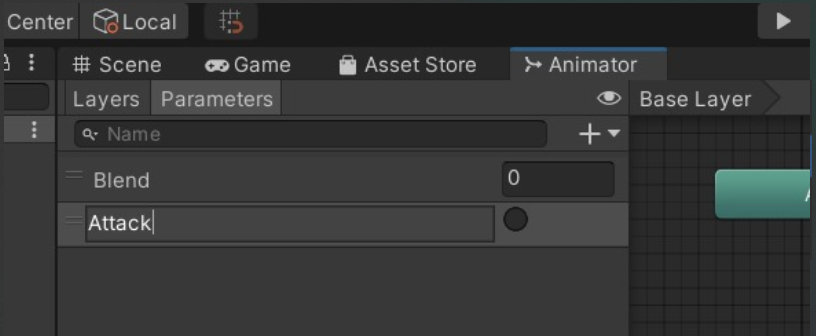
# 10 Respawn and AI Enemy Attack

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118003 |
| **Nama** | : | Kevin Majesta Ivano |
| **Kelas** | : | A |
| **Asisten Lab** | : | M. Rafi Faddilani (2118144) |

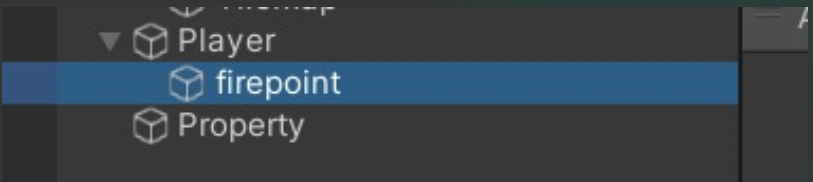
## 1.1 Tugas 1 : Membuat Mekanisme Attack, Enemy Attack AI, dan Respawn

1. **Mekanisme Attack**
2. Kemudian pada menu Tab Animator Tambahkan Parameter Trigger, Rename Menjadi Attack



### 1.1 *Add* Parameter

1. Setelah menambahkan parameter *Attack*, Langkah selanjutnya adalah membuat Layer Game object baru didalam player, Klik kanan pilih *Create Empty* lalu *Rename* menjadi *Firepoint*



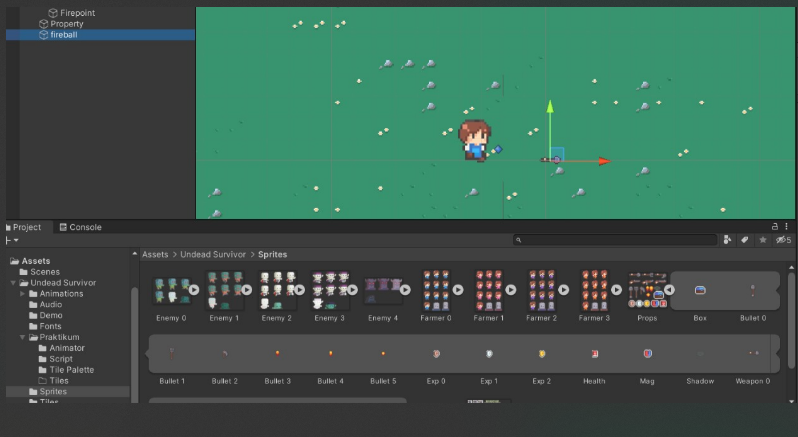
### 1.2 Tambah folder baru

1. Pada menu Hierarchy klik Firepoint untuk setting pada Inspector, Ubah Icon Menjadi titik, atur letak titik didepan player



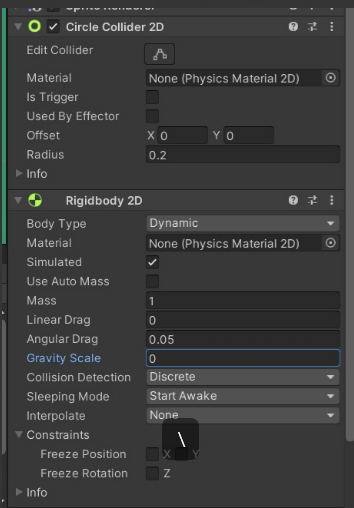
### 1.3 Atur letak bullet

1. Pada menu Hierarchy Tambahkan item-feedback-1, di folder Sprites > Fx > item-feedback-1 , *rename* menjadi *fireball*



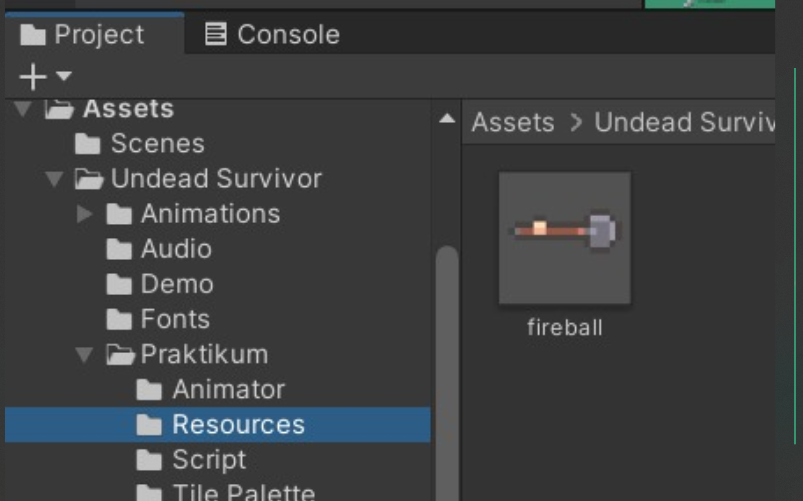
### 1.4 Tambahkan item baru

1. Klik item-feedback-1 untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



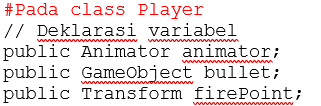
### 1.5 Tambah komponen

1. Buat Folder baru *Resources* di menu Project, kemudian drag and drop fireball kedalam folder Resources, dan hapus *fireball* pada *Hierarchy*



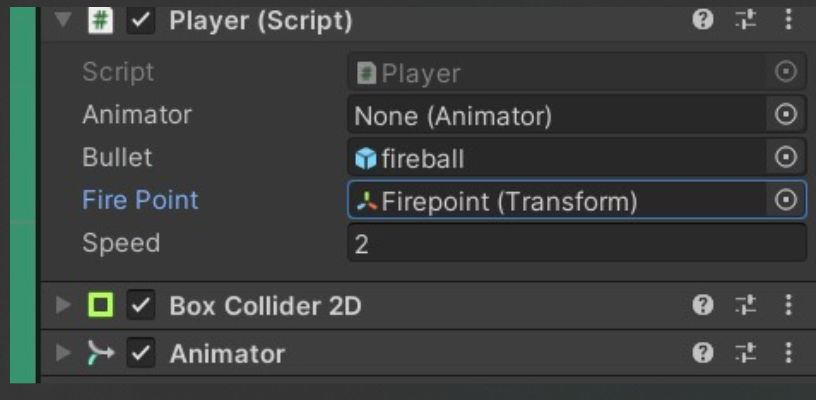
### 1.6 Panel animation

1. Pada Script Player Tambahkan Script dibawah ini



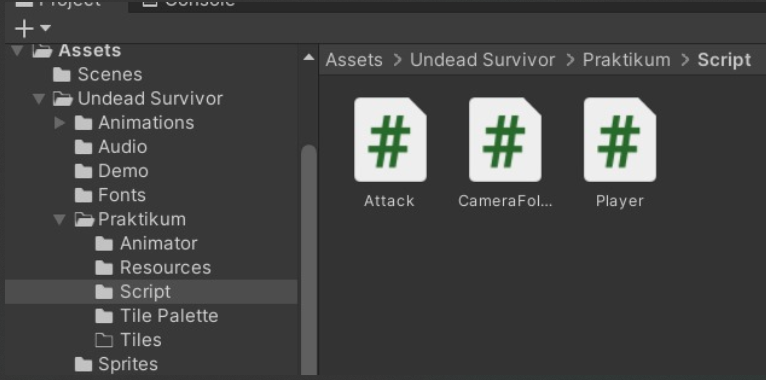
|  |
| --- |
| #Tambahkan dibawah fungsi fixedUpdate  IEnumerator Attack()  {    animator.SetTrigger("Attack");  yield return new WaitForSeconds(0.25f);    float direction = 1f;    GameObject fireball = Instantiate(bullet, firePoint.position, Quaternion.identity);  fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 10f, 0);    Destroy(fireball, 2f);  }  #Tambakan pada Function Void Update  if (Input.GetKeyDown(KeyCode.C))  {  StartCoroutine(Attack());  } |

1. Pada Inspector Player, Ubah seperti dibawah ini, Dimana Bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak pertama



### 1.7 Isi nilai

1. Buat Script Attack pada folder Script

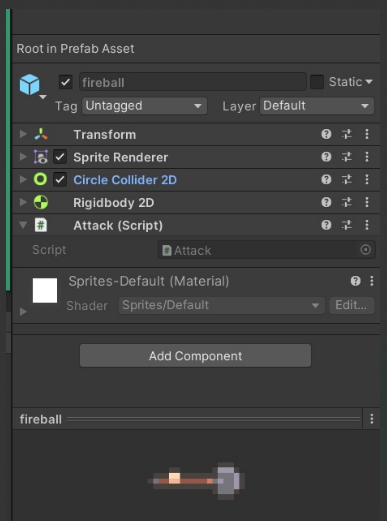


### 1.8 Buat *script attack*

1. Tambahkan Script Attack dibawah ini

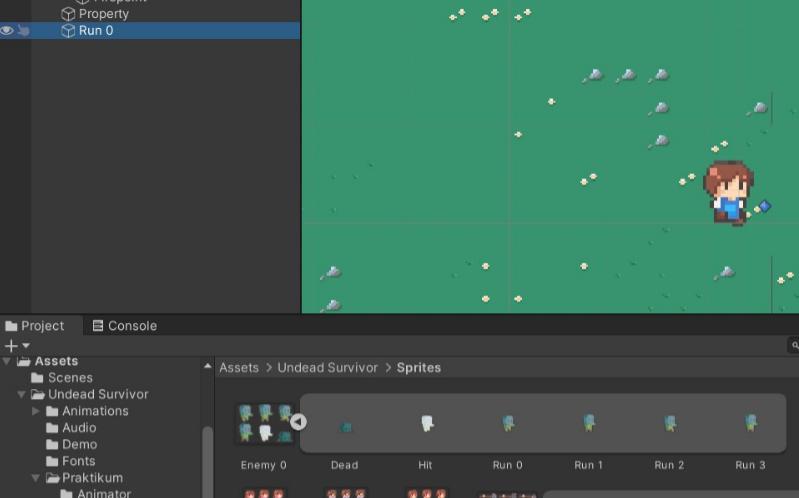
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Attack : MonoBehaviour  {  private void OnTriggerEnter2D(Collider2D collision)  {  if (collision.gameObject.CompareTag("Enemy"))  {  Destroy(gameObject);  Destroy(collision.gameObject);  }  }  } |

1. Didalam folder *resource* Tambahkan Script Attack di *Prefab fireball*, dengan cara Klik *fireball* kemudian pada menu Inspector arahkan *Script Attack* kedalam *Inspector*



### 1.9 Masukkan *script*

1. Tambahkan Enemy pada hierarchy di folder Sprites



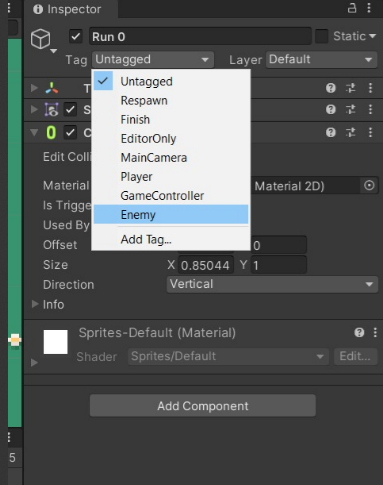
### 1.10 Tambhkan enemy

1. Kemudian klik pada enemy, lalu pada menu tab inspector tambahkan collider 2D untuk mendeteksinya



### 1.11 Atur timeline

1. Tambahkan Tag *Enemy* dengan cara Pilih *Add Tag*, kemudian *add tag* *to the list*, Tuliskan *Enemy*



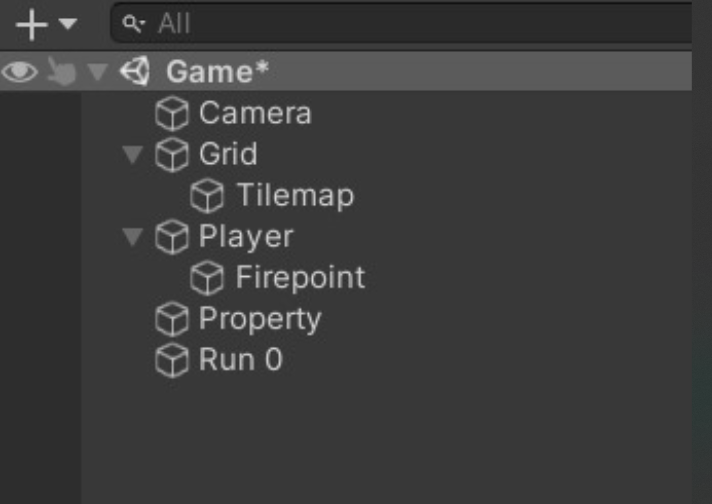
### 1.12 Tambah tag

1. Tembak Enemey dengan menekan Tombol C untuk menghancurkan musuh



### 1.13 Hasil

1. **Enemy AI**
2. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama ‘run 0’. Tambahkan ‘run-0' pada Hierarchy



### 1.14 *Add* Hierarchy

1. Buat Script Enemy\_AI pada folder Praktikum - Script

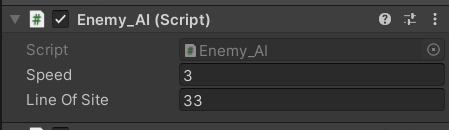


### 1.15Tambah script baru

1. Tambahkan Script dibawah ini

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_AI : MonoBehaviour  {  public float speed; // Kecepatan gerakan musuh  public float lineOfSite; // Jarak penglihatan musuh  private Transform player; // Transform dari pemain  private Vector2 initialPosition; // Posisi awal musuh  // Use this for initialization  void Start()  {  // Mencari pemain berdasarkan tag  player = GameObject.FindGameObjectWithTag("Player").transform;  // Menyimpan posisi awal musuh  initialPosition = GetComponent<Transform>().position;  }  // Update is called once per frame  void Update()  {  // Menghitung jarak antara musuh dan pemain  float distanceToPlayer = Vector2.Distance(player.position, transform.position);  // Jika pemain berada dalam jarak penglihatan musuh  if (distanceToPlayer < lineOfSite)  {  // Musuh bergerak menuju pemain  transform.position = Vector2.MoveTowards(this.transform.position, player.position, speed \* Time.deltaTime);  }  else  {  // Musuh kembali ke posisi awal  transform.position = Vector2.MoveTowards(transform.position, initialPosition, speed \* Time.deltaTime);  }  }  // Untuk menggambar jarak penglihatan musuh di editor  private void OnDrawGizmosSelected()  {  Gizmos.color = Color.red;  Gizmos.DrawWireSphere(transform.position, lineOfSite);  }  } |

1. Pada Inspector Enemy\_Ai, Atur Speed juga Line of Site untuk menentukan jarak dan speed pada enemy



### 1.16 Atur inspector

1. Running Game, maka enemy akan mengikuti Gerakan Player



### 1.17 Tambahkan item baru

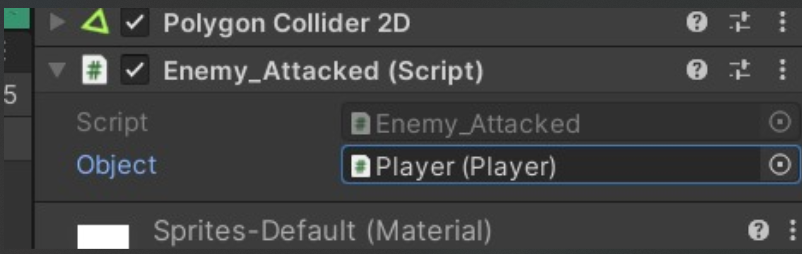
1. **Respawn**
2. Buka file script (Player.cs) tambahkan seperti dibawah ini

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {      public Animator animator;      public GameObject bullet;      public Transform firePoint;      public int nyawa;      [SerializeField] Vector3 respawn\_loc;      public bool play\_again;      Rigidbody2D rb;      [SerializeField] float speed = 1;      float horizontalValue;      float verticalValue;      bool facingRight = true;      bool facingUp = false;      private void Awake()      {          rb = GetComponent<Rigidbody2D>();          animator = GetComponent<Animator>();          respawn\_loc = transform.position;      }      void playagain()      {          if (play\_again == true)          {              nyawa = 3;              transform.position = respawn\_loc;              play\_again = false;          }      }      void Update()      {          horizontalValue = Input.GetAxisRaw("Horizontal");          verticalValue = Input.GetAxisRaw("Vertical");          if (Input.GetKeyDown(KeyCode.C))          {              StartCoroutine(Attack());          }          // playagain          if (nyawa < 0)          {              playagain();          }          if (transform.position.y < -10)          {              play\_again = true;              playagain();          }      }      void FixedUpdate()      {          Move(horizontalValue, verticalValue);          animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));      }      IEnumerator Attack()      {          animator.SetTrigger("Attack");          yield return new WaitForSeconds(0.25f);          Vector2 direction = Vector2.right; // Arah default ke kanan          float rotationAngle = 0f;          if (horizontalValue > 0)          {              direction = Vector2.right;              rotationAngle = 0f;          }          else if (horizontalValue < 0)          {              direction = Vector2.left;              rotationAngle = 180f;          }          else if (verticalValue > 0)          {              direction = Vector2.up;              rotationAngle = 90f;          }          else if (verticalValue < 0)          {              direction = Vector2.down;              rotationAngle = -90f;          }          GameObject fireball = Instantiate(bullet, firePoint.position, Quaternion.Euler(0, 0, rotationAngle));          fireball.GetComponent<Rigidbody2D>().velocity = direction \* 10f;          Destroy(fireball, 2f);      }      void Move(float horizontalDir, float verticalDir)      {          #region gerak kanan kiri dan atas bawah          float xVal = horizontalDir \* speed \* 100 \* Time.fixedDeltaTime;          float yVal = verticalDir \* speed \* 100 \* Time.fixedDeltaTime;          Vector2 targetVelocity = new Vector2(xVal, yVal);          rb.velocity = targetVelocity;          if (facingRight && horizontalDir < 0)          {              transform.localScale = new Vector3(-2, 2, 1);              facingRight = false;          }          else if (!facingRight && horizontalDir > 0)          {              transform.localScale = new Vector3(2, 2, 1);              facingRight = true;          }          if (!facingUp && verticalDir > 0)          {              transform.localScale = new Vector3(2, 2, 1);  // Sesuaikan ini jika Anda memiliki sprite yang berbeda untuk menghadap ke atas/bawah              facingUp = true;          }          else if (facingUp && verticalDir < 0)          {              transform.localScale = new Vector3(2, 2, 1);  // Sesuaikan ini jika Anda memiliki sprite yang berbeda untuk menghadap ke atas/bawah              facingUp = false;          }          #endregion      }  } |

1. Tambahkan file script (Enemy\_Attacked.cs) dan isikan source code dibawah ini

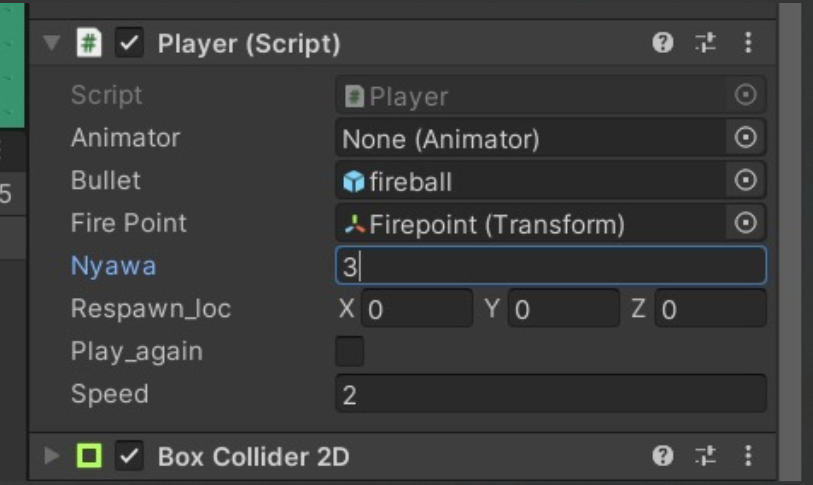
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_attacked : MonoBehaviour  {  [SerializeField] private Player Object;  void Start()  {  if (Object == null)  {  Object = GameObject.FindWithTag("Player").GetComponent<Player>();  }  }  void OnTriggerEnter2D(Collider2D other)  {  if (other.CompareTag("Player"))  {  Object.nyawa--;  if (Object.nyawa < 0)  {  Object.play\_again = true;  }  }  }  } |

1. Pada hierarchy enemy Tambahkan Script enemy attack



### 1.18 Ganti object

1. Klik game object Player, pergi ke Inspector dan ubah nilai Nyawa menjadi 3 pada Player(Script)



### 1.19 Ubah nilai

1. Jika di play, Player mengenai atau menyentuh opposum-1 sebanyak 3 kali maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan reswpawn ke titik awal



### 1.20 Hasil jadi

1. **Repository GitHub**

Kevinmajesta/2118003\_PRAK\_ANIGAME

1. **Kuis**

Lengkapi Source code dibawah ini :

|  |
| --- |
| using UnityEngine;  public class PlayerAttack : MonoBehaviour  {  public int atackRange = 2.0f;  public int attacDamage = 10;  void Update()  {  if (InputGetButtonDown("Fire1"))  {  PerformMeleeAttack();  }  }  void PerformMeleeAttack()  {  RaycastHit hit;  if (Physics.Raycast(transform.position, transform.forward, out hit, attackRange))  {  // Lengkapi kode di sini untuk mengenai musuh dan mengurangi health mereka  }  }  } |

Source code yang lengkap :

|  |
| --- |
| using UnityEngine;  public class PlayerAttack : MonoBehaviour  {  public float attackRange = 2.0f;  public int attackDamage = 10;  void Update()  {  if (Input.GetButtonDown("Fire1"))  {  PerformMeleeAttack();  }  }  void PerformMeleeAttack()  {  RaycastHit hit;  if (Physics.Raycast(transform.position, transform.forward, out hit, attackRange))  {  EnemyHealth enemyHealth = hit.transform.GetComponent<EnemyHealth>();  if (enemyHealth != null)  {  enemyHealth.TakeDamage(attackDamage);  }  }  }  } |

Penjelasan :

Kode diatas yang saya tambah mengambil komponen EnemyHealth dari objek yang terkena oleh serangan atau tindakan pemain, dan jika komponen tersebut ada, maka metode TakeDamage() dari komponen tersebut dipanggil untuk memberikan kerusakan sesuai dengan nilai attackDamage.

Dan ini merupakan source code yang sudah diperbaiki

|  |
| --- |
| void HandleJumpInput()  {  if (Input.GetKeyDown(KeyCode.Space))  {  animator.SetBool("isJumping", true); // Set to true when jumping starts  rb.AddForce(Vector2.up \* jumpForce, ForceMode2D.Impulse);  }  else if (Input.GetKey(KeyCode.Space))  {  animator.SetBool("isJumping", true); // Set to true while the space key is held down  }  else  {  animator.SetBool("isJumping", false); // Set to false when the space key is released  }  }  void HandleMovementInput()  {  float move = Input.GetAxis("Horizontal");  if (move == 0)  {  animator.SetBool("isIdle", true);  animator.SetBool("isWalking", false);  }  else  {  animator.SetBool("isIdle", false);  animator.SetBool("isWalking", true);    transform.Translate(Vector3.right \* move \* Time.deltaTime); // Corrected to Vector3.right  }  if (move < 0)  {  transform.localScale = new Vector3(-1, 1, 1); // Correct scaling for left direction  }  else if (move > 0)  {  transform.localScale = new Vector3(1, 1, 1); // Correct scaling for right direction  }  } |

Penjelasan :

Coding diatas adalah bagian dari fungsi-fungsi untuk mengatur input pemain dalam sebuah game. Fungsi HandleJumpInput() menangani input loncatan, dimana ketika tombol spasi ditekan, karakter akan meloncat dengan menerapkan gaya impuls ke atas, sementara HandleMovementInput() mengatur pergerakan horizontal karakter berdasarkan input pemain. Jika tidak ada input, karakter akan diam, jika input positif, karakter akan bergerak ke kanan, dan sebaliknya. Animasi karakter diatur berdasarkan input dengan mengatur parameter animasi seperti isJumping, isIdle, dan isWalking.