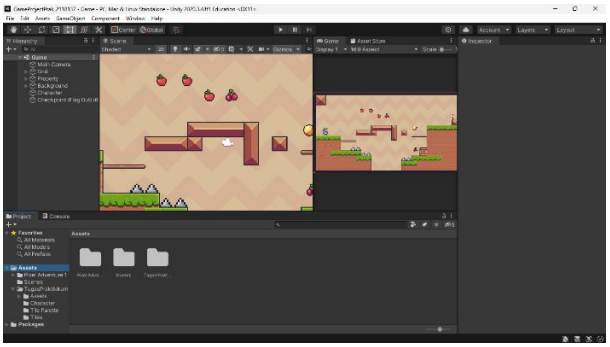
CAMERA & CHARACTER MOVEMENT

NIM	:	2118137
Nama	:	Rian Setya Budi
Kelas	:	C
Asisten Lab	:	Bagas Anardi Surya W (2118004)

8.1 Tugas 8 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

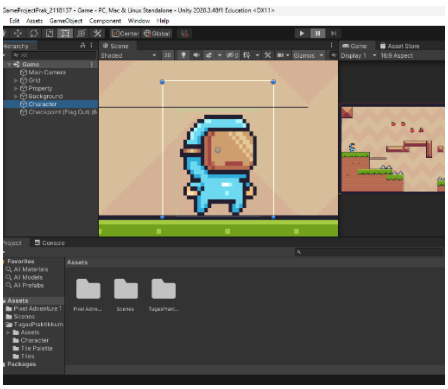
A. Membuat Pergerakan Player

1. Buka project Unity sebelumnya yang telah diimpor.



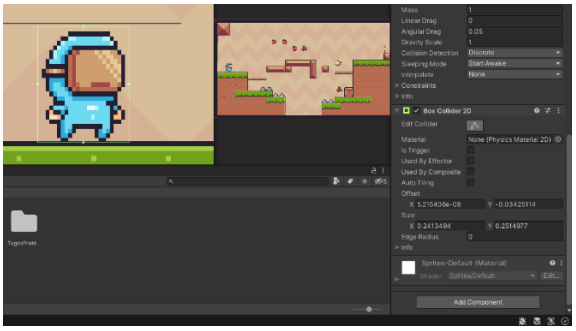
Gambar 8.1 Tampilan *Project Unity*

2. Langkah selanjutnya adalah menambahkan player pada asset yang sudah disediakan, pilih yang *idle*. Masukkan kedalam ke *Hierarchy* dengan cara di *drag and drop.*



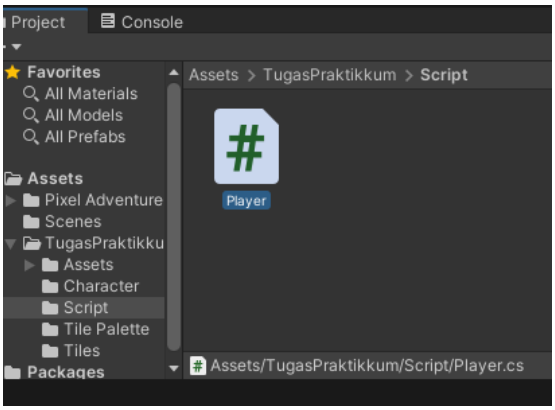
Gambar 8.2 Tampilan *folder* Tugas praktikum

3. Selanjutnya adalah menambahkan komponen *Rigidbody* 2D pada player. Setelah itu juga tambahkan komponen *Capsule Collider*, dan *edit collider* sesuaikan dengan ukuran objek player.



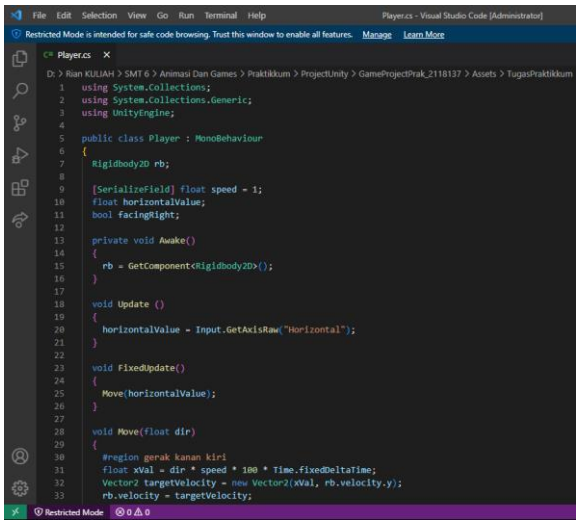
Gambar 8.3 Ca*psule Collider P*layer

4. Langkah berikutnya adalah membuat *script* untuk player, dengan membuat *folder* bernama *script* pada folder tugas praktikkum. Dan didalam folder tersebut buat scriptnya berikan nama Player.



Gambar 8.4 Membuat *Script*

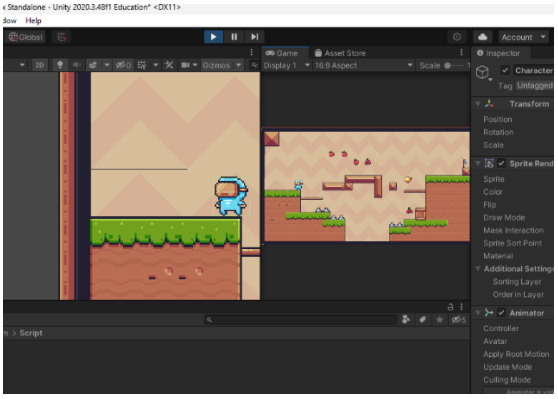
5. Kita buka script dan berikan source code yang sudah disediakan oleh modul. Kemudian drag and drop ke hirarki pada objek player.



```
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;
```

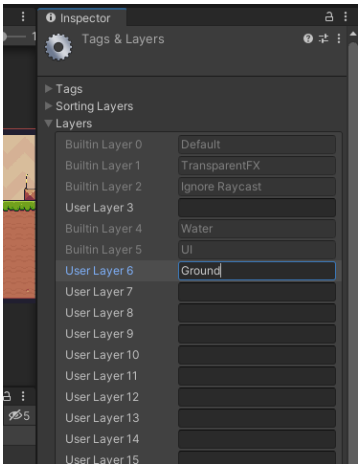
Gambar 8.5 Souce code

6. Kita coba kode nya berhasil atau tidak dengan menekan tombol keyboard a, d, arah kiri dan kanan.



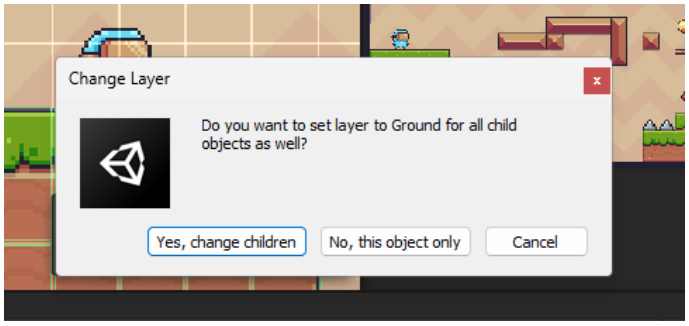
Gambar 8.6 Mencoba So*urce Cod*e

7. Langkah berikutnya adalah membuat player bisa loncat Ketika menekan spasi. Pertama membuat GroundCheck dengan cara pada inspector Grid pilih layer, kemudian klik add layer. Lalu isi “Ground” pada layer 6.



Gambar 8.7 GroundCheck Layer 6

8. Kemudian ubah layer menjadi Ground, jika muncul pop up Change Layer, klik yes aja.



Gambar 8.8 *Change Layer*

9. Selanjutnya adalah membuat objek *GroundCheck* pada hirarki player. Klik Grou*ndChe*ck ubah posisi menggunakan *Move To*ols untuk memindahkan ke bagian bawah player. Dan pada *script layer* tambahkan *source code* ini.

```
public class Player : MonoBehaviour
{
  Rigidbody2D rb;

  [SerializeField] Transform groundcheckCollider;
  [SerializeField] LayerMask groundLayer;

  const float groundCheckRadius = 0.2f; // +
  [SerializeField] float speed = 1;
  float horizontalValue;

  [SerializeField] bool isGrounded; // +
  bool facingRight;
```

Gambar 8.9 Sc*ipt GroundChec*k

10. Tambahkan juga pada script player pada bagian voidFixedUpdate tambahkan source code ini.

```
void FixedUpdate()
{
GroundCheck();
Move(horizontalValue);
}

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

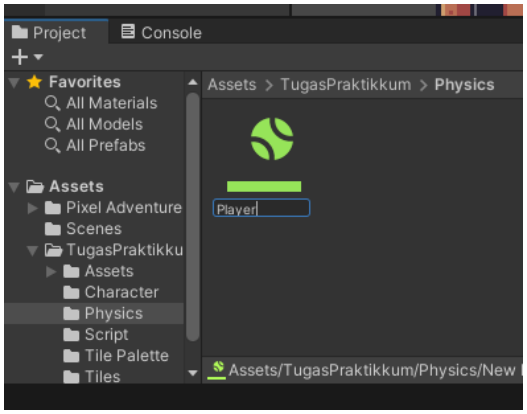
Gambar 8.10 *Source code*

11. Setelah itu untuk membuat player bisa lompat tambahkan source code ini.

```
const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
[SerializeField] float jumpPower = 100;
float horizontalValue;

[SerializeField] bool isGrounded; // +
bool facingRight;
bool jump;
```

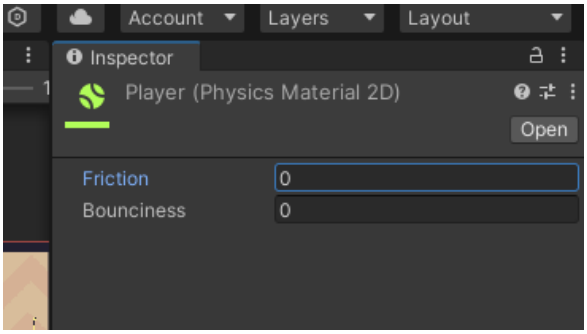
Gambar 8.11 *Source code* untuk lompat

12. Langkah berikutnya adalah membuat folder baru di TugasPraktikkum berikan nama Physics lalu buat PhysicalMaterial 2D dan berikan nama Player.



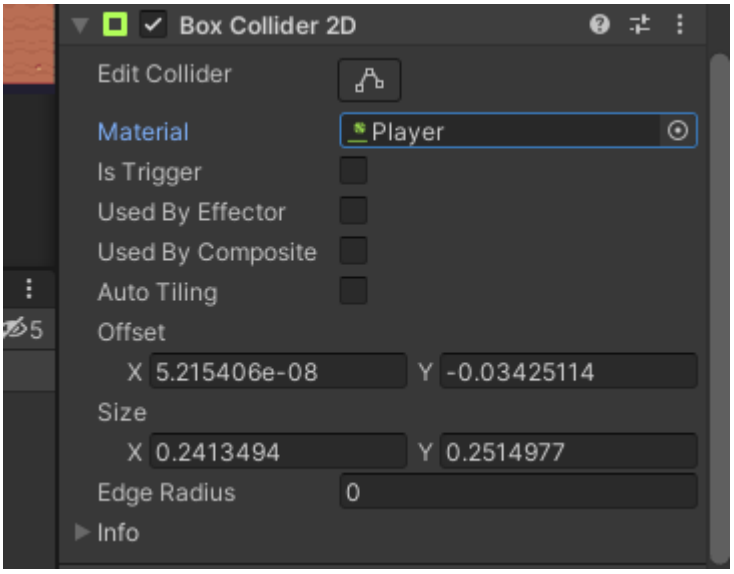
Gambar 8.12 PhysicalMaterial 2D

13. Pada inspector Physics Material 2D ubah *friction* dan *Bounciness* menjadi 0.



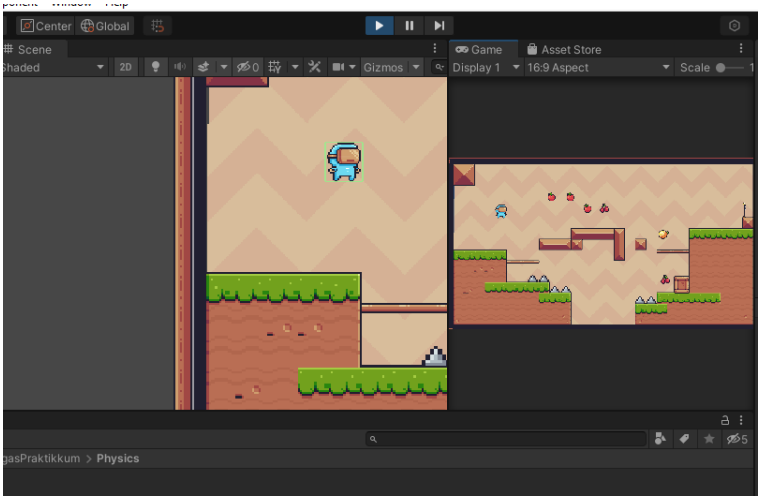
Gambar 8.13 Inspector Physics Material 2D

14. Pada hirarki pilih layer player, kemudian pada inspector bagian Box Collider 2D, kita ubah material menjadi player.



Gambar 8.14 Box Collider 2D Player

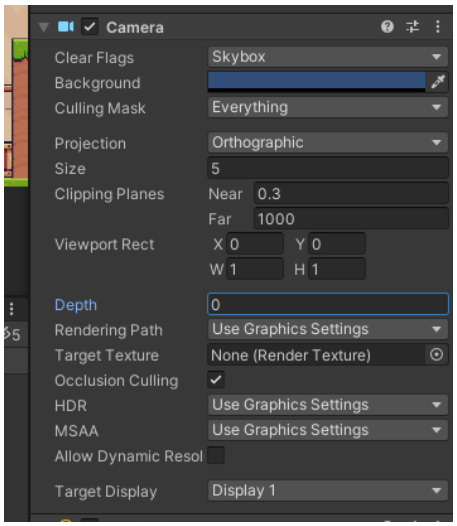
15. Tekan play kemudian tekan spasi maka player bisa melompat.



Gambar 8.14 Spasi untuk melompat

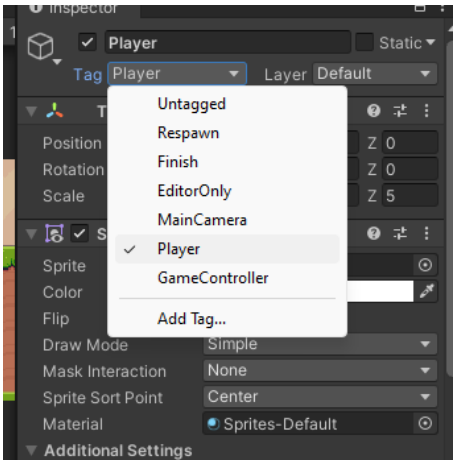
B. Camera Movement

1. Pertama dalam pembuatan kamera movement membuat pada hirarki player objek baru dan beri nama camera, kemudian pada inspectornya sesuaikan setting layer camera seperti pada gambar dibawah ini.



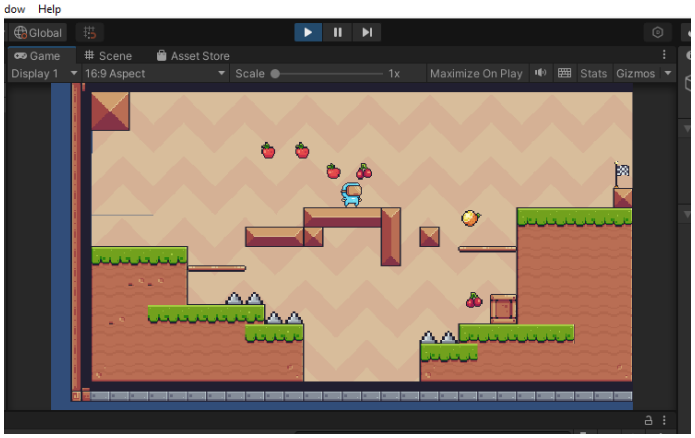
Gambar 8.15 Layer Camera

2. Langkah berikutnya adalah membuat script untuk cameraFollow dan berikan source code yang sudah diberikan dari modul. Kemudian drag and drop ke hirarki camera. Lalu pada inspector tag nya ubah ke player.



Gambar 8.16 Camera Follow

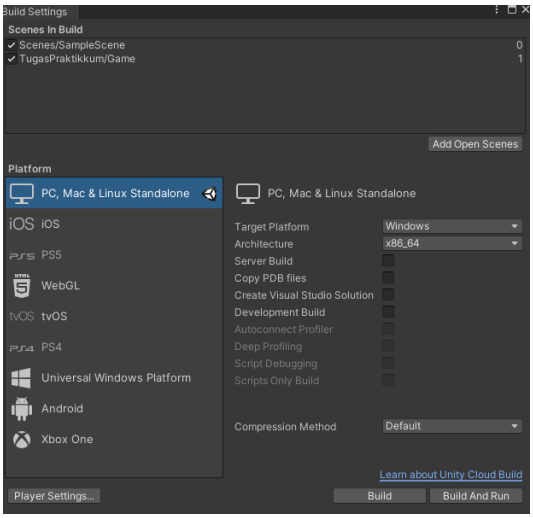
3. Dan yang terakhir adalah play projectnya.



Gambar 8.17 Play Project

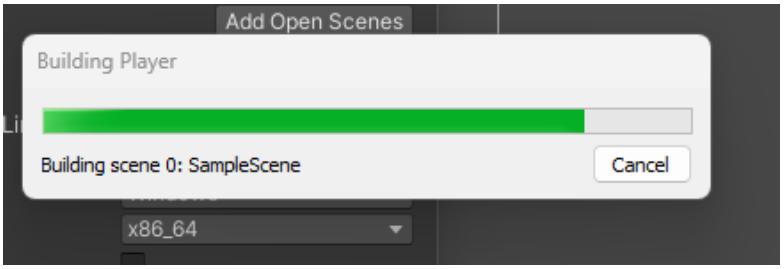
C. Render

1. Untuk proses merender pada file kemudian pilih Build Settings atau menekan Ctrl + Shift + B, setelah masuk Setting Build pilih PC, Mac dan Linux. Jangan lupa pada scenes in build mencentang project kita, kalo belum ada tekan Add Open Scenes.



Gambar 8.18 Render Settings

2. Setelah itu kita klik Build dan kita pilih project jadinya akan disimpan dimana. Dan tunggu hasilnya.



Gambar 8.19 Building Render

D. Link Pengumpulan Github

Link : https://github.com/riansetyabudi/2118137_PRAK_ANIGAME

KUIS

Menjelaskan Source Code dibawah ini :

Soal Kuis Bab 8

```
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);
    }
}
```

Analisa :

Pada source code di atas, pertama-tama kita mengimpor namespace System.Collections, System.Collections.Generic, dan UnityEngine. Kemudian, kelas camerafollow dideklarasikan sebagai publik dan mewarisi MonoBehaviour. Variabel player dideklarasikan sebagai private. Selanjutnya, method update dipanggil sekali per frame untuk memperbarui posisi kamera. Nilai x diatur berdasarkan posisi pemain (player.position.x), sedangkan nilai y dan z tetap menggunakan posisi kamera saat ini (transform.position.y dan transform.position.z).

Kuis Camera Follow

```
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);
    }
}
```

Analisa :

Pada source code di atas, pertama-tama diimpor namespace yang diperlukan dan didefinisikan sebuah kelas CameraFollow yang mewarisi MonoBehaviour. Selanjutnya, variabel player dideklarasikan sebagai tipe transform dan diatur agar bersifat private. Method update kemudian dipanggil sekali setiap frame.