



## TUGAS PERTEMUAN: 10

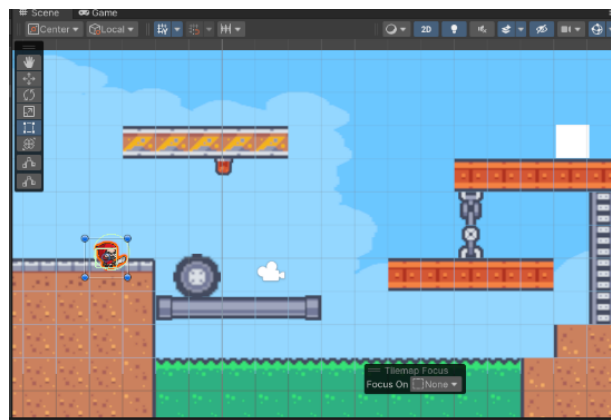
### RESPAWN AND ENEMY AI ATTACK

|                    |   |                              |
|--------------------|---|------------------------------|
| <b>NIM</b>         | : | 2118117                      |
| <b>Nama</b>        | : | Yohanes Yudha Saputra Bangko |
| <b>Kelas</b>       | : | C                            |
| <b>Asisten Lab</b> | : | Nayaka Apta N (2218102)      |

#### 10.1 Tugas 10 : Respawn and Enemy AI Attack

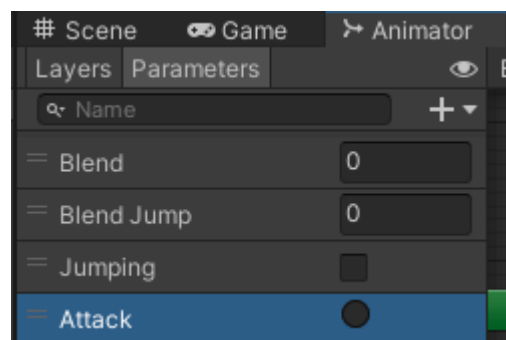
##### A. Membuat Mekanisme Attack

1. Buka project sebelumnya pada unity



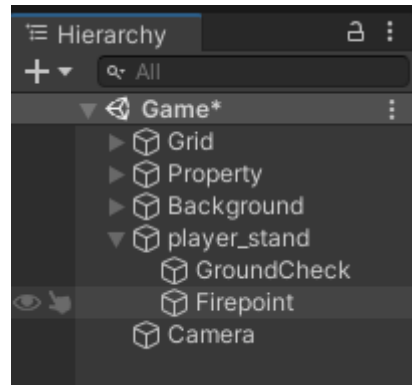
Gambar 1.1 Membuka project sebelumnya

2. Tambahkan variabel baru di tab parameters dengan tipe data Trigger



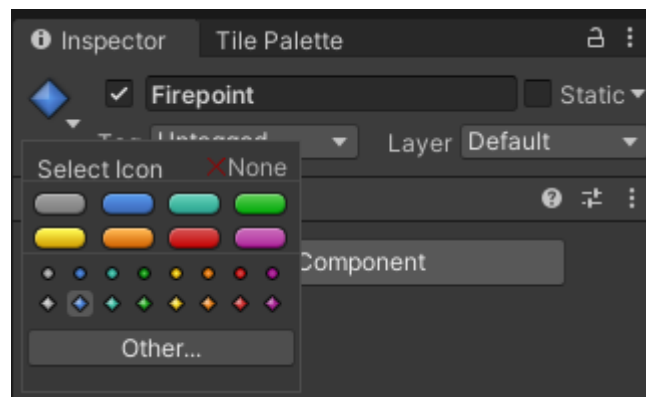
Gambar 1.2 Membuat variabel baru

3. Setelah itu pada hierarchy buat lah object baru dengan nama Firepoint.



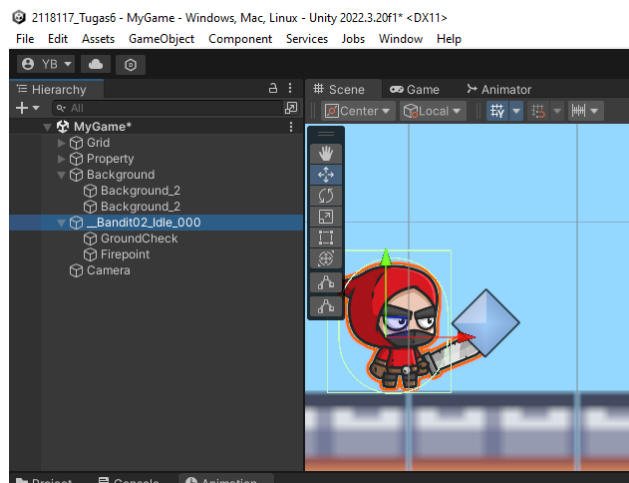
Gambar 1.3 Membuat Firepoint

4. Lalu ubahlah icon Firepointnya



Gambar 1.4 Mengubah icon firepoint

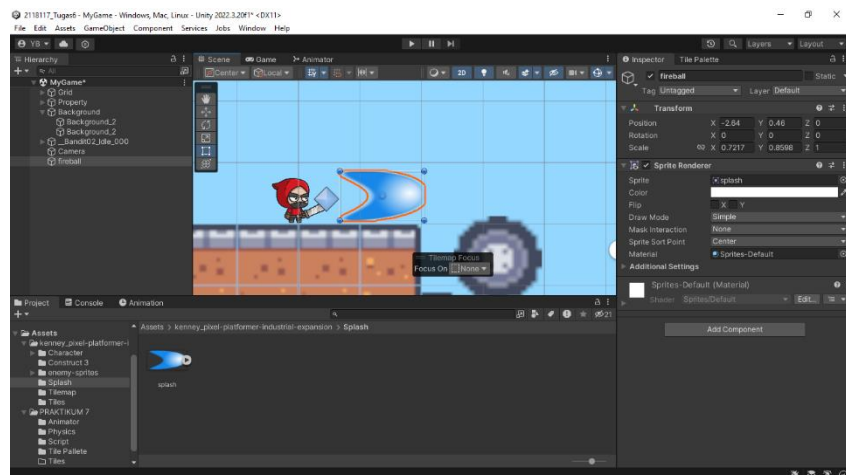
5. Taruh firepoint di depan karakter seperti berikut.



Gambar 1.5 Meletakkan firepoint

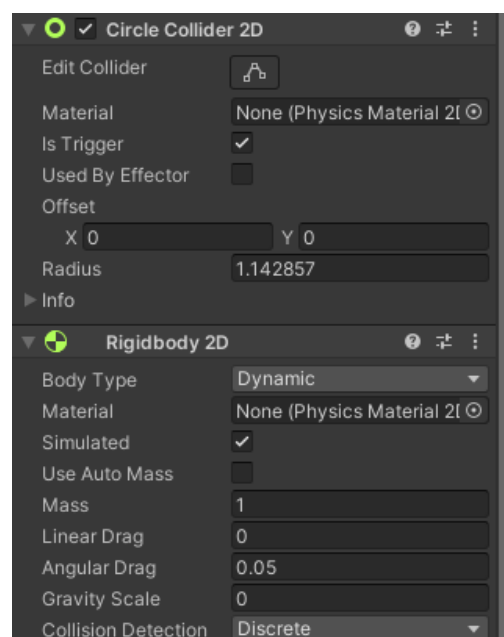


6. Drag and drop assets splash ke dalam scene dan beri nama menjadi fireball.



Gambar 1.6 Drag and drop splash

7. Pada objek fireball beri component berupa circle collider dan rigidbody 2d.



Gambar 1.7 Menambah component

8. Masukkan source code di bawah ini untuk bagian Player.cs.

```
public GameObject bullet;  
public Transform firePoint;  
  
//letakkan di bawah void FixedUpdate()  
IEnumerator Attack(){  
    animator.SetTrigger("Attack");  
    yield return new WaitForSeconds(0.25f);  
}
```



```
float direction = facingRight ? 1f : -1f;
float rotationAngle = facingRight ? -90f : 90f;

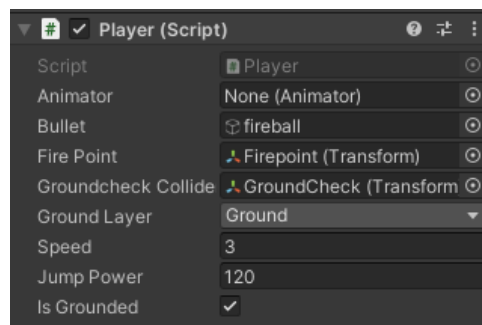
Quaternion rotation = Quaternion.Euler(0, 0,
rotationAngle);

GameObject fireball = Instantiate(bullet,
firePoint.position, rotation);
fireball.GetComponent<Rigidbody2D>().velocity =
new Vector2(direction * 10f, 0);

Destroy(fireball, 2f);}

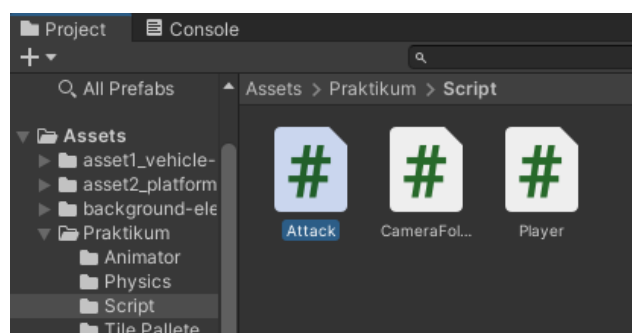
//tambahkan pada void Update()
if (Input.GetKeyDown(KeyCode.C)) {
    StartCoroutine(Attack());
}
```

9. Pada objek *player* bagian *script* di *inspector*, atur *bullet* dan *firepointnya* seperti berikut.



Gambar 1.8 Mengatur *bullet* dan *firepoint*

10. Buat file script baru bernama *Attack*.



Gambar 1.9 *Script Attack*

11. Lalu masukkan *code* berikut pada *file script attack*

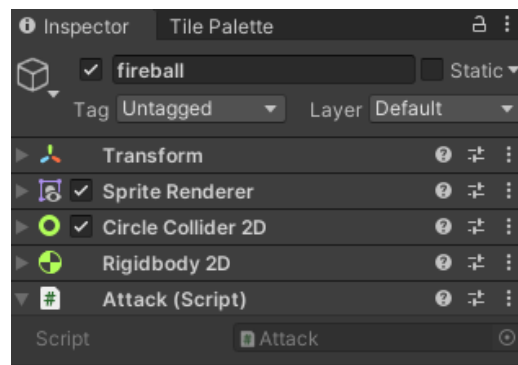
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Attack : MonoBehaviour
{
```



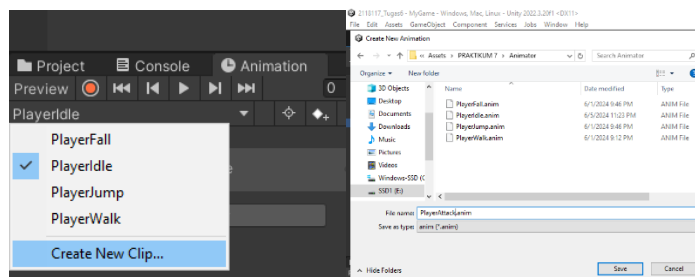
```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.CompareTag("Enemy"))
    {
        Destroy(gameObject);
        Destroy(collision.gameObject);
    }
}
```

12. Drag and Drop file script attack ke dalam objek fireball.



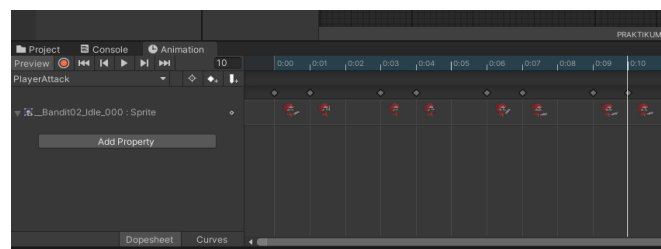
Gambar 1.10 *Drag and drop script*

13. Pada menu animation buat clip baru dan simpan dengan nama PlayerAttack.



Gambar 1.11 *Create new clip*

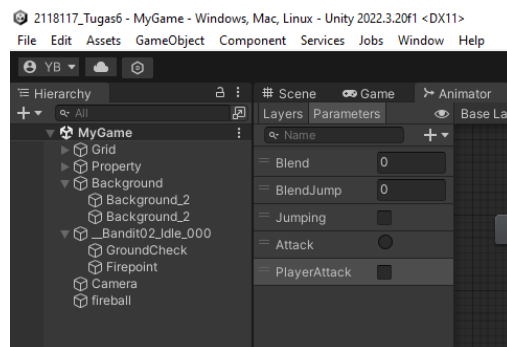
14. Setelah itu drag and drop assets sprites attack ke dalam tab animation sehingga muncul di timeline seperti berikut.



Gambar 1.12 *Drag and drop assets*

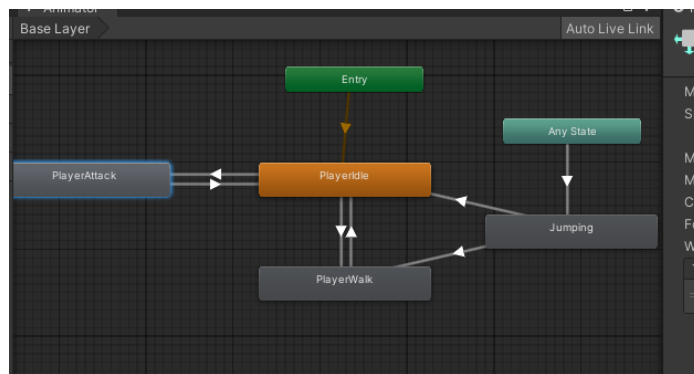


15. Setelah itu buat variabel baru dengan nama PlayerAttack bertipe data Trigger.



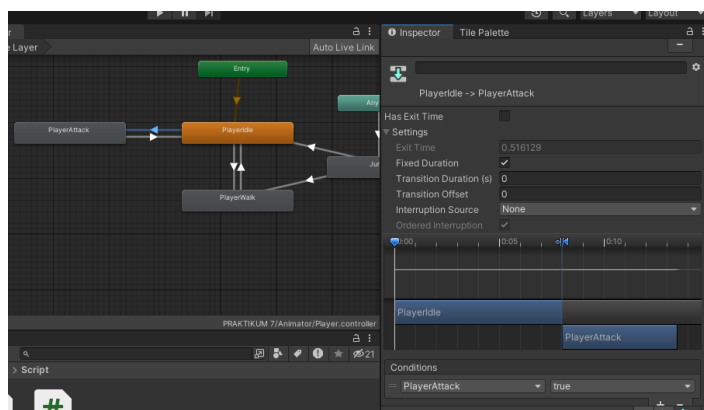
Gambar 1.13 Membuat variabel

16. Make transition antara PlayerIdle dan PlayerAttack begitu juga sebaliknya.



Gambar 1.14 Membuat transisi

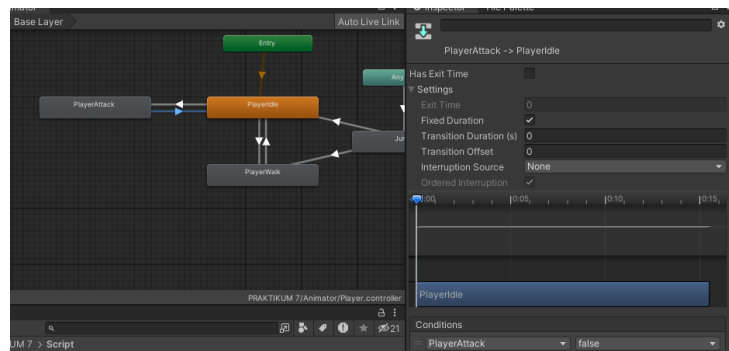
17. Setelah itu pada tanda panah dari PlayerIdle ke PlayerAttack atur conditions dan settings seperti berikut.



Gambar 1.15 Mengatur conditions dan settings



18. Setelah itu pada tanda panah PlayerAttack ke PlayerIdle atur conditions dan settings seperti berikut.



Gambar 1.16 Mengatur conditions dan settings

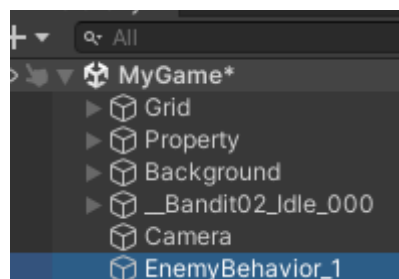
19. Lalu jalankan game maka karakter akan dapat melakukan serangan dan terdapat animasi saat melakukan serangan.



Gambar 1.17 Hasil running

## B. Membuat Enemy Behaviour

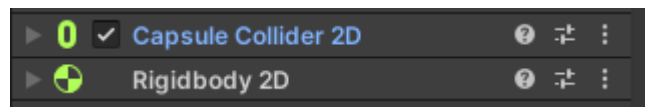
1. Tambahkan objek untuk digunakan sebagai enemy behaviour dengan cara drag and drop ke jendela hierarchy.



Gambar 1.18 Menambahkan *enemy behaviour*



2. Pada objek enemy behaviour tersebut tambahkan comopnent berupa Capsule Collider 2D dan Rigidbody 2D

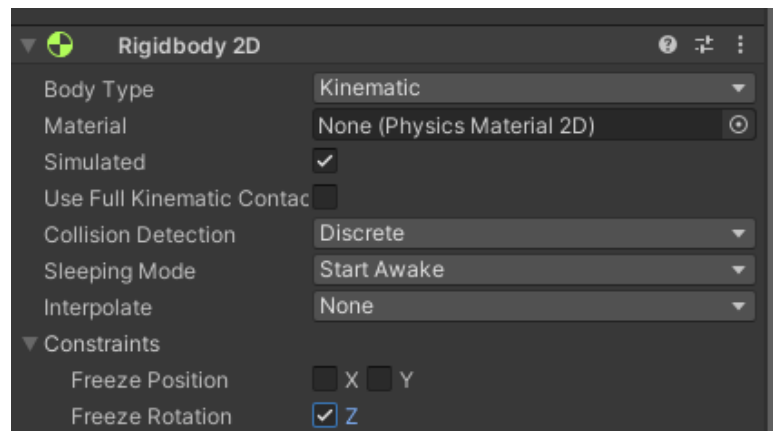


Gambar 1.19 Menambahkan component



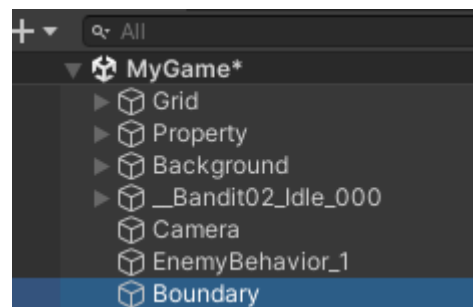


3. Pada komponen Rigidbody 2D ubah body type menjadi Kinematic dan centan Freeze Rotation Z.

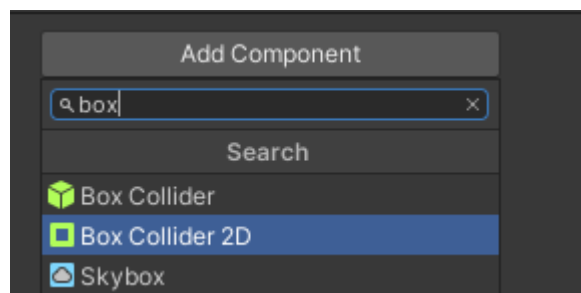


Gambar 1.20 Mengatur Rigidbody 2D

4. Tambahkan objek kosong dengan cara klik kanan pada hierarchy lalu pilih create empty, dan ubah nama menjadi Boundary.

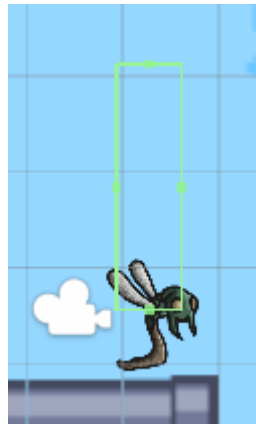


5. Pada objek boundary tersebut, tambahkan komponen Box Collider 2D



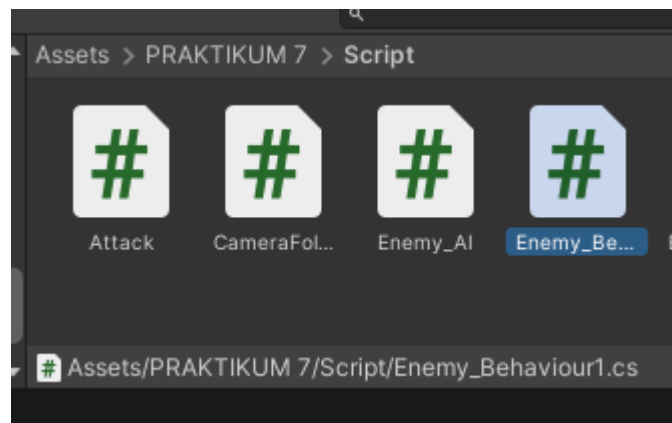


6. Aturlah ukuran collider agar musuh bergerak ke atas tidak terlalu jauh dan juga tidak terlalu dekat.



Gambar 1.21 Mengatur *boundary*

7. Buat file script baru bernama Enemy\_Behavior1



Gambar 1.22 Membuat *script behaviour 1*

8. Tambahkan source code berikut pada file script Enemy\_Behavior.cs

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

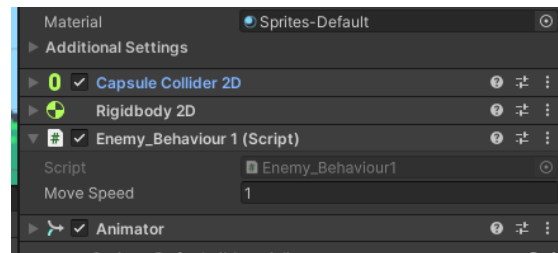
public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 1f;
    Rigidbody2D rb;
    bool isMovingUp = true;

    void Start() {
        rb = GetComponent<Rigidbody2D>();
    }
    void Update() {
        if (isMovingUp) {
            rb.velocity = new Vector2(0f, moveSpeed);
        } else {
            rb.velocity = new Vector2(0f, -moveSpeed);
        }
    }
}
```



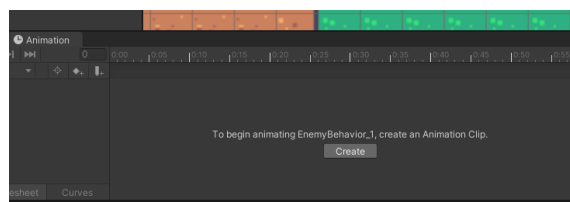
```
}  
private void OnTriggerExit2D(Collider2D  
collision){  
    isMovingUp = !isMovingUp;  
}  
}
```

9. *Drag and drop file script* Enemy\_Behavior1 ke objek Enemy Behaviour 1.



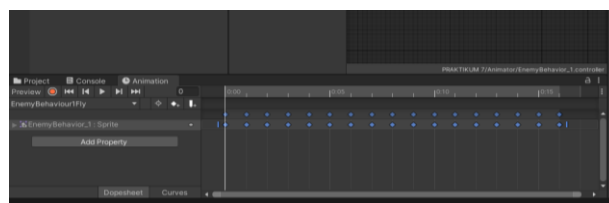
Gambar 1.23 *Drag and drop script*

10. Pada menu animation buat clip baru dengan nama EnemyBehaviour1Fly.



Gambar 1.24 Create new clip

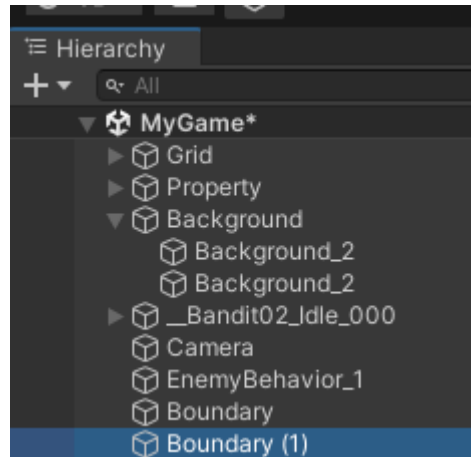
11. *Drag and drop assets sprite* ke dalam tab *animation* sehingga muncul di *timeline* nya.



Gambar 1.25 *Drag and drop dan Make Transitions*



12. Ulangi langkah ke-1 hingga ke-11 untuk objek yang lain yaitu Enemy Behaviour 2. Namun pada boundary buatlah lagi boundary yang baru karena dalam boundary baru ini akan diberi script yang berbeda seperti sebelumnya.



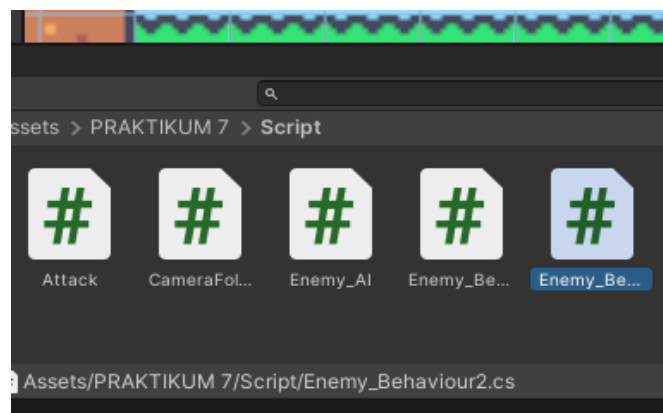
Gambar 1.26 Membuat boundary

13. Atur posisi box colider pada boundary yang baru seperti berikut.



Gambar 1.27 Mengatur posisi boundary

14. Buatlah file script baru untuk enemy behaviour 2 karena pergerakannya berbeda dengan enemy behaviour 1.



Gambar 1.28 Membuat file script enemy behaviour 2



15. Masukkan script berikut untuk enemy behaviour 2.

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

public class Enemy_Behaviour2 : 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);
    }
}
```

16. *Running game* maka akan ada *enemy behaviour* yang bergerak.



Gambar 1.29 Hasil *running*

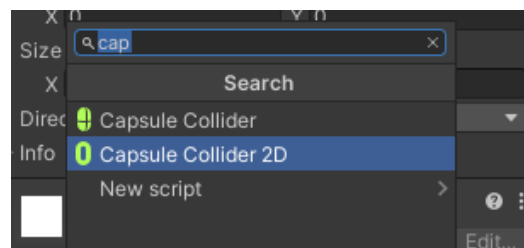
### C. Membuat Enemy AI

1. Tambahkan *asset enemy ai 1* pada *scene game* dan atur posisinya.



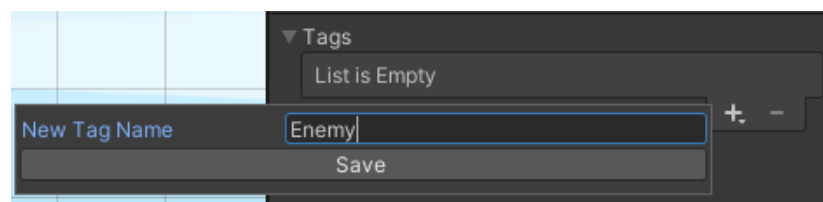
Gambar 1.30 Menambahkan *enemy ai 1*

2. Pada objek *enemy ai 1* tersebut tambahkan komponen *capsule collider 2D*



Gambar 1.31 Menambah komponen

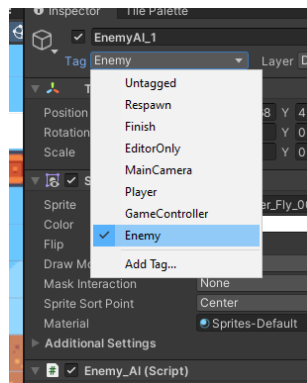
3. Setelah itu tambahkan tag dengan nama *enemy*





Gambar 1.32 Menambahkan *tag*

4. Pada objek *enemy ai 1* pilihlah *tag enemy*



Gambar 1.33 Memilih *tag enemy*

5. Buatlah *file script* dengan nama *Enemy\_AI*.



Gambar 1.34 Membuat *file script enemy ai*

6. Berilah *source code* berikut ke dalam *script Enemy\_AI* tersebut.

```
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
    private bool facingRight = true; // Menyimpan arah hadap musuh

    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);
        FacePlayer();
    }
    else
    {
        // Musuh kembali ke posisi awal
        transform.position =
Vector2.MoveTowards(transform.position,
initialPosition, speed * Time.deltaTime);
        FaceInitialPosition();
    }
}

private void FacePlayer()
{
    if (player.position.x > transform.position.x
    && !facingRight)
```



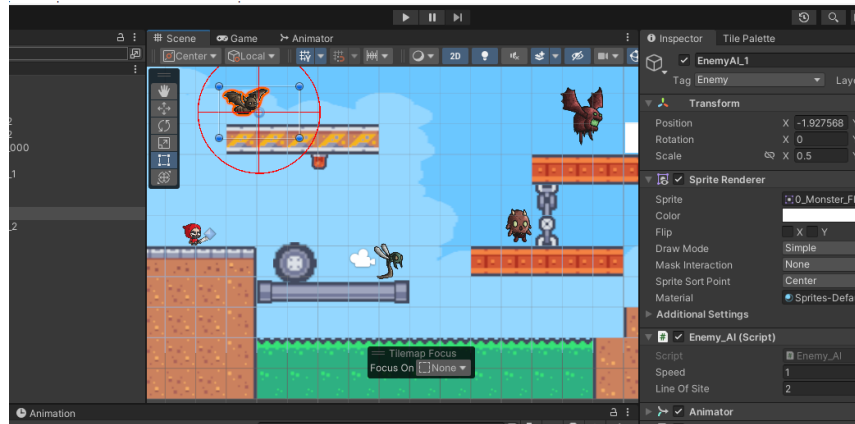


```
{
    Flip();
}
else if (player.position.x <
transform.position.x && facingRight)
{
    Flip();
}

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);
}}
```

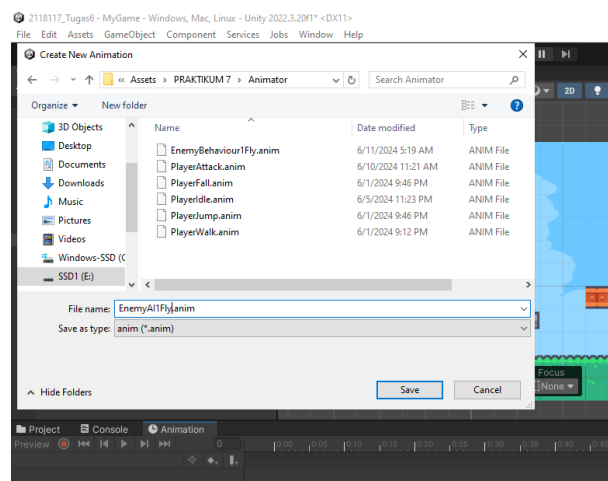


7. *Drag and drop script* ke dalam objek Enemy AI 1 lalu atur *line of site* serta *speednya* seperti berikut.



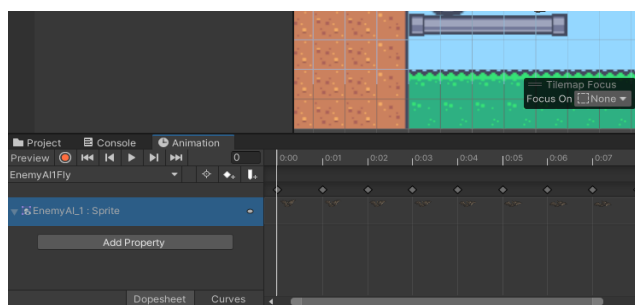
Gambar 1.35 Mengatur *line of site* dan *speed*

8. Pada menu *animation* buat *clip* baru dengan nama *EnemyAIFly* seperti berikut.



Gambar 1.36 *Create new clip*

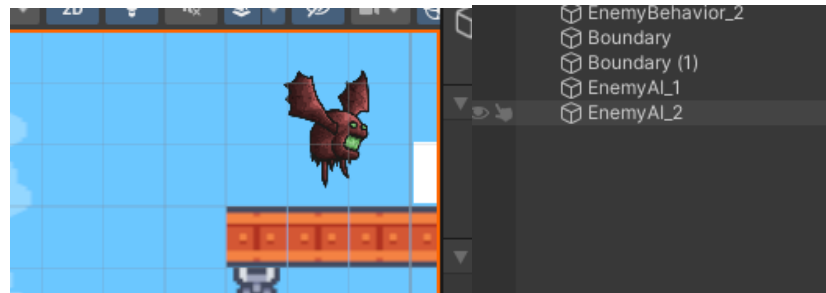
9. Setelah itu *drag and drop sprite enemy ai 1* ke dalam menu *animation* sehingga muncul di *timeline*.



Gambar 1.37 *Drag and drop assets sprite*



10. Tambahkan lagi objek EnemyAI\_2 lalu ulangi langkah 1 hingga 9.



Gambar 1.38 Enemy AI 2

#### D. Respawn

1. Tambahkan source code berikut pada file script Player.cs

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

2. Tambahkan source code berikut pada void Awake()

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

3. Buat void playagain() dan letakkan di bawah void Awake()

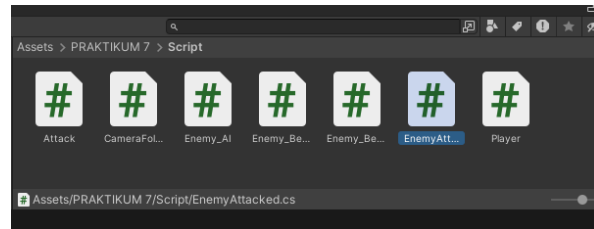
```
void playagain(){  
    if (play_again == true){  
        nyawa = 3;  
        transform.position = respawn_loc;  
        play_again = false;  
    }  
}
```

4. Tambahkan source code berikut pada void Update()

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



5. Buat file script baru bernama *EnemyAttacked*.



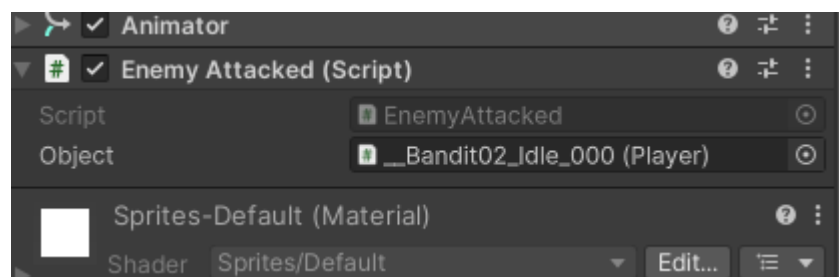
Gambar 1.39 *Enemy Attacked*

6. Tambahkan *source code* berikut pada *script EnemyAttacked.cs*

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

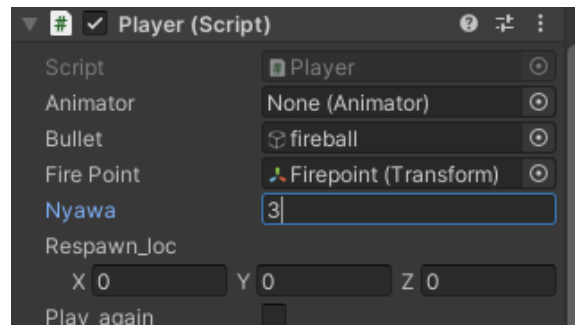
7. *Drag and drop script* pada *enemy behaviour* dan ubah objek ke *player*.



Gambar 1.40 Mengubah objek ke *player*

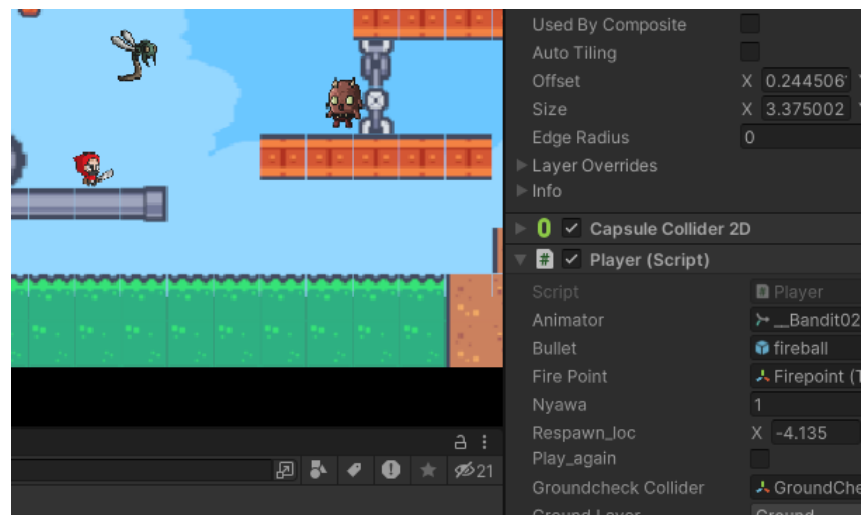


8. Pada objek *Player* ubah nyawa menjadi 3



Gambar 1.41 Mengubah nyawa

9. Jalankan *game* maka ketika *player* mengenai *enemy* maka nyawanya berkurang 1 dan jika nyawanya sama dengan 0 maka *player* akan *respawn*.



Gambar 1.42 Hasil *running*



## F. Kuis Bab 10

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    public int attackDamage = 10;
    public string enemyTag = "Enemy"; // Tag untuk mengidentifikasi musuh

    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))
            {
                // Musuh terkena serangan, kurangi health mereka
                Health healthComponent = hit.collider.GetComponent<Health>();
                if (healthComponent != null)
                {
                    healthComponent.TakeDamage(attackDamage);
                }
            }
        }
    }
}
```

### Analisa :

Pada code di atas Variabel attackRange diubah dari tipe data int menjadi float untuk lebih akurat merepresentasikan nilai jarak serangan. Typo pada InputGetButtonDown diperbaiki menjadi Input.GetButtonDown, dan attacDamage diperbaiki menjadi attackDamage. Penambahan variabel enemyTag memungkinkan identifikasi musuh berdasarkan tag, sehingga hanya musuh yang akan terkena serangan. Dalam metode PerformMeleeAttack, ditambahkan percabangan untuk memastikan bahwa objek yang terkena raycast



memiliki komponen Health, yang bertanggung jawab untuk mengurangi health musuh dan menangani kematian mereka

**G. Link Github Pengumpulan**

<https://github.com/Yohanes-Yudha/2118117-PRAK-ANIGAME>