



## TUGAS PERTEMUAN: 10

### ENEMY AI AND ATTACK

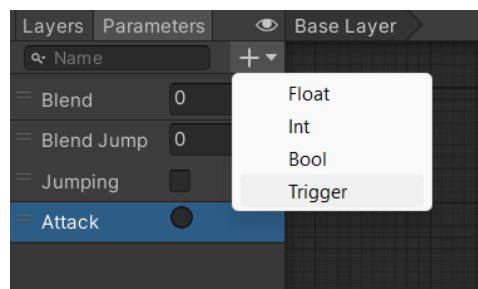
NIM	:	2118128
Nama	:	Arika Nindayu Saputri
Kelas	:	D
Asisten Lab	:	WISANDO BERLIAN PANDENSOLANG (2218095)

#### 1.1 Tugas 1 : Membuat Enemy AI dan Attack

##### 1.1 Langkah Langkah Membuat Mekanisme Attack, Enemy Behavior dan Enemy AI

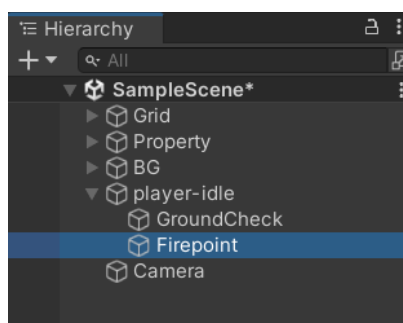
###### A. Mekanisme Attack

1. Buka *Project Unity* pada 2118128\_BAB9. Dan buat 1 buah parameter dengan tipe data *Trigger* dan rename menjadi *Attack*.



Gambar 1.1 Tambah Parameter *Attack*

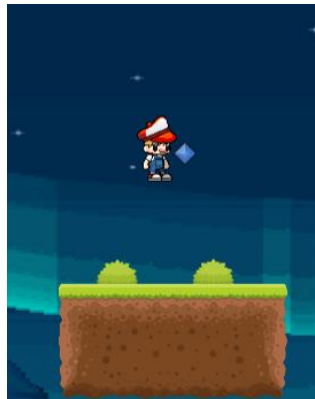
2. Pada *Hierarchy* buat *GameObject* dan di dalam *player\_idle* dan beri nama *Firepoint*.



Gambar 1.2 *GameObject* Dengan Nama *Firepoint*

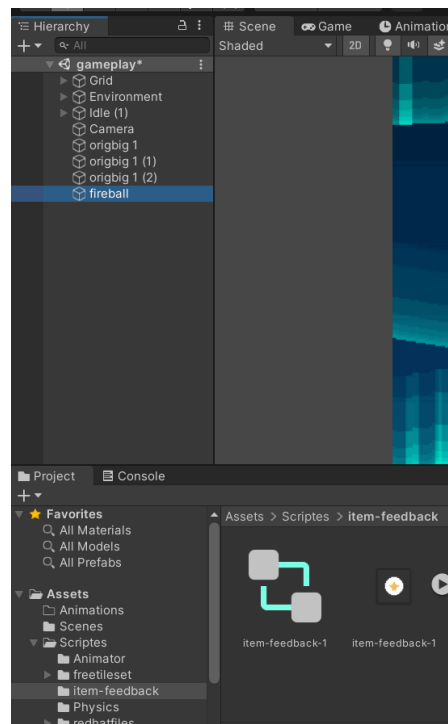


3. *Inspector* pada *Firepoint*, ubah *Icon* menjadi seperti pada gambar dibawah ini. Dan posisikan tepat di depan *player*.



Gambar 1.3 Ubah *Icon Firepoint*

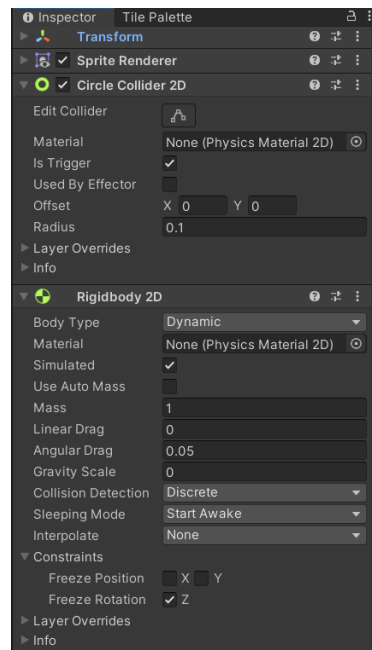
4. *Drag and drop* 'portal\_yellow' ke dalam *Hierarchy YellowBullet*.  
Portal\_yellow ini akan digunakan sebagai peluru.



Gambar 1.4 *Drag and Drop Bullet* ke *fireball*



5. Add Component Circle Collider 2D dan Riggibody 2D pada *Hierarchy YellowBullet*. Dan atur sesuai dengan gambar dibawah ini.



Gambar 1.5 *Inspector YellowBullet*

6. Buka *script Player* dan tambahkan *source code* berikut di dalam kelas *Player* paling atas.

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

7. Tambahkan *script* untuk peluru dan letakkan dibawah fungsi *fixedUpdate*.

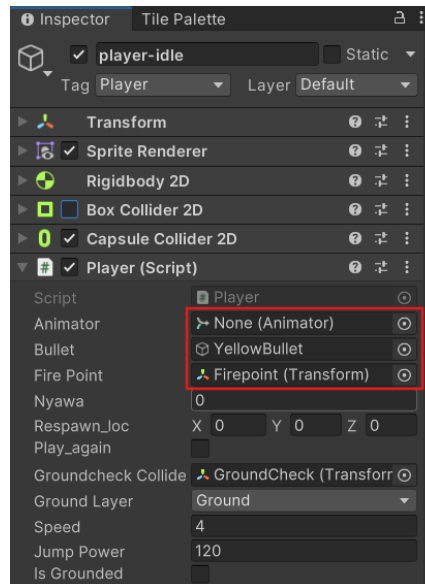
```
IEnumerator Attack() {  
    animator.SetTrigger("Attack");  
    yield return new WaitForSeconds(0.25f);  
    float direction = 1f;  
    GameObject yellowbullet = Instantiate(bullet,  
    firePoint.position, Quaternion.identity);  
  
    yellowbullet.GetComponent<Rigidbody2D>().velocity =  
    new Vector2(direction * 10f, 0);  
    Destroy(yellowbullet, 2f);  
}
```

8. Dan tambahkan *code* berikut pada *Function Void Update*.

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

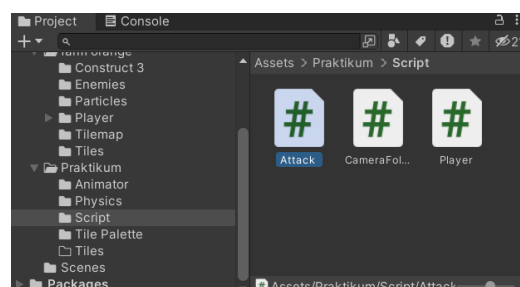


9. Pada *inspector* player-idle atur *component* *Player* (*Script*), ubah *Bullet* menjadi *YellowBullet*. Dan *Firepoint* menjadi *Firepoint* (*Transform*).



Gambar 1.6 Mengatur *Inspector* *Player-idle*

10. Tambahkan file *script* dengan nama *Attack* di dalam folder *script*.



Gambar 1.7 Membuat File *Script* *Attack*

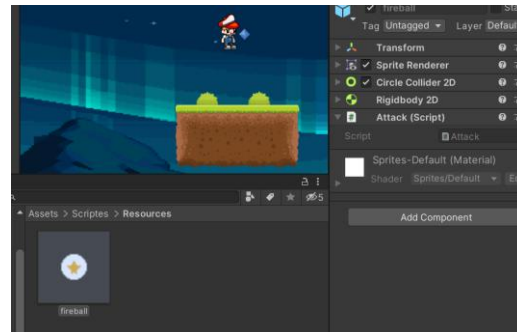
11. Tambahkan *script* *Attack* dibawah ini.

```
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);
        }
    }
}
```

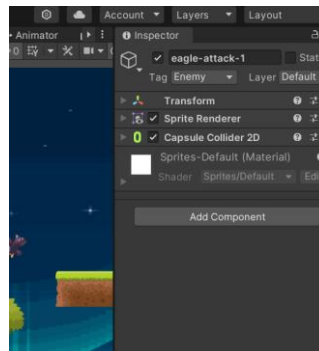


12. Masukkan file *script Attack* ke dalam *Hierarchy YellowBullet*.



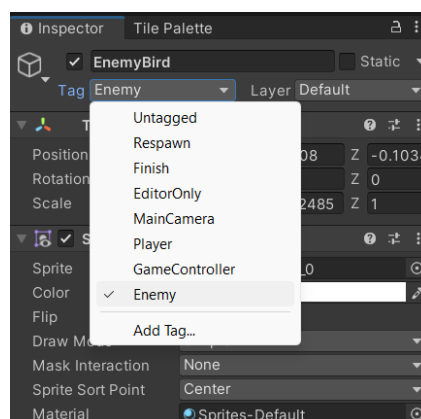
Gambar 1.8 Drag and Drop Script Attack ke YellowBullet

13. Kemudian tambahkan *component Capsule Collider 2D* pada *inspector* 'EnemyAI\_Bird'. Dan pada component *Sprite Renderer* pastikan *sprite* sudah di isi dengan *enemy* yang akan digunakan.



Gambar 1.9 Add Component Capsule Collider 2D

14. Add tag dengan nama *Enemy* pada *inspector* 'EnemyAI\_Bird'. Dan gunakan *tag Enemy* pada 'EnemyAI\_Bird'.



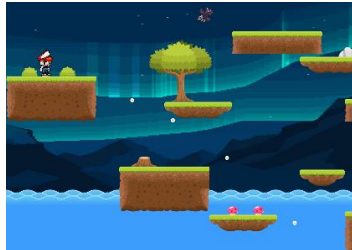
Gambar 1.10 Ubah Tag Pada EnemyAI\_Bird



16. Ubah *script* pada *script* peluru dengan *code* dibawah ini untuk membuat peluru dapat ditembakkan sesuai posisi dari *player*.

```
float direction = facingRight ? 1f : -1f;
```

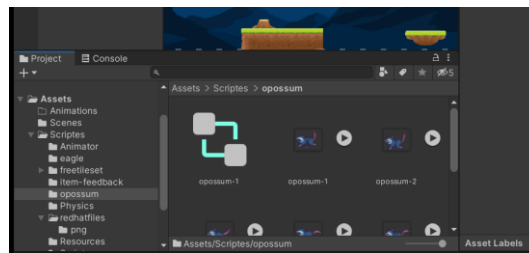
17. Tampilan pada saat di *play*.



Gambar 1.11 Tampilan Saat Di *Play*

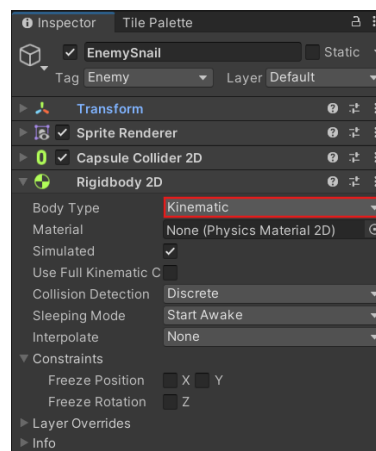
## B. Enemy Behavior

1. *Create Empty* pada *Hierarchy* dengan nama “EnemySnail”. Dan berikan *component* Sprite Renderer dan Capsule Collider 2D. Lalu atur tag menjadi *Enemy*.



Gambar 1.12 *Create Empty* Dengan Nama *EnemySnail*

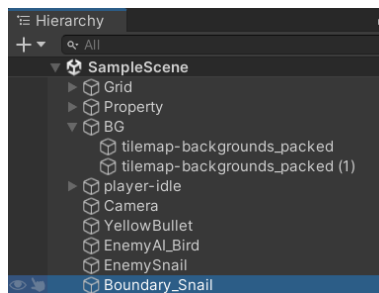
2. Tambahkan *component* Rigidbody 2D pada dan atur *Body Type* nya menjadi ‘Kinematic’.



Gambar 1.13 Atur *Component* Pada *EnemySnail*

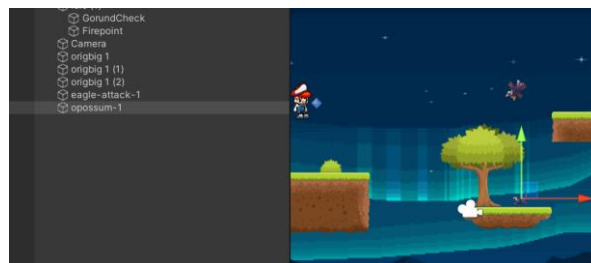


3. *Create Empty* pada *Hierarchy* dan *rename* menjadi “Boundary\_Snail”.



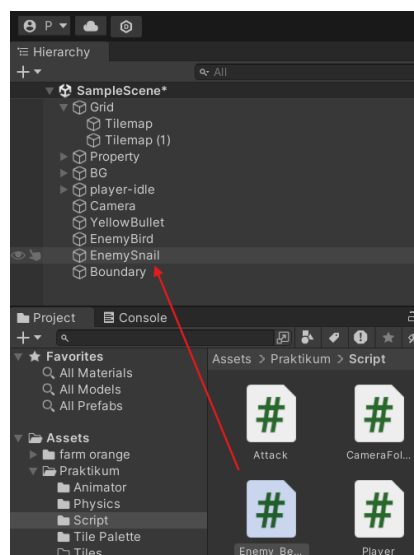
Gambar 1.14 *Create Empty* Dengan Nama Boundary\_Snail

4. *Add Component* Box Collider 2D pada Boundary\_Snail, atur posisi *boundary* dan centang *Is Trigger*.



Gambar 1.15 Atur *Boundary* Pada Boundary\_Snail

5. Buat sebuah file *script* didalam folder *Script* beri nama “Enemy\_Snail”. Kemudian *drag and drop* ke dalam *Hierarchy* EnemySnail



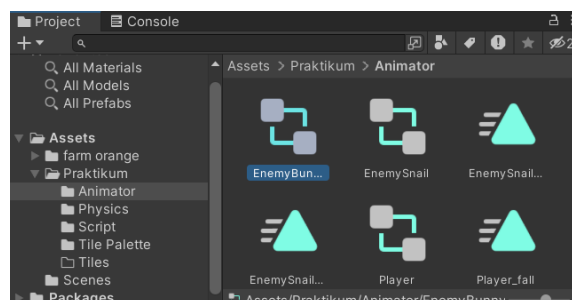
Gambar 1.16 *Drag and Drop* Script *Enemy\_Snail* ke *EnemySnail*



6. Tambahkan *script* Enemy\_Snail dibawah ini.

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

7. Buat File *Animator Controller* di dalam folder *Animator* dan beri nama “EnemySnail”.

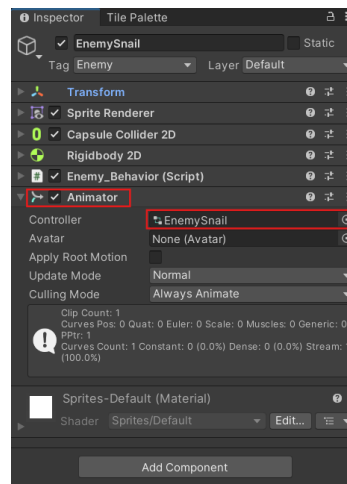


Gambar 1.17 *Animator Controller* Bernama EnemyBunny





8. *Add Component Animator* pada EnemySnail dan atur *Controller* nya menjadi EnemySnail.



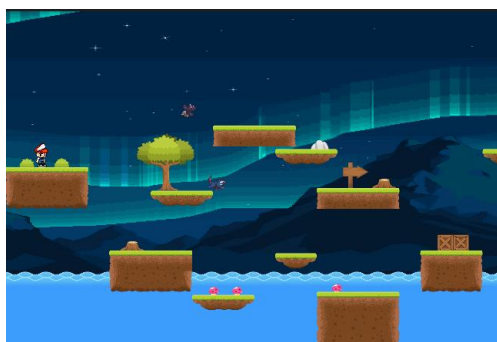
Gambar 1.18 *Add Component Animator* di EnemySnail

9. Tampilan pada saat di *play*.



Gambar 1.19 Tampilan Saat Di *Play*

10. Ulangi langkah 1-9 dan tekan icon play. Gamabr dibawah ini adalah tampilan pada saat di *play*.

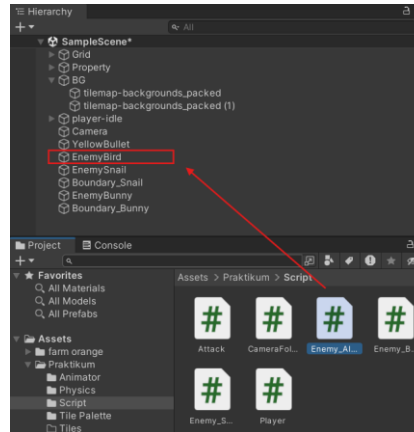


Gambar 1.20 Tampilan Saat di *Play*



### C. Enemy AI

1. Masukkan *script* Enemy\_AI\_Bird ke dalam *Hierarchy* EnemyAI\_Bird.



Gambar 1.21 *Drag and Drop Script* ke EnemyAI\_Bird

2. Tambahkan *Script* Enemy\_AI\_Bird dibawah ini.

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

public class Enemy_AI_Bird : 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

    void Start()
    {
        player =
        GameObject.FindGameObjectWithTag("Player").transform;
        initialPosition =
        GetComponent<Transform>().position;
    }

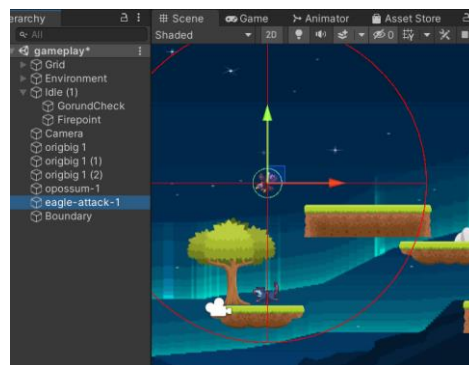
    void Update()
    {
        float distanceToPlayer =
        Vector2.Distance(player.position,
        transform.position);
        if (distanceToPlayer < lineOfSite)
        {
            transform.position =
            Vector2.MoveTowards(this.transform.position,
            player.position, speed * Time.deltaTime);
        }
        else
        {

```



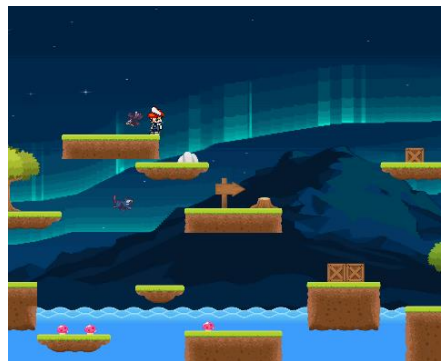
```
transform.position  
Vector2.MoveTowards(transform.position,  
initialPosition, speed * Time.deltaTime);  
}  
}  
private void OnDrawGizmosSelected()  
{  
    Gizmos.color = Color.red;  
    Gizmos.DrawWireSphere(transform.position,  
lineOfSite);  
}  
}
```

3. Buka *Enemy\_AI\_Bird (Script)* pada *inspector* *EnemyAI\_Bird* dan atur seperti pada gambar dibawah ini.



Gambar 1.22 Atur *Inspector* *EnemyAI\_Bird*

4. Tampilan pada saat di *play*.



Gambar 1.23 Tampilan Saat di *Play*



5. Ulangi langkah 1-3 dan tekan icon play. Gamabr dibawah ini adalah tampilan pada saat di *play*.



Gambar 1.24 Tampilan Saat di *Play*

#### D. Respawn

1. Buka file *script* (Player.cs) dan tambahkan variabel nyawa seperti dibawah ini. Letakkan di dalam kelas *player*.

```
public int nyawa;  
[SerializeField] Vector3 respawn_loc;  
public bool play again;
```

2. Tambah *code* dibawah ini untuk mengatur posisi *respawn* agar sesuai dengan posisi awal permainan dimulai.

```
respawn_loc = transform.position;
```

3. Kemudian tambahkan kode dibawah ini di dalam *void update* Player.cs agar ketika nyawa *player* dibawah 0 maka akan melakukan respawn

```
if (nyawa < 0){  
    playagain();}
```

4. Selanjutnya tambahkan kode berikut dibawah *code* sebelumnya agar ketika *player* jatuh dibawah *platform* akan melakukan *respawn*

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

5. Tambahkan fungsi *playagain()* dalam script Player.cs

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

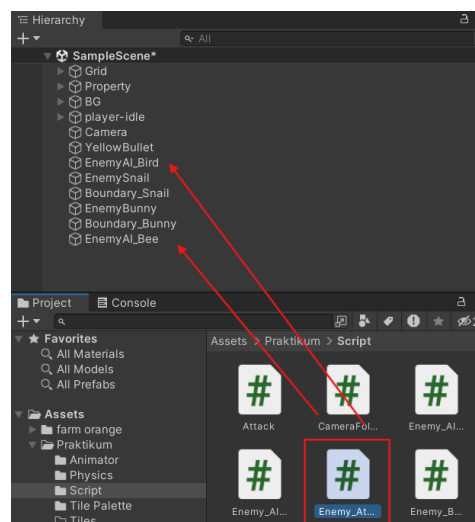


6. Tambah file *script* (Enemy\_Attacked.cs) dan isikan *source code* dibawah ini.

```
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;
            }
        }
    }
}
```

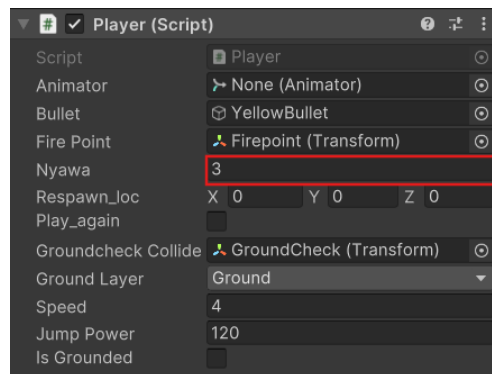
7. *Drag and drop script* Enemy\_Attack ke dalam *Hierarchy* 'EnemyAI\_Bee dan 'EnemyAI\_Bird'.



Gambar 1.25 *Drag and Drop Script*



8. Pada *inspector* player-idle di dalam *Player (Script)* berikan nilai pada Nyawa menjadi 3.



Gambar 1.26 Memberikan Nyawa Pada Player

A.