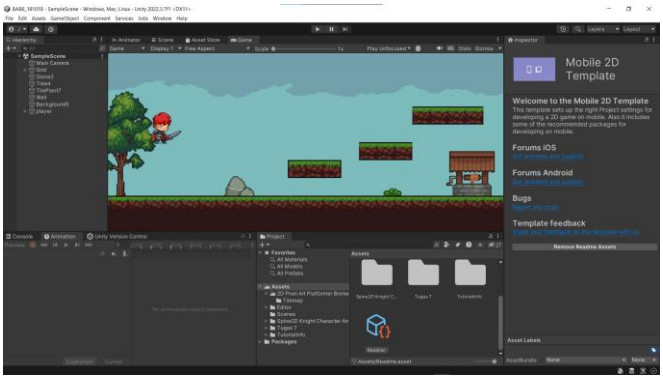
TUGAS PERTEMUAN: 10

Respawn and AI Enemy Attack

NIM	:	1818018
Nama	:	Muhamad Fauzan Nashir
Kelas	:	E
Asisten Lab	:	Difa Fisabilillah (2118052)
Tugas	:	Membuat Enemy AI & Attack

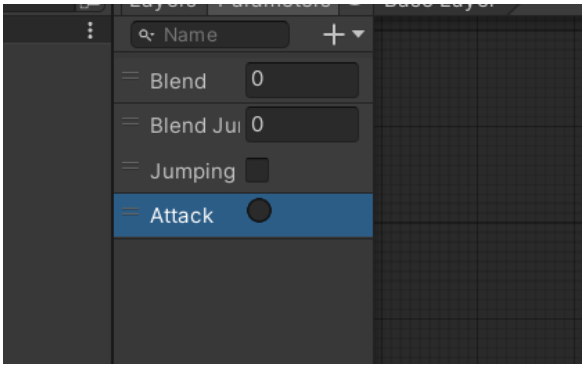
A. Membuat Enemy AI

1. Buka Project Bab 9 Untuk melanjutkannya



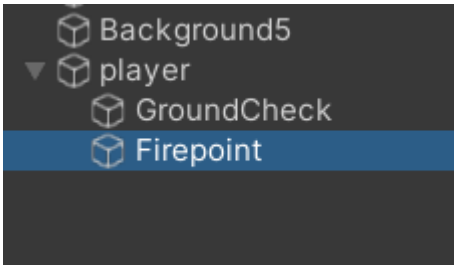
Gambar 1.1 tampilan bab 9

2. Kemudian pada menu Tab **Animator** Tambahkan Parameter Trigger, Rename Menjadi ***Attack***



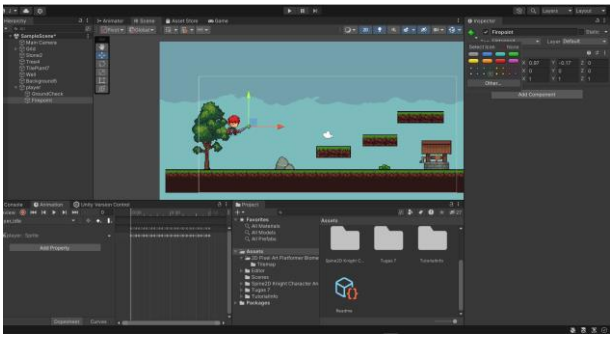
Gambar 1.2 Tampilan menambahkam parameter

3. Setelah menambahkan parameter Attack, Langkah selanjutnya adalah membuat Layer *Game object* baru didalam *player*, Klik kanan pilih *Create Empty* lalu Rename menjadi *Firepoint*



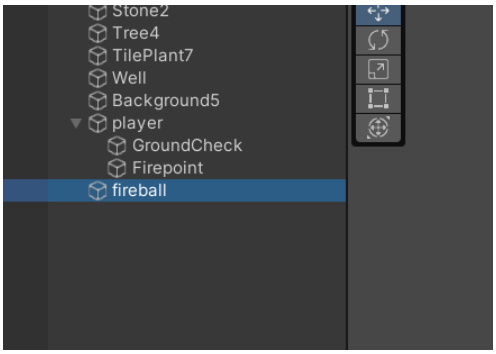
Gambar 1.3 Tampilan membuay firepoint

4. Pada menu *Hierarchy* klik Firepoint untuk setting pada Inspector, Ubah *Icon* Menjadi titik, atur letak titik didepan player



Gambar 1.4 memberi symbol firepoint

5. Pada menu Hierarchy Tambahkan item-feedback-1, di folder Sprites > Fx > item-feedback-2 , *rename* menjadi *fireball*

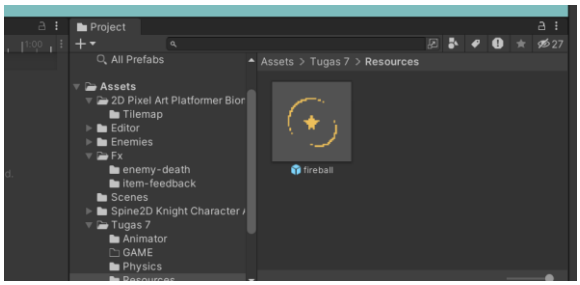


Gambar 1.5 memasukan fireball

6. Klik item-feedback-2 untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini

Gambar 1.6 memberi collider 2d

7. Buat Folder baru *Resources* di menu Tugas7, kemudian drag and drop fireball kedalam folder Resources, dan hapus *fireball* pada *Hierarchy*

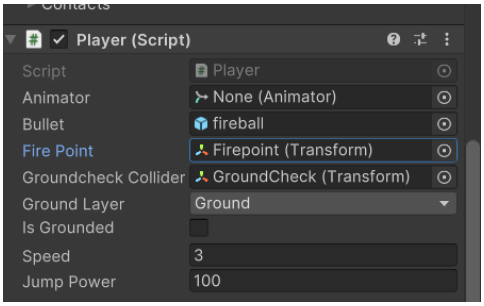


Gambar 1.7 drag and drop fireball

8. Pada Script Player Tambahkan Script dibawah ini

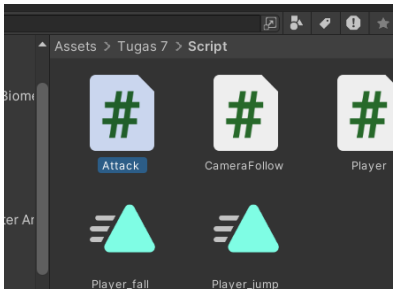
```
#Tambahkan dibawah fungsi fixedUpdate
IEnumerator Attack()
{

     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);
}
#Tambakan pada Function Void Update
if (Input.GetKeyDown(KeyCode.C))
{
    StartCoroutine(Attack());}
```

9. Pada Inspector Player, Ubah seperti dibawah ini, Dimana Bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak pertama



Gambar 1.8 mengatur script pada unity

10. Buat Script Attack pada folder Script

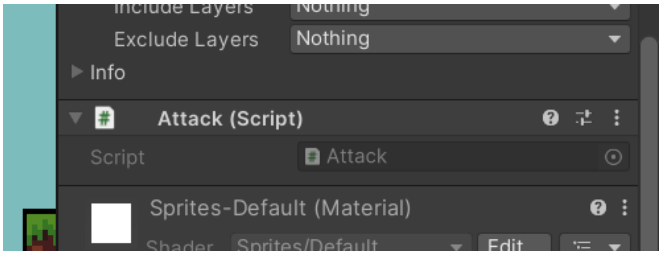


Gambar 1.9 membuat script attact

11. Tambahkan Script Attack dibawah ini

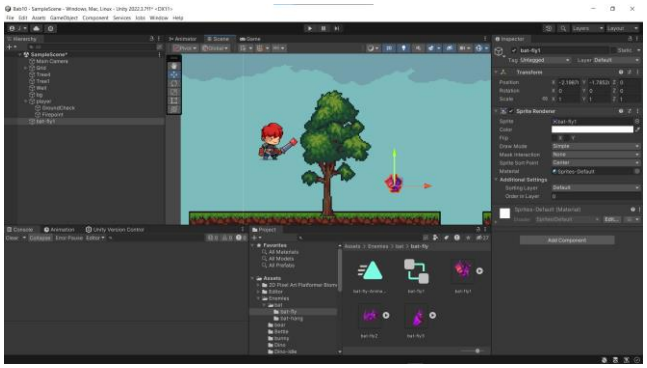
```
        using System.Collections;
        using System.Collections.Generic;
        using UnityEngine;

        public class Attack : MonoBehaviour
        {
            private   void   OnTriggerEnter2D(Collider2D
        collision)
            {
                if
        (collision.gameObject.CompareTag("Enemy"))
                {
                    Destroy(gameObject);
                    Destroy(collision.gameObject);
                }
            }
}
```

12. Didalam folder resource Tambahkan Script Attack di Prefab fireball, dengan cara Klik fireball kemudian pada menu Inspector arahkan Script Attack kedalam Inspector



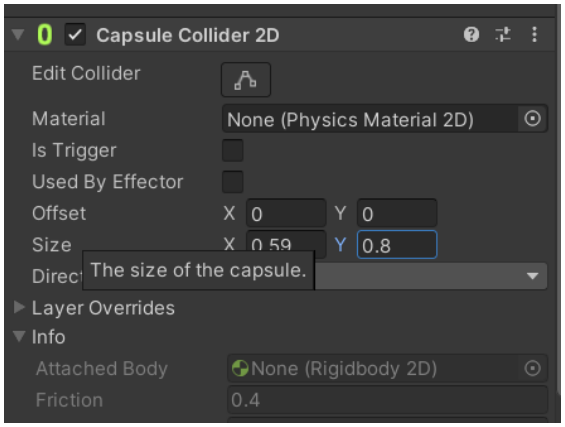
Gambar 1.10 memasukan scrip attack kedalam inspector

13. Tambahkan Enemy vulture pada hierarchy di folder Sprites,



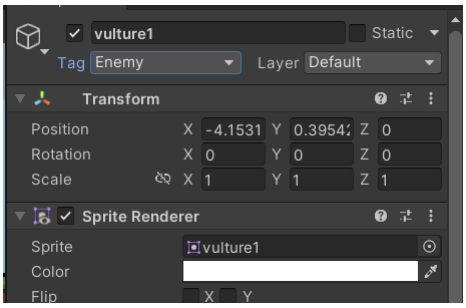
Gambar 1.11 menambahkan objek vulture

14. Kemudian klik pada *vulture*, lalu pada menu tab *inspector* tambahkan *collider 2D* untuk mendeteksinya



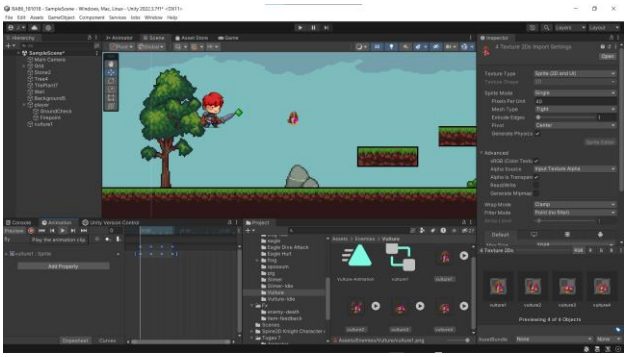
Gambar 1.12 menambahkan *collider 2d*

15. Tambahkan Tag *Enemy* dengan cara Pilih Add Tag, kemudian add tag to the list, Tuliskan *Enemy*



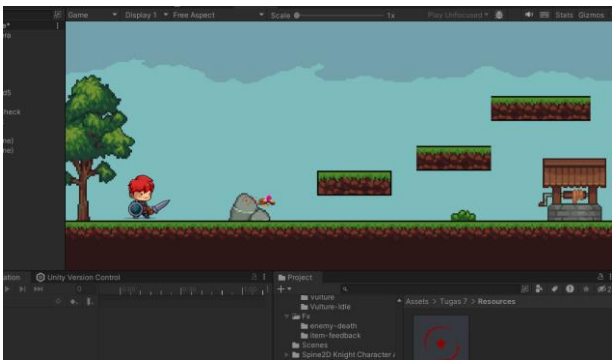
Gambar 1.13 menambahkan enmy tag

16. Tambahkan animasi dengan cara klik animation kemudian masukan *asset enemy*.



Gambar 1.14 Menambahkan animasi pada *enemy*

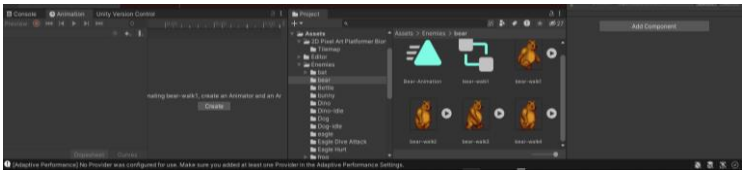
17. Tembak *Enemey* dengan menekan Tombol C untuk menghancurkan musuh



Gambar 1.15 menjalankan program

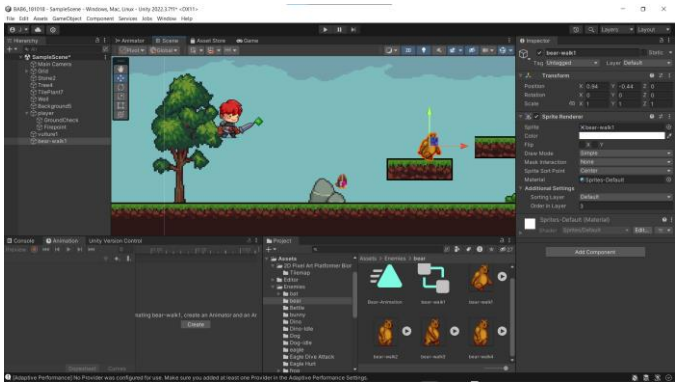
B. Enemy Behavior NPC

1. Cari sebuah sprite pack Bernama enemy dan buka folder bernama “bear”



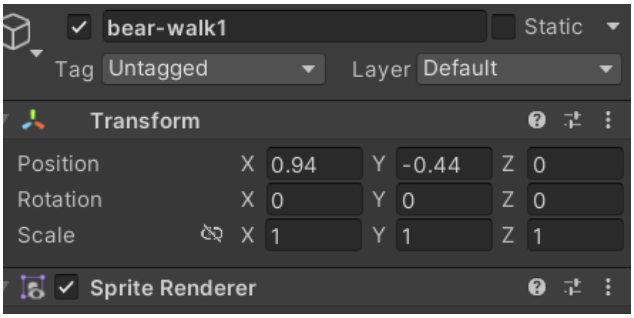
Gambar 1.16 membuka folder asset

2. Tambahkan “bear” ke Hierarchy.



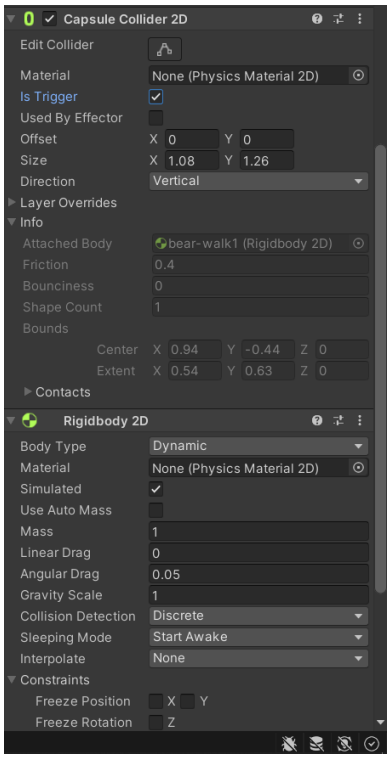
Gambar 1.17 menambahkan asset bear

3. Pada inspector atur transform scale menjadi seperti berikut



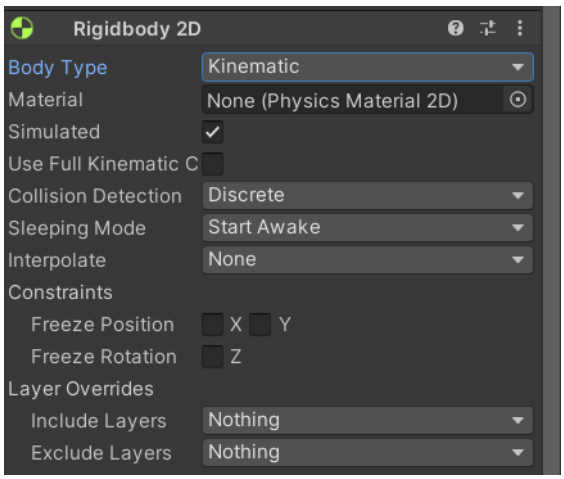
Gambar 1.18 mengatur transform

4. Tambahkan sebuah komponen bernama Capsule Colider 2D dan Rigidbody dalam inspector game objek dog-1



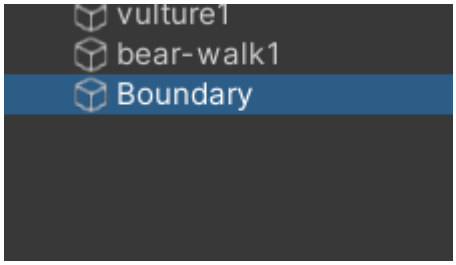
Gambar 1.19 menambahkan collider dan rigi body 2d

5. Atur sedikit collider tersebut seperti ukurannya diubah jika terlalu besar, dan pada *Body Type* Ubah menjadi *Kinematic*



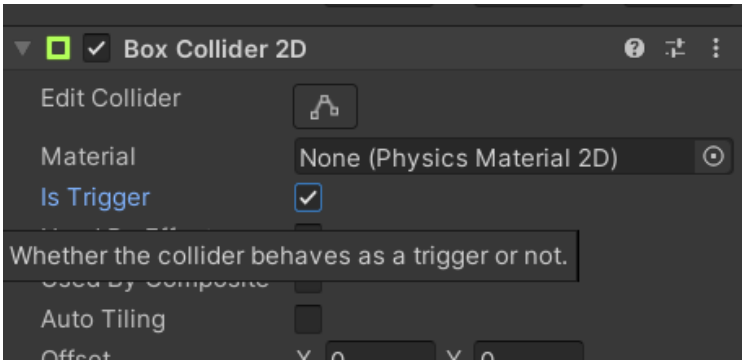
Gambar 1.20 mengatur ukuran collider

6. Create Empty object pada Hierarchy, Rename Menjadi *Boundary*



Gambar 1.21 menambahkan boundary

7. Tambahkan Box Collider 2d pada Boundary, centang pada Is Trigger lalu atur sesuai keinginan pada size dan offside



Gambar 1.22 mengatur box collider 2d

8. Buat sebuah file script didalam folder Script beri nama "Enemy_Behavior", kemudian drag dan masukkan ke dalam game object "bear"

Gambar 1.23 menambahkan script enemy behavior

9. Tambahkan Script dibawah ini

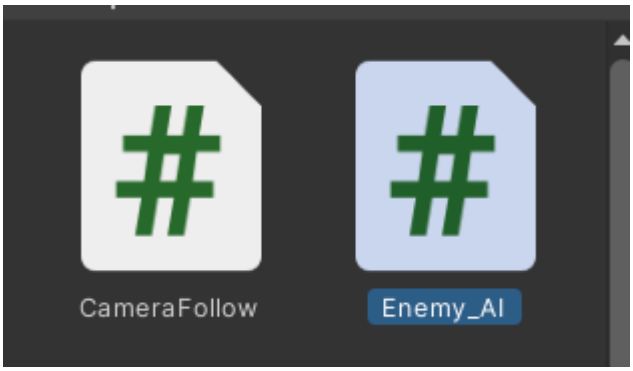
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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);
    }
}
```

C. Enemy AI

1. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama vulture. Tambahkan bat pada Hierarchy

Gambar 1.24 menambahkan asset enemy

2. Buat Script Enemy_AI pada folder Tugas7 - Script



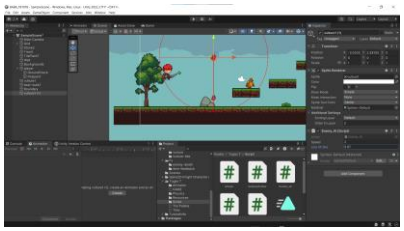
Gambar 1.25 membuat script enmy

3. Tambahkan Script dibawah 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
```

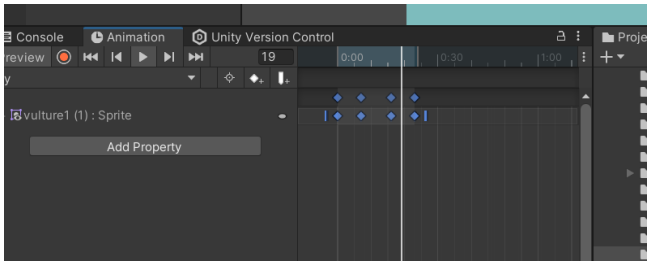
```
            player                                          =
    GameObject.FindGameObjectWithTag("Player").tran
    sform;
            // Menyimpan posisi awal musuh
            initialPosition                                 =
    GetComponent<Transform>().position;
        }

        // 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);
        }
}
```

4. Pada *Inspector Enemy_Ai*, Atur *Speed* juga *Line of Site* untuk menentukan jarak dan *speed* pada *enemy*



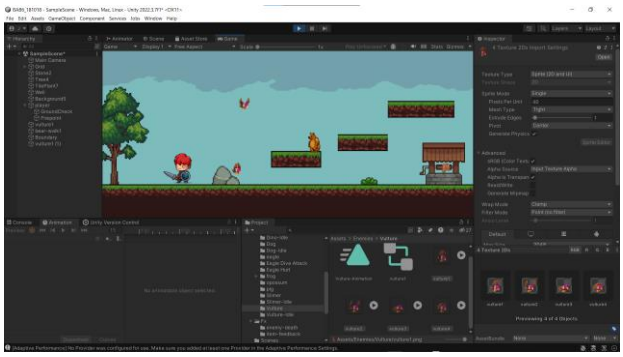
Gambar 1.26 mengatur *script* pada *inspector enemy ai*

5. Tambahkan animasi pada *vulture* dengan cara create animation dan drag asset



Gambar 1.27 menambahkan animasi

6. Running Game, maka vulture akan mengikuti Gerakan Player



Gambar 10. 1 menjalankan program

D. Respawn

1. Buka file script (Player.cs) tambahkan variabel nyawa seperti dibawah ini

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

Gambar 1.28 file script

2. Tambahkan kode dibawah ini untuk mengatur posisi respawn sesuai dengan posisi awal permainan dimulai

```
private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
    animator = GetComponent<Animator>();

    respawn_loc = transform.position;
}
```

Gambar 1.29 posisi respawn

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

```
void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
    if (Input.GetButtonDown("Jump"))...
    else if (Input.GetButtonUp("Jump"))...

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

Gambar 1.30 void update Player.cs

4. Tambahkan juga kode berikut dibawah code sebelumnya agar ketika player jatuh dibawah platform akan melakukan respawn

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

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

Gambar 1.31 player jatuh dibawah platform

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

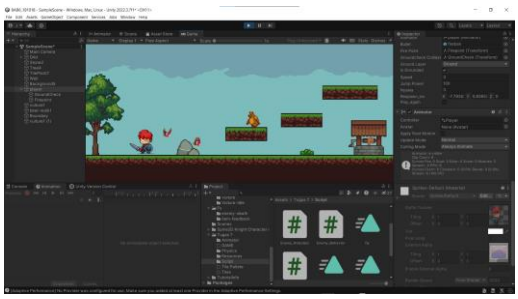
```
27
28      private void Awake()...
34
35      void playagain()
36      {
37          if (play_again == true)
38          {
39              nyawa = 3;
40              transform.position = respawn_loc;
41              play_again = false;
42          }
43      }
44
```

Gambar 1.32 fungsi playagain

6. Tambahkan 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<P
layer>();
        }
    }

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

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

7. Jika di play, Player mengenai atau menyentuh opposum-1 sebanyak 3 kali maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan reswpawn ke titik awal



Gambar 1.33 menjalankan program

1.2 Kesimpulan

1. Dalam Video *Game*, kecerdasan buatan digunakan untuk membuat perilaku cerdas yang biasanya terletak pada *non*-player *characters* (*NPCs*), dan seringnya mensimulasikan seperti kecerdasan manusia.
2. *layer-experience modelling*: memahami kemampuan dan kondisi emosional pemain, agar menyesuaikan *game* dengan benar
3. *Procedural-content generation*: membuat elemen dari sebuah lingkungan game seperti kondisi lingkungan, tingkatan, dan bahkan music dengan secara otomatis.