



TUGAS PERTEMUAN: 10

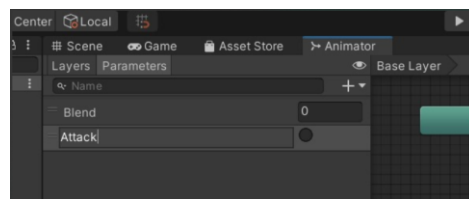
Respawn and AI Enemy Attack

NIM	:	2118003
Nama	:	Kevin Majesta Ivano
Kelas	:	A
Asisten Lab	:	M. Rafi Faddilani (2118144)

1.1 Tugas 1 : Membuat Mekanisme Attack, Enemy Attack AI, dan Respawn

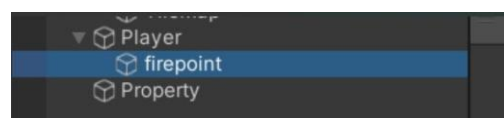
A. Mekanisme Attack

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



Gambar 1.1 Add Parameter

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

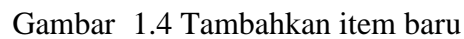
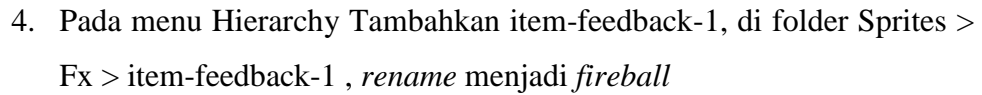


Gambar 1.2 Tambah folder baru

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



Gambar 1.3 Atur letak bullet



-
- The screenshot shows the Unity Inspector window with two components selected for a circle object:
- Circle Collider 2D**
 - Edit Collider:
 - Material: None (Physics Material 2D)
 - Is Trigger: ☐
 - Used By Effector: ☐
 - Offset: X 0 Y 0
 - Radius: 0.2
 - Rigidbody 2D**
 - Body Type: Dynamic
 - Material: None (Physics Material 2D)
 - Simulated: ☒
 - Use Auto Mass: ☐
 - Mass: 1
 - Linear Drag: 0
 - Angular Drag: 0.05
 - Gravity Scale: 0
 - Collision Detection: Discrete
 - Sleeping Mode: Start Awake
 - Interpolate: None
 - Constraints**
 - Freeze Position: ☒ X ☒ Y
 - Freeze Rotation: ☒ Z

Gambar 1.5 Tambah komponen

-

Gambar 1.6 Panel animation



7. Pada Script Player Tambahkan Script dibawah ini

```
#Pada class Player
// Deklarasi variabel
public Animator animator;
public GameObject bullet;
public Transform firePoint;

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

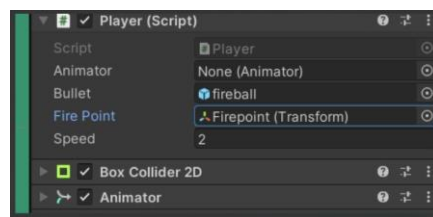
    float direction = 1f;

    GameObject fireball = Instantiate(bullet,
    firePoint.position, Quaternion.identity);

    fireball.GetComponent<Rigidbody2D>().velocity =
    new Vector2(direction * 10f, 0);

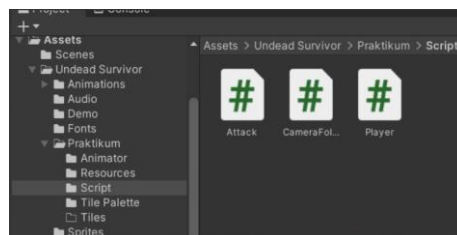
    Destroy(fireball, 2f);
}
#Tambahkan pada Function Void Update
if (Input.GetKeyDown(KeyCode.C))
{
    StartCoroutine(Attack());
}
```

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



Gambar 1.7 Isi nilai

9. Buat Script Attack pada folder Script



Gambar 1.8 Buat script attack

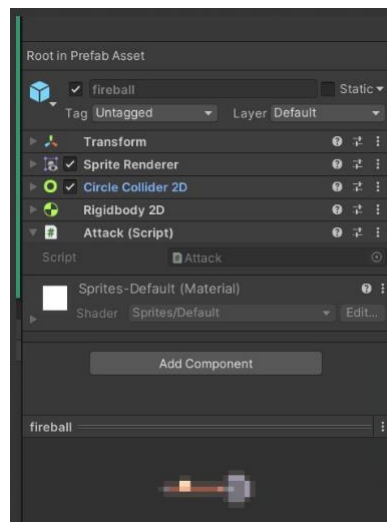


10. Tambahkan Script Attack dibawah ini

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

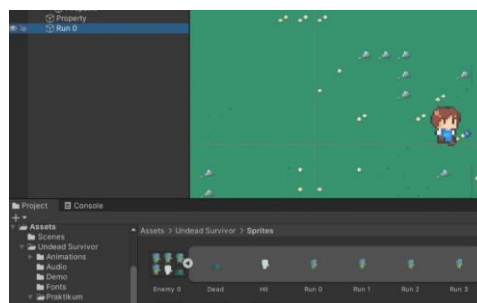
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```

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



Gambar 1.9 Masukkan *script*

12. Tambahkan Enemy pada hierarchy di folder Sprites



Gambar 1.10 Tambah enemy

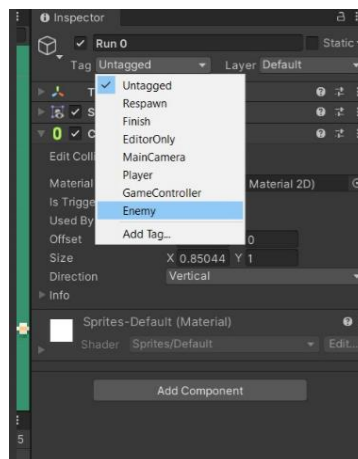


13. Kemudian klik pada enemy, lalu pada menu tab inspector tambahkan collider 2D untuk mendeteksinya



Gambar 1.11 Atur timeline

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



Gambar 1.12 Tambah tag

15. Tembak Enemy dengan menekan Tombol C untuk menghancurkan musuh

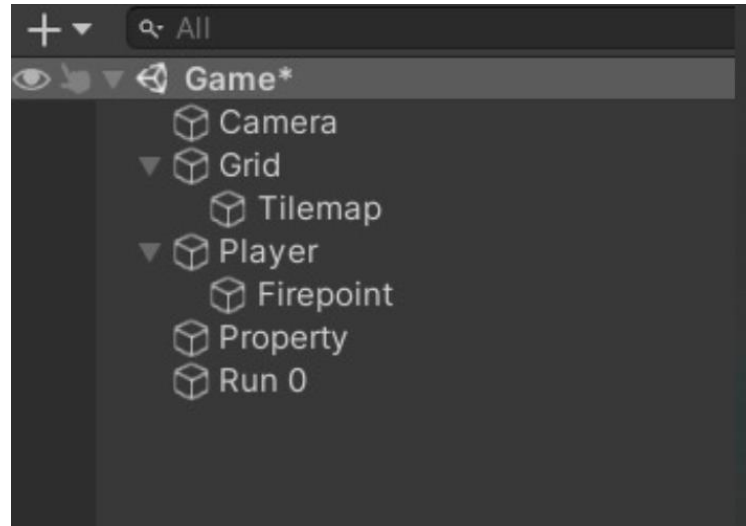


Gambar 1.13 Hasil



B. Enemy AI

1. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama 'run 0'. Tambahkan 'run-0' pada Hierarchy



Gambar 1.14 Add Hierarchy

2. Buat Script Enemy_AI pada folder Praktikum - Script



Gambar 1.15 Tambah script baru

3. 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
    }
}
```



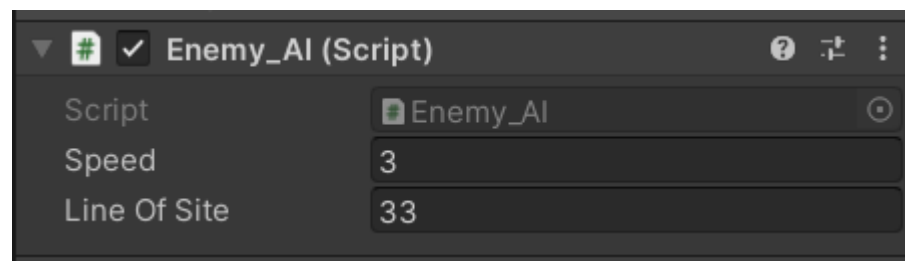
```
        initialPosition = GetComponent<Transform>().position;
    }

    // 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);
    }
}
```

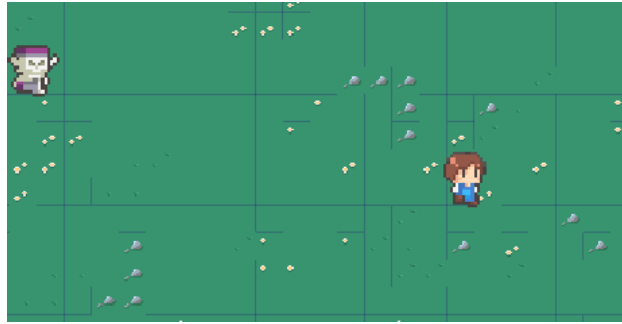
4. Pada Inspector Enemy_Ai, Atur Speed juga Line of Site untuk menentukan jarak dan speed pada enemy



Gambar 1.16 Atur inspector



5. Running Game, maka enemy akan mengikuti Gerakan Player



Gambar 1.17 Tambahkan item baru

C. Respawn

1. Buka file script (Player.cs) tambahkan seperti dibawah ini

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

public class Player : MonoBehaviour
{
    public Animator animator;
    public GameObject bullet;
    public Transform firePoint;
    public int nyawa;
    [SerializeField] Vector3 respawn_loc;
    public bool play_again;

    Rigidbody2D rb;

    [SerializeField] float speed = 1;
    float horizontalValue;
    float verticalValue;
    bool facingRight = true;
    bool facingUp = false;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();

        respawn_loc = transform.position;
    }

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

    void Update()
```




```
{
    horizontalValue =
Input.GetAxisRaw("Horizontal");
    verticalValue = Input.GetAxisRaw("Vertical");

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

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

void FixedUpdate()
{
    Move(horizontalValue, verticalValue);

    animator.SetFloat("Blend",
Mathf.Abs(rb.velocity.x));
}

IEnumerator Attack()
{
    animator.SetTrigger("Attack");
    yield return new WaitForSeconds(0.25f);

    Vector2 direction = Vector2.right; // Arah
default ke kanan
    float rotationAngle = 0f;

    if (horizontalValue > 0)
    {
        direction = Vector2.right;
        rotationAngle = 0f;
    }
    else if (horizontalValue < 0)
    {
        direction = Vector2.left;
        rotationAngle = 180f;
    }
    else if (verticalValue > 0)
    {
        direction = Vector2.up;
        rotationAngle = 90f;
    }
    else if (verticalValue < 0)
    {
        direction = Vector2.down;
        rotationAngle = -90f;
    }
}
```



```
        GameObject fireball = Instantiate(bullet,
firePoint.position, Quaternion.Euler(0, 0,
rotationAngle));
        fireball.GetComponent<Rigidbody2D>().velocity =
direction * 10f;

        Destroy(fireball, 2f);
    }

    void Move(float horizontalDir, float verticalDir)
    {
        #region gerak kanan kiri dan atas bawah
        float xVal = horizontalDir * speed * 100 *
Time.fixedDeltaTime;
        float yVal = verticalDir * speed * 100 *
Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal, yVal);
        rb.velocity = targetVelocity;

        if (facingRight && horizontalDir < 0)
        {
            transform.localScale = new Vector3(-2, 2,
1);
            facingRight = false;
        }
        else if (!facingRight && horizontalDir > 0)
        {
            transform.localScale = new Vector3(2, 2, 1);
            facingRight = true;
        }

        if (!facingUp && verticalDir > 0)
        {
            transform.localScale = new Vector3(2, 2,
1); // Sesuaikan ini jika Anda memiliki sprite yang
berbeda untuk menghadap ke atas/bawah
            facingUp = true;
        }
        else if (facingUp && verticalDir < 0)
        {
            transform.localScale = new Vector3(2, 2,
1); // Sesuaikan ini jika Anda memiliki sprite yang
berbeda untuk menghadap ke atas/bawah
            facingUp = false;
        }
        #endregion
    }
}
```

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

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



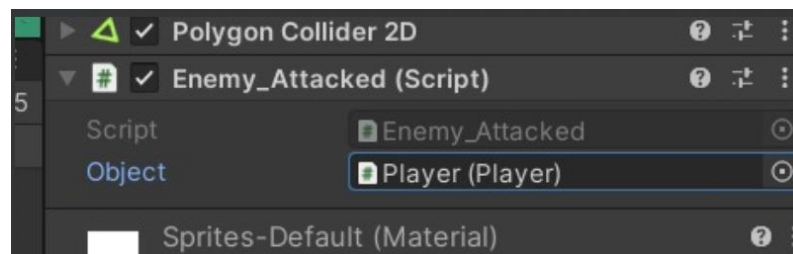
```
public class Enemy_attacked : MonoBehaviour
{
    [SerializeField] private Player Object;

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

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

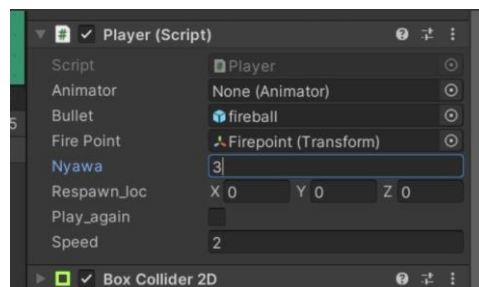
            if (Object.nyawa < 0)
            {
                Object.play_again = true;
            }
        }
    }
}
```

3. Pada hierarchy enemy Tambahkan Script enemy attack



Gambar 1.18 Ganti object

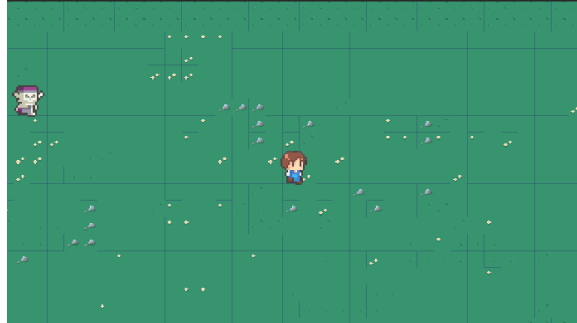
4. Klik game object Player, pergi ke Inspector dan ubah nilai Nyawa menjadi 3 pada Player(Script)



Gambar 1.19 Ubah nilai



5. Jika di play, Player mengenai atau menyentuh opossum-1 sebanyak 3 kali maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan reswpawn ke titik awal



Gambar 1.20 Hasil jadi

D. Repository GitHub

Kevinmajesta/2118003_PRAK_ANIGAME

E. Kuis

Lengkapi Source code dibawah ini :

```
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))
        {
            // Lengkapi kode di sini untuk mengenai musuh
            dan mengurangi health mereka
        }
    }
}
```

Source code yang lengkap :

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

Penjelasan :

Kode diatas yang saya tambah mengambil komponen EnemyHealth dari objek yang terkena oleh serangan atau tindakan pemain, dan jika komponen tersebut ada, maka metode TakeDamage() dari komponen tersebut dipanggil untuk memberikan kerusakan sesuai dengan nilai attackDamage.



Dan ini merupakan source code yang sudah diperbaiki

```
void HandleJumpInput()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        animator.SetBool("isJumping", true); // Set to true
        when jumping starts
        rb.AddForce(Vector2.up * jumpForce,
        ForceMode2D.Impulse);
    }
    else if (Input.GetKey(KeyCode.Space))
    {
        animator.SetBool("isJumping", true); // Set to true
        while the space key is held down
    }
    else
    {
        animator.SetBool("isJumping", false); // Set to
        false when the space key is released
    }
}

void HandleMovementInput()
{
    float move = Input.GetAxis("Horizontal");

    if (move == 0)
    {
        animator.SetBool("isIdle", true);
        animator.SetBool("isWalking", false);
    }
    else
    {
        animator.SetBool("isIdle", false);
        animator.SetBool("isWalking", true);

        transform.Translate(Vector3.right * move *
        Time.deltaTime); // Corrected to Vector3.right
    }

    if (move < 0)
    {
        transform.localScale = new Vector3(-1, 1, 1); //
        Correct scaling for left direction
    }
    else if (move > 0)
    {
        transform.localScale = new Vector3(1, 1, 1); //
        Correct scaling for right direction
    }
}
```

Penjelasan :

Coding diatas adalah bagian dari fungsi-fungsi untuk mengatur input pemain dalam sebuah game. Fungsi HandleJumpInput() menangani input



loncatan, dimana ketika tombol spasi ditekan, karakter akan meloncat dengan menerapkan gaya impuls ke atas, sementara `HandleMovementInput()` mengatur pergerakan horizontal karakter berdasarkan input pemain. Jika tidak ada input, karakter akan diam, jika input positif, karakter akan bergerak ke kanan, dan sebaliknya. Animasi karakter diatur berdasarkan input dengan mengatur parameter animasi seperti `isJumping`, `isIdle`, dan `isWalking`.