TUGAS PERTEMUAN: 10

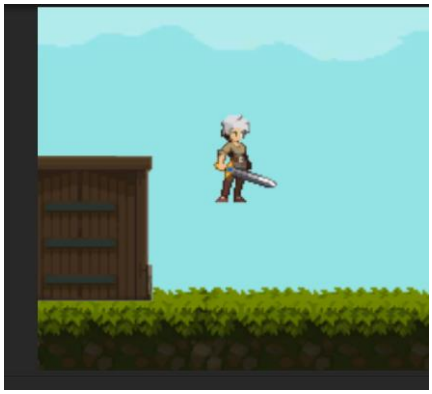
Respawn and AI Enemy Attack

NIM	:	2118104
Nama	:	Andhika Wira S
Kelas	:	D
Asisten Lab	:	2218106 - Rifal Rifqi Rhomadon

10.1 Tugas 10: Respawn, Menyerang dan Diserang Musuh

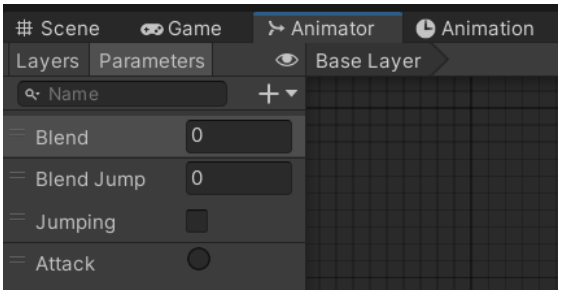
A. Menyerang Musuh

1. Buka proyek sebelumnya



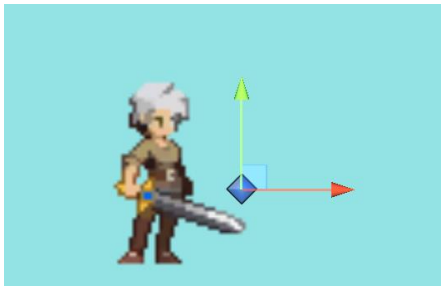
Gambar 10.1 Proyek Sebelumnya

2. Tambahkan parameter *Attack* di menu *Animator*



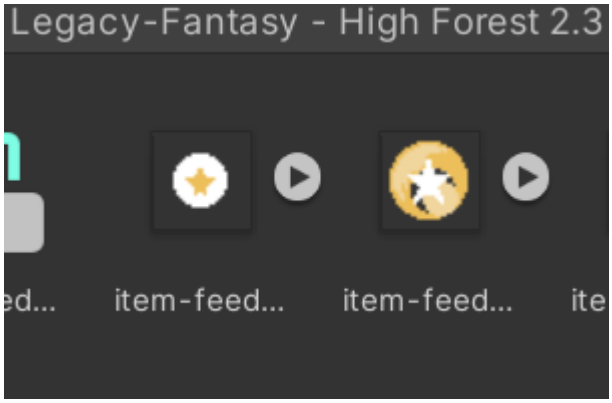
Gambar 10.2 Menambah Parameter *Attack*

3. Buat hierarki ‘Firepoint’ dan masukkan ke hierarki Player. Ubah icon menjadi kristal biru dan posisikan di depan karakter



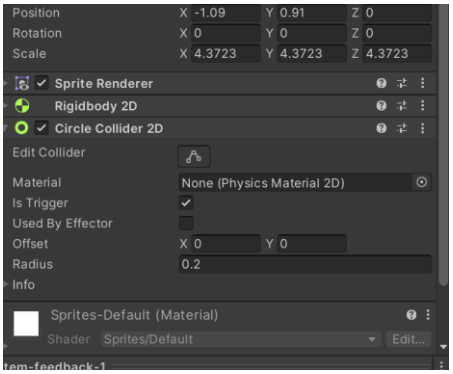
Gambar 10.3 Menambah Firepoint pada Player

4. Salin aset roket di *2D Player* > *Scene* > *Example* > *Rocket* ke direktori *Resources*



Gambar 10.4 Menyalin Aset Roket

5. Tambahkan komponen *Rigidbody 2D* dan *Collider* pada roket. Atur konfigurasinya seperti gambar 10.5



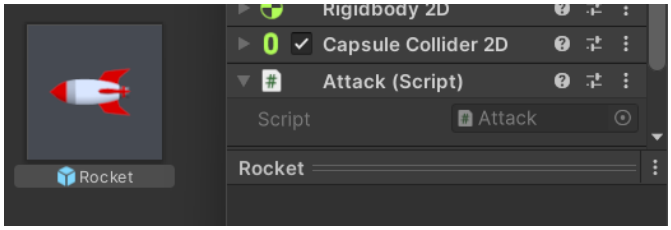
Gambar 10.5 Komponen Roket

6. Buat kodek *Attack* di dalam direktori Script dan tuliskan kodeknya

```
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. Tambahkan kodek *Attack* di komponen roket

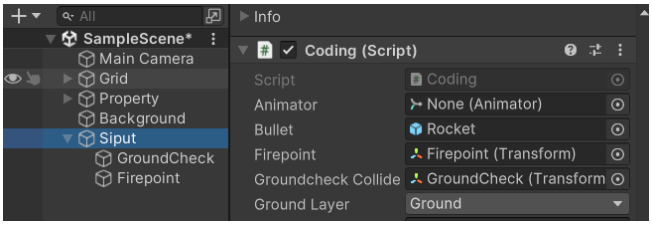


Gambar 10.6 Menambahkan Kodek *Attack* di Komponen Roket

8. Tambahkan beberapa kodek pada berkas Coding.cs agar Player dapat melontarkan roket

```
public GameObject bullet; // di dalam class Coding
public Transform Firepoint;
IEnumerator Attack(){
        animator.SetTrigger("Attack");
        yield return new WaitForSeconds(0.25f);
        float direction = facingRight ? 1f : -1f;
       float rotationAngle = facingRight ? 0f : 180f;
        Quaternion rotation = Quaternion.Euler(0, 0,
rotationAngle);
          GameObject Rocket = Instantiate(bullet,
Firepoint.position, rotation);
         Rocket.GetComponent<Rigidbody2D>().velocity
= new Vector2(direction * 5f, 0);
        Destroy(Rocket, 2f);
    }
// di dalam void Update()
if (Input.GetKeyDown(KeyCode.C)){
            StartCoroutine(Attack());
            animator.SetBool("AttackPlayer", true);
        } else if (Input.GetKeyUp(KeyCode.C)) {
            animator.SetBool("AttackPlayer", false);
        }
```

9. Atur konfigurasi kodek Coding di komponen Player dan pilih Roket yang ada di direktori Resources sebagai *bullet* dan *Firepoint* di dalam hierarki Player sebagai *Firepoint*

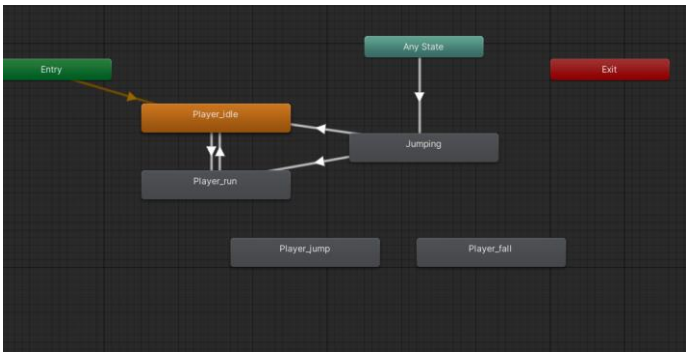


Gambar 10.7 Menambahkan *Bullet* dan *Firepoint* di Player

10. Buat *clip* baru dengan nama ‘player_attack’ dan *drag drop* seluruh aset VICTORY_Player ke halaman kerja *Animation.* Beri rentang hingga 0:30

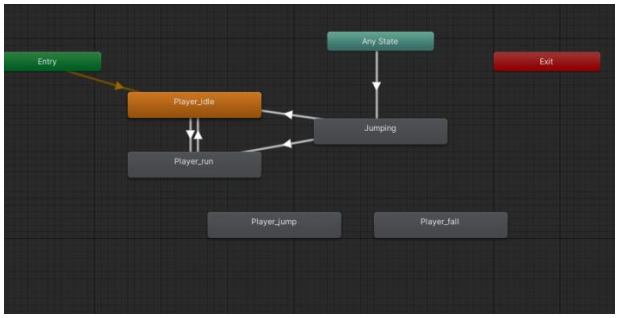
Gambar 10.8 Membuat Animasi Player Menyerang

11. Buat parameter *AttackPlayer* tipe data bool dan hubungkan player_idle ke player_attack dengan konfigurasi seperti gambar 10.9



Gambar 10.9 Menghubungkan player_idle ke player_attack

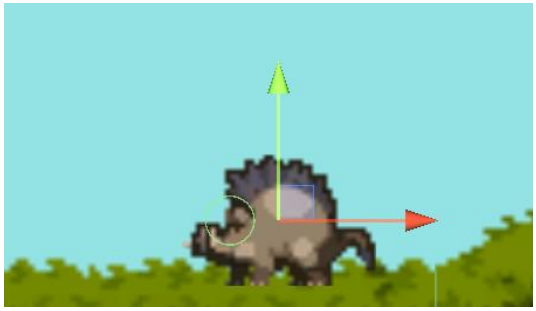
12. Hubungkan player_attack ke player_idle dengan konfigurasi seperti gambar 10.10



Gambar 10.10 Menghubungkan player_attack ke player_idle

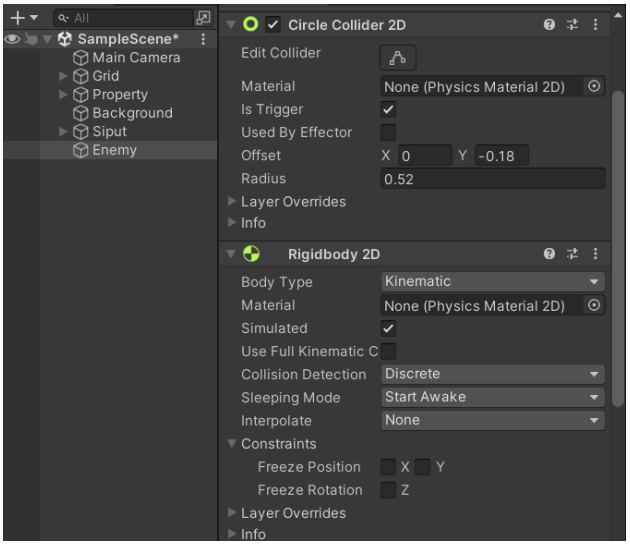
B. Enemy Behavior

1. Tambahkan musuh dari aset DEATH_Player dan beri nama *Enemy*



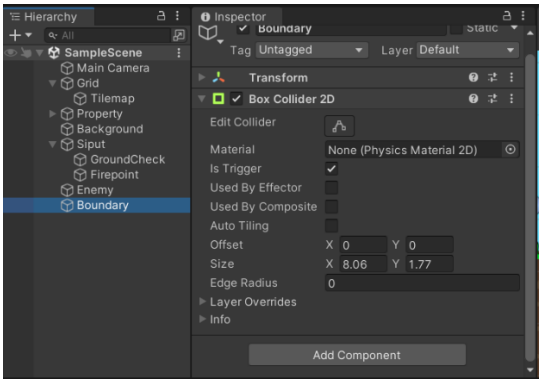
Gambar 10.11 Menambahkan Musuh

2. Tambahkan komponen *Rigidbody 2D* dan *Collider* dengan konfigurasi seperti gambar 10.12



Gambar 10.12 Komponen *Enemy*

3. Buat hierarki *Boundary* dan tambahkan komponen *Box Collider* 2D. Atur konfigurasinya seperti gambar 10.13



Gambar 10.13 Menambah Hierarki *Boundary*

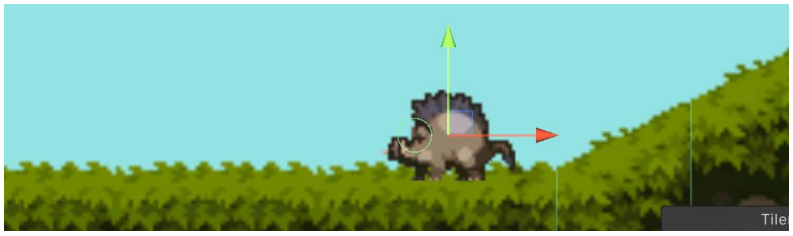
4. Pada hierarki *Enemy*, tambahkan animasi enemy_behavior. Drag drop aset DEATH_Player dari 0060 hingga 00130. Atur rentang animasi hingga 0:20

Gambar 10.14 Menambahkan Animasi Musuh

5. Buat kodek *Enemy_Behavior* di dalam direktori Script dan tuliskan kodeknya. Kemudian tambahkan ke komponen *Enemy*

```
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);}}
```

6. Duplikat hierarki *Enemy* dan *Boundary*. Posisikan agak berjauhan



Gambar 10.15 Menduplikat *Enemy* dan *Boundary*

C. Enemy AI

1. Tambahkan musuh dari aset DEATH_Player dan beri nama *Enemy3*

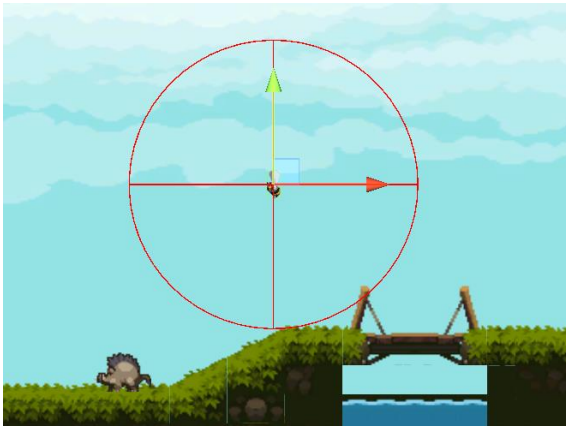
Gambar 10.16 Menambahkan *Enemy3*

2. Buat kodek Enemy_AI di dalam direktori Script dan tuliskan kodeknya

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_AI : MonoBehaviour{
    public float speed;
    public float lineOfSite;
    private Transform player;
    private Vector2 initialPosition;
    private bool facingRight = true;
    void Start(){
player                                                  =
GameObject.FindGameObjectWithTag("Player").transfor
m;
initialPosition=GetComponent<Transform>().position;
}
    void Update(){
                        float    distanceToPlayer    =
Vector2.Distance(player.position,
transform.position);
        if (distanceToPlayer < lineOfSite){
                                    transform.position    =
Vector2.MoveTowards(this.transform.position,
player.position, speed * Time.deltaTime);
            FacePlayer();
        }else{
```

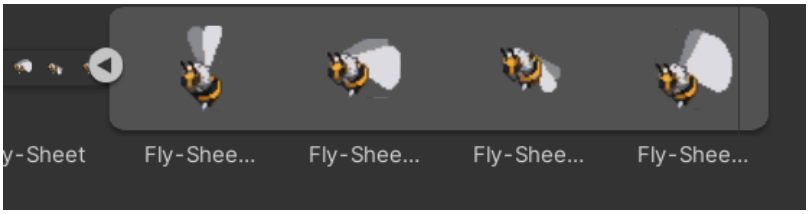
```
                                transform.position    =
Vector2.MoveTowards(transform.position,
initialPosition, speed * Time.deltaTime);
            FaceInitialPosition();}}
    private void FacePlayer(){
        if (player.position.x < transform.position.x
&& !facingRight){
            Flip();
                }else   if   (player.position.x   >
transform.position.x && facingRight){
            Flip();}}
    private void FaceInitialPosition(){
        if (initialPosition.x > transform.position.x
&& !facingRight){
            Flip();
                }else   if   (initialPosition.x   <
transform.position.x && facingRight){
            Flip();}}
    private void Flip(){
        facingRight = !facingRight;
        Vector3 theScale = transform.localScale;
        theScale.x *= -1;
        transform.localScale = theScale;}
    private void OnDrawGizmosSelected(){
        Gizmos.color = Color.red;
           Gizmos.DrawWireSphere(transform.position,
lineOfSite);}}
```

3. Tambahkan komponen kodek Enemy_AI dan *Circle Collider*. Atur konfigurasi seperti gambar 10.17



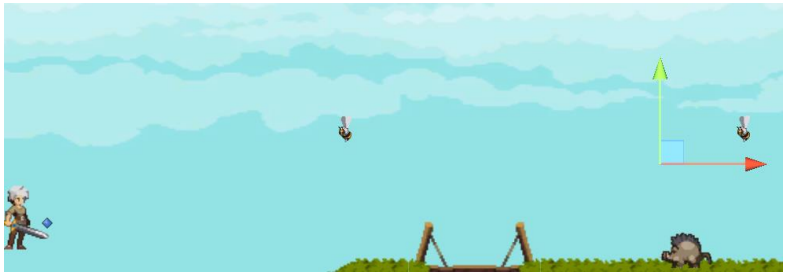
Gambar 10.17 Menambahkan Komponen *Enemy3*

4. Pada hierarki *Enemy3*, tambahkan animasi enemy_ai. *Drag drop* aset DEATH_Player dari 00140 hingga 00160. Atur rentang animasi hingga 0:10



Gambar 10.18 Menambahkan Animasi *Enemy3*

5. Duplikat *Enemy3* dan beri jarak agak berjauhan



Gambar 10.19 Menduplikat *Enemy3*

D. Respawn

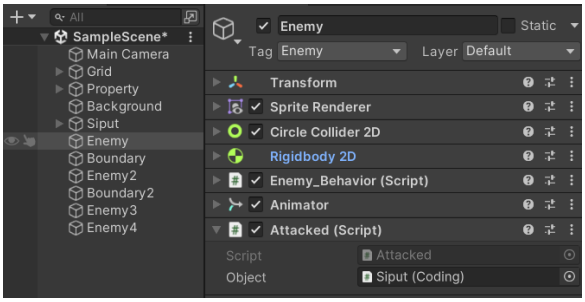
1. Tambahkan beberapa kodek pada berkas Coding.cs agar Player dapat kembali ke titik awal apabila *health point* habis

```
public int HP; // di dalam class Coding
public bool play_again;
[SerializeField] Vector3 respawn_loc;
void playagain(){
        if(play_again == true){
            HP = 3;
            transform.position = respawn_loc;
            play_again = false;}}
// di dalam void awake()
respawn_loc = transform.position;
// di dalam void Update()
if(HP < 0){
            playagain();}
```

2. Buat kodek *Attacked* di dalam direktori Script dan tuliskan kodeknya

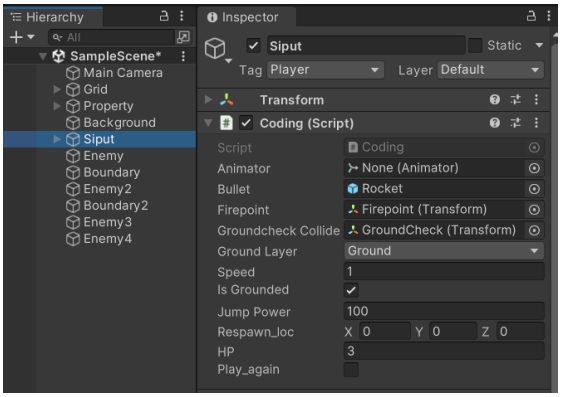
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Attacked : MonoBehaviour{
    [SerializeField] private Coding Object;
    void Start(){
        if (Object == null){
                                            Object      =
GameObject.FindWithTag("Player").GetComponent<Codin
g>();}}
    void OnTriggerEnter2D(Collider2D other){
        if (other.CompareTag("Player")){
            Object.HP--;
Object.animator.SetBool("Attacked", true);
            if (Object.HP <= 0){
                Object.play_again = true;}
        } else {
       Object.animator.SetBool("Attacked", false);}
    }}
```

3. Tambahkan komponen *Attacked* ke semua musuh dan atur *Object* ke Player agar HP Player berkurang jika menyentuh musuh



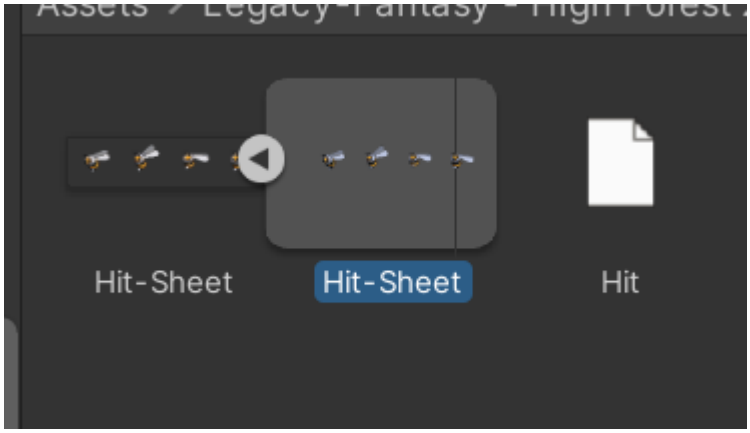
Gambar 10.20 Menambah Komponen Attacked ke Semua Enemy

4. Atur *health point* Player menjadi 3



Gambar 10.21 Mengatur HP Player

5. Buat *clip* baru dengan nama ‘player_attacked’ dan *drag drop* seluruh aset HIT_Player ke halaman kerja *Animation.* Beri rentang hingga 0:30



Gambar 10.22 Membuat Animasi Player Diserang

E. Link Github Pengumpulan

https://github.com/andhikatok/praktikum-animasi-game/tree/main/TUGAS%210

F. Kuis

```
using UnityEngine;
public class PlayerAttack : MonoBehaviour{
    public float attackRange = 2.0f;
    public int attackDamage = 10;
    public string enemyTag = "Enemy";
    void Update(){
        if (Input.GetButtonDown("Fire1")){
            PerformMeleeAttack();
        }
    }
    void PerformMeleeAttack(){
        RaycastHit hit;
        if          (Physics.Raycast(transform.position,
transform.forward, out hit, attackRange)){
            if (hit.collider.CompareTag(enemyTag)){
                Health        healthComponent        =
hit.collider.GetComponent<Health>();
                if (healthComponent != null){
healthComponent.TakeDamage(attackDamage);
                }
            }
        }
    }
}
```

Analisis

Kodek di atas diperbaiki pada *void PerformMeleeAttack*(). Pertama, tipe variabel *attackRange* diubah dari *int* menjadi *float* untuk mencerminkan penggunaannya sebagai nilai jarak serangan. Kedua, terdapat kesalahan ketik pada *InputGetButtonDown* diperbaiki menjadi *Input.GetButtonDown*, dan *attacDamage* diperbaiki menjadi *attackDamage*. Kemudian, penambahan tag *enemyTag* memungkinkan identifikasi musuh melalui *tak* dan memastikan hanya musuh yang terkena serangan. Dalam *void PerformMeleeAttack*, ditambahkan pemeriksaan untuk memastikan bahwa obyek yang terkena *raycast* memiliki komponen *Health* yang berfungsi mengurangi *health* musuh. Dengan perubahan ini, kodek mampu mengurangi *health* musuh tanpa memerlukan kodek tambahan terpisah untuk *EnemyHealth*.