



TUGAS PERTEMUAN: 10

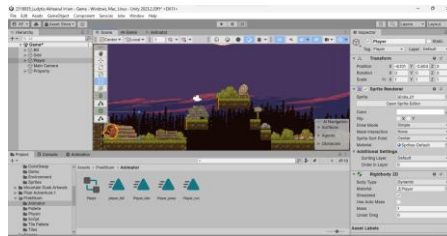
Respawn and AI Enemy Attack

NIM	:	2118035
Nama	:	Ludytio Akhsanul In'am
Kelas	:	A
Asisten Lab	:	Mohammad Akbar Ilham (2118091)

1.1 Tugas 1 : Respawn and AI Enemy Attack

A. Membuat Mekanisme Attack

1. Pertama buka *project* yang telah dikerjakan di materi sebelumnya.



Gambar 1.1 Membuka Project

2. Setelah itu pada tab animator buat parameter trigger dan beri nama “Attack”. Lalu tambahkan *create empty* pada player dan beri nama “Firepoint”. Jika sudah maka pergi ke *inspector* pada bagian *firepoint* dan ubah iconnya dan letakkan berada di depan player.



Gambar 1.2 Menambahkan Firepoint



3. Setelah itu tambahkan satu item untuk sebagai peluru. Lalu pergi ke bagian *inspector*. Jika sudah maka tambahkan komponen *circle collider 2D* dan *rigidbody 2D*.



Gambar 1.3 Menambahkan Peluru

4. Setelah itu buka *script player*. Jika sudah maka tambahkan source code dibawah ini

```
public Animator animator;
public GameObject bullet;
public Transform firePoint;
float direction;

void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
    if (Input.GetKeyDown(KeyCode.C))
    {
        StartCoroutine(Attack());
    }

    if (Input.GetButtonDown("Jump"))
    {
        animator.SetBool("Jumping", true);
        jump = true;
    }
    else if (Input.GetButtonUp("Jump"))
        jump = false;
}

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);
}

void Move(float dir, bool jumpflag)
{
    if(isGrounded && jumpflag)
    {
        isGrounded = false;
        jumpflag = false;
        rb.AddForce(new Vector2(0f, jumpPower));
    }
}
```



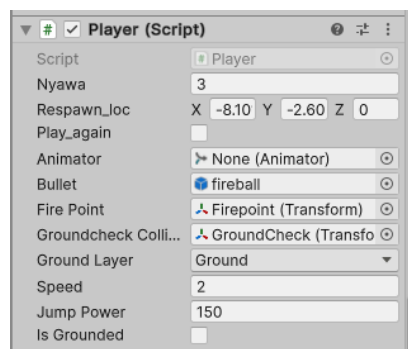
```
#region gerak kanan kiri
float xVal = dir * speed * 100 * Time.fixedDeltaTime;
Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
rb.velocity = targetVelocity;

if (facingRight && dir < 0)
{
    // ukuran player
    transform.localScale = new Vector3(-1, 1, 1);
    facingRight = false;
    direction = -1f;
}

else if (!facingRight && dir > 0)
{
    // ukuran player
    transform.localScale = new Vector3(1, 1, 1);
    facingRight = true;
    direction = 1f;
}

#endregion
}
```

5. Setelah itu klik karakter. Pergi ke bagian *inspector*. Lalu cari *bullet* dan beri objek *fireball* untuk yang akan ditembak. Lalu bagian *fire point* isi dengan *firepoint* untuk titik tembak pertamanya.



Gambar 1.4 Setting Bullet dan Firepoint

6. Setelah itu buat script baru. Beri nama *script* tersebut “Attack”. Jika sudah maka tambahkan *source code* dibawah ini.

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

public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {

```



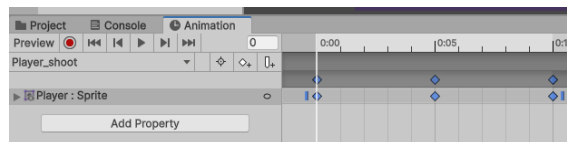
```
if (collision.gameObject.CompareTag("Enemy"))
{
    Destroy(gameObject);
    Destroy(collision.gameObject);
}
}
```

7. Berikutnya *script* yang telah dibuat tadi *drag and drop* pada *fireball*. Jika sudah maka ketika dicek pada bagian *inspector* maka hasilnya akan seperti gambar dibawah ini. Jika sudah maka karakter sudah bisa menembak ke kiri dan ke kanan.



Gambar 1.5 Membuat Karakter Bisa Menembak

8. Setelah itu tambahkan animsi untuk ketika menembak. Klik pada player dan buat clip baru. Beri nama “Player_shoot”. Tambahkan animasi yang ingin digunakan ketika menembak. Jika sudah buat transisinya. Lalu pada bagian tanda panah dari *idle* ke *player shoot* buat kondisinya menjadi *attack* dan beri transisi durasinya 0,2. Lalu pada panah sebaliknya beri kondisi blend dan less dengan valuenya 0,01.



Gambar 1.6 Menambahkan Animasi Ketika Menembak



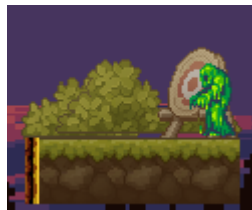
9. Lalu tambahkan musuh. Klik pada kelelawar, dan pergi ke *inspector*. Tambahkan *capsule collider 2D*. Tambahkan tag *enemy* dan ubah tagnya sebagai *enemy*. Diaktakan berhasil jika musuh ditembak dia akan menghilang.



Gambar 1.7 Uji Coba Musuh

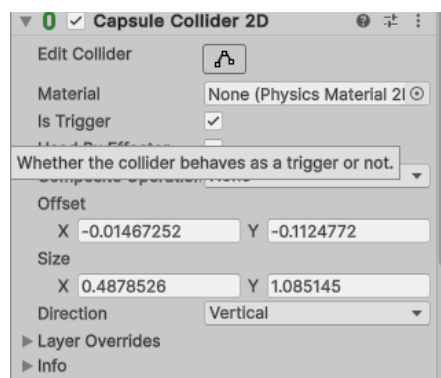
B. Membuat Enemy Behavior NPC

1. Berikutnya, tambahkan satu musuh lagi. Cari asetnya, lalu letakkan pada *hierarcy*. Jika sudah hasilnya akan seperti gambar dibawah ini. Jangan lupa untuk diatur ukurannya sesuai keinginan.



Gambar 1.8 Menambahkan Musuh

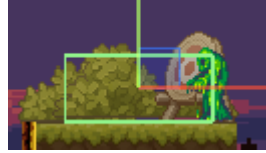
2. Setelah itu klik pada bagian musuh. Tambahkan komponen *Capsule collider 2D*, dan *Rigidbody 2D*. Jika sudah atur collidernya sesuai dengan keinginan.



Gambar 1.9 Menambahkan Capsule collider 2D, dan Rigidbody 2D



3. Lalu buat sebuah *empty object* pada hirarki. Jika sudah ubah Namanya menjadi “*Boundary*”. Lalu sesuaikan ukurannya untuk sebagai batasan pergerakan karakter musuh.



Gambar 1.10 Mengatur Boundary Untuk Musuh

4. Setelah itu buat sebuah *script* baru. Buat nama scriptnya menjadi “*Enemy_Behavior*”. Tambahkan *source code* berikut.

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

public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 1f;
    Rigidbody2D rb;

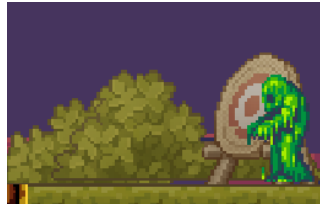
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

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

5. Lalu lakukan *drag and drop* pada *script*. Letakkan *script* pada bagian musuh. Jika sudah maka hasilnya akan seperti gambar dibawah ini.



Gambar 1.11 Menambahkan Transisi dan Tipe Data

6. Selanjutnya tambahkan animasi berjalan pada musuh. Klik pada musuhnya. Lalu buat clip baru. Jika sudah tambahkan animasi berjalan. Jika sudah atur waktunya sesuai dengan keinginan. Maka animasi sudah ditambahkan.



Gambar 1.12 Menambahkan Animasi pada Musuh

C. Membuat Enemy AI

1. Setelah itu buat *script*. Beri nama “Enemy_AI”. Jika sudah tambahkan *source code* berikut 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);
}
}
```

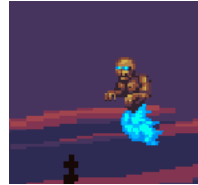
2. Setelah itu *drag and drop script* yang barusan dibuat tadi pada musuh yang telah dibuat untuk testing tadi. Lalu atur jarak dan kecepatan musuhnya..



Gambar 1.13 Mengatur Transisi Run ke Idle



3. Selanjutnya tambahkan animasi berjalan pada musuh. Klik pada musuhnya. Lalu buat clip baru. Jika sudah tambahkan animasi berjalannya. Jika sudah atur waktunya sesuai dengan keinginan. Maka animasi sudah ditambahkan.



Gambar 1.14 Menambahkan Animasi pada Karakter Musuh

D. Respawn

1. Selanjutnya buka *script player*. Lalu tambahkan kode berikut untuk membuat *spawn*. Source ini akan berfungsi ketika *player* jatuh atau terkena musuh sebanyak 3x maka akan kembali ke titik awal.

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

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");
    if (Input.GetKeyDown(KeyCode.C))
    {
        StartCoroutine(Attack());
    }
    if (Input.GetButtonDown("Jump"))
    {
        animator.SetBool("Jumping", true);
        jump = true;
    }
    else if (Input.GetButtonUp("Jump"))
        jump = false;
    //plat again
```



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

2. Selanjutnya buat *script* baru. Beri nama *script* tersebut “Enemy_Attacked”. Jika sudah tambahkan *source code* dibawah ini.

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

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

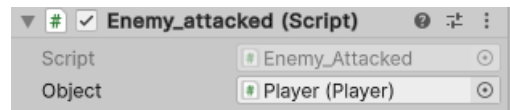
    void Start()
    {
        if (Object == null)
        {
            Object =
GameObject.FindWithTag("Player").GetComponent<Player>()
;
        }
    }

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

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

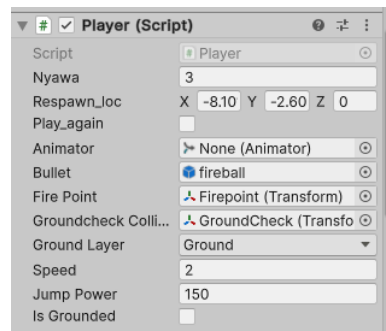


3. Lalu tambahkan *script* tersebut pada semua musuh yang ada. Agar ketika musuh mengenai karakter maka, karakter bisa mati.



Gambar 1.15 Menambahkan Penyerangan pada Aset Musuh

4. Setelah itu atur nyawa player. Jika didalam *setting* ini maka, nyawanya terdapat 3x kesempatan. Jika sudah habis nyawanya, maka akan kembali ke Lokasi awal.



Gambar 1.16 Menambahkan File Player Fall

5. Jika sudah selesai maka game selesai dibuat. Render game di tempat yang sudah disediakan dan tunggu render hingga selesai. Setelah selesai di render hasilnya akan seperti gambar dibawah ini.



Gambar 1.17 Hasil Render



E. Kuis Pertemuan 10

```
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))
        {
            // Mengecek apakah objek terkena serangan
            Health enemyHealth =
            hit.transform.GetComponent<Health>();
            if (enemyHealth != null)
            {
                // Mengurangi health musuh
                enemyHealth.TakeDamage(attackDamage);
            }
        }
    }
}
```

Penjelasan Source code

Pada source code diawali dengan deklarasi bahwa value dari jarak dan damage dari senjata. Lalu dilanjutkan dengan jika menekan button fire1 maka akan melakukan serangan melee. Lalu akan terjadi pengecekan pada objek musuh, apakah musuh memiliki nyawa atau tidak, jika memiliki nyawa ketika terkena serangan maka, nyawa musuh akan berkurang.