



## TUGAS PERTEMUAN: 10

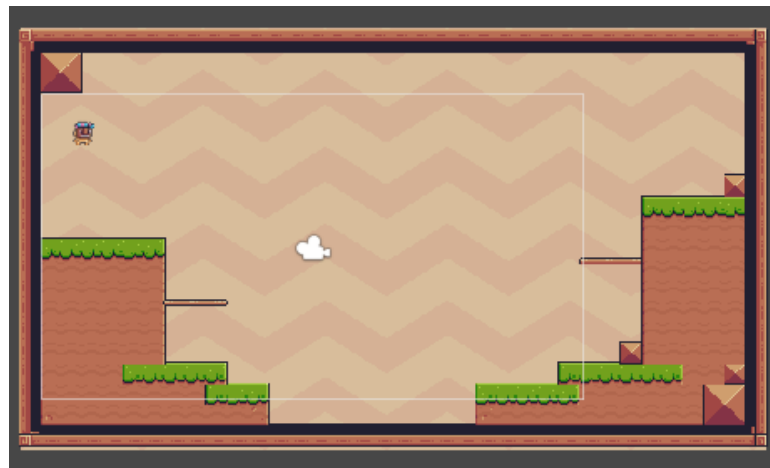
### RESPAWN AND AI ENEMY ATTACK

NIM	:	2118027
Nama	:	RAFID ARTUR PRASETYA
Kelas	:	C
Asisten Lab	:	APRILLIA DWI DYAH S. (2118143)

#### 1.1 Tugas 1 : Membuat Respawn dan AI Enemy Attack

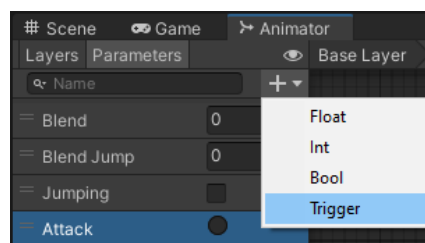
##### A. Membuat Mekanisme Attack

1. Buka kembali project pada bab 9.



Membuka Project Bab 9

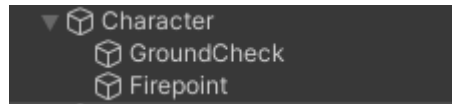
2. Buat tipe data baru yaitu Trigger kemudian rename menjadi 'Attack'.



Membuat Tipe Data Attack

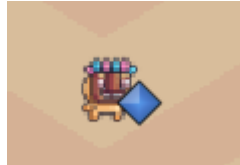


3. Create Empty pada Character, kemudian rename menjadi 'Firepoint'.



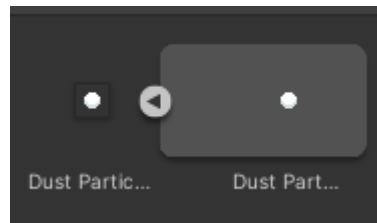
#### Membuat Layer Firepoint

4. Pada menu Hirarki, klik pada Firepoint untuk setting pada Inspector, ubah Icon menjadi titik, atur letak titik di depan player.



#### Menambahkan Icon di Depan Karakter

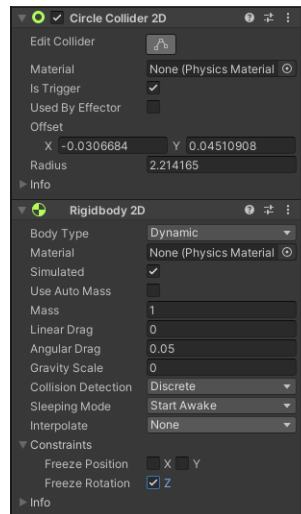
5. Pada menu Hirarki tambahkan energyball yang ada di folder Sprites.



#### Menambahkan Layer Fireball

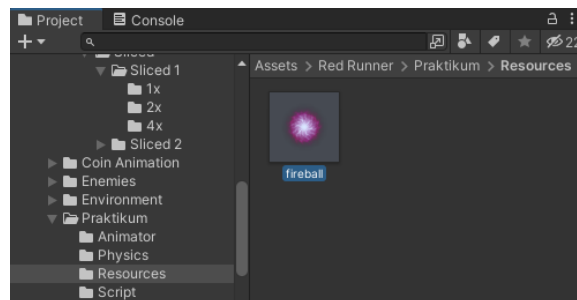


- Klik fireball dan tambahkan komponen Circle Collider 2D kemudian centang pada bagian Is Trigger dan atur radiusnya sesuai gambar. Tambahkan juga komponen Rigidbody 2D, ubah Gravity Scale menjadi 0 dan centang pada sumbu Z di Freeze Rotation.



#### Setting Inspector pada Fireball

- Buat folder baru dan beri nama Resources, lalu drag and drop fireball ke dalam folder Resources, dan hapus fireball pada Hirarki.



#### Tampilan Resources Folder

- Tambahkan script pada class player di Script Player.

```
public GameObject bullet;  
public Transform firePoint;
```

- Tambahkan script di bawah ini di bawah 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);  
}
```

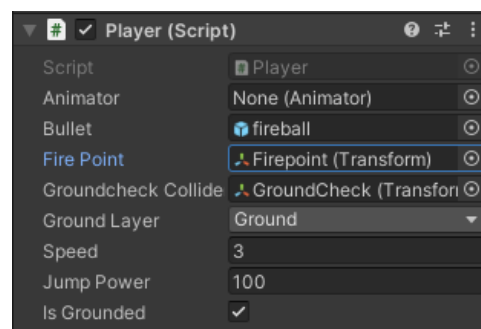


```
Destroy(fireball, 2f);  
}
```

10. Pada script Player tambahkan script pada Function Void Update.

```
if (Input.GetKeyDown(KeyCode.C))  
{  
    StartCoroutine(Attack());  
}
```

11. Pada Inspector Player, ubah seperti di bawah ini. Dimana Bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak yang pertama.



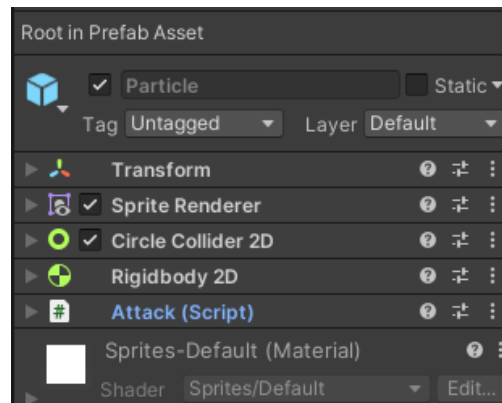
Menambahkan Object pada Buller dan Fire Point

12. Buka script Attack pada folder Script.

```
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);  
        }  
    }  
}
```

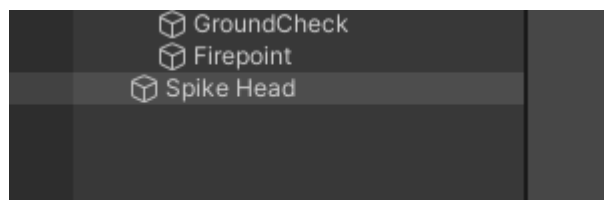


14. Di dalam folder resource. Tambahkan script Attack di Prefab fireball, dengan cara Klik fireball kemudian pada menu Inspector arahkan Script Attack ke dalam Inspector.



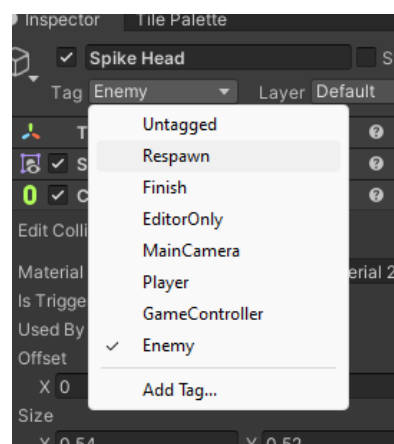
Menambahkan Script pada Resource Fireball

15. Tambahkan Enemy Spike Head pada Hirarki.



Menambahkan Enemy

16. Kemudian klik pada Spike Head, lalu pada menu tab inspector tambahkan capsule collider 2D untuk mendeteksinya.



Menambahkan Tag Enemy



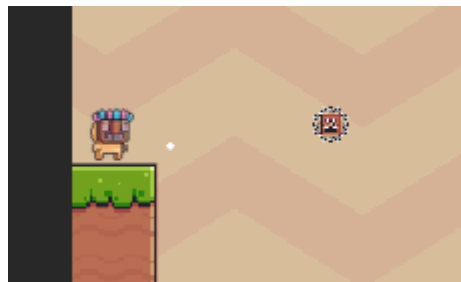
17. Untuk membuat tembakan fireball sesuai dengan arah pergerakan dari hadap player, perlu ditambahkan script di dalam function Awake.

```
private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
    animator = GetComponent<Animator>();
    facingRight = true;
}
```

18. Kemudian ubah script pada function IEnumerator

```
IEnumerator Attack()
{
    animator.SetTrigger("Attack");
    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);
}
```

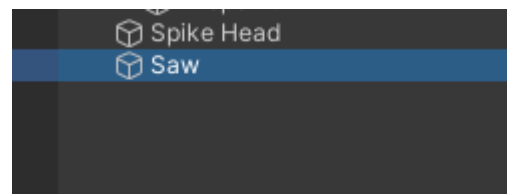
19. Tembak Enemy dengan menekan tombol C untuk menghancurkan musuh.



Hasil Tampilan Menembak Enemy

## B. Enemy Behavior NPC

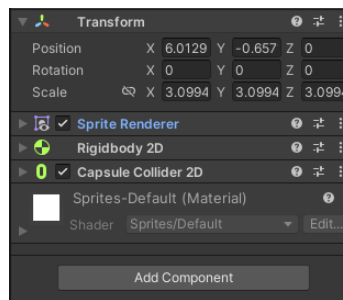
1. Cari sprites di folder Enemies, png, 64x64 kemudian tambahkan "Saw" ke Hirarki.



Menambahkan Enemy

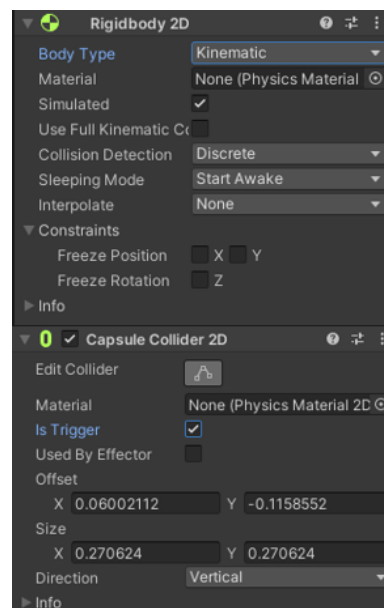


2. Klik pada layer Saw kemudian pergi ke menu Inspector. Tambahkan komponen bernama Capsule Collider 2D dan Rigidbody 2D.



Menambahkan Komponen

3. Pada collider 2D centang pada bagian Is Trigger dan pada Rigidbody 2D ganti menjadi Kinematic pada Body Type.



Atur Capsule Collider 2D dan Rigidbody 2D

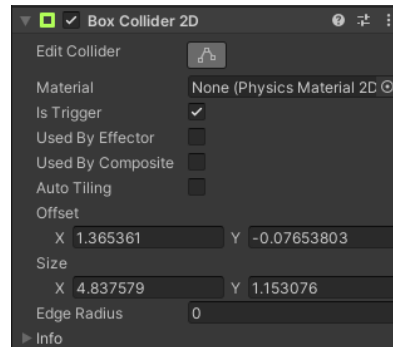
4. Create Empty object pada Hirarki, rename menjadi Boundary.



Membuat Boundary

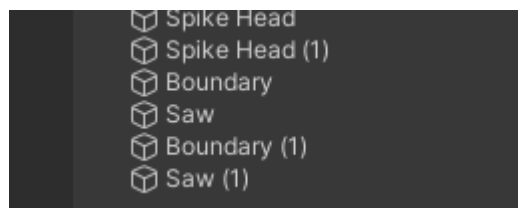


5. Tambahkan Box Collider 2D pada Boundary, centang pada Is Trigger.



Menambah Box Collider 2D pada Boundary

6. Buat file script di dalam folder Scrip dan beri nama menjadi 'EnemyBehavior'. Kemudian drag and drop ke dalam layer 'Saw'.

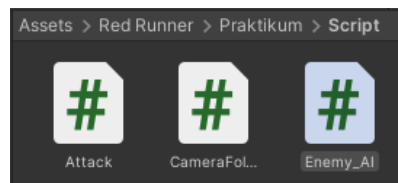


Memasukkan Script Enemy\_Behavior

7. Ulangi langkah ke-1 hingga ke 8 untuk Enemy yang lain.

### C. Membuat Enemy AI

1. Membuat file script dengan nama Enemy\_AI.



Membuat File Script Enemy\_AI

2. Klik 2 kali pada file script Enemy\_AI dan masukkan source code di bawah 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);
}
}
```

3. Drag and drop file script ke objek Spike Head, kemudian atur juga untuk line of site dan juga speed-nya.



Drag and Drop File Scrip ke Objek Spike Head



4. Tampilkan maka hasilnya seperti berikut, Spike Head akan mengikuti pergerakan dari player.



Hasil Running

#### D. Respawn

1. Tambahkan source code berikut pada Player.cs.

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

2. Untuk mengatur posisi respawn sesuai dengan posisi awal permainan dimulai yaitu dengan menambahkan source code berikut di dalam fungsi void Awake().

```
respawn_loc = transform.position;
```

3. Tambahkan kode di bawah ini dalam fungsi void Update, agar jika nyawa player di bawah 0 maka akan melakukan respawn.

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

4. Masukkan source code berikut di bawah code sebelumnya agar ketika player jatuh di bawah platform maka akan melakukan respawn.

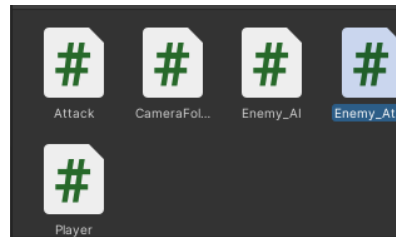
```
if (transform.position.y < -10)  
{  
    play_again = true;  
    playagain();  
}
```

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

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



6. Buat file script baru dan beri nama 'Enemy\_Attacked'.



### Membuat File Script Enemy\_Attacked

7. Kemudian klik 2 kali pada file script Enemy\_Attacked, dan masukkan source code di bawah ini.

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

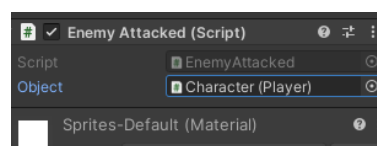
public class Enemy_attacked : MonoBehaviour
{
    [SerializeField] private Player Object;

    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;
            }
        }
    }
}
```

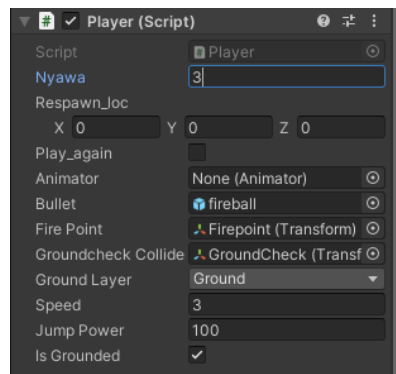
8. Drag and drop file script pada objek Spike Head dan ubah objek ke player.



### Mengubah Objek ke Player



9. Pada objek player, pergi ke menu Inspector dan di Player (Script) ubah nyawanya menjadi 3.



Mengubah Nyawa Player

10. Jalankan hasilnya seperti pada gambar berikut.



Hasil Running



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))
        {
            // Cek apakah objek yang terkena adalah musuh
            EnemyHealth enemyHealth =
            hit.collider.GetComponent<EnemyHealth>();
            if (enemyHealth != null)
            {
                // Kurangi health musuh
                enemyHealth.TakeDamage(attackDamage);
            }
        }
    }
}

// Skrip untuk mengelola health musuh
public class EnemyHealth : MonoBehaviour
{
    public int health = 100;

    public void TakeDamage(int damage)
    {
        health -= damage;
        if (health <= 0)
        {
            Die();
        }
    }

    void Die()
    {
        Destroy(gameObject);
    }
}
```



#### Analisa :

Dalam game, script di atas mengatur serangan melee pemain dan kesehatan musuh. "PlayerAttack" memeriksa input pemain di setiap frame untuk mengetahui ketika tombol serangan ditekan. Kemudian, menggunakan "Physics.Raycast" untuk menentukan apakah serangan mengenai musuh dalam jarak tertentu (attackRange). Nilai kesehatan (kesehatan) musuh akan dikurangi sebesar kerusakan serangan oleh komponen "EnemyHealth" musuh. Jika kesehatan musuh kurang dari nol, musuh akan dihancurkan dengan metode "Die()", yang menghancurkan game object musuh.