



TUGAS PERTEMUAN: 10

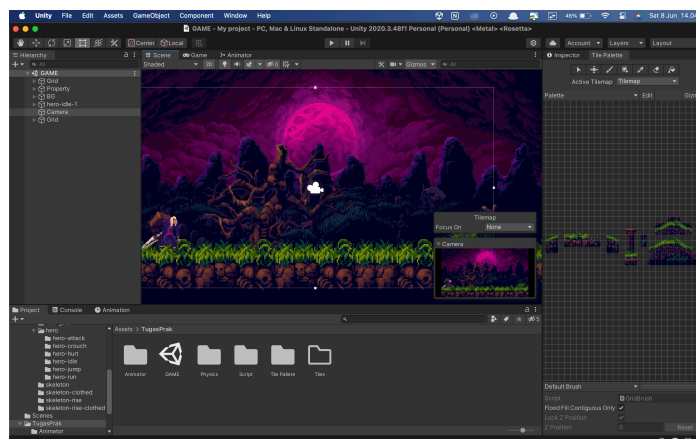
Respawn and AI Enemy Attack

NIM	:	Miftakhussurur Al Maliki
Nama	:	2118103
Kelas	:	C
Asisten Lab	:	Rifal Rifqi Rhomadon (2218106)
Baju Adat	:	Baju Safari (Bali-Indonesia Timur)
Referensi	:	https://www.inilah.com/8-jenis-pakaian-adat-bali-yang-memiliki-makna-dan-filosofi

10.1 Tugas 10 : Membuat Respawn and AI Enemy Attack

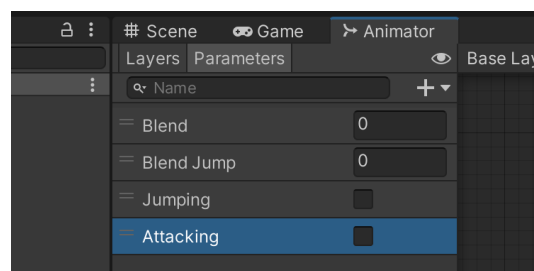
A. Membuat Mekanisme Attack

1. Buka file proyek Unity sebelumnya pada bab 9



Gambar 1.1 Membuka project sebelumnya

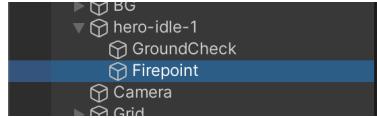
2. Kemudian pada menu Tab Animator Tambahkan Parameter Bool, Rename Menjadi *Attack*



Gambar 1.2 Menambah parameter

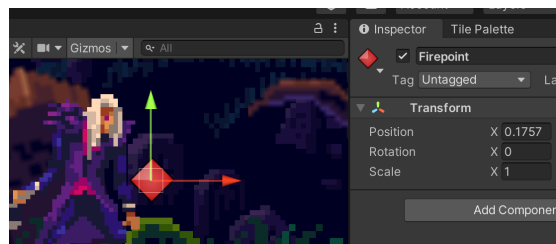


3. Setelah menambahkan parameter attack, langkah selanjutnya adalah membuat layer *game object* baru didalam *hero-idle-1*, klik kanan pilih *create empty* lalu rename menjadi *Firepoint*



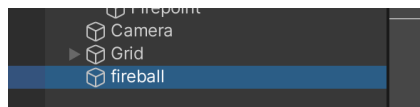
Gambar 1.3 Menambah object Fire point

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



Gambar 1.4 Mengubah icon firepoint

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



Gambar 1.5 Menambahkan fireball

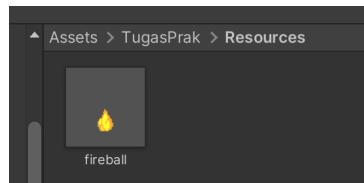
6. Klik fireball tambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



Gambar 1.6 Menambahkan component pada fireball

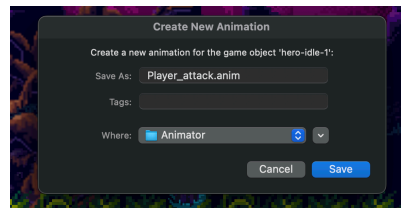


7. Buat Folder baru *Resources* di menu Project, kemudian drag and drop fireball kedalam folder Resources, dan hapus *fireball* pada *Hierarchy*



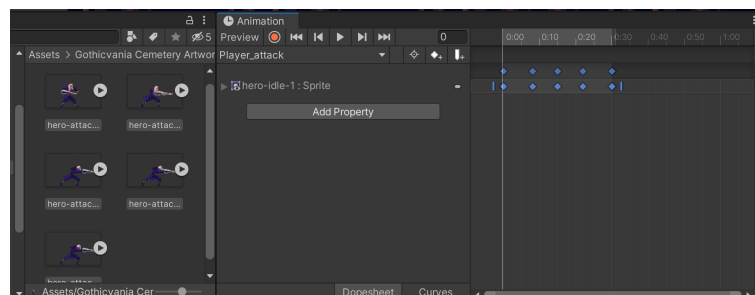
Gambar 1.7 Memindahkan fireball ke resources

8. Pada hero-idle buat animasi baru dan beri nama “Player_attack”



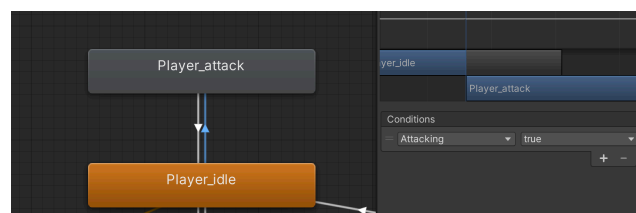
Gambar 1.8 Menambahkan animasi baru

9. Pada menu project buka folder hero lalu pilih hero-attack dan pilih gambar hero-attack-1 sampai hero-attack-5, kemudian drag ke tab Animation lalu geser keyframe sampai frame 0:30.



Gambar 1.9 Memasukkan assets hero attack ke timeline animation

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.



Gambar 1.10 Menambahkan transisi dari player idle ke player attack



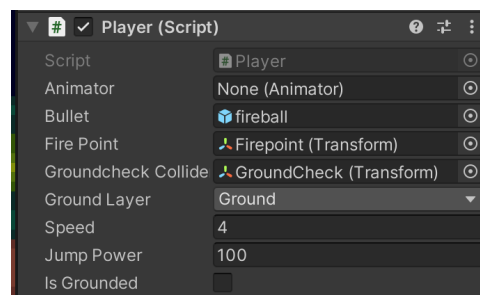
11. Pada script player ubah script dibawah ini

```
// 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;

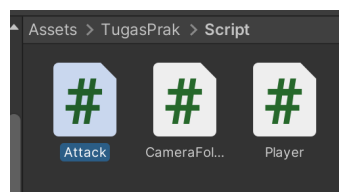
    GameObject fireball = Instantiate(bullet,
    firePoint.position, Quaternion.identity);
    fireball.GetComponent<Rigidbody2D>().velocity =
    new Vector2(direction * 10f, 0);
    Destroy(fireball, 2f);
    animator.SetBool("Attacking", false);
}
//Tambahkan pada Function Void Update
if (Input.GetKeyDown(KeyCode.C))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

12. Pada inspector player, ubah seperti dibawah ini, dimana bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak pertama



Gambar 1.11 Mengatur transisi player idle ke player attack

13. Buat Script Attack pada folder Script



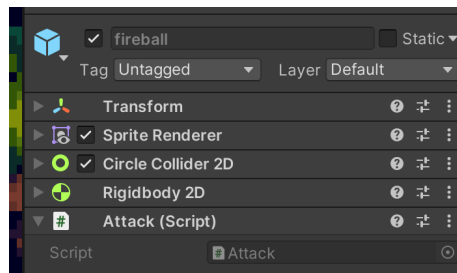
Gambar 1.12 Menambahkan script attack



14. Tambahkan script attack dibawah ini

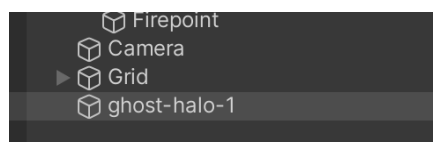
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.gameObject.CompareTag("Enemy")) {
            Destroy(gameObject);
            Destroy(collision.gameObject);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

15. Didalam folder resource tambahkan script attack di prefab fireball, dengan cara klik fireball kemudian pada menu inspector arahkan script attack kedalam inspector



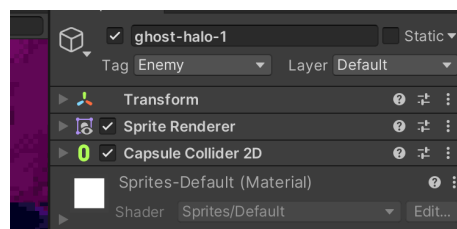
Gambar 1.13 Menambahkan script attack pada fireball

16. Tambahkan enemy ghost-halo pada hierarchy



Gambar 1.14 Mengatur enemy ghost-halo

17. Kemudian klik pada ghost-halo, lalu pada menu tab inspector tambahkan collider 2D untuk mendeteksinya. Tambahkan tag enemy, dan ubah tag ghost menjadi enemy.



Gambar 1.15 Menambahkan capsule dan mengganti tag ghost-halo



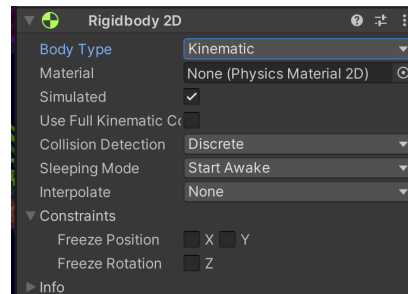
18. Tembak enemy dengan menekan tombol c untuk menghancurkan musuh



Gambar 1.16 Hasil enemy attack

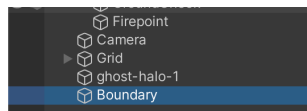
B. Membuat Enemy AI

1. Pada ghost-halo tambahkan Component Rigidbody 2D, ubah settingnya seperti berikut



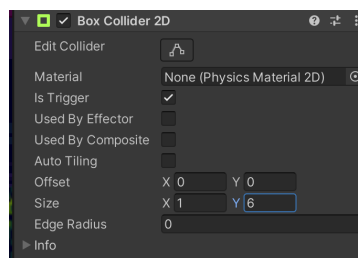
Gambar 1.17 Menambahkan rigidbody pada ghost-halo

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



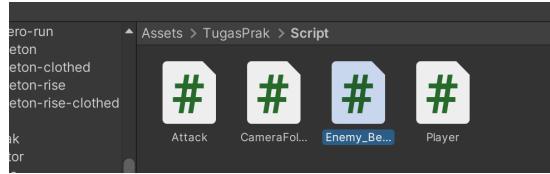
Gambar 1.18 Menambahkan boundary pada hierarchy

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



Gambar 1.19 Mengatur ukuran box collider boundary

4. Buat sebuah file script didalam folder script beri nama “Enemy_Behavior”, kemudian drag dan masukkan ke dalam game object “ghost-halo”



Gambar 1.20 Membuat script enemy behavior

5. Tambahkan script dibawah ini

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

public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 2f;
    Rigidbody2D rb;
    bool movingUp = true;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        if (movingUp)
        {
            rb.velocity = new Vector2(0f, moveSpeed);
        }
        else
        {
            rb.velocity = new Vector2(0f, -
moveSpeed);
        }
    }

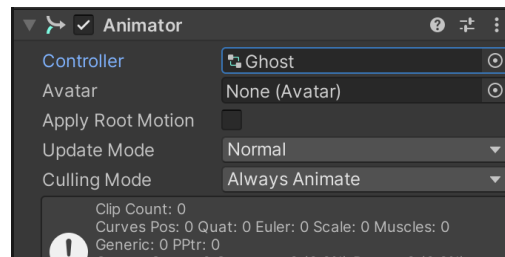
    private void OnTriggerExit2D(Collider2D
collision)
    {
        movingUp = !movingUp;
    }
}
```

6. Buat File Animator Controller pada folder Animator , ubah namanya menjadi Ghost



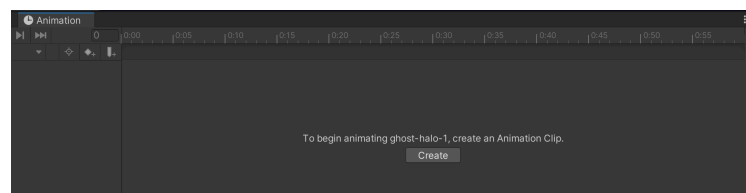
Gambar 1.21 Menambah animator controller ghost

7. Klik ghost pada hierarchy, kemudian cari component animator, pada setting controller ubah menjadi Ghost



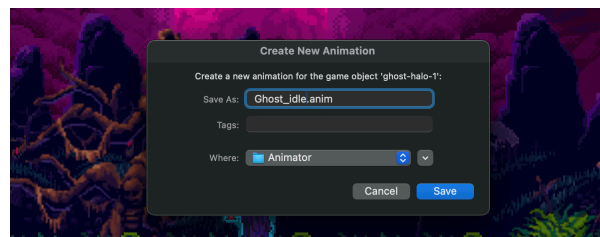
Gambar 1.22 Mengatur animator ghost pada ghost-halo

8. Untuk membuat animasi klik ghost-halo pada Hierarchy, kemudian ke menu panel Animation, pilih Create



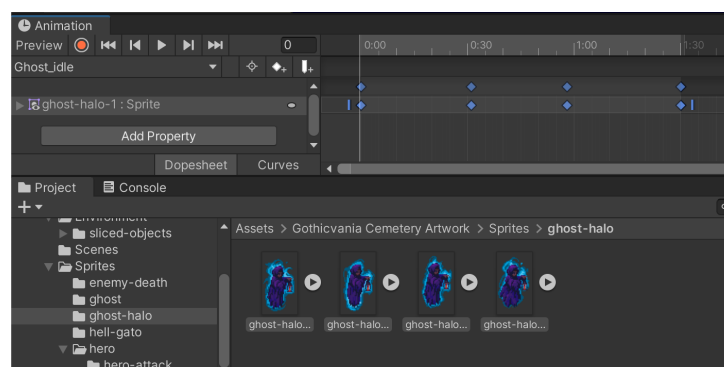
Gambar 1.23 Menambahkan animation clip ghost halo

9. Simpan pada folder Animator dan beri nama "Ghost_idle"



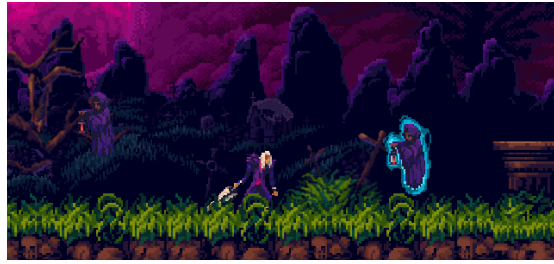
Gambar 1.24 Menyimpan animasi clip ghost-idle

10. Pada menu project buka folder ghost lalu pilih ghost-idle dan pilih gambar ghost-halo-1 sampai ghost-halo-4, kemudian drag ke tab Animation lalu geser keyframe sampai frame 1:30.



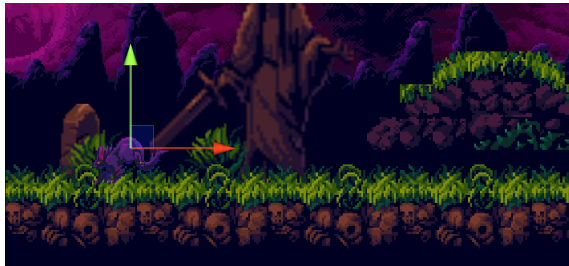
Gambar 1.25 Memasukkan asset ghost idle dan mengatur frame

11. Ulangi langkah-langkah diatas pada enemy yang lain.



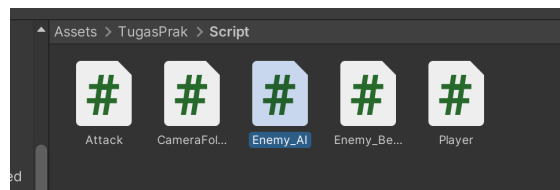
Gambar 1.26 Hasil enemy npc dengan animasi

12. Untuk membuat enemy yang menggunakan AI, tambahkan hell-gato-1 pada Hierarchy



Gambar 1.27 Menambahkan asset hell-gato

13. Buat Script Enemy_AI pada folder TugasPrak – Script



Gambar 1.28 Membuat script enemy_ai

14. Tambahkan Script dibawah ini

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

public class Enemy_AI : MonoBehaviour
{
    public float speed;
    public float lineOfSite;
    private Transform player;
    private Vector2 initialPosition;
    private bool facingRight = false;

    void Start()
    {
        player =
        GameObject.FindGameObjectWithTag("Player").transform;
        initialPosition =
        GetComponent<Transform>().position;
    }

    void Update()
```



```
{
    float distanceToPlayerX =
Mathf.Abs(player.position.x - transform.position.x);

    if (distanceToPlayerX < lineOfSite)
    {
        Vector2 targetPosition = new
Vector2(player.position.x, transform.position.y);
        transform.position =
Vector2.MoveTowards(transform.position,
targetPosition, speed * Time.deltaTime);

        if (player.position.x >
transform.position.x && !facingRight)
        {
            Flip();
        }
        else if (player.position.x <
transform.position.x && facingRight)
        {
            Flip();
        }
    }
    else
    {
        Vector2 targetPosition = new
Vector2(initialPosition.x, transform.position.y);
        transform.position =
Vector2.MoveTowards(transform.position,
targetPosition, speed * Time.deltaTime);

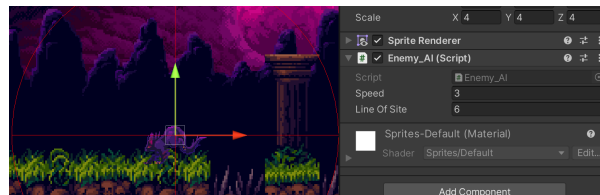
        if (initialPosition.x >
transform.position.x && !facingRight)
        {
            Flip();
        }
        else if (initialPosition.x <
transform.position.x && facingRight)
        {
            Flip();
        }
    }
}

void Flip()
{
    facingRight = !facingRight;
    Vector3 theScale = transform.localScale;
    theScale.x *= -1;
    transform.localScale = theScale;
}

private void OnDrawGizmosSelected()
{
    Gizmos.color = Color.red;
    Gizmos.DrawWireSphere(transform.position,
lineOfSite);
}
}
```

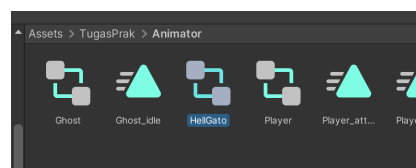


15. Tambahkan script Enemy_AI pada hell-gato-1 dan atur speed juga line of site untuk menentukan jarak dan speed pada enemy



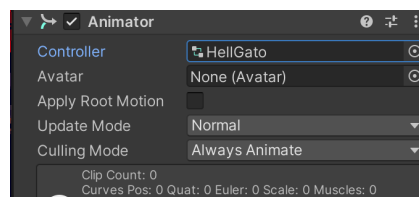
Gambar 1.29 Menambahkan script enemy_ai ke hell-gato

16. Buat File Animator Controller pada folder Animator , ubah namanya menjadi HellGato



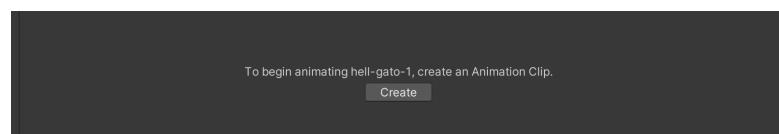
Gambar 1.30 Membuat animator controller hellgato

17. Klik hell-gato pada hierarchy, kemudian cari component animator, pada setting controller ubah menjadi HellGato



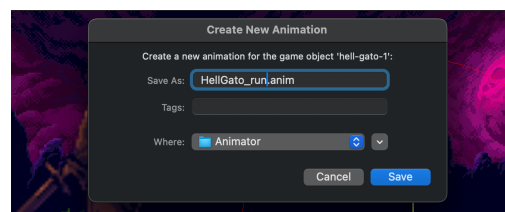
Gambar 1.31 Mengatur animator pada hellgato

18. Untuk membuat animasi klik ghost-halo pada Hierarchy, kemudian ke menu panel Animation, pilih Create



Gambar 1.32 Menambahkan animation clip hell gato

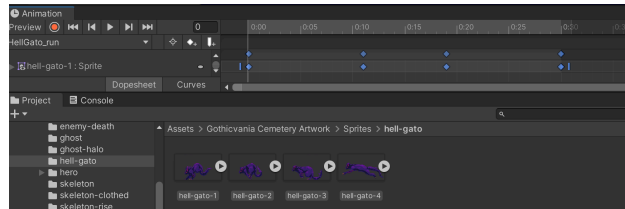
19. Simpan pada folder Animator dan beri nama “HellGato_run”



Gambar 1.33 Menyimpan animasi clip hellgato-run



20. Pada menu project buka folder ghost lalu pilih ghost-idle dan pilih gambar ghost-halo-1 sampai ghost-halo-4, kemudian drag ke tab Animation lalu geser keyframe sampai frame 1:30.



Gambar 1.34 Memasukkan asset hell-gato dan mengatur frame

21. Maka enemy akan menghampiri ketika player mendekati enemy. Lakukan juga langkah diatas pada enemy yang ingin diberikan AI.



Gambar 1.35 Hasil enemy AI

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 pada fungsi Awake() untuk mengatur posisi respawn sesuai dengan posisi awal permainan dimulai

```
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

```
//play again  
if (nyawa < 0){  
    playagain();  
}  
// player yg jatuh akan respawn  
if (transform.position.y < -10){  
    play_again = true;  
    playagain();  
}
```



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

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

5. Tambahkan file script (Enemy_Attacked.cs) dan isikan source code dibawah ini

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

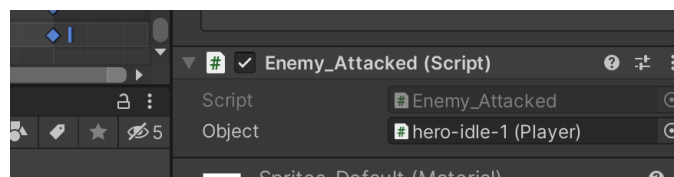
public class Enemy_attacked : MonoBehaviour
{
    [SerializeField] private Player Object;

    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;
            }
        }
    }
}
```

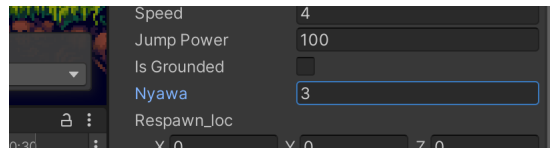
6. Pada hierarchy hell-gato-1 tambahkan script enemy attack, arahkan object pada hero-idle-1



Gambar 1.36 Menambahkan script enemy_attacked pada hell-gato



7. Klik game object player, pergi ke inspector dan ubah nilai nyawa menjadi 3 pada player(script)



Gambar 1.37 Mengatur nyawa player

8. Jika di play, player mengenai atau menyentuh hell-gato-1 maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan respawn ke titik awal



Gambar 1.38 Hasil akhir game

10.2 Kuis

```
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);
            }
        }
    }
}
```



Analisa

Pada source diatas metode Update memeriksa apakah pemain menekan tombol "Fire1", yang akan memicu metode PerformMeleeAttack. Metode ini menggunakan raycast dari posisi pemain ke arah depan untuk mendeteksi objek yang terkena dalam jarak serangan yang ditentukan. Jika raycast mengenai suatu objek, skrip akan memeriksa apakah objek tersebut memiliki komponen EnemyHealth. Jika ada, skrip akan memanggil metode TakeDamage pada komponen EnemyHealth, yang mengurangi health musuh sebesar nilai kerusakan serangan yang ditentukan.