



TUGAS PERTEMUAN: 10

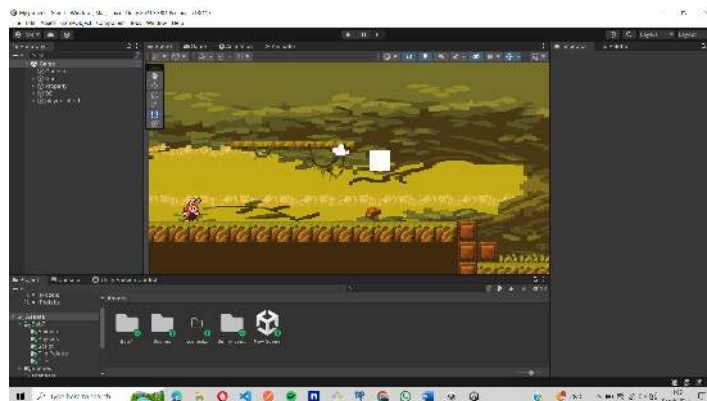
Respawn and AI Enemy Attack

NIM	:	2118100
Nama	:	Moch Arif Rochmanullah
Kelas	:	C
Asisten Lab	:	RIFAL RIFQI RHOMADON (2218106)
Baju Adat	:	
Referensi	:	

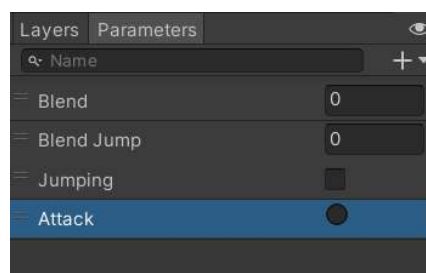
9.1 Tugas 9 : Membuat Mekanisme Attack

A. Membuat Mekanisme Attack

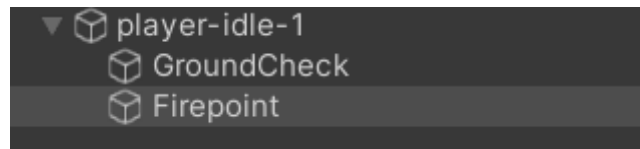
1. Buka Project Bab 9 Untuk melanjutkannya



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



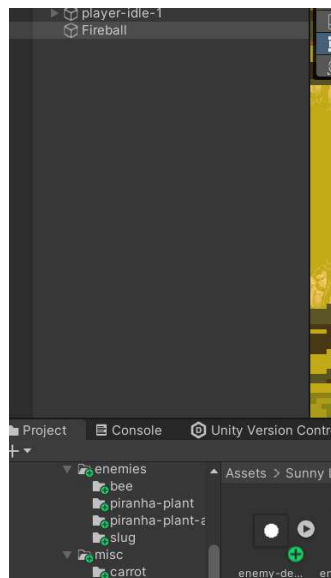
3. Setelah menambahkan parameter Attack, Langkah selanjutnya adalah membuat Layer *Game object* baru didalam *player-idle-1*, Klik kanan pilih *Create Empty* lalu Rename menjadi *Firepoint*



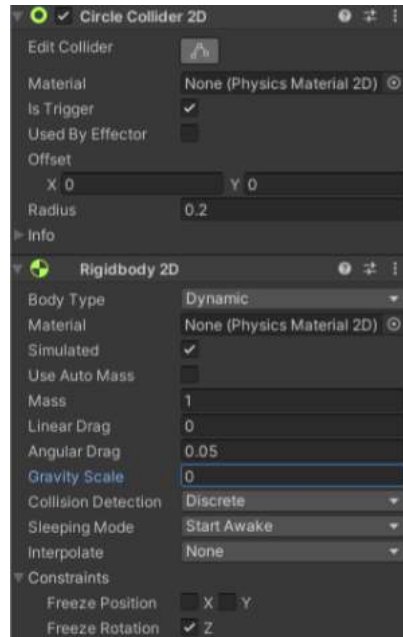
4. Pada menu *Hierarchy* klik Firepoint untuk setting pada Inspector, Ubah *Icon* Menjadi titik, atur letak titik didepan player



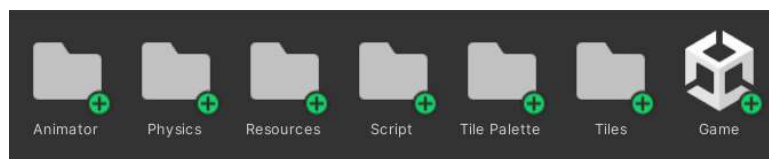
5. Pada menu Hierarchy Tambahkan item-feedback-1, di folder Sprites > misc > enemy-death , *rename* menjadi *fireball*



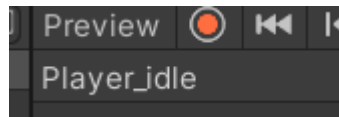
6. Klik item-feedback-1 untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



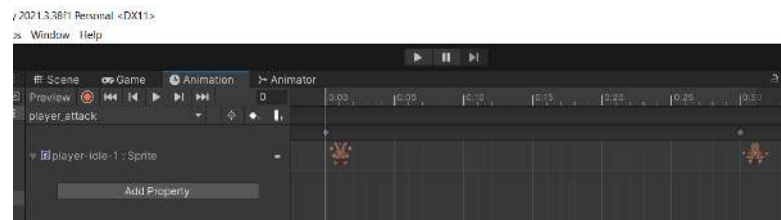
7. Buat Folder baru *Resources* di menu Project, kemudian drag and drop fireball kedalam folder Resources, dan hapus *fireball* pada *Hierarchy*



8. Pada player-idle buat animasi baru dan beri nama *player_attack*



9. Pada menu project buka folder player lalu pilih player-hurt dan pilih gambar player-hurt-1 sampai player-hurt-2, kemudian drag ke tab Animation lalu geser keyframe sampai frame 0:30.



10. Buat transisi dari *player_idle* ke *player_attack* dan atur pada bagian conditions tambahkan condition Attacking dengan nilai true. Buat juga transisi dari *player_attack* ke *player_idle* dan atur pada bagian conditions tambahkan condition Attacking dengan nilai false.



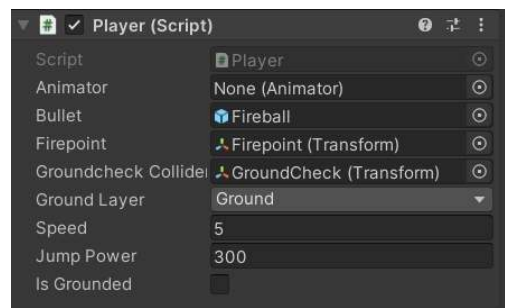
11. Pada Script Player Tambahkan Script dibawah ini

```
// Deklarasi variable
public Animator animator;
public GameObject bullet;
public Transform firePoint;
//Ubah inisialisasi nilai dari facingRight;
bool facingRight = true;

//Tambahkan dibawah fungsi fixedUpdate
IEnumerator Attack()
{
    animator.SetBool("Attacking", true);
    yield return new WaitForSeconds(0.25f);
    float direction = facingRight ? 1f : -1f;

    GameObject fireball = Instantiate(bullet,
    firePoint.position, Quaternion.identity);
    fireball.GetComponent<Rigidbody2D>().velocity = new
    Vector2(direction * 10f, 0);
    Destroy(fireball, 2f);
    animator.SetBool("Attacking", false);
}
//Tambahkan pada Function Void Update
if (Input.GetKeyDown(KeyCode.C))
{
    StartCoroutine(Attack());
}
```

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



13. Buat Script Attack pada folder Script

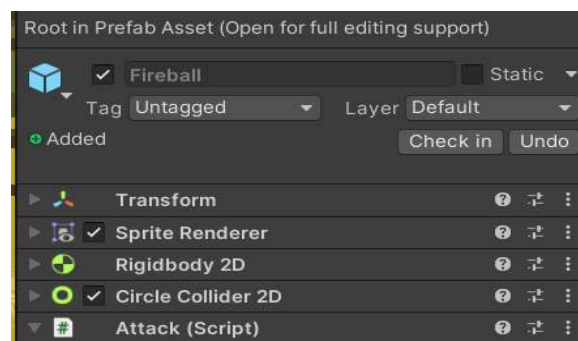


14. Tambahkan Script Attack dibawah ini

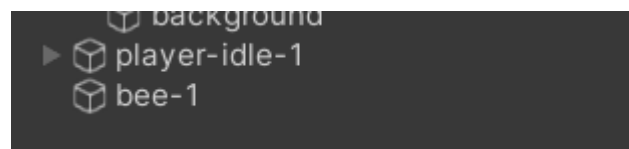
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```

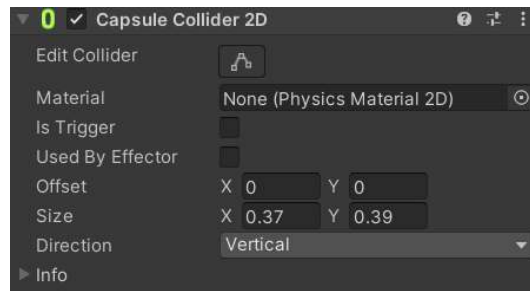
15. Didalam folder resource Tambahkan Script Attack di Prefab fireball, dengan cara Klik fireball kemudian pada menu Inspector arahkan Script Attack kedalam Inspector



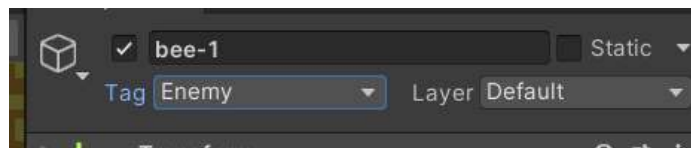
16. Tambahkan Enemy bee-1 pada hierarchy di folder enemey



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



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

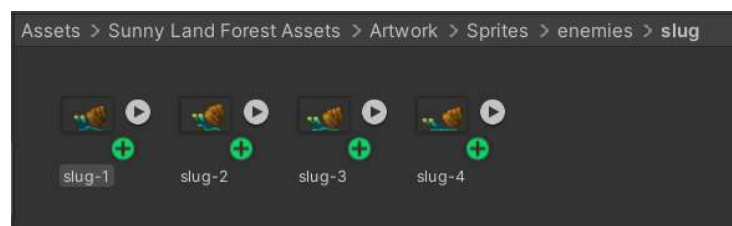


19. Tembak Enemy dengan menekan Tombol C untuk menghancurkan musuh



B. Membuat enemy AI

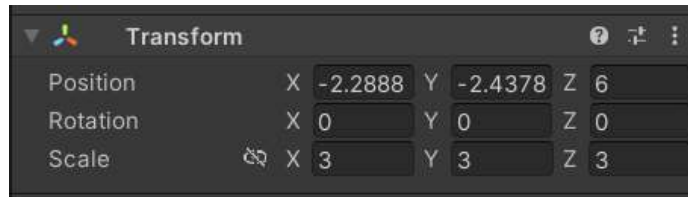
1. Cari sebuah sprite pack Bernama enemy dan buka folder bernama “slug”



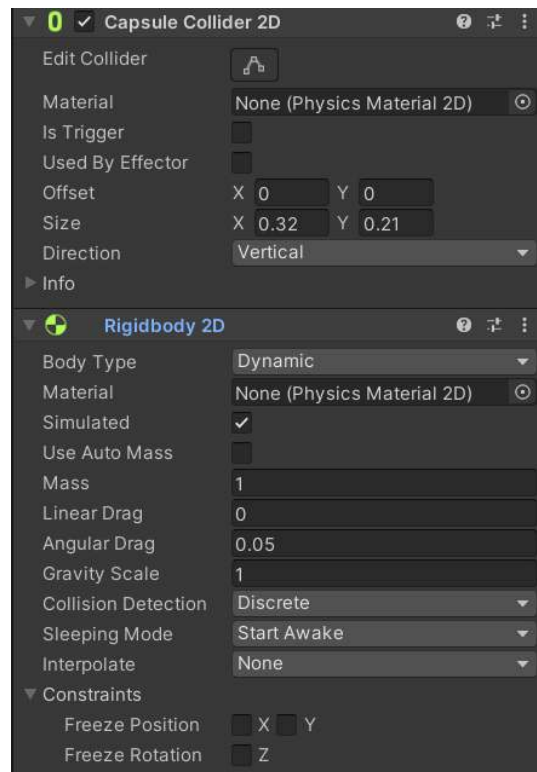
2. Tambahkan “slug-1” ke Hierarchy.



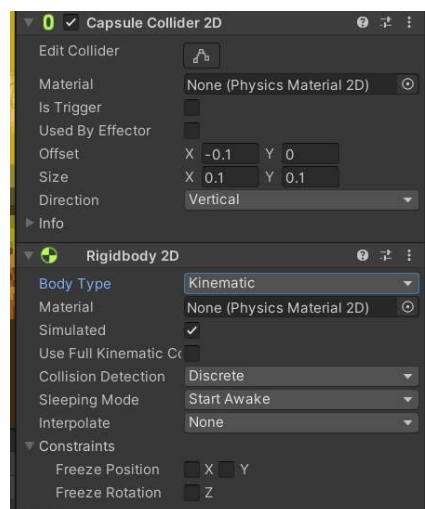
3. Pada inspector atur transform scale menjadi seperti berikut



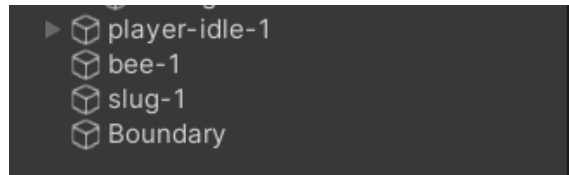
4. Tambahkan sebuah komponen bernama Capsule Collider 2D dan Rigidbody dalam inspector game objek slug-1



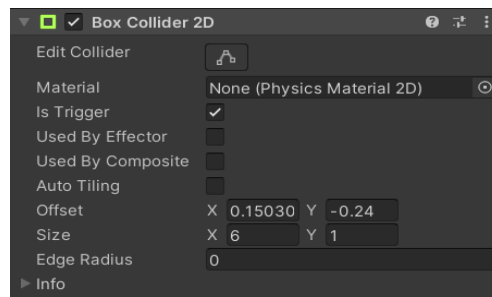
5. Atur sedikit collider tersebut seperti ukurannya diubah jika terlalu besar, dan pada *Body Type* Ubah menjadi *Kinematic*



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



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



8. Buat sebuah file script didalam folder Script beri nama “Enemy_Behavior”, kemudian drag dan masukkan ke dalam game object “slug-1”



9. Tambahkan Script dibawah ini

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Enemy_Behavior : 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);
        }
    }
}
```

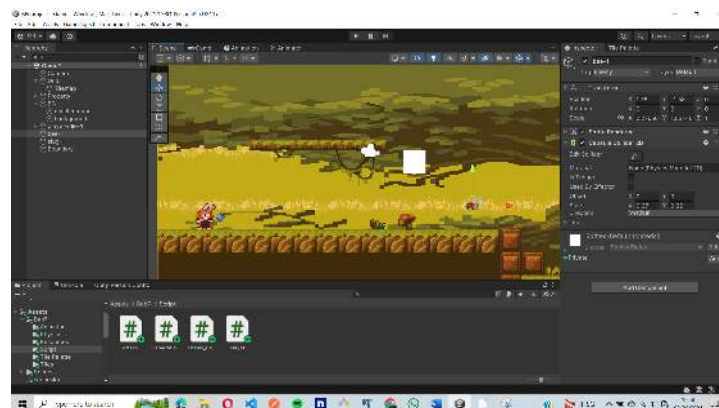



```
}  
  
private bool isFacingRight()  
{  
    return transform.localScale.x > Mathf.Epsilon;  
}  
  
private void OnTriggerExit2D(Collider2D collision)  
{  
    transform.localScale = new Vector2(-  
transform.localScale.x, transform.localScale.y);  
}  
}
```

10. Jalankan Program



11. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama bee-1'. Tambahkan bee-1' pada Hierarchy



12. Buat Script Enemy_AI pada folder Bab7 – Script





13. 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);
    }
}
```

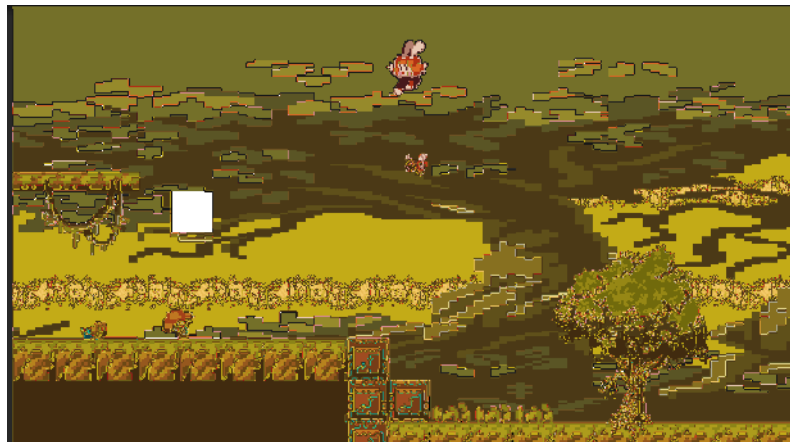


```
}
```

14. Pada Inspector Enemy_Ai, Atur Speed juga Line of Site untuk menentukan jarak dan speed pada enemy



15. Running Game, maka eagle akan mengikuti Gerakan Player



C. Respawn

1. Buka file script (Player.cs) tambahkan variabel nyawa seperti dibawah ini

```
public int nyawa;  
[SerializeField] Vector3 respawn_loc;  
public bool play_again;
```

2. Tambahkan kode dibawah ini untuk mengatur posisi respawn sesuai dengan posisi awal permainan dimulai

```
private void Awake()  
{  
    rb = GetComponent<Rigidbody2D>();  
    animator = GetComponent<Animator>();  
    respawn_loc = transform.position;  
}
```

3. Tambahkan kode dibawah ini di dalam void update Player.cs agar ketika nyawa player dibawah 0 maka akan melakukan respawn



```
void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
    if (Input.GetButtonDown("Jump"))...
    else if (Input.GetButtonUp("Jump"))...

    // playagain
    if (nyawa < 0)
    {
        playagain();
    }
}
```

4. Tambahkan juga kode berikut dibawah code sebelumnya agar ketika player jatuh dibawah platform akan melakukan respawn

```
if (nyawa < 0)
{
    playagain();
}

if (transform.position.y < -10)
{
    play_again = true;
    playagain();
}
```

5. Tambahkan fungsi playagain() dalam script Player.cs

```
private void Awake()...

void playagain()
{
    if (play_again == true)
    {
        nyawa = 3;
        transform.position = respawn_loc;
        play_again = false;
    }
}
```

6. 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--;

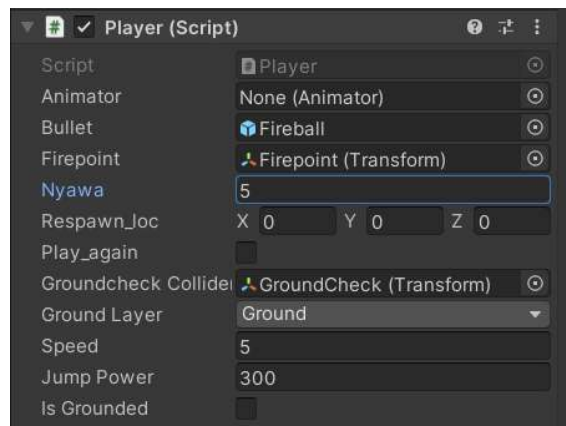
            if (Object.nyawa < 0)
            {
                Object.play_again = true;
            }
        }
    }
}
```



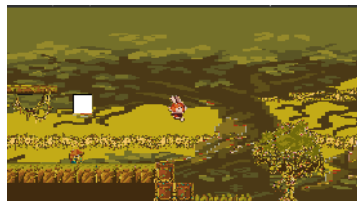
7. Pada hierarchy bee-1 Tambahkan Script enemy attack, arahkan object pada player-idle-1



8. Klik game object Player, pergi ke Inspector dan ubah nilai Nyawa menjadi 5 pada Player(Script)



9. 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 respawn ke titik awal



D. Quiz

```
using UnityEngine;
public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    public int attackDamage = 10;
    public string enemyTag = "Enemy";
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PerformMeleeAttack();
        }
    }
    void PerformMeleeAttack()
    {
        RaycastHit hit;
```



```
        if (Physics.Raycast(transform.position,
transform.forward, out hit, attackRange))
        {
            if (hit.collider.CompareTag(enemyTag))
            {Health healthComponent =
hit.collider.GetComponent<Health>();
                if (healthComponent != null)
                {
                    healthComponent.TakeDamage(attackDamage);
                }
            }
        }
    }
}
```

Penjelasan

Source code di atas telah diperbaiki pada metode `PerformMeleeAttack()`. Pertama, tipe variabel `attackRange` diubah dari `int` menjadi `float` untuk mencerminkan penggunaannya sebagai nilai jarak serangan. Kedua, kesalahan ketik pada `InputGetButtonDown` diperbaiki menjadi `Input.GetButtonDown`, dan `attacDamage` diperbaiki menjadi `attackDamage`. Penambahan tag `enemyTag` memungkinkan identifikasi musuh melalui tag, memastikan hanya musuh yang terkena serangan. Dalam metode `PerformMeleeAttack`, ditambahkan pemeriksaan untuk memastikan bahwa objek yang terkena raycast memiliki komponen `Health`, yang bertanggung jawab untuk mengurangi health musuh dan menangani kematian mereka.

E. Link Github Pengumpulan

https://github.com/Rachmanullah/2118100_PRAK_ANIGAME.git