# PRAKTIKUM ANIMASI DAN GAME PERTEMUAN :8 Camera & Character Movement

|  |  |  |
| --- | --- | --- |
| **NIM** | **:** |  |
| **NAMA** | **:** |  |
| **MATERI** | **:** | Camera & Character Movement |
| **KELAS** | **:** | E |
| **TUGAS** | **:** | Membuat pergerakan karakter player |

## Tujuan

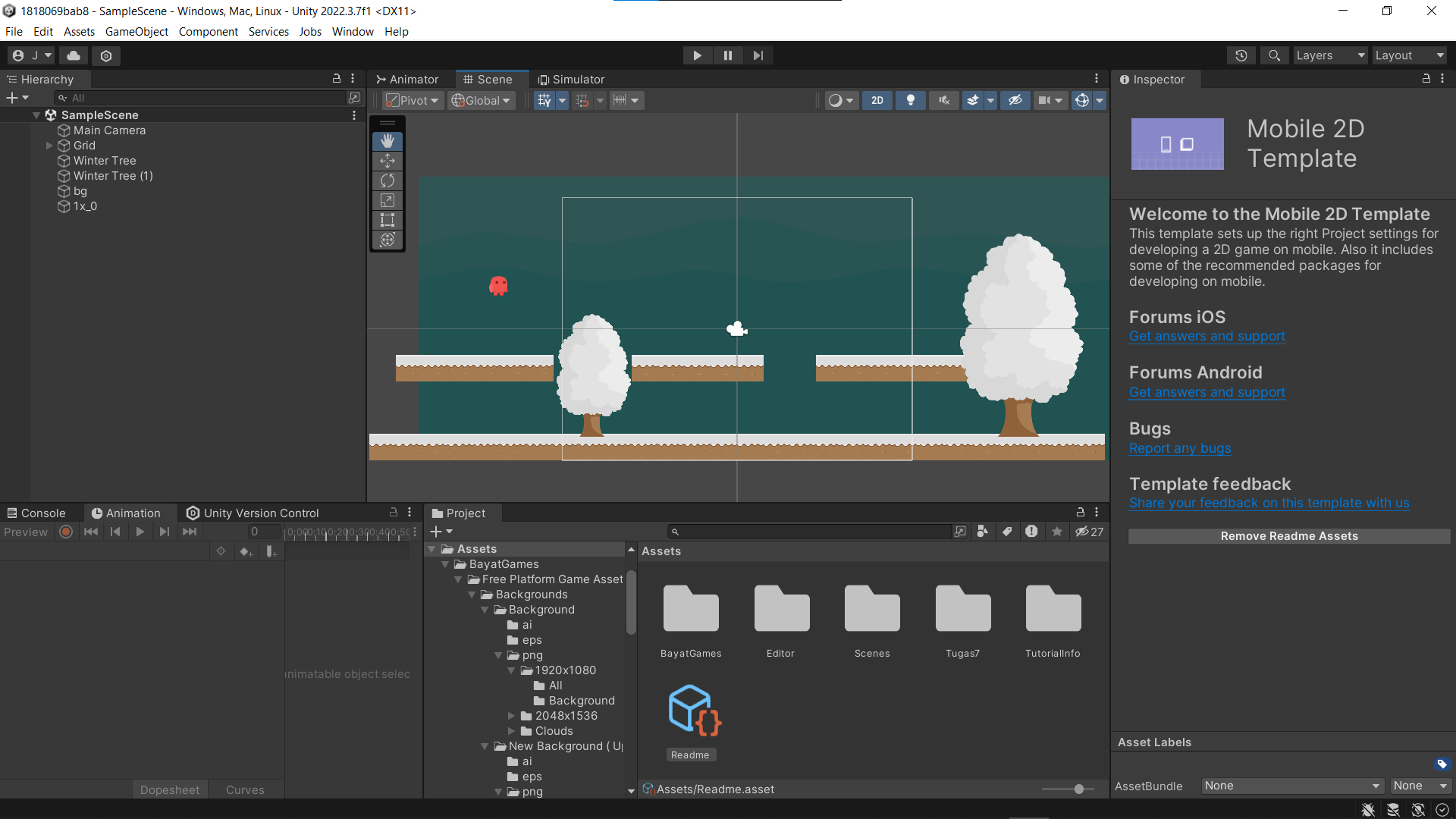
* + 1. Praktikan dapat mengetahui *Game* 2D dan *Game* 3D.
    2. Praktikan dapat mengetahui jenis *Game.*
    3. Praktikan dapat menerapkan *Game* pada Unity.

## Alat dan Bahan

* + 1. Laptop/pc.
    2. Modul Praktikum Animasi dan *Game* 2023.
    3. Unity 2017.

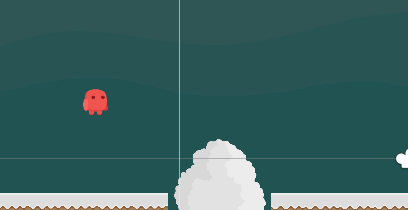
## Langkah-Langkah Membuat Tugas

1. **Membuat Pergerakan Player**
2. Buat file *projek Unity* Tugas 7



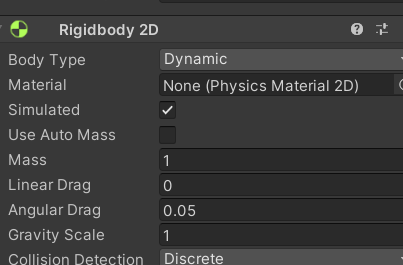
### Buka projek tugas 7

1. Tambahkan player, pilih yang idle, Import kedalam Hirarki.



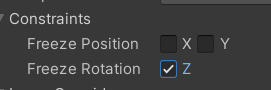
### Tampilan menambahkan karakter

1. Klik karakter tersebut, pergi ke *Inspector* dan klik *Add Component*, kemudian cari komponen bernama *RigidBody2D*, komponen tersebut berguna untuk memberikan efek gravitasi pada objek

****

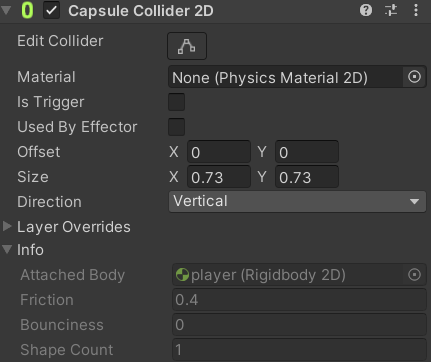
### Menambahkan komponen *RigidBody* *2d*

1. sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



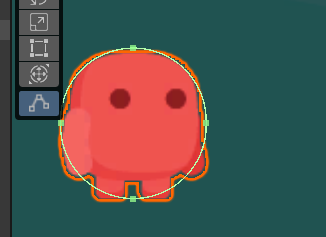
### Centang Rotation Z

1. Lalu tambahkan komponen Capsule Colider di karakter, lalu klik icon sebelah kanan edit colider

****

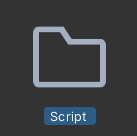
### Menambahkan komponen *capsule collider 2d*

1. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya



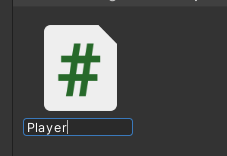
### Mengatur gari oval

1. Buka Folder Tugas7, lalu bikin folder baru bernama Script

****

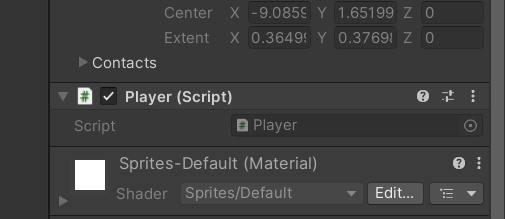
### Membuat folder script

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



### Membuat script

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

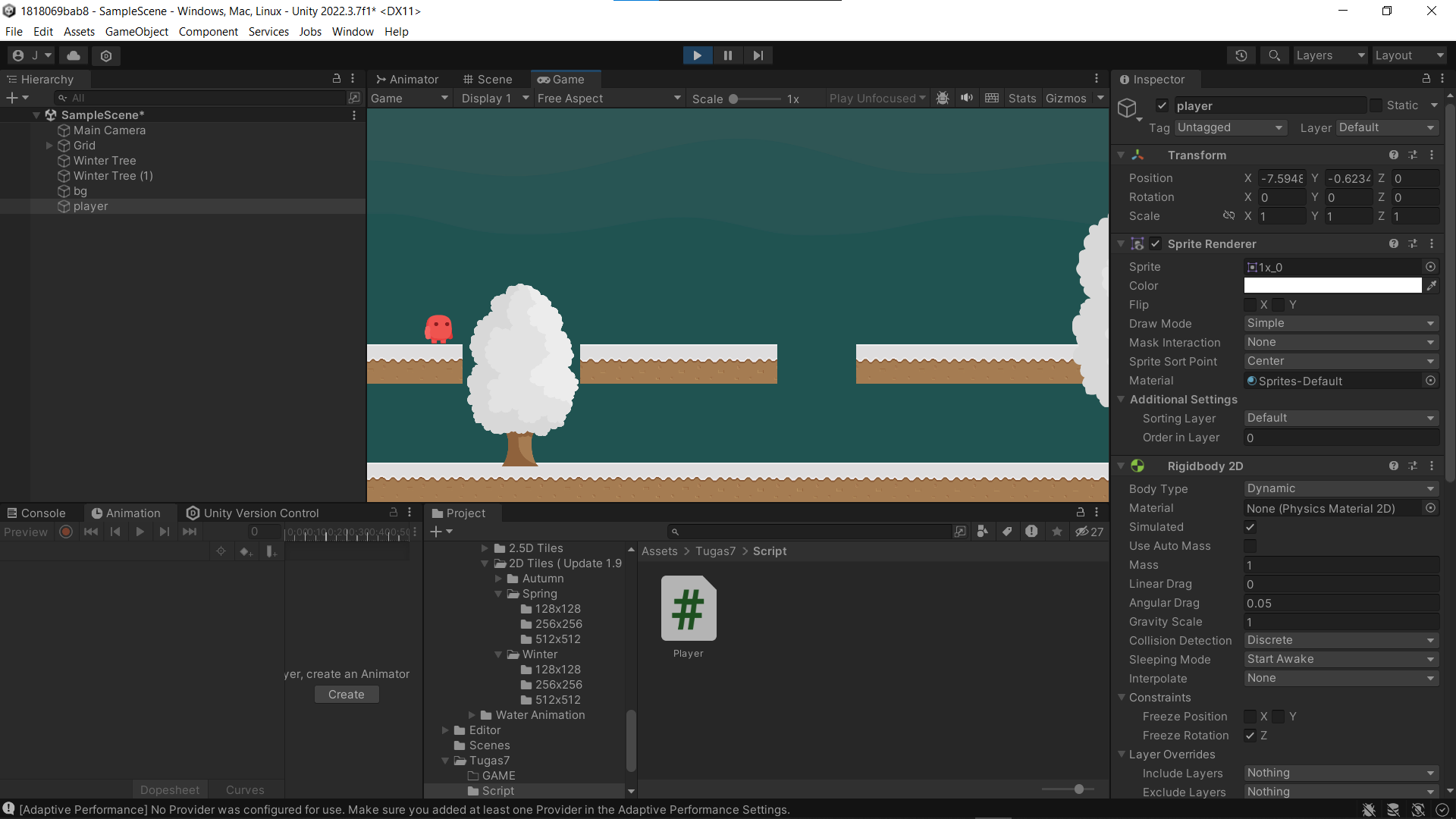


### Memasukan Script Player

### Lalu pada script Player masukkan Source Berikut

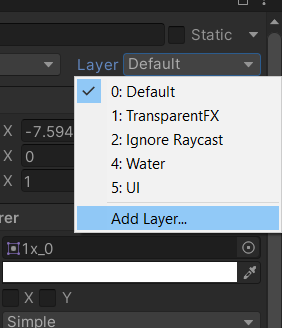
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  Rigidbody2D rb;  [SerializeField] float speed = 1;  float horizontalValue;  bool facingRight;  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  }  void Update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



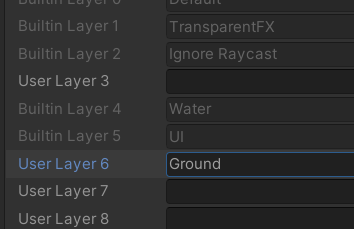
### Menjalankan program

1. Untuk membuat player loncat menggunakan spasi, kita perlu membuat GorundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer



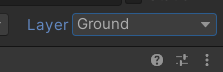
### Add Layer

1. Isikan *User Layer* 6 dengan nama *Ground*



### Mengisi *layer 6* menjadi *Ground*

1. Ganti *Layer Default* menjadi *Ground*



### Menggati layer menjadi *Ground*

1. Klik kanan *Warrior* kemudian pilih *Create Empty* dan ubah namanya menjadi *GroundCheck*



### Membuat *GroundCheck*

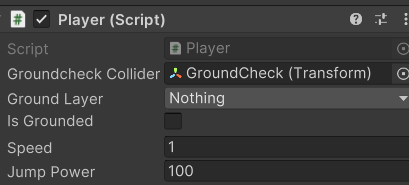
1. Kembali ke script Player tambahkan source code seperti ini

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

1. Buat void ground check dibawah void fixedUpdate & tambahkan GorundCheck(); pada void fixedUpdate

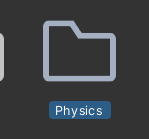
|  |
| --- |
| void FixedUpdate()  {  GroundCheck();  Move(horizontalValue);  }  void GroundCheck()  {  isGrounded = false;  Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);  if (colliders.Length > 0)  isGrounded = true;  } |

1. drag *GroundCheck* arahkan kedalam *Groundcheck* Collider yang ada pada *Inspector*



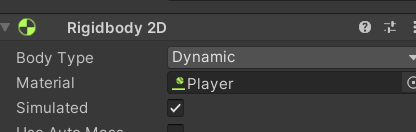
### Penyesuaian *GroundCheck*

1. Membuat folder Physics



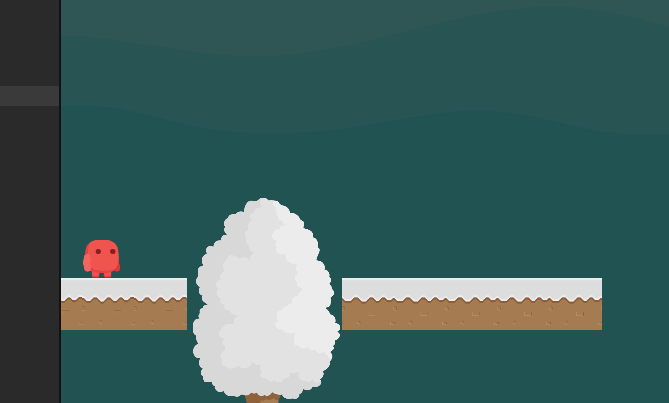
### Membuat folder Physics

1. Klik kanan folder “*Physics*” pilih *Create>Physics Material 2D* dan beri nama “Player” lalu ubah nilai Friction dan Bouncies menjadi 0



### Tampilan *Physics* karakter

1. Tekan play, maka player bisa melompat dengan menekan spasi



### Tampilan *Physics* karakter

1. **Camera Movement**
2. Tambahkan script dalam folder Script dan beri nama *“CameraFollow.cs”*



### Membuat dan Menambahkan *Source Code*

1. Tambahkan *Source Code* berikut pada Script *CameraFollow*

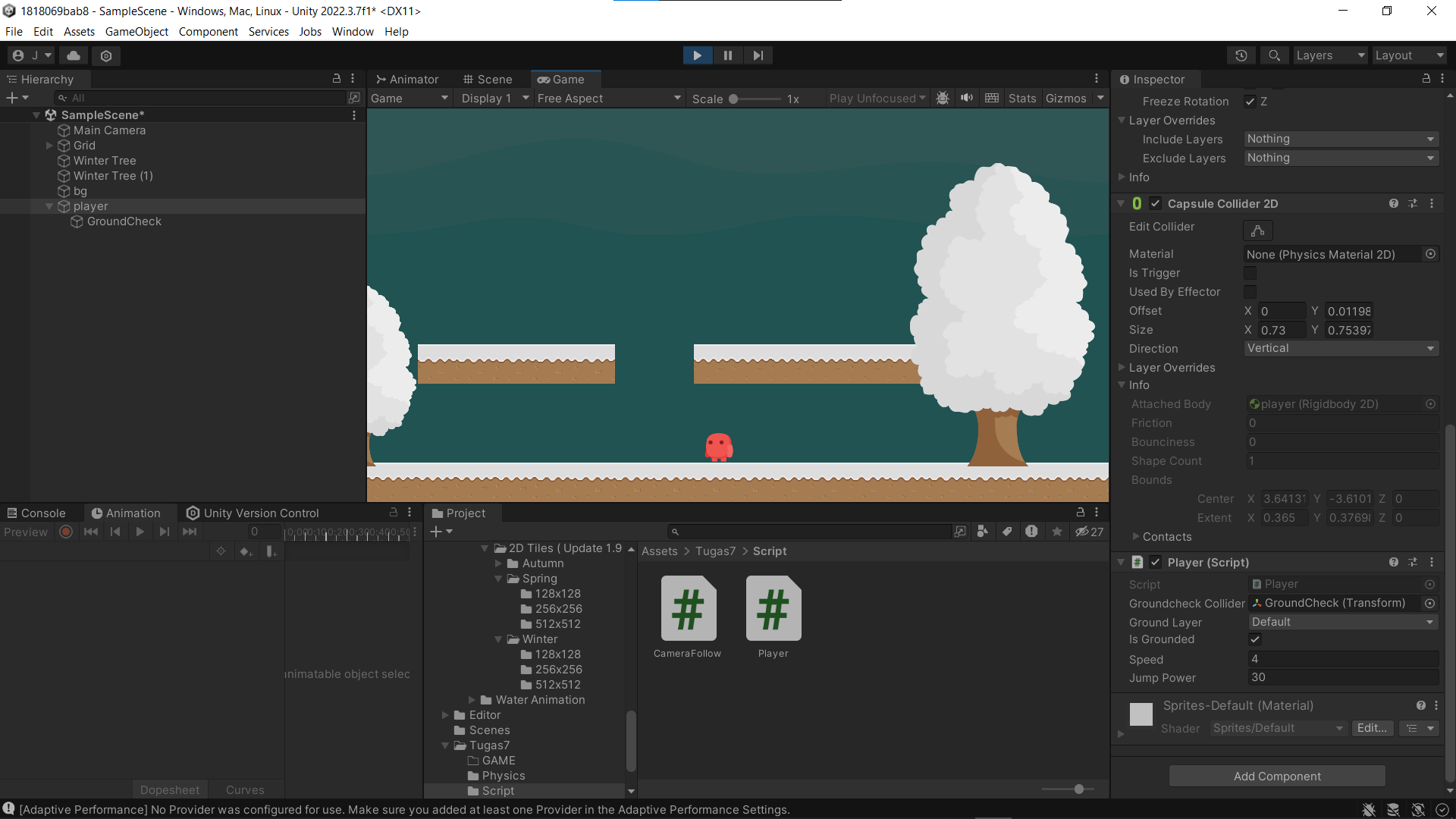
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraFollow : MonoBehaviour  {  public float xMargin = 0.5f;  public float yMargin = 0.5f;  public float xSmooth = 4f;  public float ySmooth = 4f;  public Vector2 maxXAndY;  public Vector2 minXAndY;  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. Pergi ke *Inspector Main* Camera kemudian lakukan setting *Camera Follow* seperti gambar berikut



### Mensetting *Camera Follow*

1. Jika di *Play*, maka camera akan mengikuti pergerakan setiap karakter.



### Tampilan game ketika di Run

## Kesimpulan

* + 1. Bahasa C# adalah sebuah Bahasa pemrograman modern yang bersifat generalpurpose, berorientasi objek, yang dapat digunakan untuk membuat program diatas arsitektur Microsoft .Net. Framework. Dalam Unity Bahasa C# adalah Bahasa pemrograman default untuk penegmabangan game menggunakan Unity
    2. Komponen RigidBody2D menempatkan objek dibawah kendali sifat fisika. Objek hanya dapat bergerak di bidang XY dan hanya dapat memutar pada sumbu yang tegak lurus terhadap bidang itu
    3. Camera adalah perangkat yang menangkap dan menampikan dunia kepada pemain. Dengan menyesuaikan dan memanipulasi kamera. Setiap game dapat memiliki jumlah kamera yang tida terbatas dalam sebuah scene. Meraka dapat diatur untuk menyajikan dalam urutan apapun

|  |  |
| --- | --- |
| **Nilai** | **Asisten Laboratorium** |
|  | M. AKBAR ILHAM  2118091 |