



TUGAS PERTEMUAN: 10

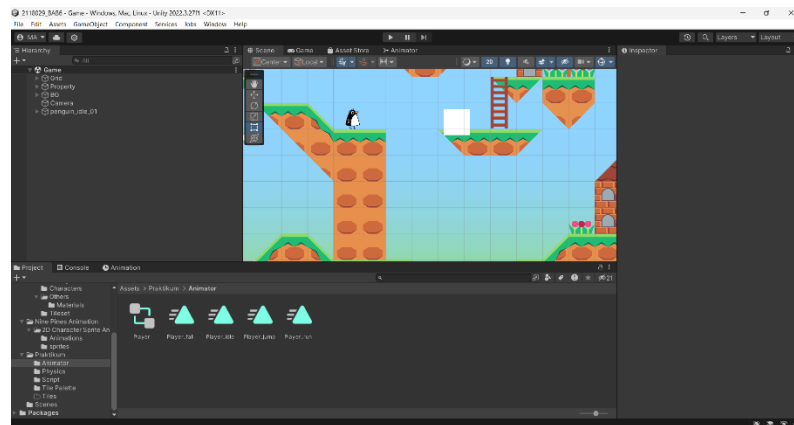
RESPAWN AND AI ENEMY ATTACK

NIM	:	2118029
Nama	:	Mohammad Arvin Manda Aradhana
Kelas	:	A
Asisten Lab	:	APRILLIA DWI DYAH S. (2118143)

1.1 Tugas 1 : Membuat Respawn And AI Enemy Attack

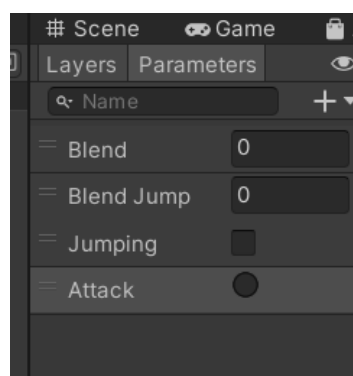
A. Mempersiapkan Mekanisme Menyerang

1. Buka project Unity sebelumnya yang telah ditambahkan game animation.



Gambar 10.1 Membuka File Project

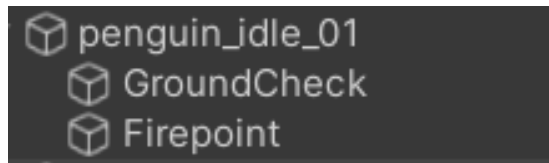
2. Buka tab Animator, tambahkan parameter Trigger dengan nama Attack.



Gambar 10.2 Menambah Parameter

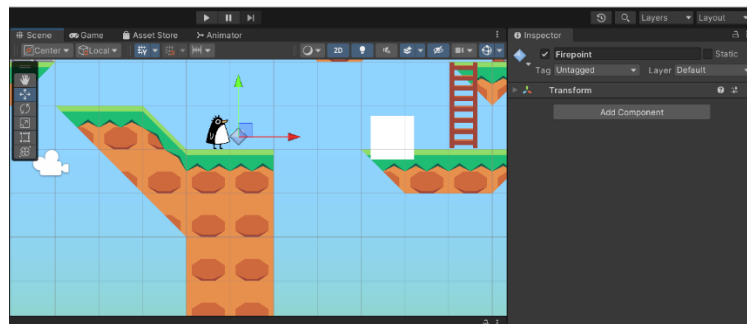


3. Berikutnya, tambahkan game object baru dalam layer penguin_idle_01. Klik kanan, pilih Create Empty, lalu ubah namanya menjadi Firepoint.



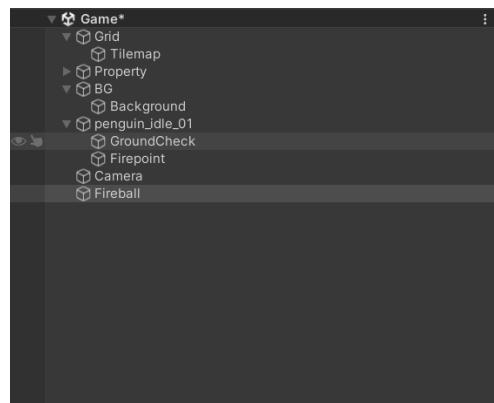
Gambar 10.3 Membuat Game Object Baru

4. Selanjutnya, atur ikon Firepoint menjadi titik melalui tab Inspector dan posisikan di depan pemain.



Gambar 10.4 Memindahkan Firepoint

5. Berikutnya, tambahkan satu item sebagai peluru dalam folder Enemies. Pada folder charged, pilih charged4 dan ganti namanya menjadi fireball.



Gambar 6.5 Menambahkan tembakan

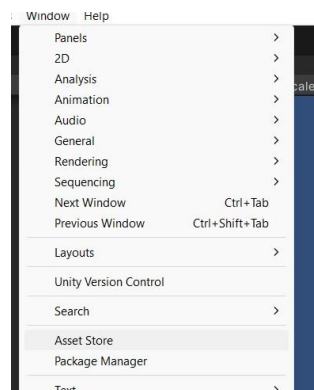


6. Selanjutnya, tambahkan komponen Circle Collider 2D dan Rigidbody 2D, dan atur propertinya seperti berikut.



Gambar 10.6 Menambahkan Komponen

7. Buat folder baru bernama Resources di dalam folder project. Selanjutnya, seret dan lepaskan fireball ke dalam folder Resources, dan hapus fireball dari Hierarchy.



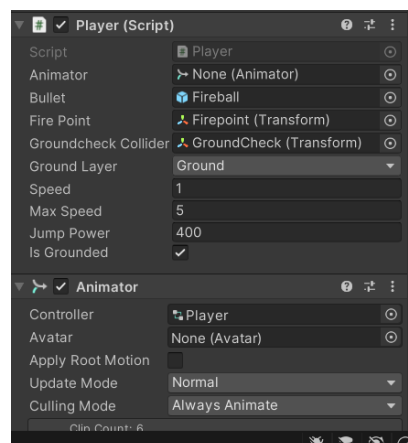
Gambar 10.7 Membuat Folder Resources



8. Pada script Player tambahkan script berikut.

```
public Animator animator;  
public GameObject bullet;  
public Transform firePoint;  
IEnumerator Attack()  
{  
    animator.SetTrigger("Attack");  
    yield return new WaitForSeconds(0.25f);  
  
    // Menentukan arah peluru berdasarkan arah karakter  
    float direction = facingRight ? 1f : -1f;  
  
    // Membuat peluru baru dan mengatur arahnya  
    GameObject fireball = Instantiate(bullet,  
    firePoint.position, Quaternion.identity);  
    fireball.GetComponent<Rigidbody2D>().velocity = new  
    Vector2(direction * 10f, 0);  
  
    // Membalikkan skala peluru jika menghadap ke kiri  
    if (!facingRight)  
    {  
        Vector3 theScale =  
        fireball.transform.localScale;  
        theScale.x *= -1;  
        fireball.transform.localScale = theScale;  
    }  
  
    Destroy(fireball, 2f);  
}
```

9. Di tab Inspector Player, ubah Bullet dengan objek yang akan ditembak, sementara Fire Point menjadi titik tembak pertama.



Gambar 10.8 Mengatur Player

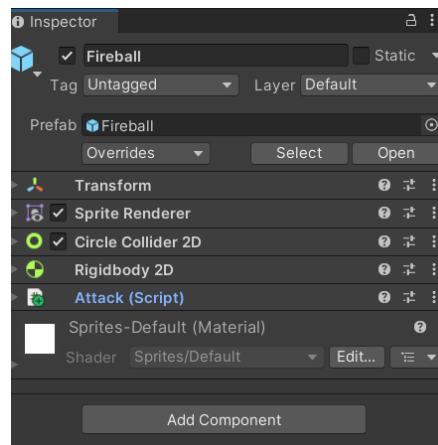


10. Buat script Attack pada folder Script dan tambahkan script berikut.

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

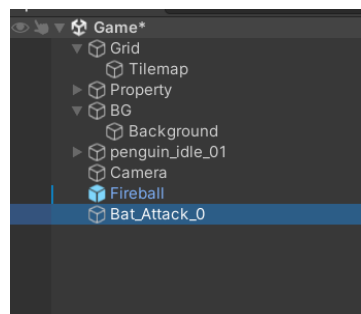
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```

11. Di dalam folder Resources, tambahkan script Attack ke dalam Prefab fireball. Caranya adalah dengan mengklik fireball, kemudian di tab Inspector, arahkan script Attack ke dalam Inspector.



Gambar 10.9 Menambahkan Script Attack

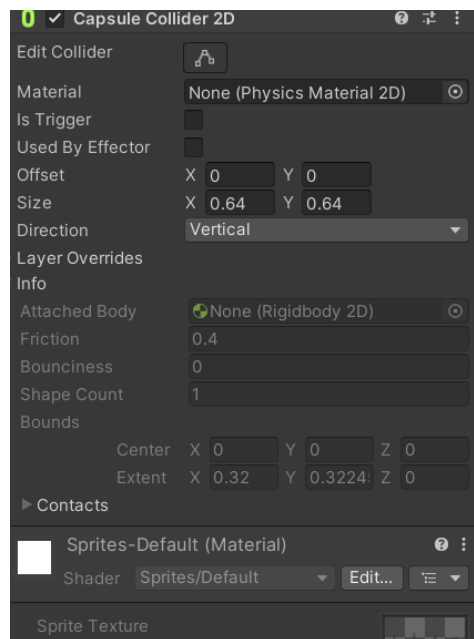
12. Masukkan Enemy Bat_attack_0 ke dalam hierarki di folder Sprites, Bat_attack_0.



Gambar 10.9 Menambahkan Enemy

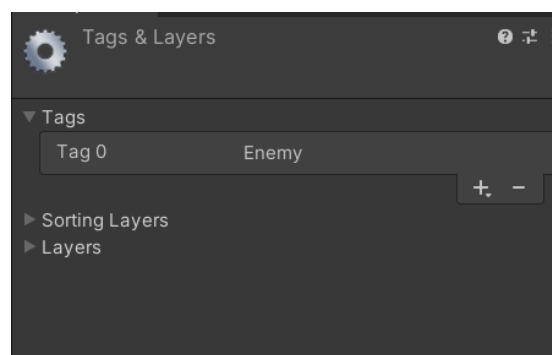


13. Setelah itu, klik pada eagle-attack, dan di tab Inspector, tambahkan Collider 2D untuk mendeteksinya.



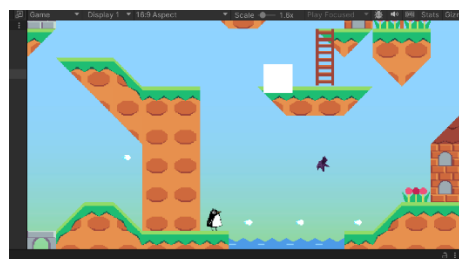
Gambar 10.10 Menambahkan capsule colider

14. Tambahkan tag Enemy dengan cara pilih Add Tag, lalu tambahkan tag ke dalam daftar. Tuliskan "Enemy".



Gambar 10.11 Menambahkan tag enemy

15. Hancurkan musuh dengan menembaknya menggunakan tombol C.



Gambar 10.12 karakter menjadi menembak



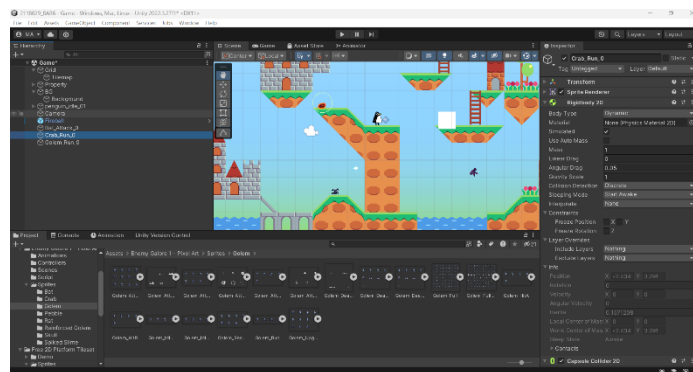
B. Membuat Enemy Behavior

1. Masukkan sprite Run ke dalam hirarki dari folder Enemies, terlebih dahulu pada crab, dan kemudian pada folder golem.



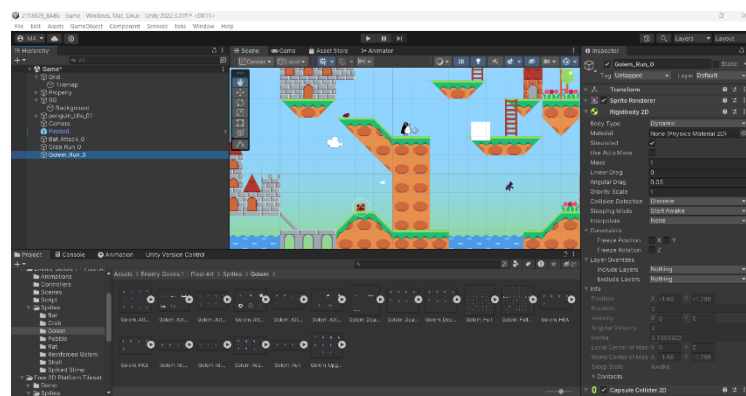
Gambar 10.13 Menambahkan Sprite

2. Tambahkan komponen Capsule Collider 2D dan Rigidbody 2D pada kedua musuh tersebut. Selanjutnya, atur propertinya pada musuh 1 sebagai berikut.



Gambar 10.14 Pengaturan Enemy 1

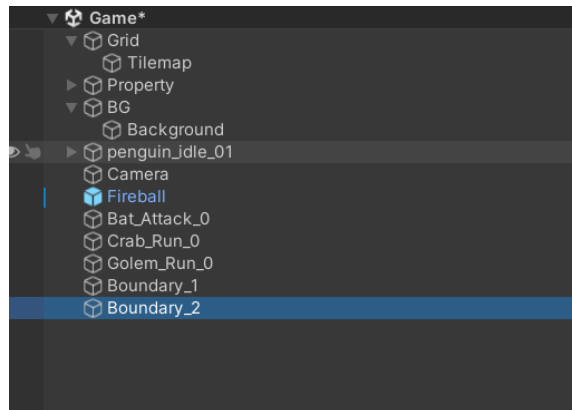
3. Kemudian untuk enemy 2 sebagai berikut.



Gambar 10.15 Pengaturan Enemy 2

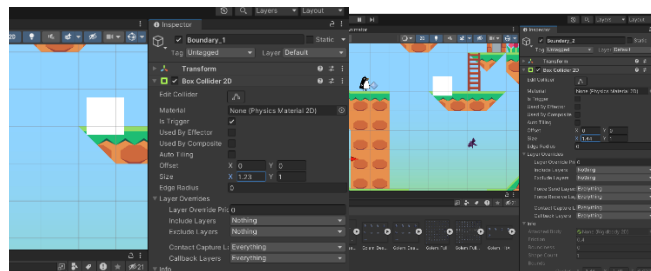


4. Buatlah dua objek kosong pada hirarki dan beri nama Boundary_1 dan Boundary_2.



Gambar 10.16 Membuat Empty Object

5. Sertakan Box Collider 2D pada masing-masing Boundary, aktifkan Is Trigger, dan atur size serta offset sesuai dengan batas pergerakan musuh. Lalu, rotasikan musuh sebesar 180 derajat pada sumbu Y.



Gambar 10.17 Menambahkan Box Collider

6. Lalu buat sebuah script baru dengan nama Enemy_Behavior dan tambahkan script berikut.

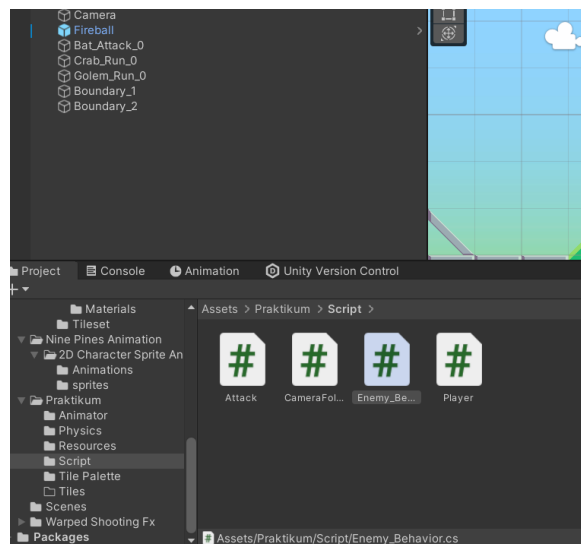
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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,
```



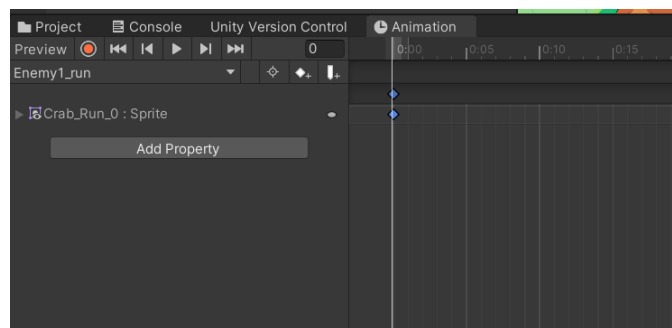

```
Of);  
}  
}  
private bool isFacingRight()  
{  
    return transform.localScale.x >  
    Mathf.Epsilon;  
}  
private void OnTriggerExit2D(Collider2D  
collision)  
{  
    transform.localScale = new Vector2(-  
transform.localScale.x, transform.localScale.y);  
}  
}
```

7. Kemudian drag and drop script pada enemy 1 dan enemy 2.



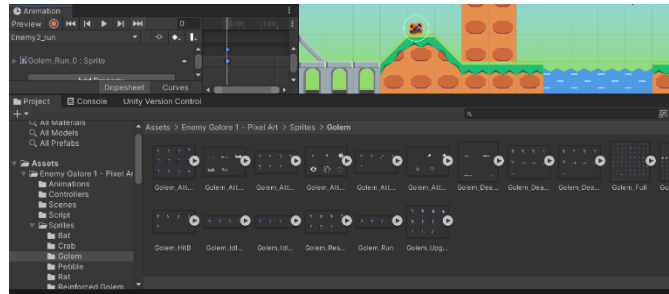
Gambar 10.18 Menambahkan Script

8. Selanjutnya, tambahkan animasi berjalan pada musuh 1 dengan membuat klip baru. Masukkan animasi sprite berjalan ke dalam timeline dan atur waktunya.



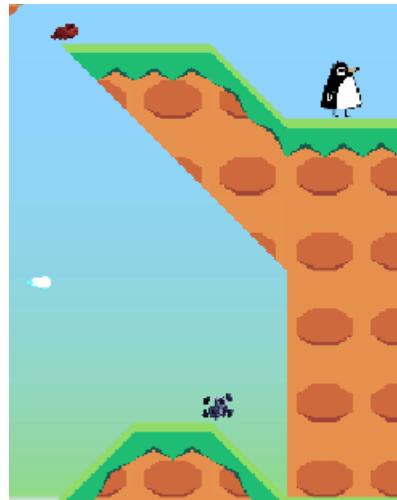
Gambar 10.19 Menambah Animasi Enemy 1

9. Berikutnya, tambahkan animasi berjalan pada musuh 2 dengan membuat klip baru. Tambahkan animasi sprite berjalan ke dalam timeline dan atur waktunya.



Gambar 10.20 Menambah Animasi Enemy 2

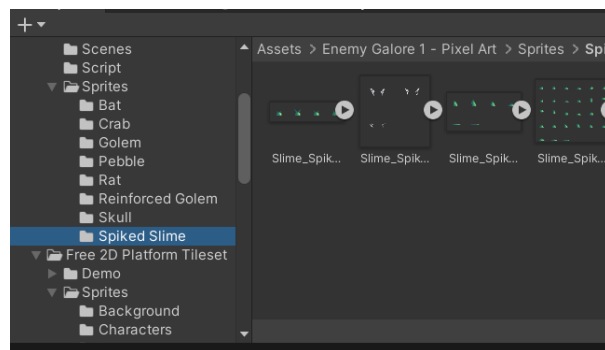
10. Kemudian jalankan game dan pastikan enemy 1 dan 2 berjalan.



Gambar 10.21 Menjalankan Game

C. Membuat Enemy AI

1. Temukan sprite dengan nama Spiked Slime, lalu tambahkan ke dalam hirarki.



Gambar 10.22 Menambahkan Sprite



2. Buat script Enemy_AI dan tambahkan script berikut.

```
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;
    }

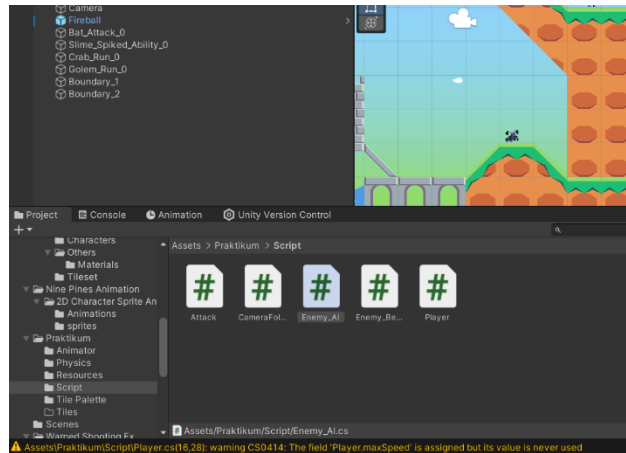
    // Update is called once per frame
    void Update()
    {
        // Menghitung jarak antara musuh dan pemain
        float distanceToPlayer =
        Vector2.Distance(player.position, transform.position);

        // 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);
    }
}
```

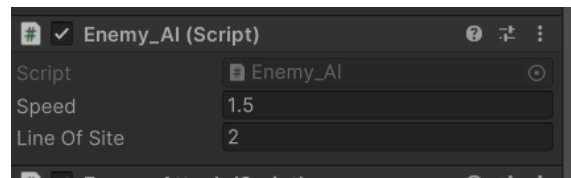


3. Selanjutnya, seret dan lepaskan skrip Enemy_AI ke musuh 3 dan 4.



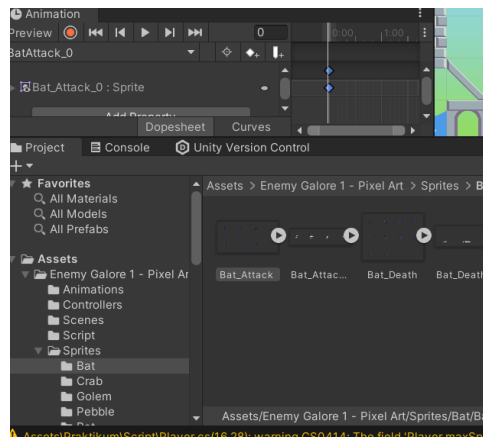
Gambar 10.23 Menambahkan Script

4. Ubah kalimatnya agar tidak sama dan tidak perlu di garis tebal



Gambar 10.24 Mengatur Kecepatan Dan Jarak

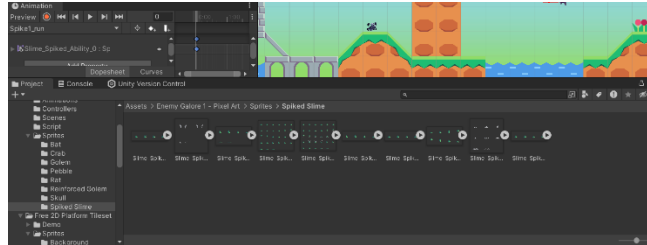
5. Selanjutnya, tambahkan animasi bergerak pada musuh 3 dengan membuat klip baru. Masukkan animasi sprite bergerak ke dalam timeline dan atur waktunya.



Gambar 10.25 Menambah Animasi Enemy 3



6. Berikutnya, tambahkan animasi bergerak pada musuh 4 dengan membuat klip baru. Masukkan animasi sprite bergerak ke dalam timeline dan atur waktunya.



Gambar 10.26 Menambah Animasi Enemy 4

D. Respawn

1. Sisipkan script berikut ke dalam script Player untuk mengatur pemain kembali ke lokasi awalnya ketika jatuh atau nyawanya berkurang di bawah 0.

```
public int nyawa;
[SerializeField] Vector3 respawn_loc;
public bool play_again;
private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
    rb.freezeRotation = true; // Mencegah rotasi
    karakter
    animator = GetComponent<Animator>();

    // Menentukan tempant respawn
    respawn_loc = transform.position;
}
void Update()
{
    horizontalValue =
    Input.GetAxisRaw("Horizontal");
    if (Input.GetButtonDown("Jump"))
    {
        animator.SetBool("Jumping", true);
        jump = true;
    }
    else if (Input.GetButtonUp("Jump"))
        jump = false;

    if (Input.GetKeyDown(KeyCode.C))
    {
        StartCoroutine(Attack());
    }
    if (nyawa < 0)
    {
        playagain();
    }
    if (transform.position.y < -10)
    {
        play_again = true;
    }
}
```

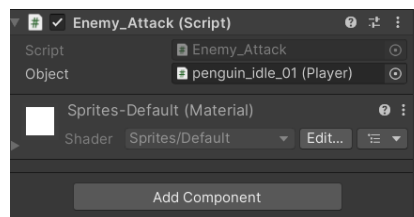


```
        playagain();
    }
}
void playagain()
{
    if (play_again == true)
    {
        nyawa = 3;
        transform.position = respawn_loc;
        play_again = false;
    }
}
```

2. Selanjutnya, buatlah sebuah script baru dengan nama `Enemy_Attack`, kemudian tambahkan skrip berikut ke dalamnya.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_Attack : 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;
            }
        }
    }
}
```

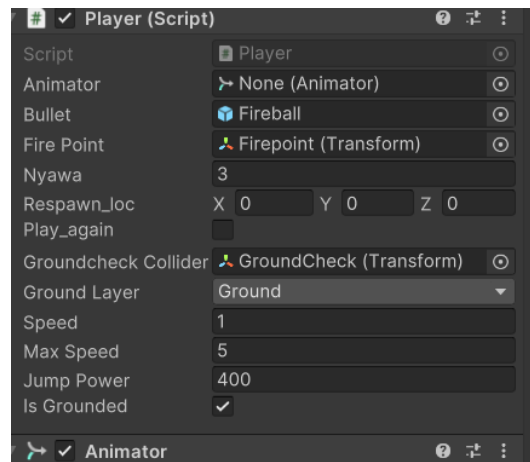
3. Lalu drag and drop script `Enemy_Attack` ke semua musuh dan arahkan object pada `penguin_idle_0`.



Gambar 10.27 Menambahkan Script

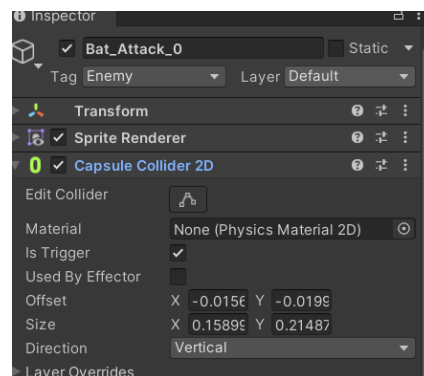


4. Beralih ke tab Inspector pada game object Player, kemudian ubah nilai Nyawa menjadi 3 pada bagian Player (Script).



Gambar 10.28 Mengubah Nilai Nyawa

5. Pastikan bahwa setiap musuh telah diberi tag "Enemy" dan memiliki komponen Capsule Collider 2D dengan opsi Is Trigger tercentang.



Gambar 10.29 Tab Inspector Enemy

6. Selanjutnya, coba jalankan game. Ketika pemain bersentuhan dengan musuh, nyawa akan berkurang 1. Jika nyawa kurang dari 0 atau pemain jatuh, maka akan respawn kembali ke titik awal.

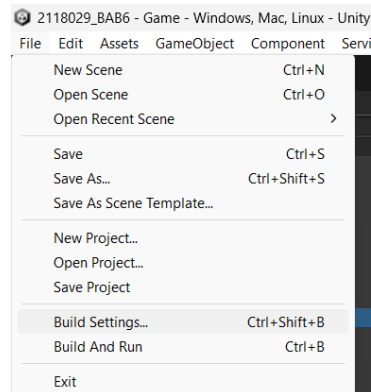


Gambar 10.30 Menjalankan Game



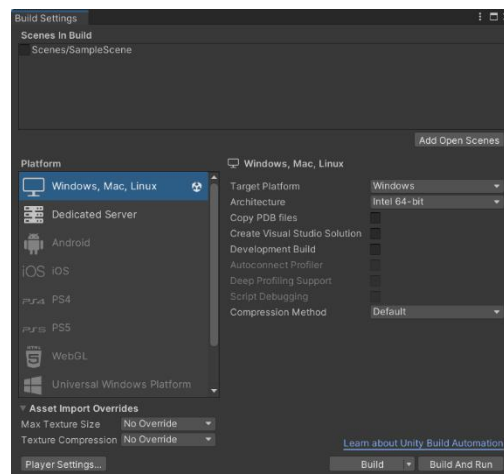
E. Render game

1. Untuk merender game, buka menu File dan pilih Build Settings.



Gambar 10.31 Menu File

2. Kemudian tambahkan Scene yang ingin dirender dengan memilih Add Open Scenes pada Build Settings. Setelah itu, pilih Build dan simpan hasilnya ke dalam folder yang diinginkan.



Gambar 10.32 Build Settings

F. Kuis : Melengkapi Source Code

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




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

Penjelasan :

Pada script PlayerAttack merupakan sebuah kelas dalam Unity yang mengatur serangan melee pemain. Saat tombol "Fire1" ditekan, fungsi PerformMeleeAttack() dipanggil. Fungsi ini menggunakan Raycast untuk mendeteksi apakah ada musuh di depan pemain dalam jarak serangan. Jika ada, maka serangan akan diterapkan ke musuh tersebut dengan memanggil fungsi TakeDamage() dari komponen Health musuh.