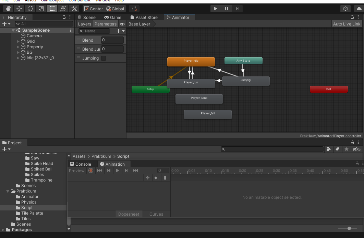
TUGAS PERTEMUAN: 10

Respawn and AI Enemey Attack

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 1918028 |
| **Nama** | : | Darmawansyah |
| **Kelas** | : | D |
| **Asisten Lab** | : | M. Rafi Faddilani (2118144) |

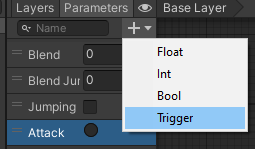
* 1. **Tugas 1 : Membuat Respawn and AI Enemey Attack**

1. **Character Animation**
2. Buka *Project* Bab 9 Untuk melanjutkannya



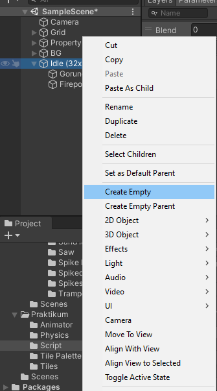
### Gambar 10. 1 Tampilan *Project* Bab 9

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



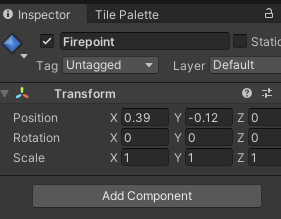
### Gambar 10. 2 Tampilan Parameter *Trigger.*

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



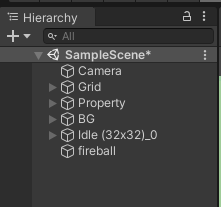
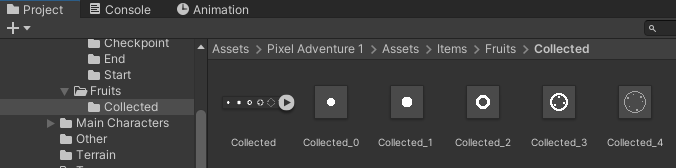
### Gambar 10. 3 Tampilan Membuat *layer game object*

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

### Gambar 10. 4 Tampilan *Hierachy*

1. Pada menu Hierarchy Tambahkan item- Collected-3, di folder Sprites > Fruits > item- Collected-3, *rename* menjadi *fireball*

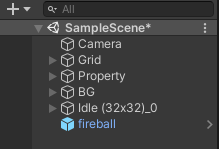
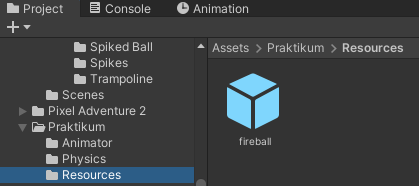
### Gambar 10. 5 Tampilan Membuat *layer game object*

1. Klik item- Collected-3 untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



### Gambar 10. 6 Tampilan Menambahkan *Component*

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

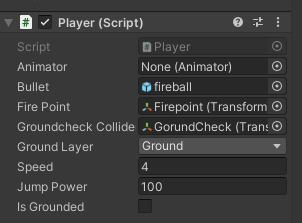
 

### Gambar 10. 7 Tampilan *folder Resources*

1. Pada script *Player* tambahkan *source code* dibawah ini.

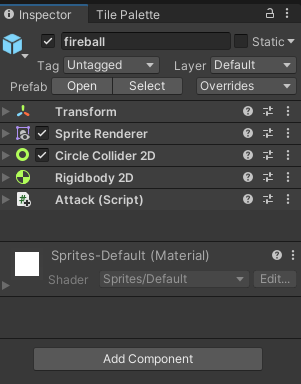
|  |
| --- |
| //tambahkan source code berikut untuk deklarasi variabel  public Animator animator;  public GameObject bullet;  public Transform firePoint;  //tambahkan dibawah fungsi FixedUpdate  IEnumerator Attack(){  animator.SetTrigger("Attack");  yield return new WaitForSeconds(0.25f);    // agar player dapat menembak ke kanan dan kiri  float direction = facingRight ? 1f : -1f;  GameObject peluru = Instantiate(bullet, firePoint.position, Quaternion.identity);  peluru.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 10f, 0);    Destroy(peluru, 2f);  }  //tambahkan didalam 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



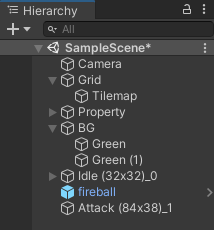
### Gambar 10. 8 Tampilan *inspector Player*

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



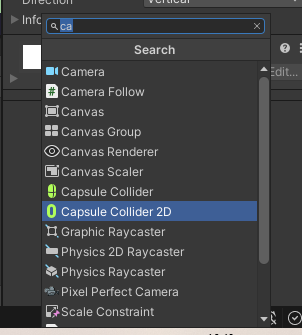
### Gambar 10. 9 Tampilan Menambahkan *Scrip Attact*

1. Tambahkan Enemy bluebird-attack pada hierarchy di folder Sprites, eagle



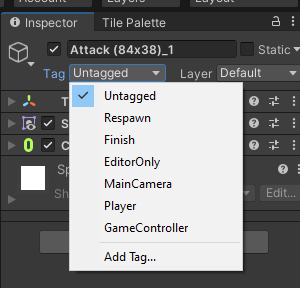
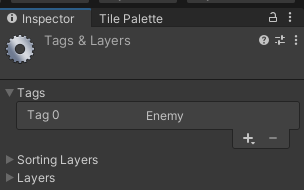
### Gambar 10. 10 Tampilan *folder Enemy bluebird-attack*

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



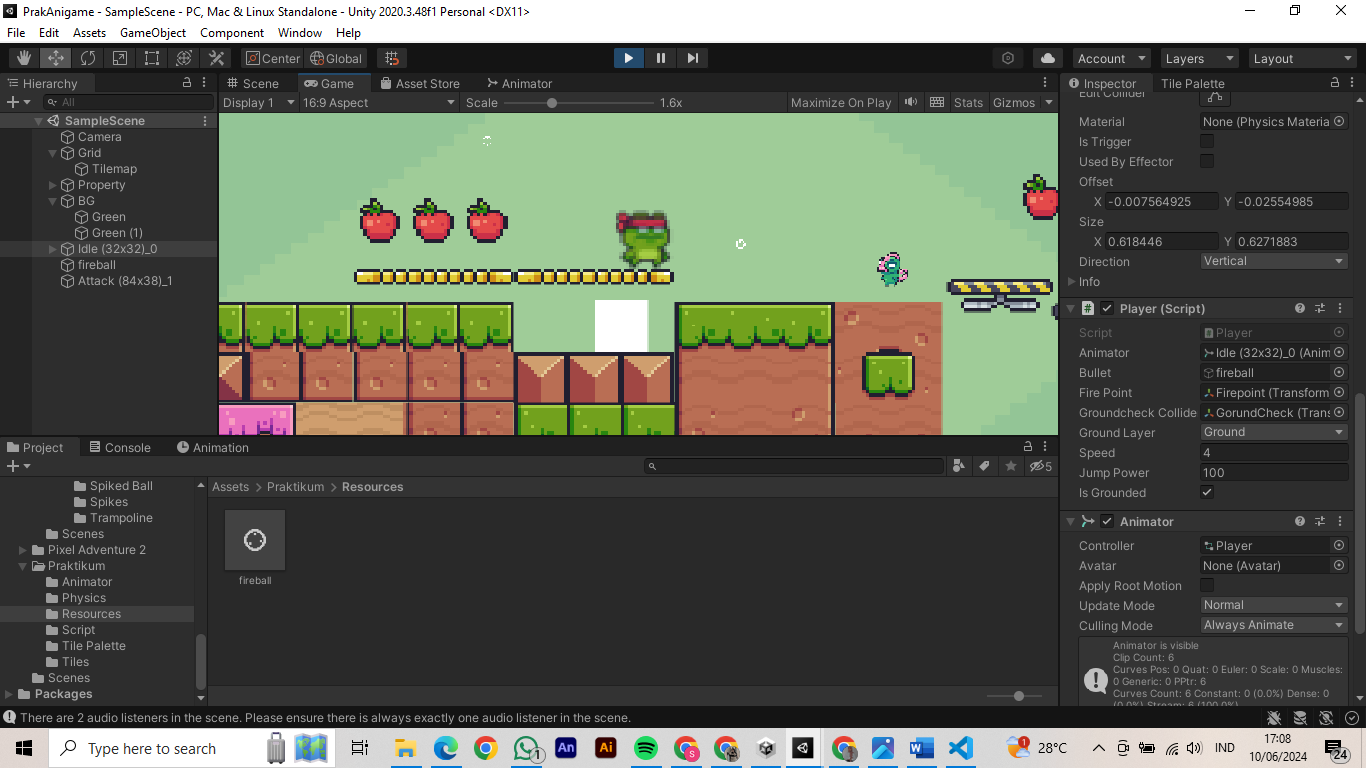
### Gambar 10. 11 Tampilan *collider 2d*

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

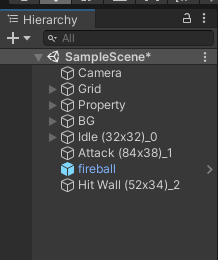
### Gambar 10. 12 Tampilan *add tag*

1. Tembak Enemey dengan menekan Tombol C untuk menghancurkan musuh



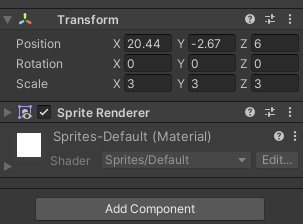
### Gambar 10. 13 Tampilan tembak *ennemy*

1. Kemudian Cari sebuah sprite pack Bernama enemy dan buka folder bernama “Hitwall”Lalu Tambahkan “Hitwall -1” ke Hierarchy.



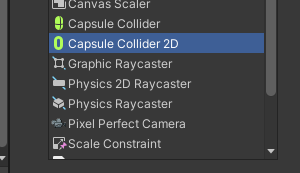
### Gambar 10. 14 Tampilan *folder hitwall*

1. Kemudian Pada inspector atur transform scale menjadi seperti berikut



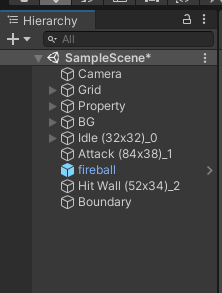
### Gambar 10. 15 Tampilan *inspector*

1. Setelah Itu Tambahkan sebuah komponen bernama *Capsule Colider 2D* dan *Rigidbody* dalam *inspector* *game* objek Hitwall -1



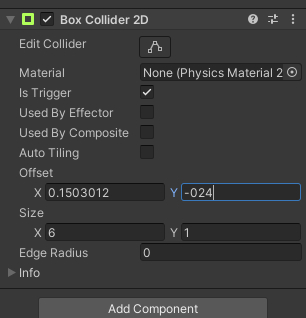
### Gambar 10. 16 Tampilan *add component*

1. *Create* Empty *object* pada *Hierarchy*, *Rename* Menjadi *Boundary*



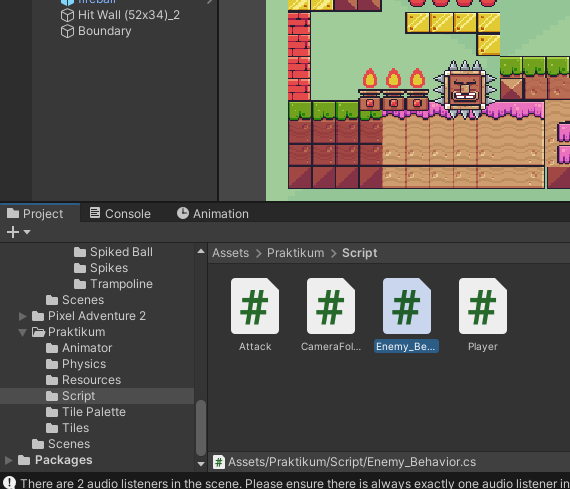
### Gambar 10. 17 Tampilan *create empty*

1. Tambahkan *Box Collider 2d* pada *Boundary*, centang pada Is Trigger lalu atur sesuai keinginan pada *size* dan *offside*



### Gambar 10. 18 Tampilan *box collider 2d*

1. Buat sebuah *file script* didalam folder Script beri nama “*Enemy\_Behavior*”, kemudian drag dan masukkan ke dalam game *object* “*Hitwall* -1”



### Gambar 10. 19 Tampilan *file script*

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. Buat *script* baru dan beri nama *Enemy*\_*Behavior*Lalu tambahkan *source code* dibawah ini. *Drag and drop script* tersebut ke *Hitwall* *-*1.

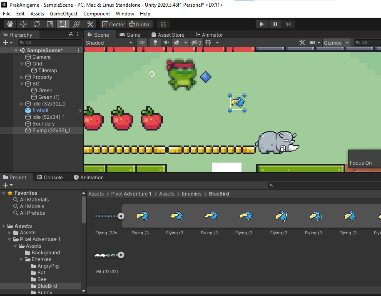
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_Behavior\_Crab : MonoBehaviour{  [SerializeField] float moveSpeed = 1f;  Rigidbody2D rb;  void Start(){  rb = GetComponent<Rigidbody2D>();  }  void Update(){  if (isFacingRight()){  rb.velocity = new Vector2(-moveSpeed, 0f);  }else{  rb.velocity = new Vector2(moveSpeed, 0f);  }  }  private bool isFacingRight(){  return transform.localScale.x > Mathf.Epsilon;  }  private void OnTriggerExit2D(Collider2D collision){  transform.localScale = new Vector2(-transform.localScale.x, transform.localScale.y);  }  } |

1. Kemudian Coba Jalankan Program



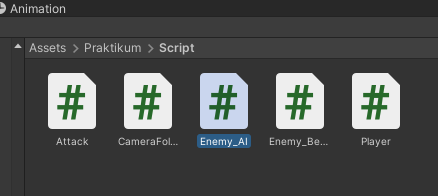
### Gambar 10. 20 Tampilan jalankan *program*

1. Langkah Selanjutnya Cari sebuah *sprite pack* bernama *'enemy'* dan buka *folder* bernama *Flying* (32x32)\_1'. Tambahkan *Flying* (32x32)\_1' pada *Hierarchy*

****

### Gambar 10. 21 Tampilan *tambah fliying*

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

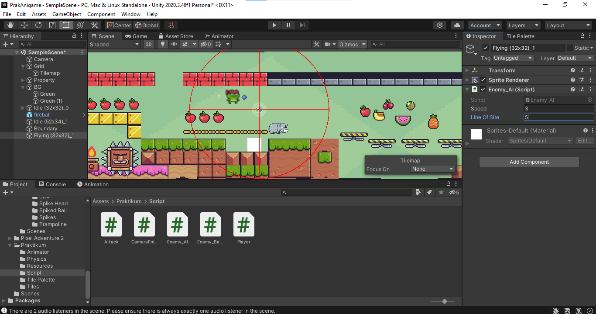


### Gambar 10. 22 Tampilan *folder script*

1. Tambahkan source *code* berikut didalam *script Enemy\_AI*

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_AI\_Octopus : MonoBehaviour{  public float speed;  public float lineOfSite;  private Transform player;  private Vector2 initialPosition;  void Start(){  player = GameObject.FindGameObjectWithTag("Player").transform;  initialPosition = GetComponent<Transform>().position;  }  void Update(){  float distanceToPlayer = Vector2.Distance(player.position, transform.position);  Vector2 targetPosition;  if (distanceToPlayer < lineOfSite){  targetPosition = player.position;  }else{  targetPosition = initialPosition;  }  transform.position = Vector2.MoveTowards(this.transform.position, targetPosition, speed \* Time.deltaTime);  if (targetPosition.x > transform.position.x){  transform.localScale = new Vector3(-7, 7, 1);  }else if (targetPosition.x < transform.position.x){  transform.localScale = new Vector3(7, 7, 1);  }  }  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*



### Gambar 10. 23 Tampilan *speed pada enemy*

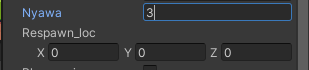
1. Lalu buka *script player* dan tambahkan berikut

|  |
| --- |
| // menambahkan variabel  public int nyawa;  [SerializeField] Vector3 respawn\_loc;  public bool play\_again;  // menambahkan posisi respawn didalam void Awake()  respawn\_loc = transform.position;  //menambahkan respawn di void Update()  if(nyawa < 0){  playagain();  }  if (transform.position.y < -10){  play\_again = true;  playagain();  }  //menambahkan fungsi playagain()  void playagain(){  if(play\_again == true){  nyawa = 3;  transform.position = respawn\_loc;  play\_again = false;  }  } |

1. Buat *script* baru dengan nama *Enemy*\_*Attacked* lalu isikan *source code* dibawah ini.

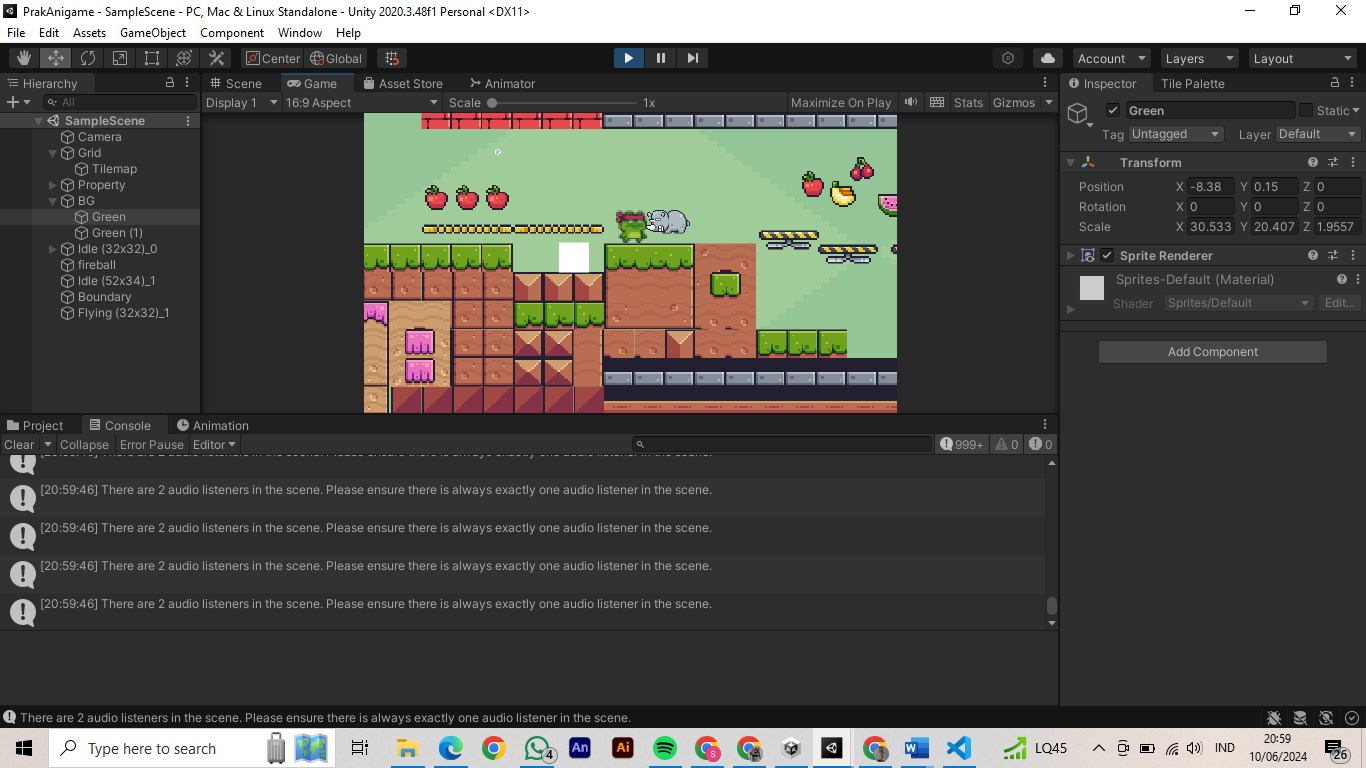
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_Attacked : MonoBehaviour{  [SerializeField] private Player Object;  void Start(){  if (Object == null)  {  Object = GameObject.FindWithTag("Player").GetComponent<Player>();  }  }  void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

1. Tambahkan *script* Enemy\_*Attacked* ke *Octopus*. Lalu klik *Player* dan ubah nyawa pada bagian *script* menjadi 3.



### Gambar 10. 24 Tampilan atur nyawa

1. Coba jalankan.



### Gambar 10. 25 Tampilan Akhir

1. **Kuis Pertemuan 10**

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 sudah dilengkapi yang kurang

|  |
| --- |
| 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);  }  }  }  } |

Tanda Merah yang menyebabkan Source code Error :

|  |
| --- |
| void HandleJumpInput()  {  if (Input.GetKeyDown(KeyCode.Space))  {  animator.SetBool("isJumping", );  rb.AddForce(Vector2.up \* jumpForce, ForceMode2D.Impulse);  }  else if (Input.GetKey(KeyCode.Space))  {  animator.SetBool("isJumping",);  }  }  void HandleMovementInput()  {  float move = Input.GetAxis("Horizontal");  if (move != 1)  {  animator.SetBool("isIdle", true);  transform.Translate(Vector3.left \* move \* Time.deltaTime);  }  else  {  animator.SetBool("isWalking", false);  }  if (move != 0)  {  transform.localScale = new Vector3(-4, 1, 1);  }  else if (move > 0)  {  transform.localScale = new Vector3(1, 2, 1);  }  } |

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  }  } |

Analisa :

Pada bagian pertama Sourcode , fungsi *PerformMeleeAttack* yang ada pada *PlayerAttack* script mengalami beberapa kekurangan. Kesalahan utamanya adalah tidak adanya kode untuk mengurangi *health* musuh setelah terkena serangan. Hal ini disebabkan oleh tidak adanya pemeriksaan komponen musuh (*EnemyHealth*) yang mengelola health mereka. Selain itu, dalam *script* *HandleJumpInput*, kesalahan utama adalah tidak adanya nilai *boolean* pada *animator*.*SetBool*("*isJumping*", );. Kondisi ini menyebabkan *error* kompilasi karena *SetBool* membutuhkan dua parameter: nama parameter animasi dan nilai *boolean* yang akan diatur. Pada *script* *HandleMovementInput*, terdapat beberapa kesalahan *logika*. Pertama, kondisi *if (move != 1)* yang seharusnya memeriksa apakah nilai *move* tidak sama dengan nol (0) untuk mendeteksi gerakan. Kedua, arah *transform.Translate(Vector3.left \* move \* Time.deltaTime);* yang tidak sesuai dengan arah gerakan yang benar.

Setelah perbaikan, pada fungsi *PerformMeleeAttack*, telah ditambahkan logika untuk memeriksa apakah objek yang terkena adalah musuh dan mengurangi *health* mereka menggunakan komponen *EnemyHealth*. *Script HandleJumpInput* telah diperbaiki dengan menambahkan nilai *boolean* yang sesuai (*true atau false*) pada *animator*.*SetBool*, sehingga sekarang dapat berfungsi dengan benar. Pada *script* *HandleMovementInput*, *kondisi if (move != 1)* telah diperbaiki menjadi *if (move == 0)* untuk mendeteksi jika tidak ada pergerakan. Selain itu, arah gerakan telah diperbaiki dengan menggunakan *Vector3.right* untuk mencerminkan arah yang benar berdasarkan nilai *move*. Logika animasi juga telah diperbaiki dengan mengatur *animator.SetBool("isWalking", true)* ketika ada gerakan dan *animator.SetBool("isIdle", false)* ketika tidak ada. Terakhir, pengaturan *transform.localScale* telah disesuaikan untuk mencerminkan arah karakter yang menghadap kiri atau kanan dengan benar, berdasarkan nilai *move*.

1. **Link Github Pengumpulan**

https://github.com/darmawansyah01/1918028\_PRAK\_ANIGAME.git