



## TUGAS PERTEMUAN: 10

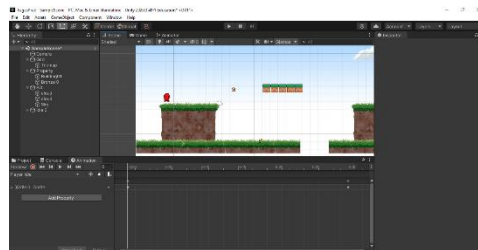
### Respawn and AI Enemy Attack

NIM	:	2118139
Nama	:	Farhan Maulana Ridho
Kelas	:	D
Asisten Lab	:	2118004 - Bagas Anardi Surya W.

#### 10.1 Tugas 10: Respawn, Menyerang dan Diserang Musuh

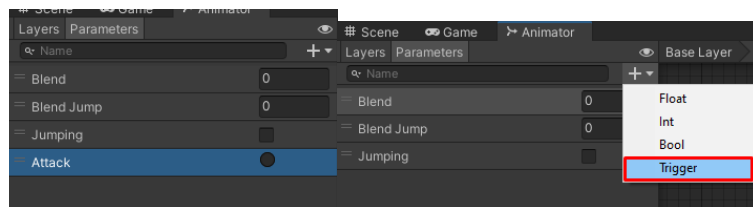
##### A. Menyerang Musuh

1. Buka *project Unity* sebelumnya bab 9.



Gambar 10.1 Proyek Sebelumnya

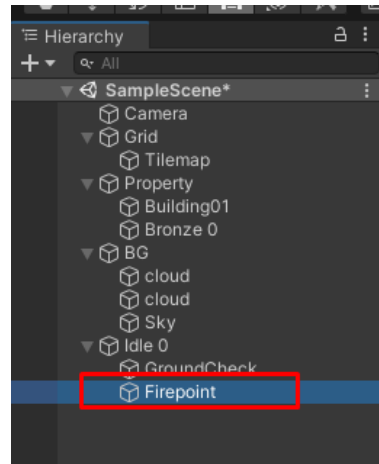
2. Tambahkan parameter dengan tipe *Trigger* dan beri nama *Attack* di menu *Animator*.



Gambar 10.2 Parameter *Attack*

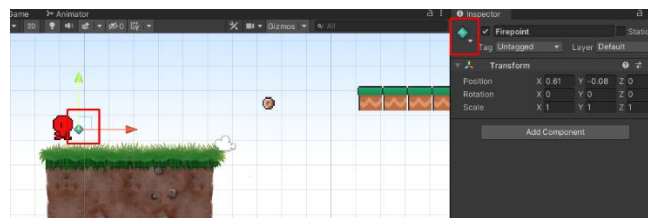


3. Pada hierarki buat 'Firepoint' dan masukkan ke hierarki Player.



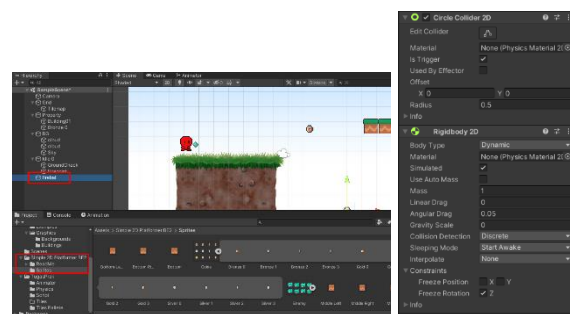
Gambar 10.3 Menambah Firepoint

4. Setelah itu pada firepoint rubahlah icon dan atur depan pada player.



Gambar 10.4 Firepoint

5. Tambahkan item-feedback-1 di folder Sprite. Tambahkan komponen *Rigidbody 2D* dan *Collider* pada roket. Atur konfigurasinya seperti berikut.



Gambar 10.5 Komponen Roket



6. Pada script player tambahkan seperti berikut

```
public class Player : MonoBehaviour
{
    public Animator animator;
    public GameObject bullet;
    public Transform firePoint;
```

Lalu tambahkan dibawah fungsi fixedUpdate

```
IEnumerator Attack()
{
    // Mengaktifkan animasi serangan
    animator.SetTrigger("Attack");

    // Menunggu selama 0.25 detik sebelum
    melanjutkan eksekusi kode berikutnya
    yield return new WaitForSeconds(0.25f);

    // Menentukan arah tembakan berdasarkan arah
    hadapan pemain
    float direction = facingRight ? 1f : -1f;

    // Membuat objek fireball di posisi
    firePoint dengan rotasi identitas (tanpa rotasi)
    GameObject fireball = Instantiate(bullet,
    firePoint.position, Quaternion.identity);

    // Memberikan kecepatan pada fireball
    sehingga bergerak ke arah yang ditentukan dengan
    kecepatan 10 unit per detik

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

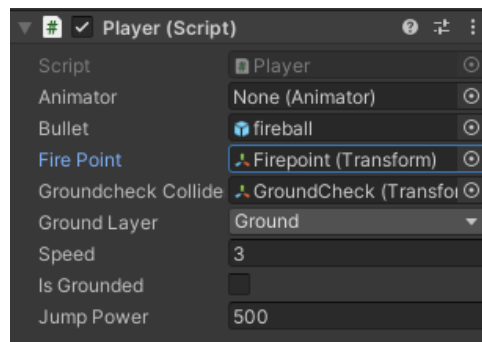
    // Menghancurkan objek fireball setelah 2
    detik untuk mengeliminasi objek yang tidak
    dibutuhkan lagi dari scene
    Destroy(fireball, 2f);
}
```

Lalu tambahkan didalam fungsi update

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

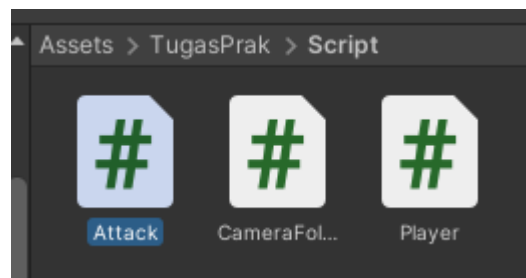


7. Pada player inspector, pada bullet isi kan dengan fireball yang sudah dibuat. Untuk firepoint isi dengan firepoint yang sudah dibuat.



Gambar 10.6 Inspector Player

8. Buat script dengan nama Attack pada folder TugasPrak>Script.



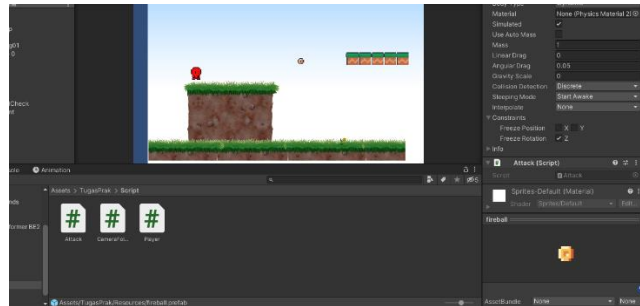
Gambar 10.7 New Script

9. Buka script Attack dan ganti kode seperti dibawah.

```
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);
        }
    }
}
```

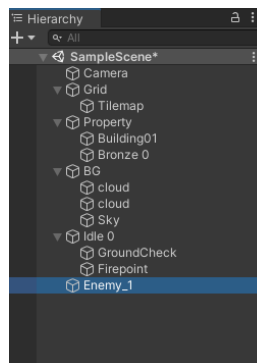


10. Kemudian di *folder resource* tambahkan *Script Attack* di *Prefab fireball*, dengan cara klik *fireball* kemudian di *inspector* arahkan *Script Attack* kedalam *Inspector*.



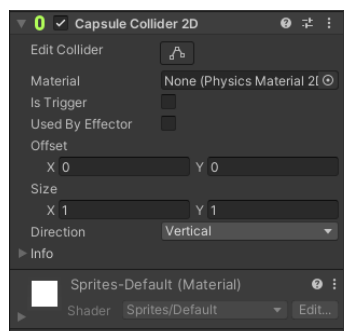
Gambar 10.7 Script Attack

11. Tambahkan Enemy pada hierarchy.



Gambar 10.8 Tambahkan Enemy

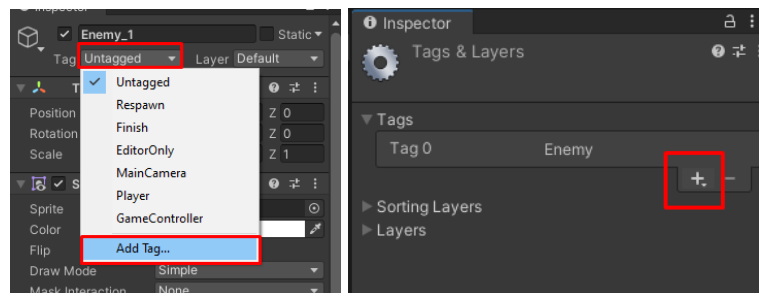
12. Pada inspector tambahkan capsule collider 2d agar bisa terdeteksi.



Gambar 10.9 Capsule Collider 2d

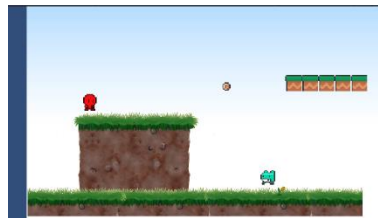


13. Pada inspector enemy pada tag, buat tag dengan Add Tag. Setelah itu pada tombol + tambahkan menjadi Enemy.



Gambar 10.10 Tag Enemy

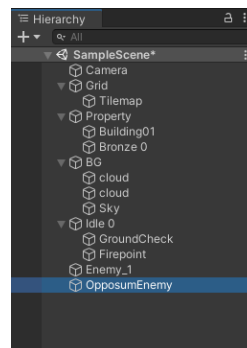
14. Setelah itu jalankan



Gambar 10.11 Play Project

## B. Enemy Behavior

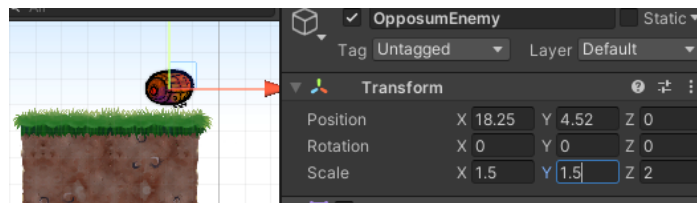
1. Tambahkan musuh dari aset dan beri nama *OpposumEnemy*.



Gambar 10.11 Menambahkan Musuh

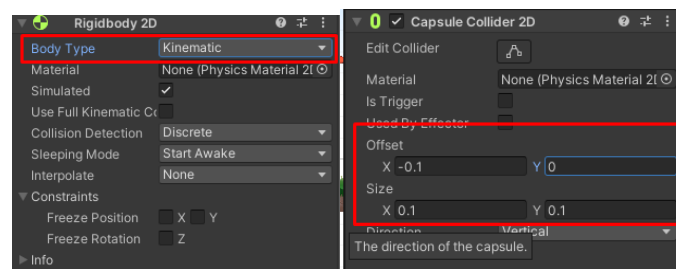


2. Pada inspector di *component transform* pada scale atur lah untuk ukuran *enemy*.



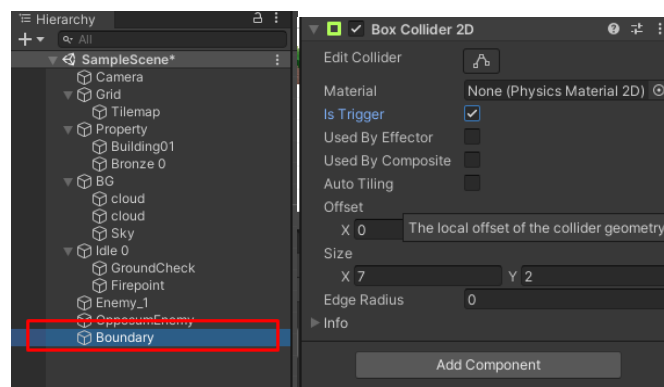
Gambar 10.12 Scalling

3. Tambahkan komponen *Rigidbody 2D* dan *Collider* dengan konfigurasi seperti berikut



Gambar 10.13 Komponen *Enemy*

4. Buat hierarki *Boundary* dan tambahkan komponen *Box Collider 2D*. Atur pada inspector.



Gambar 10.14 Menambah *Boundary*

5. Buat file script *Enemy\_Behavior* di dalam folder Script dan tuliskan script seperti dibawah. Kemudian drag dan drop ke komponen *OpposumEnemy*.

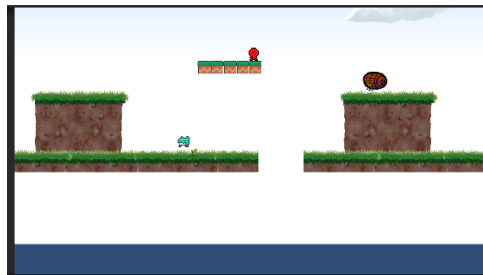
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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 OnTriggerEnter2D(Collider2D collision) {  
        transform.localScale = new Vector2(-  
transform.localScale.x, transform.localScale.y);  
    }  
}
```

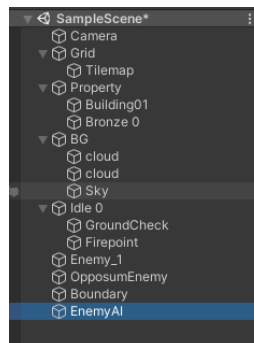
#### 6. Jalankan program



Gambar 10.15 *Enemy* dan *Boundary*

### C. Enemy AI

#### 1. Tambahkan enemy dari aset dan beri nama *EnemyAI*



Gambar 10.16 Menambahkan *EnemyAI*

#### 2. Buat file script *Enemy\_AI* di dalam folder *Script* dan tuliskan scriptnya

```
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").transform;  
    }  
}
```

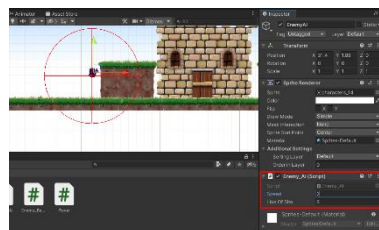




```
initialPosition=GetComponent<Transform>().position;
}

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. Pada hierarchi EnemyAI, pada script Enemy\_AI drag kemudian drop pada hierarchi EnemyAI. Setelah iu pada inspector atur pada script dan object seperti dibawah.



Gambar 10.17 Menambahkan Komponen *Enemy3*



## D. Respawn

1. Buka pada folder script>Player.cs. Tambahkan variable pada class player

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

2. Pada fungsi awake tambahkan kode seperti dibawah

```
private void Awake()  
{  
    rb = GetComponent<Rigidbody2D>();  
    animator = GetComponent<Animator>();  
  
    respawn_loc = transform.position;  
}
```

3. Pada fungsi update, tambahkan kode seperti berikut

```
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();  
    }  
}
```



4. Tambahkan fungsi playagain di script seperti berikut

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

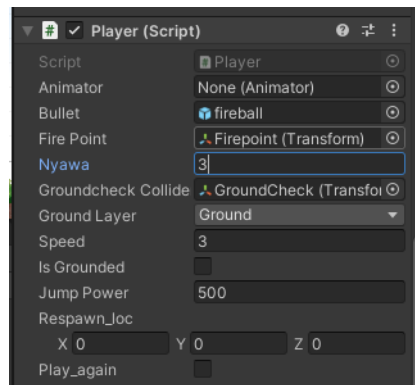
5. Buat file script *Enemy\_Attacked* di dalam folder Script dan tuliskan kode seperti berikut

```
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;
            }
        }
    }
}
```



6. Atur nyawa *point* menjadi 3



Gambar 10.21 Mengatur Nyawa

7. Jalankan game



Gambar 10.22 Run Game



## E. Link Github Pengumpulan

[https://github.com/farhan2153/2118139\\_PRAK\\_ANIGAME.git](https://github.com/farhan2153/2118139_PRAK_ANIGAME.git)

## 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

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