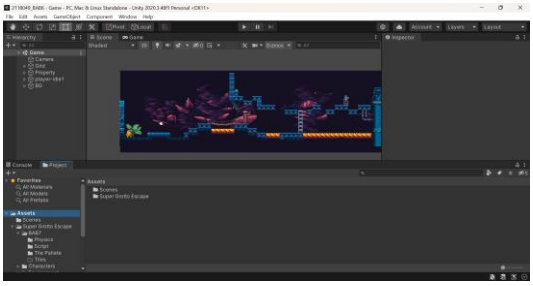
TUGAS PERTEMUAN 10

Respawn and AI Enemy Attack

NIM	:	2118049
Nama	:	Siti Aisyah
Kelas	:	B
Asisten Lab	:	Difa Fisabilillah (2118052)

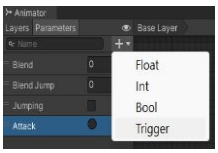
10.1 Tugas 1 : Membuat Mekanisme Enemy Attack

1. Buka *project* Unity sebelumnya untuk melanjutkan tahap pembuatan AI *enemy attack* dan *respawn*.



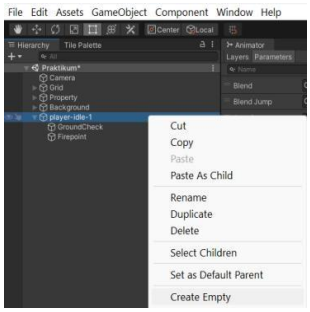
Gambar 10.1 Tampilan *Project*

2. Kemudian pada menu Tab Animator, tambahkan parameter *trigger* dan *rename* menjadi *Attack*.



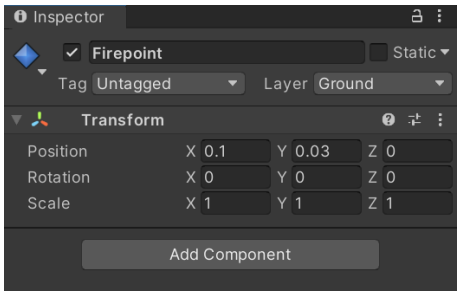
Gambar 10.2 Tampilan Menambahkan Parameter

3. Selanjutnya, buat Layer *Game object* baru didalam *player-idle-1* dengan cara klik kanan, lalu pilih *Create Empty* dan *rename* menjadi *Firepoint*.



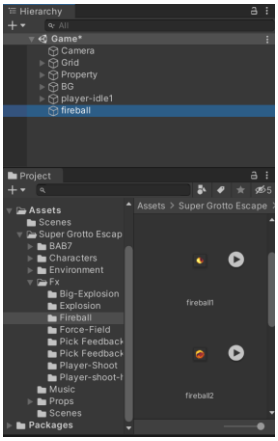
Gambar 10.3 Tampilan Add Firepoint

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



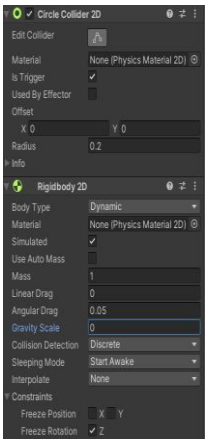
Gambar 10.4 Tampilan Mengatur *Icon*

5. Pada menu *Hierarchy*, tambahkan fireball2 dari folder Sprites, lalu folder Fx, kemudian fireball2 dengan *drag and drop* dan *rename* menjadi *fireball*.



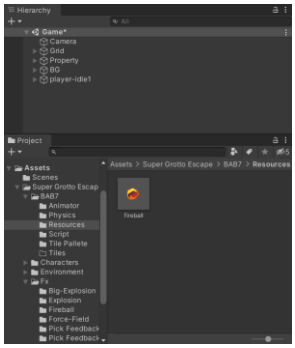
Gambar 10.5 Menambahkan *Fireball*

6. Klik *fireball* untuk menambahkan komponen *Circle Collider 2D* dan Riggidbody 2D, lalu *setting* sesuai gambar dibawah ini.



Gambar 10.6 Add Circle Collider Dan Rigidbody

7. Buat Folder baru *Resources* di dalam folder BAB7 pada menu *Project*, kemudian *drag and drop fireball* ke dalam folder *Resources* dan hapus *fireball* pada *Hierarchy*

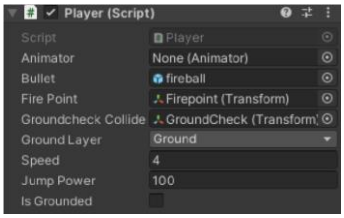


Gambar 10.7 Buat Folder Resources

8. Pada *script* Player tambahkan *Script* berikut.

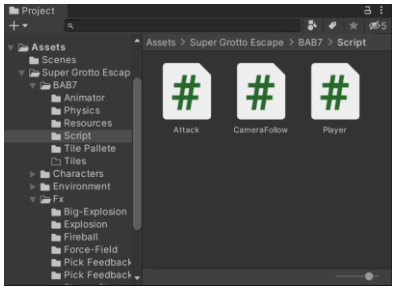
```
//Pada class Player
// Deklarasi variable
Public Animator animator;
Public GameObject bullet;
Public Transform Firepoint;

//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 objek yang akan ditembak, sedangkan *fire point* adalah titik tembak pertama.



Gambar 10.8 *Add Bullet* Objek

10. Buat *script attack* pada folder Script.

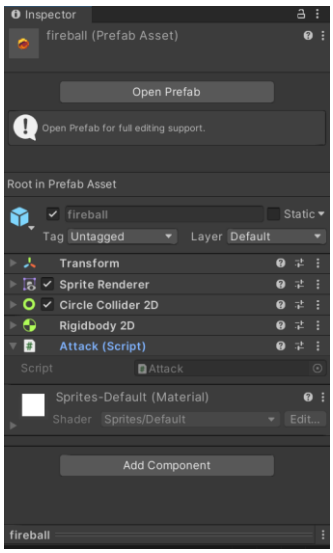


Gambar 10.9 Membuat Script Attack

11. Tambahkan Script Attack dibawah ini.

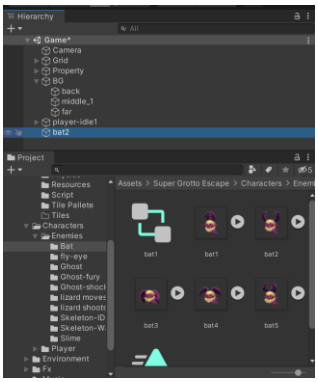
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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

12. Didalam folder *resources*, tambahkan *Script Attack* di *Prefab fireball*, dengan cara klik *fireball*, kemudian pada menu Inspector arahkan *Script Attack* kedalam Inspector.



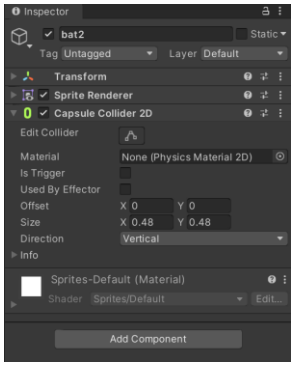
Gambar 10.10 *Add Script Attack*

13. Tambahkan Enemy Bat pada *hierarchy* dari folder Characters, lalu Enemies, kemudian bat2 dengan cara *drag and drop.*



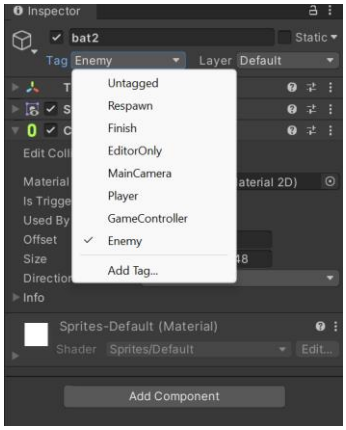
Gambar 10.11 *Add Enemy Bat*

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



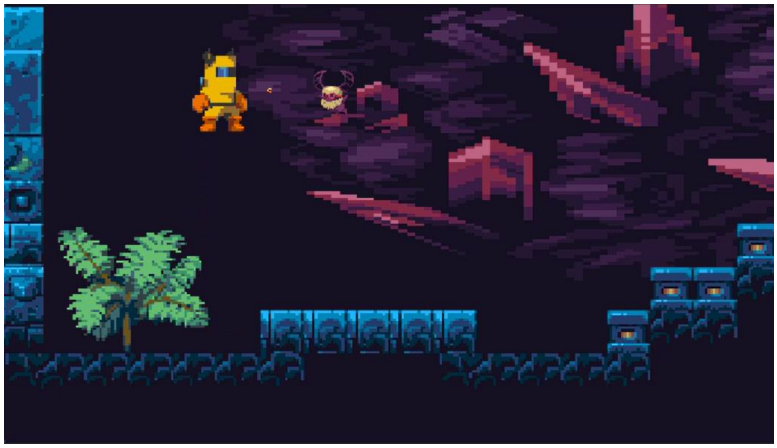
Gambar 10.12 *Add Capsule Collider* 2D

15. Tambahkan Tag *Enemy* dengan cara Pilih *Add Tag*, kemudian add tag to the list dan tuliskan *enemy.*



Gambar 10.13 *Add Tag Enemy*

16. Tembak *enemy* dengan menekan tombol C untuk menghancurkan musuh.

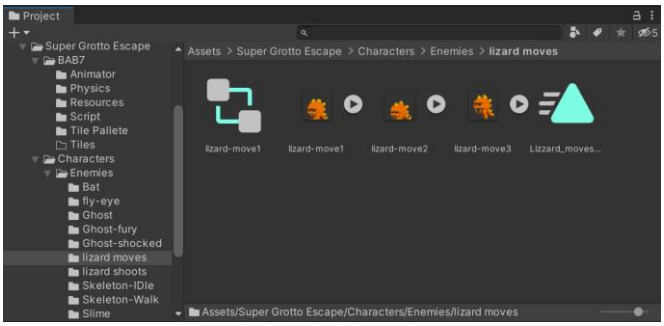


Gambar 10.14 Tampilan Menembak Lawan

10.2 Tugas 2 : Membuat Enemy AI

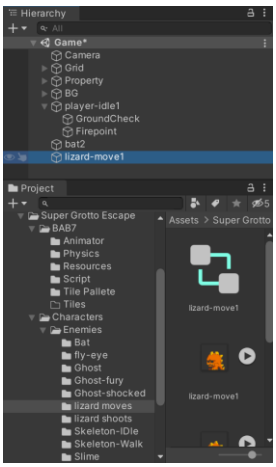
A. Enemy Behavior NPC

1. Cari sebuah *sprite pack* bernama *enemy* dan buka folder bernama "lizard moves".



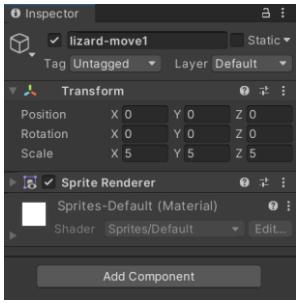
Gambar 10.15 Tampilan Enemy

2. Tambahkan "lizard-move1" ke *Hierarchy*.



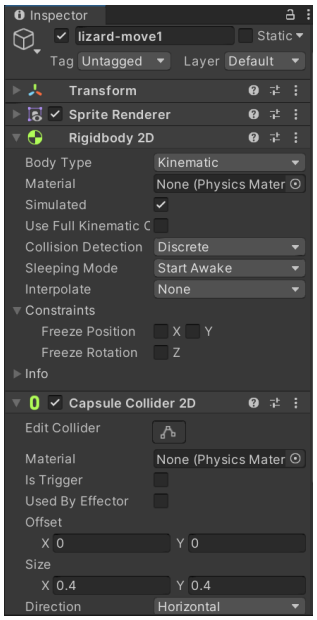
Gambar 10.16 *Add Lizard Move* 1

3. Pada *inspector* atur *transform scale* menjadi seperti berikut.



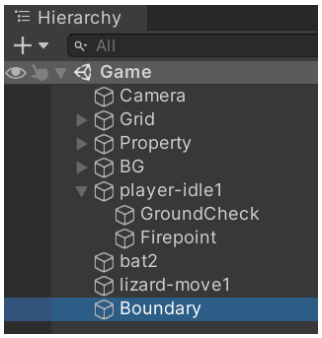
Gambar 10.17 *Setting Transform Scale*

4. Tambahkan sebuah komponen bernama *Capsule Collider* 2D dan Rigidbody 2D dalam *inspector game object* lizard-move1. Lalu atur seperti gambar berikut.



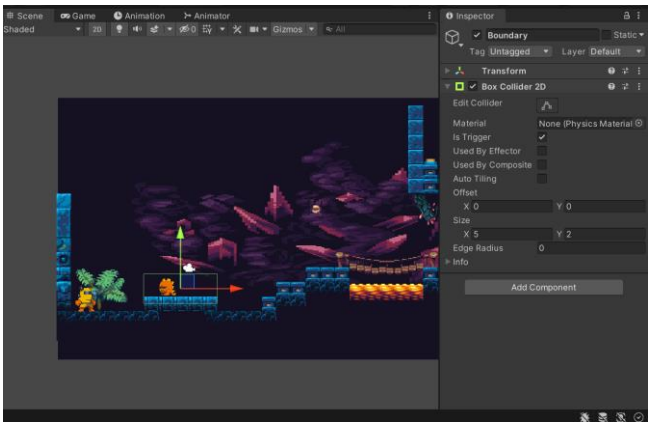
Gambar 10.18 *Add Component*

5. *Create Empty object* pada Hierarchy dan *rename* menjadi *Boundary*.



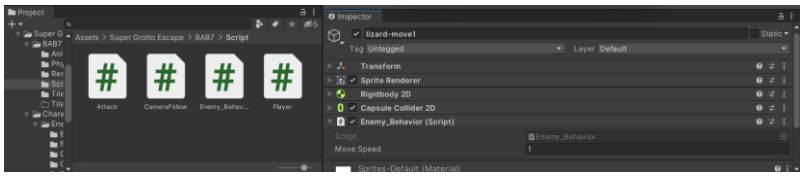
Gambar 10.19 *Create Empty Object*

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



Gambar 10.20 *Add Box Collider*

7. Buat sebuah file *script* didalam folder Script, lalu beri nama "Enemy_Behavior", kemudian *drag* dan masukkan ke dalam *game object* "lizard-move1"



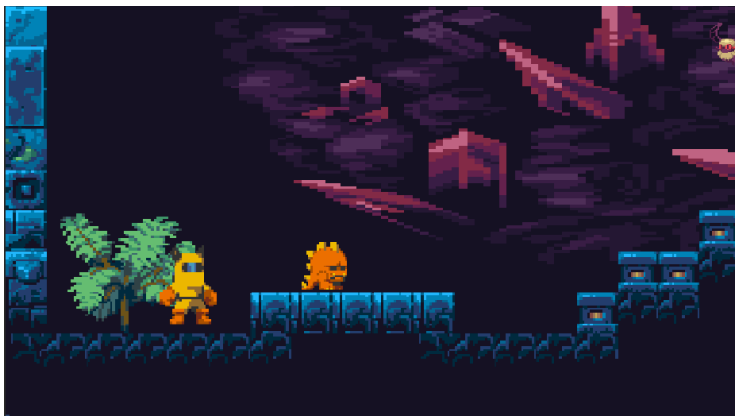
Gambar 10.21 Membuat Script Enemy Idle 25.

8. Tambahkan Script berikut di *script* Enemy_Behavior.

```
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
{
```

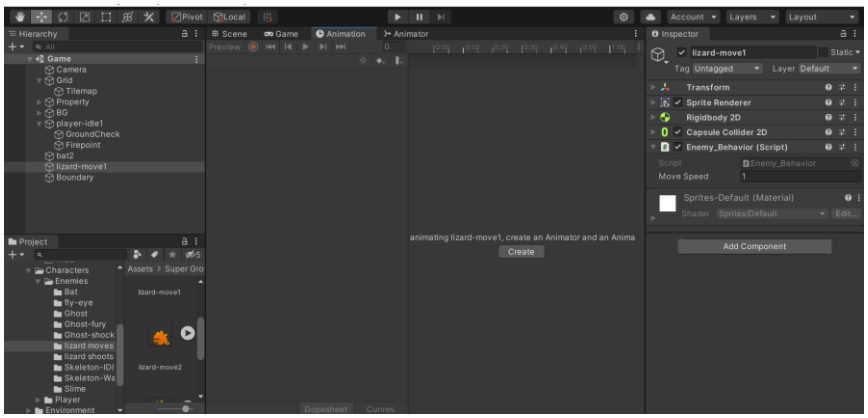
```
            rb.velocity = new Vector2(-moveSpeed,
0f);
        }

}
    private bool isFacingRight()
    {
        return transform.localScale.x >
Mathf.Epsilon;

}
    private void OnTriggerExit2D(Collider2D
collision)
    {
        transform.localScale = new Vector2(-
transform.localScale.x, transform.localScale.y);
    }
}
```

9. Jalankan Program



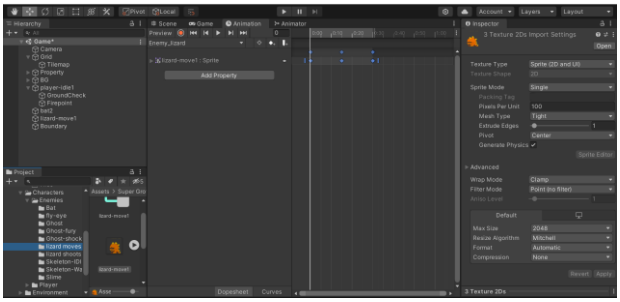
Gambar 10.22 Hasil Tampilan Lawan

10. Tambahkan animasi pada enemy_lizard dengan cara klik lizard-move1 lalu ke *tab animation*, kemudian *create* dan simpan animasi pada folder animator.



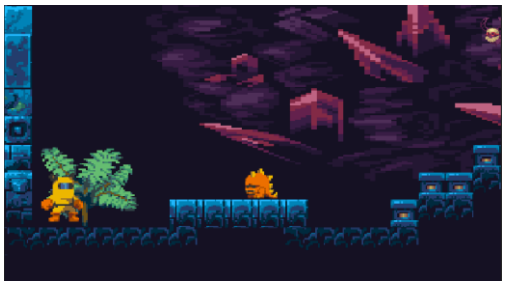
Gambar 10.23 Menambahkan Animasi

11. Lalu *drag and drop* animasi *enemy* ke tab animasi dan atur frame ke 0:30.



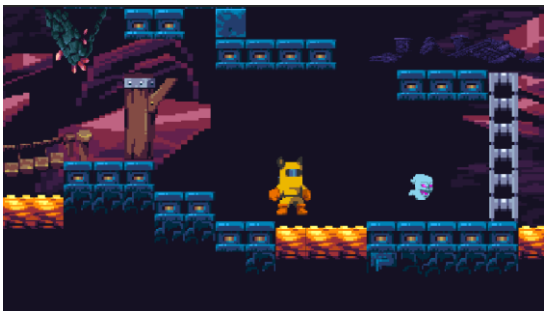
Gambar 10.24 *Drag and Drop* Animasi dan Mengatur *Frame*

12. Maka ketika di *play*, *enemy* akan memiliki animasi.



Gambar 10.25 Hasil Tampilan Animasi Lawan

13. Tambahkan *enemy* lain dengan mengulangi langkah 1 sampai 11, maka tampilan akan seperti berikut.



Gambar 10.26 Menambahkan Lawan

B. Enemy AI

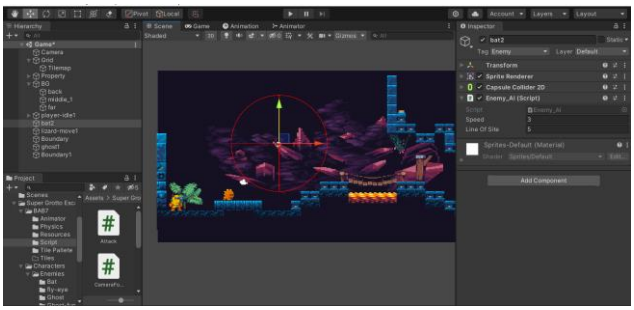
1. Buat Script Enemy_AI pada folder *Script*.

Gambar 10.27 Membuat Script AI

2. Tambahkan *script* berikut pada *script* Enemy_AI.

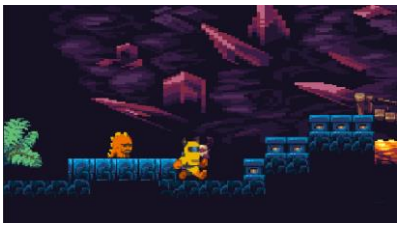
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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);
    }
}
```

3. *Drag and drop script* Enemy_AI ke dalam GameObject bat2, lalu pada *inspector* Enemy_AI, atur *speed* dan *Line of Site* untuk menentukan jarak dan *speed* pada *enemy*.



Gambar 10.28 Mengatur Speed Line

4. *Play game*, maka bat akan mengikuti gerakan player.



Gambar 10.29 Hasil Tampilan Game

5. Tambahkan animasi pada bat, maka akan seperti dibawah ini**.**

Gambar 10.30 Menambahkan Animasi *Bat*

10.3 Tugas 3 : Respawn

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

```
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 berikut di dalam void *update* Player.cs agar ketika nyawa *player* dibawah 0 atau *player* jatuh, maka akan melakukan *respawn*.

```
if (nyawa < 0)
{
  playagain();
}
if (transform.position.y < -10)
{
  play_again = true;
  playagain();
}
```

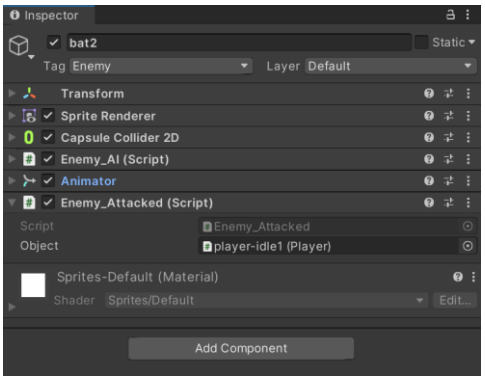
4. Tambahkan fungsi playagain() dalam *script* Player.cs.

```
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* berikut.

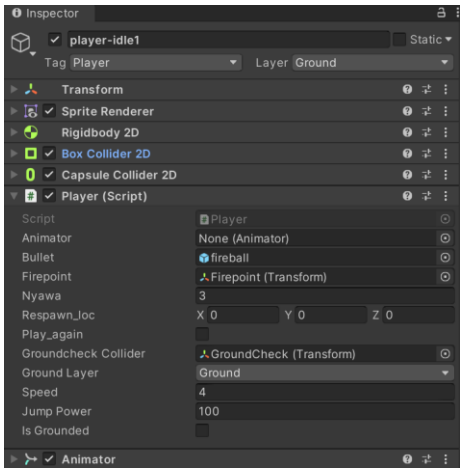
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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* bat2, tambahkan Script *enemy attack*, arahkan *object* pada player-idle-1.



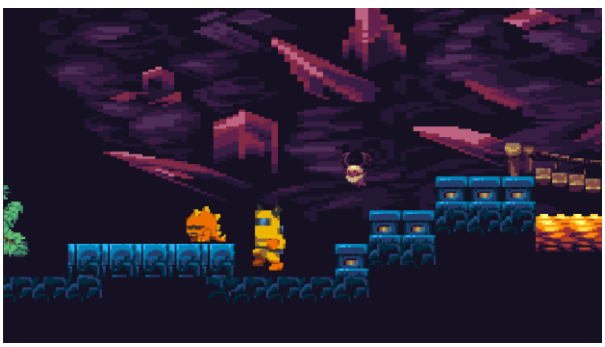
Gambar 10.31 *Add Script Enemy Attacked*

7. Klik *game object Player*, pergi ke *Inspector* dan ubah nilai nyawa menjadi 3 pada Player (Script).



Gambar 10.32 Mengatur Nyawa

8. Jika di play, apabila *player* mengenai atau menyentuh *enemy* sebanyak 3 kali, maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0, maka akan respawn ke titik awal.



Gambar 10.33 Hasil Akhir Tampilan

10.4 KUIS

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    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 dan kekurangan yang perlu diperbaiki. Pertama, tipe data untuk variabel `attackRange` harus diubah dari `int` menjadi `float`, dan penulisannya harus diperbaiki dari `atackRange` menjadi `attackRange` untuk konsistensi. Selanjutnya, variabel `attacDamage` harus diperbaiki penulisannya menjadi `attackDamage`. Kemudian, pada fungsi update terdapat metode `InputGetButtonDown` yang harus diperbaiki menjadi `Input.GetButtonDown` untuk mengikuti konvensi penamaan metode di Unity. Terakhir, di dalam fungsi `PerformMeleeAttack`, perlu ditambahkan kode untuk mengecek apakah objek yang terkena serangan

memiliki komponen `EnemyHealth`. Jika ya, maka panggil metode `TakeDamage` pada komponen tersebut untuk mengurangi *health* musuh.

10.5 Link Github

https://github.com/saasyh/2118049_PRAK_ANIGAME