



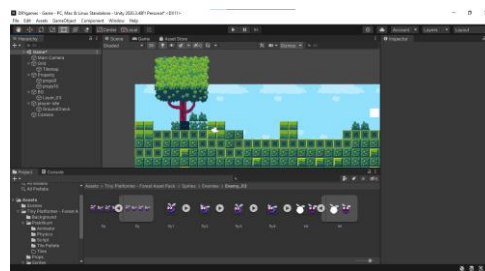
TUGAS PERTEMUAN 10

Respawn and AI Enemy Attack

NIM	:	2118090
Nama	:	Derry Frediansa
Kelas	:	C
Asisten Lab	:	Nur Aria Hibnastiar (2118078)

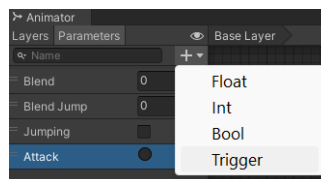
10.1 Tugas 10 : Membuat AI Enemy Attack dan Respawn Serta Kuis

1. Buka Project Bab 9 Untuk melanjutkan



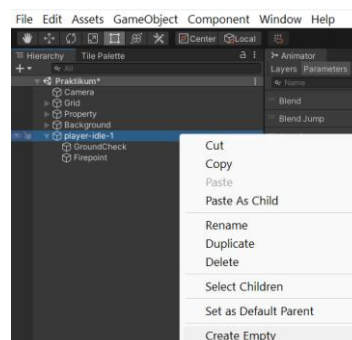
Gambar 10.1 Tampilan Bab 9

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



Gambar 10.2 Tampilan Add Parameter

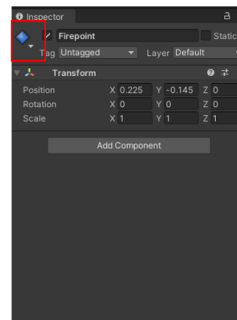
3. Setelah menambahkan parameter Attack, Langkah selanjutnya adalah membuat Layer *Game object* baru didalam *player-idle-1*, Klik kanan pilih *Create Empty* lalu Rename menjadi *Firepoint*



Gambar 10.3 Tampilan Add Firepoint

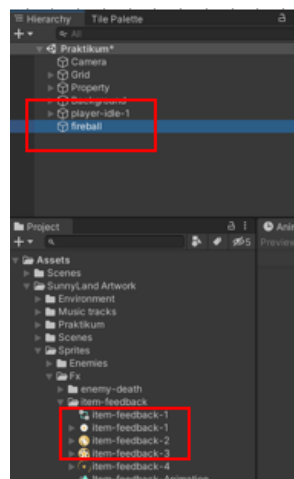


4. Pada menu *Hierarchy* klik Firepoint untuk setting pada Inspector, Ubah *Icon* Menjadi titik, atur letak titik didepan player



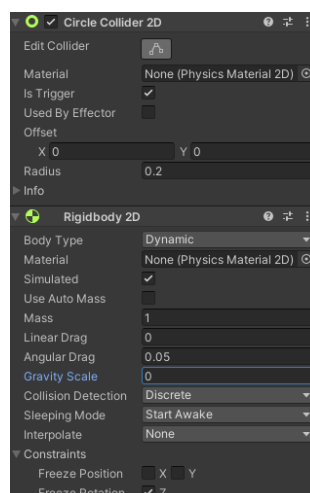
Gambar 10.4 Tampilan Mengatur Icon

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



Gambar 10.5 Menambahkan Item Feedback

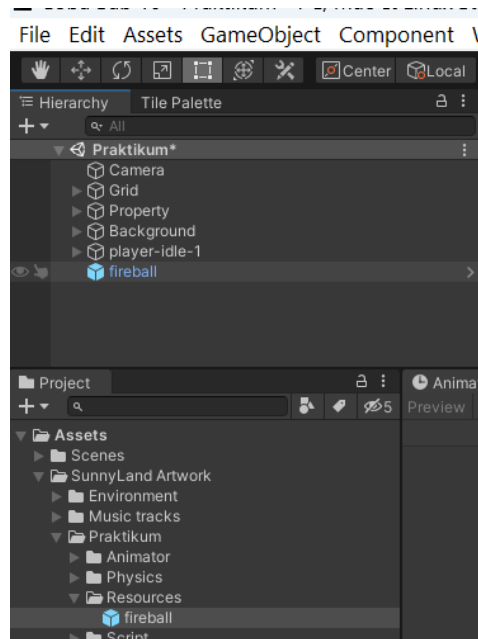
6. Klik item-feedback-1 untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



Gambar 10.6 Add Circle Collider Dan Riggidbody



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



Gambar 10.7 Buat Folder Resource

8. Pada Script Player Tambahkan Script dibawah ini

```
#Pada class Player
// Deklarasi variable
Public Animator animator;
Public GameObject bullet;
Public Transform Firepoint;

#Tambahkan dibawah fungsi fixedUpdate
IEnumerator Attack()
{
    animator.SetTrigger("Attack");
    yield return new WaitForSeconds(0.25f);

    float direction = 1f;

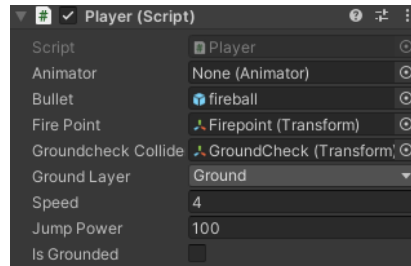
    GameObject fireball = Instantiate(bullet,
    firePoint.position, Quaternion.identity);
    fireball.GetComponent<Rigidbody2D>().velocity
    = new Vector2(direction * 10f, 0);

    Destroy(fireball, 2f);
}

#Tambahkan pada Function Void Update
if (Input.GetKeyDown(KeyCode.C))
{
    StartCoroutine(Attack());
}
```

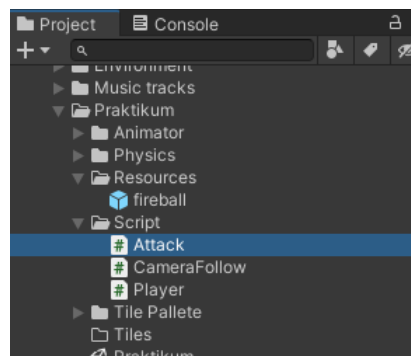


9. Pada Inspector Player, Ubah seperti dibawah ini, Dimana Bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak pertama



Gambar 10.8 Add Bullet Objek

10. Buat Script Attack pada folder Script



Gambar 10.9 Membuat Script Attack

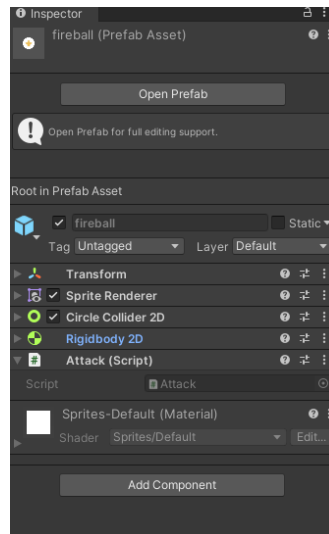
11. Tambahkan Script Attack dibawah ini

```
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);
        }
    }
}
```

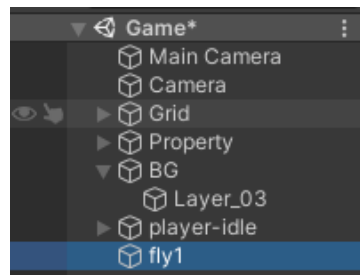


12. Didalam folder resource Tambahkan Script Attack di Prefab fireball, dengan cara Klik fireball kemudian pada menu Inspector arahkan Script Attack kedalam Inspector



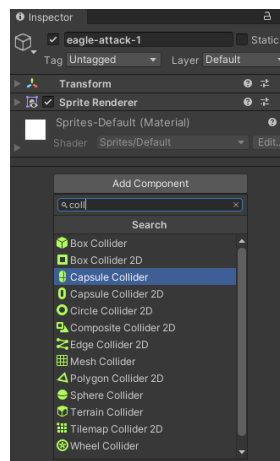
Gambar 10.10 Add Script Attack

13. Tambahkan Enemy fly1 pada hierarchy di folder Sprites, enemies03->fly



Gambar 10.11 Add Enemy Fly

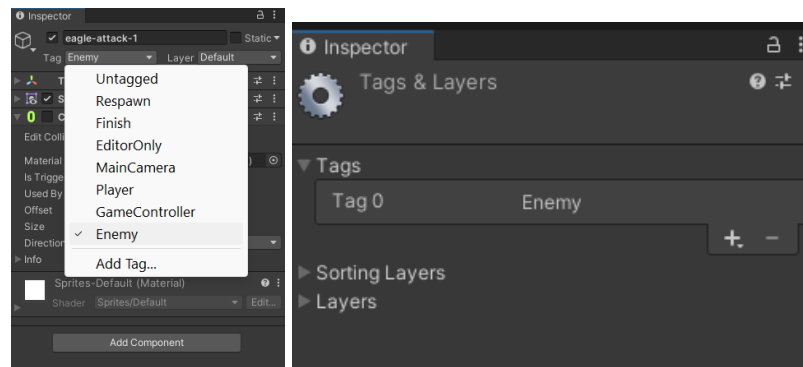
14. Kemudian klik pada fly, lalu pada menu tab inspector tambahkan collider 2D untuk mendeteksinya



Gambar 10.12 Add Capsule Collidor



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



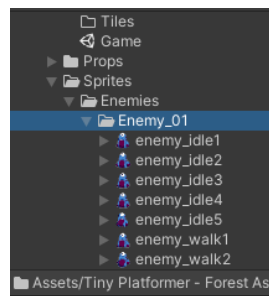
Gambar 10.13 Add Tag Enemy

16. Tembak Enemy dengan menekan Tombol C untuk menghancurkan musuh



Gambar 10.14 Tampilan Menembak Lawan

17. Cari sebuah sprite pack Bernama enemy dan buka folder bernama “enemi_01”



Gambar 10.15 Tampilan Enemy 1

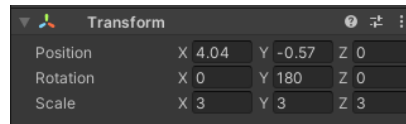
18. Tambahkan “enemy_idle1” ke Hierarchy.



Gambar 10.16 Add Enemy Idle 1

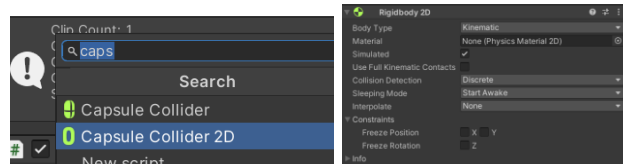


19. Pada inspector atur transform scale menjadi seperti berikut



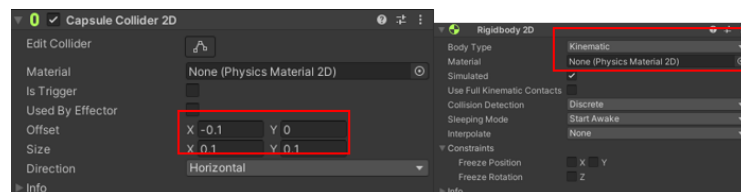
Gambar 10.17 Setting Transform Scale

20. Tambahkan sebuah komponen bernama Capsule Colider 2D dan Rigidbody dalam inspector game objek enemy_idle1



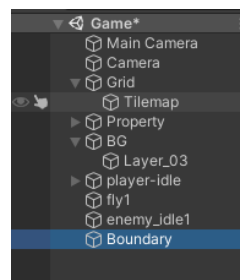
Gambar 10.18 Add Component

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



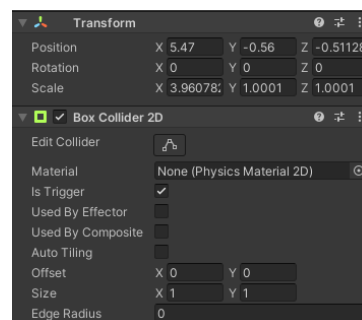
Gambar 10.19 Menyetting Komponen

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



Gambar 10.20 Create Empty Object

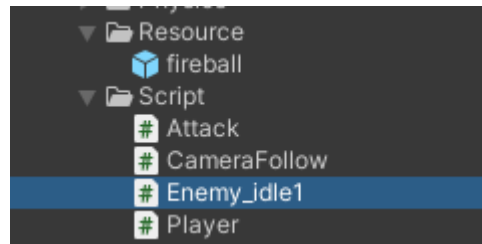
23. Tambahkan Box Collider 2d pada Boundary, centang pada Is Trigger lalu atur sesuai keinginan pada size dan offside



Gambar 10.21 Add Box Collider



24. Buat sebuah file script didalam folder Script beri nama “Enemy_idle1”, kemudian drag dan masukkan ke dalam game object “enemy_idle1”



Gambar 10.22 Membuat Script Enemy Idle

25. 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);
    }
}
```




26. Jalankan Program



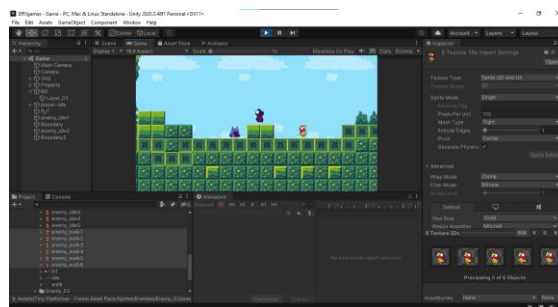
Gambar 10.23 Hasil Tampilan Lawan

27. Ulangi langkah 1 sampai 9 pakai enemy yang kedua, maka tampilan akan seperti dibawah ini



Gambar 10.24 Menambahkan Lawan

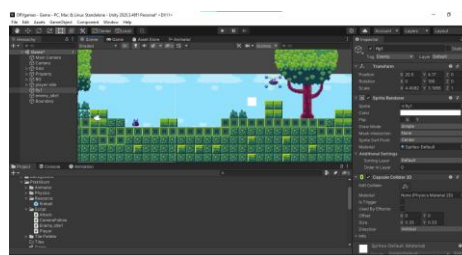
28. Tambahkan animasi pada enemy behavior maka jika berjalan aka nada animasinya seperti dibawah ini



Gambar 10.25 Menambahkan Animasi Lawan

29. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama 'fly1'.

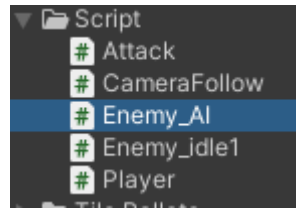
Tambahkan 'fly1' pada Hierarchy



Gambar 10.26 Add Enemy Fly



30. Buat Script Enemy_AI pada folder Praktikum - Script



Gambar 10.27 Membuat Script AI

31. 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;
    }

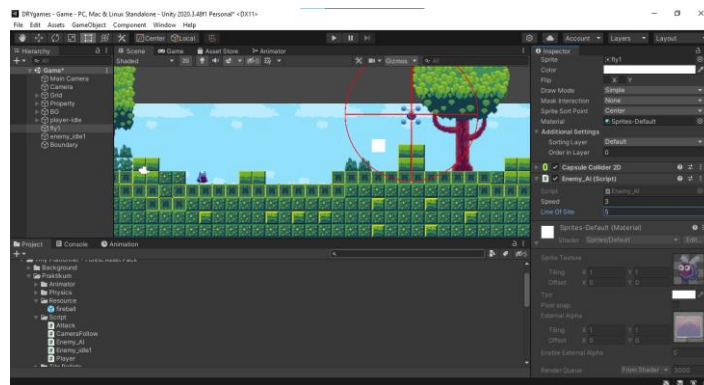
    // Update is called once per frame
    void Update()
    {
        // Menghitung jarak antara musuh dan pemain
        float distanceToPlayer = Vector2.Distance(player.position, transform.position);

        // 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);
}
}
```

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



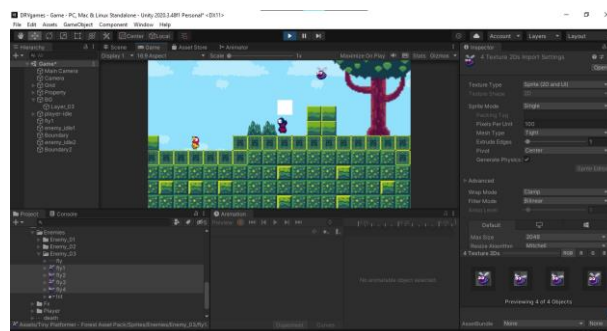
Gambar 10.28 Mengatur Speed Line

33. Running Game, maka eagle akan mengikuti Gerakan Player



Gambar 10.29 Hasil Running Game

34. Tambahkan animasi pada kupu-kupu maka akan seperti dibawah ini



Gambar 10.30 Menambahkan Animasi Kupu-Kupu



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

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

Gambar 10.31 Script Player

36. 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;  
}
```

Gambar 10.32 Script Class Awake

37. 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();  
    }  
}
```

Gambar 10.33 Script Update Playagain

38. 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();  
}
```

Gambar 10.34 Script Playagain Jatuh



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

```
27
28     private void Awake()...
34
35     void playagain()
36     {
37         if (play_again == true)
38         {
39             nyawa = 3;
40             transform.position = respawn_loc;
41             play_again = false;
42         }
43     }
44
```

Gambar 10.35 Add Fungsi Playagain

40. 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;

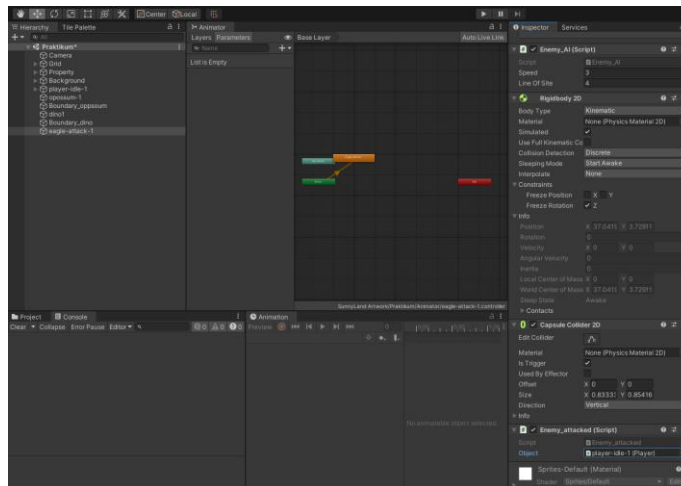
    void Start()
    {
        if (Object == null)
        {
            Object =
GameObject.FindWithTag("Player").GetComponent<Player>();
        }
    }

    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            Object.nyawa--;

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

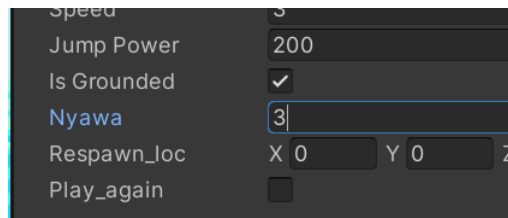


41. Pada hierarchy fly1 Tambahkan Script enemy attack, arahkan object pada player-idle-1



Gambar 10.36 Add Script Enemyattack

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



Gambar 10.37 Mengatur Nyawa

43. Jika di play, Player mengenai atau menyentuh Enemy_idle1 sebanyak 3 kali maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan respawn ke titik awal



Gambar 10.38 Hasil Akhir Running

10.2 KUIS

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f; // Mengubah tipe data
    dari int ke float
}
```



```
public int attackDamage = 10; // Mengoreksi kesalahan
penulisan variabel attacDamage menjadi attackDamage

void Update()
{
    if (Input.GetButtonDown("Fire1")) // Mengoreksi
kesalahan      penulisan      InputGetButtonDown      menjadi
Input.GetButtonDown
    {
        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
        EnemyHealth enemyHealth =
hit.transform.GetComponent<EnemyHealth>();
        if (enemyHealth != null)
        {
            enemyHealth.TakeDamage(attackDamage);
        }
    }
}
```

Analisa :

Dalam kode `PlayerAttack`, terdapat beberapa kesalahan yang perlu diperbaiki. Pertama, tipe data untuk variabel `attackRange` harus diubah dari `int` menjadi `float`, dan penulisannya harus diperbaiki dari `attackRange` menjadi `attackRange` untuk konsistensi. Kedua, variabel `attacDamage` harus diperbaiki penulisannya menjadi `attackDamage`. Selanjutnya, metode `InputGetButtonDown` harus diperbaiki menjadi `Input.GetButtonDown` untuk mengikuti konvensi penamaan metode di Unity. Terakhir, dalam fungsi `PerformMeleeAttack`, perlu ditambahkan kode untuk mengecek apakah objek yang terkena serangan memiliki komponen `EnemyHealth`, dan jika ya, panggil metode `TakeDamage` pada komponen tersebut untuk mengurangi health musuh. Dengan melakukan perbaikan ini, kode akan berjalan dengan benar tanpa error.