# 8 Character and Camera Movement

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118145 |
| **Nama** | : | Esa Arya Mahardika |
| **Kelas** | : | D |
| **Asisten Lab** | : | 2118004 - Bagas Anardi Surya W. |
| **Baju Adat** | : | Banjar Babaju Kubaya Panjang |
| **Referensi** | : | https://berita.99.co/wp-content/uploads/2023/08/banjar-babaju-kubaya-panjang.jpg |

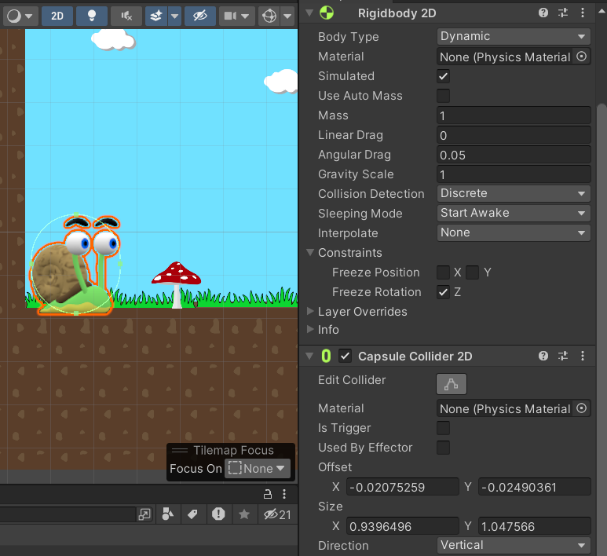
## Tugas 8: Menggerakkan Karakter dan Kamera

1. **Menggerakkan Karakter**
2. Buka proyek sebelumnya



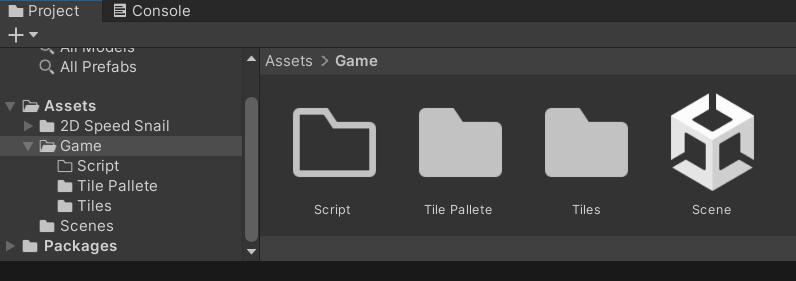
### 8.1 Proyek Sebelumnya

1. Pastikan komponen Siput hanya *Rigidbody 2D* dan *Capsule Collider 2D*



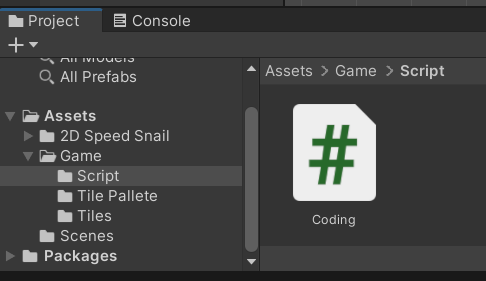
### 8.2 Komponen Karakter

1. Buat folder ‘Script’ di dalam *folder* Game



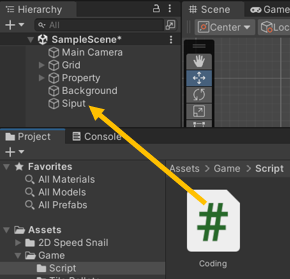
### 8.3 Membuat Folder ‘Script’

1. Buat berkas C# di dalam folder Script



### 8.4 Membuat Berkas C# Coding

1. Sisipkan berkas Coding ke Siput dengan *drag drop*

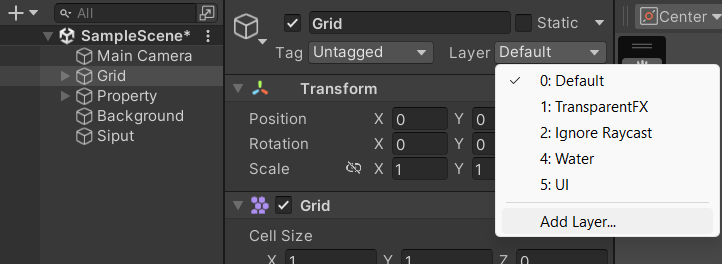


### 8.5 Menyisipkan Berkas Coding ke Siput

1. Buka berkas Coding dan sisipkan skrip di bawah ini

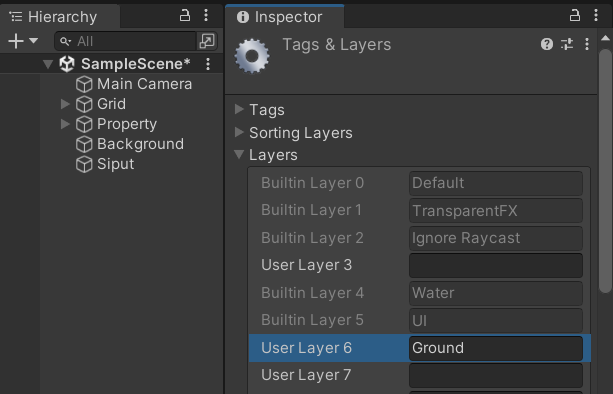
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Coding : MonoBehaviour{      Rigidbody2D rb;      [SerializeField] Transform groundcheckCollider;      [SerializeField] LayerMask groundLayer; // +      [SerializeField] float speed = 1;      [SerializeField] bool isGrounded; // +      [SerializeField] float jumpPower = 300; // ++      const float groundCheckRadius = 0.2f; // +      float horizontalValue;      bool facingRight;      bool jump; // ++      private void Awake(){          rb = GetComponent<Rigidbody2D>();}      void Update (){  horizontalValue = Input.GetAxisRaw("Horizontal");          if (Input.GetButtonDown("Jump"))              jump = true;          else if (Input.GetButtonUp("Jump"))              jump = false;}      void FixedUpdate(){          GroundCheck();          Move(horizontalValue, jump);}      void GroundCheck(){          isGrounded = false;          Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);          if (colliders.Length > 0)          isGrounded = true;}      void Move(float dir, bool jumpflag){          #region gerak kanan kiri  float xVal = dir\*speed\*100\*Time.fixedDeltaTime;          Vector2 targetVelocity = new Vector2(xVal, rb.velocity.y);          rb.velocity = targetVelocity;          if(isGrounded && jumpflag){              isGrounded = false;              jumpflag = false;            rb.AddForce(new Vector2(0f, jumpPower));}          if (facingRight && dir < 0){  transform.localScale = new Vector3(-2, 2, 2);              facingRight = false;          } else if (!facingRight && dir > 0){  transform.localScale = new Vector3(2, 2, 2);              facingRight = true;}          #endregion}} |

1. Buka kembali Grid dan tambahkan *layer* baru



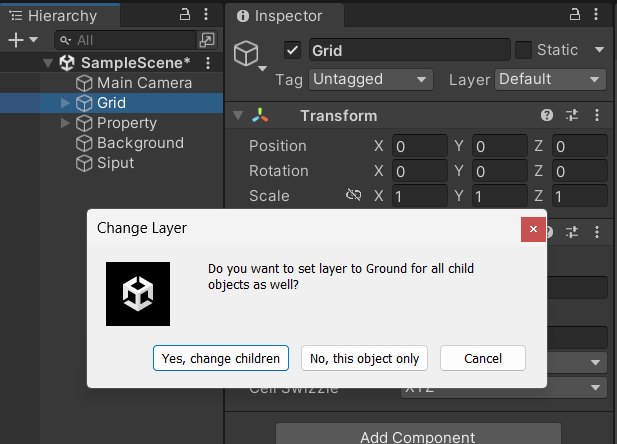
### 8.6 Membuat *Layer* Baru

1. Beri nama ‘Ground’ pada *layer* ke-6



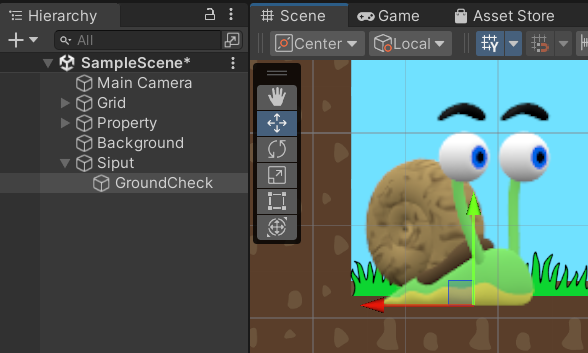
### 8.7 Memberi Nama Baru *Layer*

1. Kembali ke Grid dan ubah *layer* menjadi ‘Ground’. Ketika muncul notifikasi seperti ini, tekan ‘Yes’



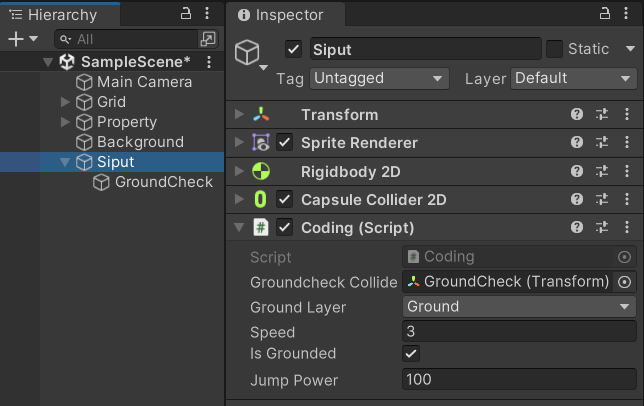
### 8.8 Mengatur *Layer* Grid

1. Buat hierarki baru di dalam Siput, beri nama ‘GroundCheck’



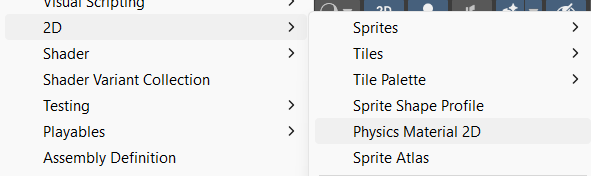
### 8.9 Menambahkan Hierarki ‘GroundCheck’

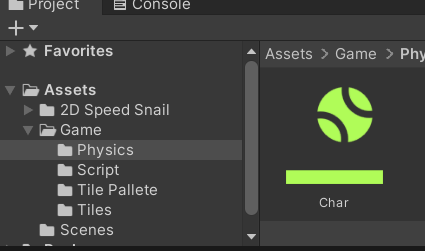
1. Atur komponen Siput bagian Coding seperti di bawah ini



### 8.10 Mengatur Komponen Siput bagian Coding

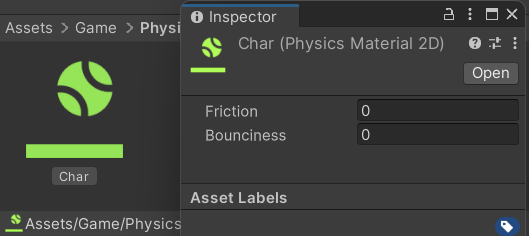
1. Buat folder ‘Physics’ dan buat *Physics Material 2D* baru





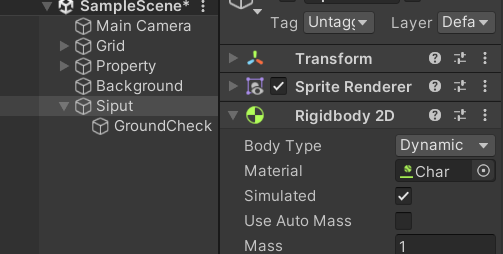
### 8.11 Membuat *Physics Material 2D* Baru

1. Konfigurasi Char di inspector dengan nilai 0 semua



### 8.12 Konfigurasi Char

1. Ganti material pada komponen *Rigidbody 2D* Siput dengan Char



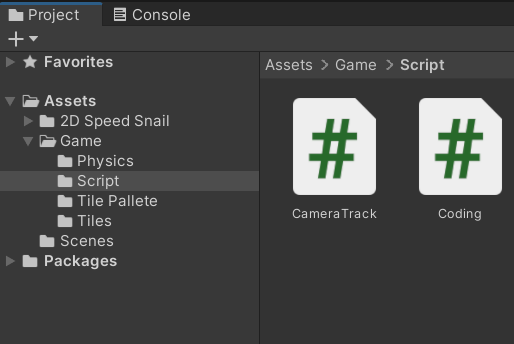
### 8.13 Mengganti Material *Rigidbody 2D*

1. Jalankan proyek dan coba untuk lompat dengan spasi



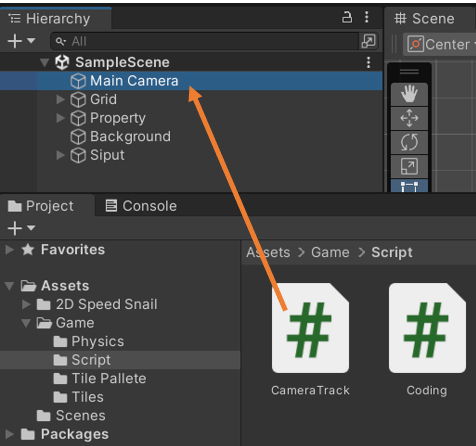
### 8.14 Uji Coba Lompat

1. **Menggerakkan Kamera**
2. Tambahkan *script* baru dengan nama ‘CameraTrack’



### 8.15 Buat berkas C# baru untuk Kamera

1. Sisipkan CameraTrack ke Main Camera

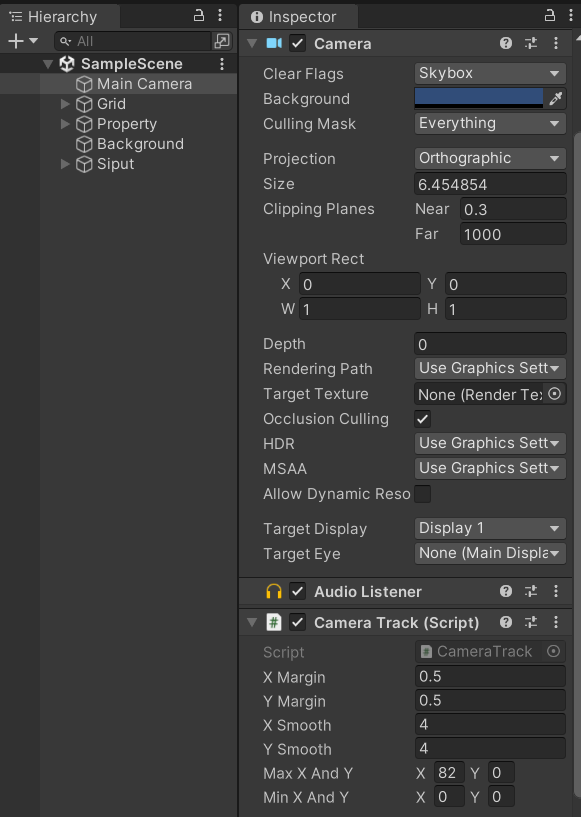


### 8.16 Menyisipkan CameraTrack ke Kamera

1. Masuk ke berkas CameraTrack dan masukkan skrip di bawah ini

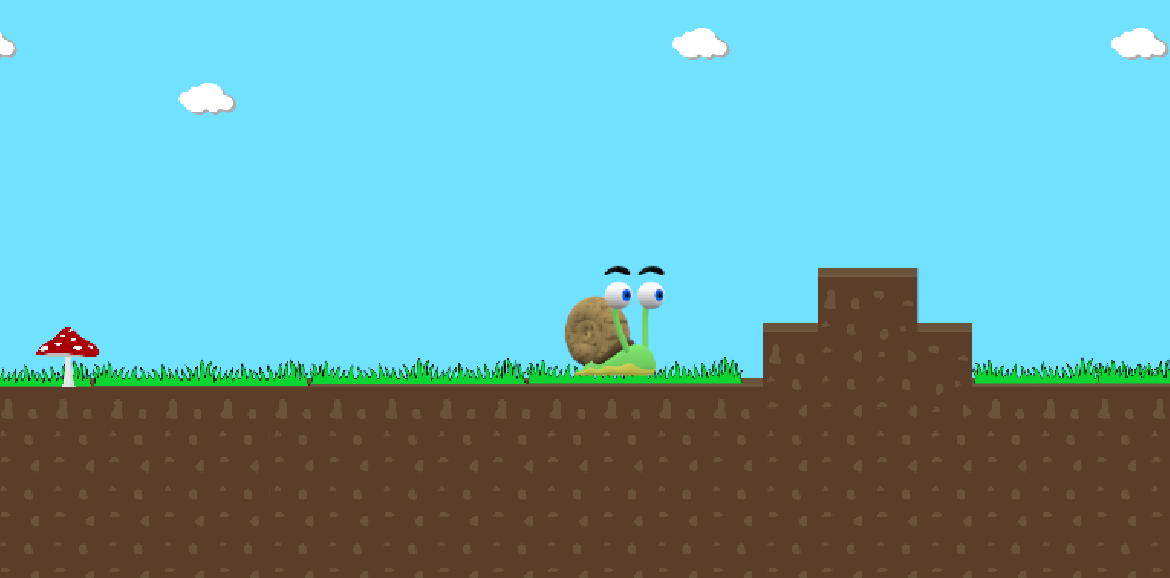
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraTrack : MonoBehaviour{      public float xMargin = 0.5f;      public float yMargin = 0.5f;      public float xSmooth = 4f;      public float ySmooth = 4f;      public Vector2 maxXAndY;      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. Atur komponen di Main Camera seperti ini



### 8.17 Menyesuaikan Komponen di Main Camera

1. Jalankan proyek dan coba untuk bergerak lebih jauh hingga kamera bergerak



### 8.18 Uji Coba Pergerakan Kamera

1. **Link Github Pengumpulan**

<https://github.com/EsaAryaMahardika/PraktikumAnimasiGame/tree/main/Pertemuan%208>

1. **Kuis**

Penjelasan source code langkah ke-6 Menggerakkan Karakter

Terdapat 5 function yang berjalan pada karakter, yaitu Awake(), Update(), FixedUpdate(), GroundCheck(), dan Move(). Func Awake() berfungsi membaca komponen Rigidbody 2D di dalam karakter Siput. Func Update() berfungsi mengatur kondisi karakter ketika lompat atau tidak. Func FixedUpdate() untuk memberikan perintah ke karakter untuk berjalan atau melompat. Func GroundCheck() berfungsi mengecek karakter bertapak atau tidak, jika tidak bertapak maka karakter akan turun. Func Move() berfungsi menggerakkan karakter khususnya maju dan mundur.

Penjelasan source code langkah ke-6 Menggerakkan Karakter

Terdapat 5 function yang bekerja pada kamera, yaitu Awake(), CheckXMargin(), CheckYMargin(), FixedUpdate(), dan TrackPlayer(). Func Awake() berfungsi mendefinisikan kamera ke obyek yang memiliki tag ‘Player’. Func CheckXMargin() berfungsi mengecek margin sumbu X kamera. Func CheckYMargin() berfungsi mengecek margin sumbu Y kamera. Func TrackPlayer() berfungsi mengecek posisi karakter dan mengikuti karakter. Func FixedUpdate() berfungsi menjalankan func TrackPlayer().