



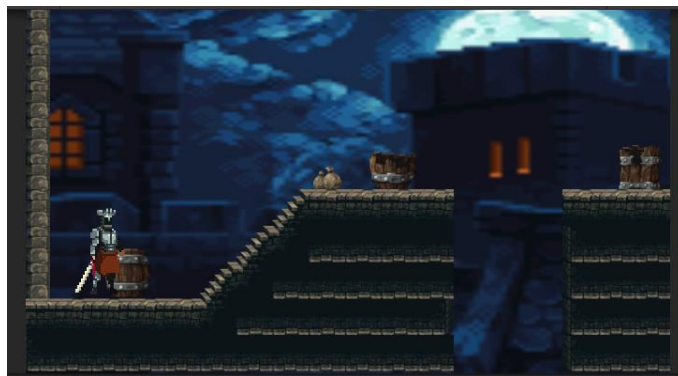
TUGAS PERTEMUAN: 10 ENEMY AI & ATTACK

NIM	:	2118080
Nama	:	Syahdan Izzur Rosuli
Kelas	:	B
Asisten Lab	:	M. Zainul Musyafa' (2118050)
Baju Adat	:	Cak dan Ning (Jawa Timur)
Referensi	:	https://cdnwpseller.gramedia.net/wp-content/uploads/2021/09/01210916/Baju-Cak-dan-Ning.jpg

10.1 Tugas 10 : Membuat Enemy AI & Attack

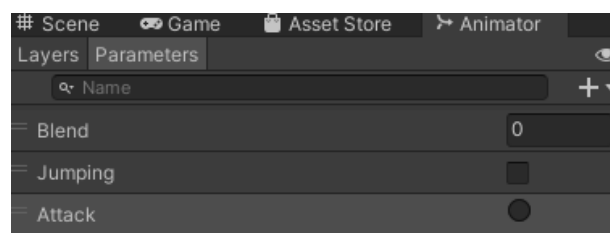
A. Membuat Mekanisme Attack

1. Buka *project Unity* sebelumnya bab 9.



Gambar 1.1 Tampilan *Project Unity*

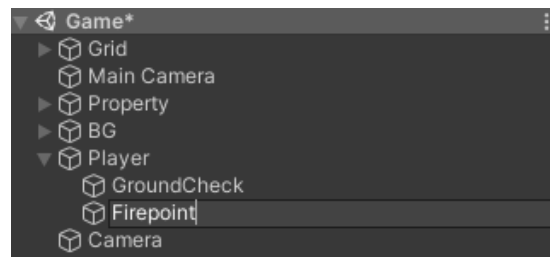
2. Kemudian pada menu *Tab Animator* tambahkan *Parameter Trigger*, *Rename* menjadi *Attack*.



Gambar 1.2 Tampilan *Parameter Attack*

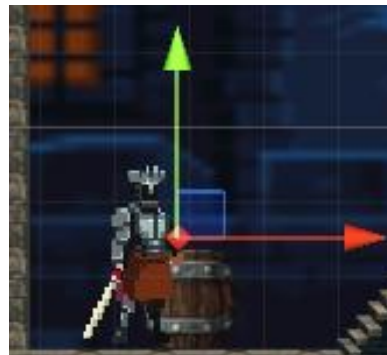


3. Lalu membuat *Layer Game Object* didalam *Player*, klik kanan pilih *Create Empty* kemudian rename menjadi *Firepoint*.



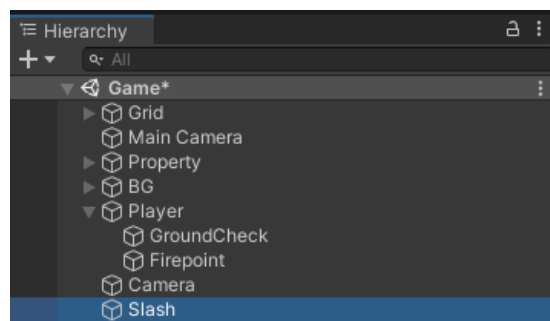
Gambar 1.3 Tampilan *FirePoint*

4. Pada menu *Hierarchy* klik *Firepoint* lalu ke *inspector*, ubah *Icon* menjadi dan atur letak titik di depan *player*.



Gambar 1.4 Tampilan atur letak titik

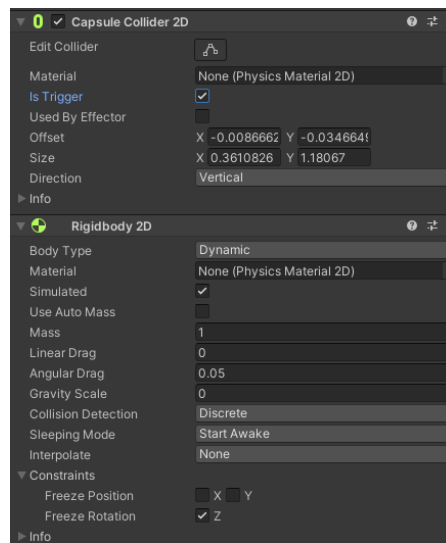
5. Lalu tambahkan *small-slash*, di *folder Assets > Fx>item-feedback-1*, *rename* menjadi *Slash* .



Gambar 1.5 Tampilan *Slash*

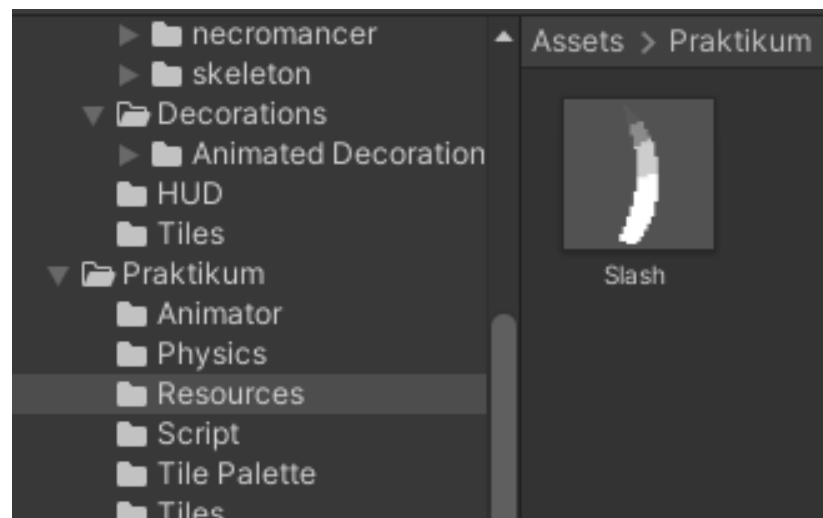


6. Klik *slash* dan pergi ke *inspector* tambahkan *component Circle Collider 2D* dan *RigidBody 2D* setting sesuai dibawah.



Gambar 1.6 Tampilan *Capsule Collider & RigidBody*

7. Kemudian buat folder baru beri nama *Resources* di menu *project*, kemudian *drag & drop slash* ke dalam *folder Resources*, kemudian hapus *slash* pada *Hierarchy*.



Gambar 1.7 Tampilan *Folder Resources*

8. Pada Script player tambahkan script dibawah ini :

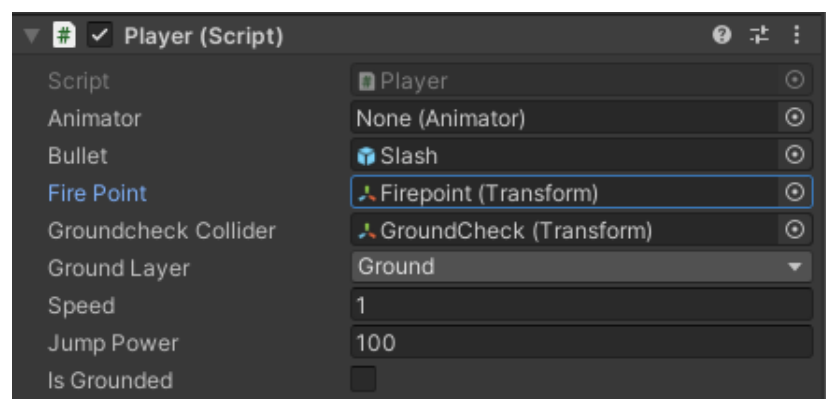
```
public class Player : MonoBehaviour
{
    public Animator animator;
    public GameObject bullet;
    public Transform firePoint;
```



9. Lalu tambahkan script fungsi dibawah fixedUpdate & Void Update seperti berikut

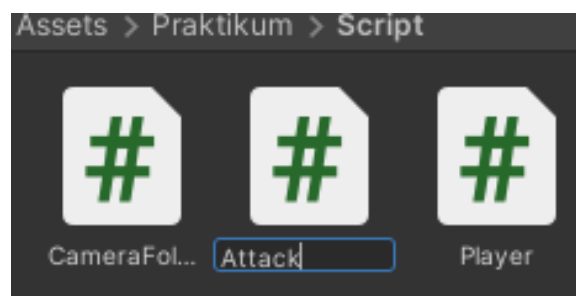
```
IEnumerator Attack()  
{  
    animator.SetTrigger("Attack");  
    yield return new WaitForSeconds(0.25f);  
    float direction = facingRight? 1f:-1f;  
  
    GameObject Slash = Instantiate(bullet,  
        firePoint.position, Quaternion.identity);  
    Slash.GetComponent<Rigidbody2D>().velocity = new  
        Vector2(direction * 10f, 0);  
    if (!facingRight)  
    {  
        Vector3 slashScale =  
            Slash.transform.localScale;  
        slashScale.x *= -1; // Membalik skala di sumbu  
        X  
        Slash.transform.localScale = slashScale;  
    }  
    Destroy(Slash, 2f);  
}
```

10. Pada *inspector* player ubah *bullet* menjadi *Slash* dan *firePoint* adalah titik tembak pertama.



Gambar 1.8 Tampilan *FirePoint Player*

11. Kemudian buat *script Attack* pada *folder Script*



Gambar 1.9 Tampilan *Script Attack*

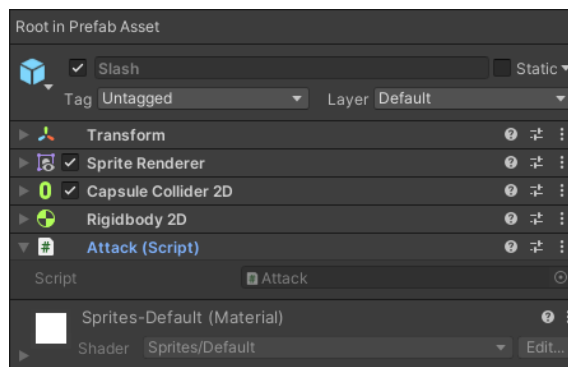


12. Lalu masukkan *Script Attack* sesuai dibawah ini

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

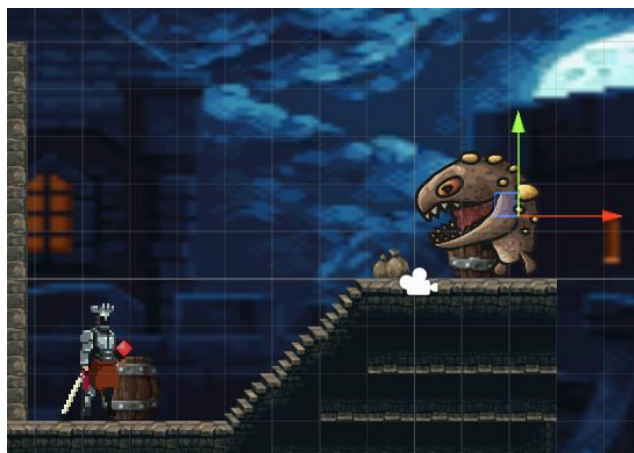
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```

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



Gambar 1.10 Tampilan *Prefab*

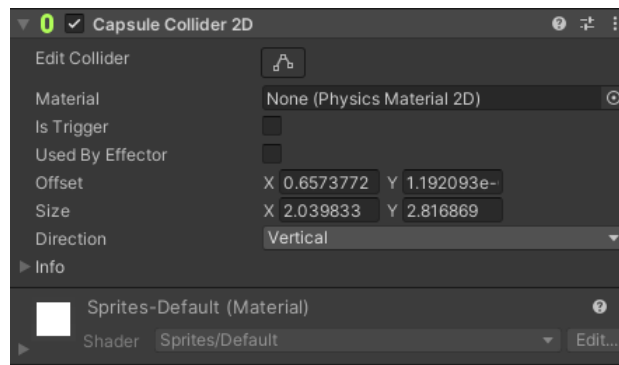
14. Tambahkan *Enemy* pada *hierarchy* di *folder Sprites*.



Gambar 1.11 Tampilan *character2*

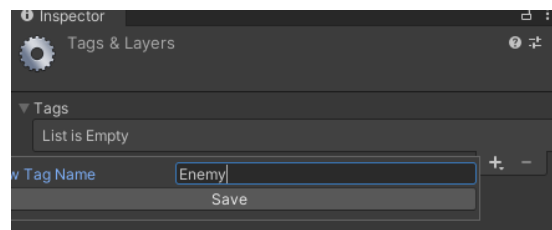


15. Klik pada *Monster*, pergi ke *inspector* tambahkan *Capsule Collider 2D* untuk mendeteksi.



Gambar 1.12 Tampilan *Capsule Collider 2D*

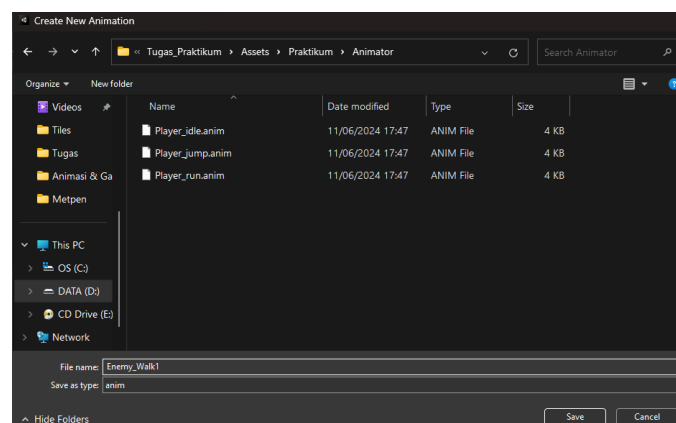
16. Tambahkan *tag enemy* dengan cara pilih *Add tag*, kemudian *add tag to the list*, lalu tuliskan *enemy*.



Gambar 1.13 Tampilan *Tag Enemy*

B. Enemy Behavior NPC

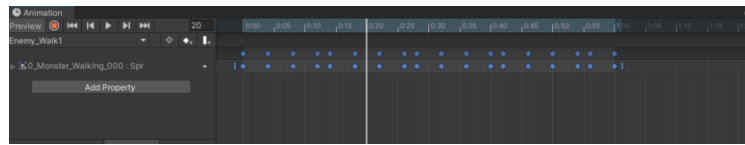
1. Tambahkan Animasi pada *Monster* berupa *Enemy_Walk1* kemudian save pada folder animator



Gambar 1.14 Tambah animasi enemy

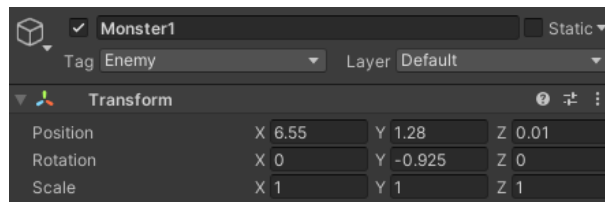


2. Tambahkan seluruh animasi berjalan ke keyframe dan sesuaikan durasinya



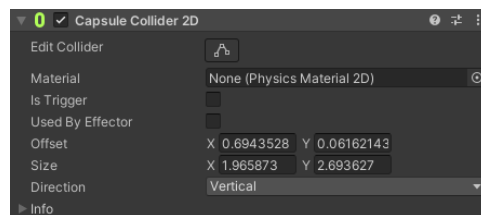
Gambar 1.15 Tampilan Keyframe

3. Pada *inspector* atur *transform Scale* menjadi seperti berikut



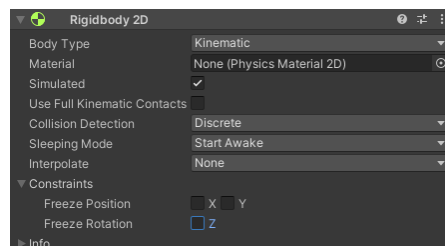
Gambar 1.16 Tampilan *Setting Transform*

4. Tambahkan *component* bernama *Capsule Collider 2D* & *RigidBody* 2D pada Monster



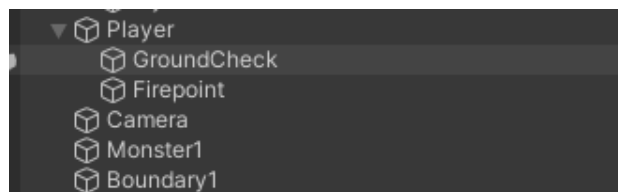
Gambar 1.17 Tampilan *Capsule & RigidBody* pada Monster

5. Atur sedikit Collider dan pada Body Type ubah menjadi Kinematic



Gambar 1.18 Tampilan *Kinematic*

6. *Create Empty object* pada *Hierarchy*, *rename* menjadi *Boundary1*



Gambar 1.19 Tampilan *Boundary*

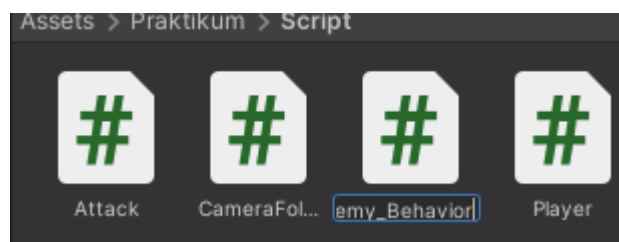


7. Tambahkan *Box Collider 2D* pada *Boundary*, lalu centang *Is Trigger* dan atur *size* dan *offset* sesuai keinginan



Gambar 1.20 Tampilan *Box Collider 2D*

8. Buat file *Script* didalam *folder script* beri nama “Enemy_Behavior” kemudian *drag* masukan ke dalam game object “Monster1”.



Gambar 1.21 Tampilan *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);
    }
}
```

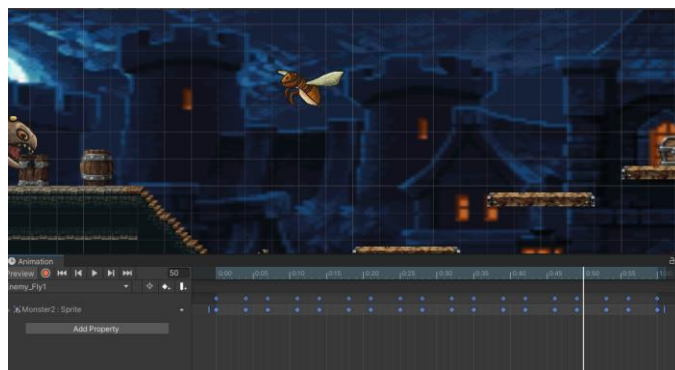
10. Akan terlihat hasil seperti dibawah dimana musuh akan bergerak sesuai dengan kotak yang sebelumnya dibuat.



Gambar 1.22 Hasil Enemy Behavior

C. Enemy AI

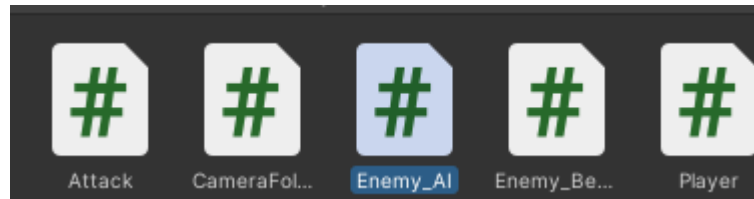
1. Buat enemy baru kemudian beri animasi.



Gambar 1.23 Buat Enemy baru



2. Buat Script “Enemy_AI” pada folder Praktikum Script



Gambar 1.24 Tampilan *Enemy AI*

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
    private bool facingRight = true; // Menunjukkan apakah musuh menghadap ke kanan

    // Use this for initialization
    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(); // Memutar musuh untuk menghadap pemain
        }
    }
}
```



```
else
{
    // Musuh kembali ke posisi awal
    transform.position =
Vector2.MoveTowards(transform.position,
initialPosition, speed * Time.deltaTime);
    FaceInitialPosition(); // Memutar musuh
    untuk menghadap posisi awal jika diperlukan
}
}

// Memutar musuh untuk menghadap pemain
void FacePlayer()
{
    if (player.position.x > transform.position.x
    && facingRight)
    {
        Flip();
    }
    else if (player.position.x <
    transform.position.x && !facingRight)
    {
        Flip();
    }
}

// Memutar musuh untuk menghadap posisi awal jika
diperlukan
void FaceInitialPosition()
{
    if (initialPosition.x < transform.position.x
    && facingRight)
    {
        Flip();
    }
    else if (initialPosition.x >
    transform.position.x && !facingRight)
    {
        Flip();
    }
}

// Membalik orientasi musuh
void Flip()
{
    facingRight = !facingRight;
    Vector3 scaler = transform.localScale;
    scaler.x *= -1;
    transform.localScale = scaler;
}

// 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 1.25 Tampilan menentukan jarak dan *Speed*

5. Maka ketika dijalankan musuh akan mengikuti player ketika ada di range yang telah ditentukan



Gambar 1.26 Tampilan Enemy AI

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;
```

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

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

3. Tambahkan didalam *void update Player.cs* agar nyawa player dibawah 0 maka akan melakukan *respawn*.

```
If (nyawa < 0)  
{  
    Playagain();}
```



4. Tambahkan kode dibawah ini jika player jatuh dibawah platform maka 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. Tambahkan *file script*(*Enemy_Attacked.cs*) dan isikan *source code* dibawah ini

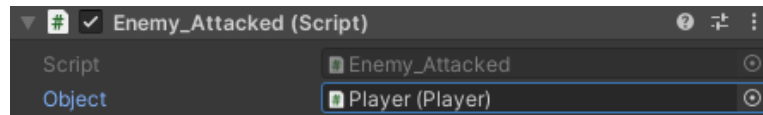
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_attacked : MonoBehaviour
{
    [SerializeField] private Player Object;

    void Start()
    {
        if (Object == null)
        {
            Object =
GameObject.FindWithTag("Player").GetComponent<Player>();
        }
    }
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            Object.nyawa--;

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

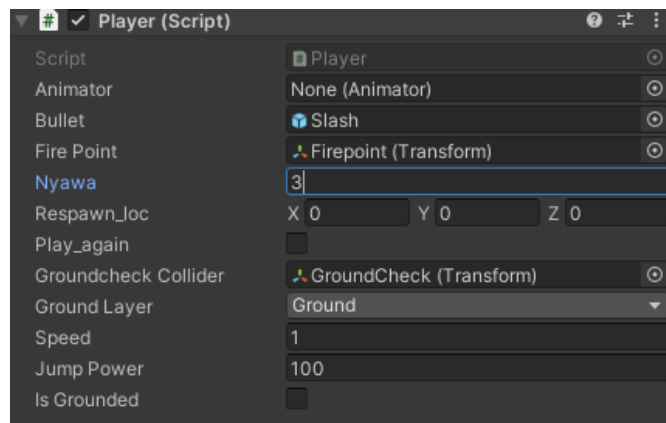


7. Tambahkan monster baru beserta animasinya kemudian Pada hierarchy Tambahkan Script enemy attack, arahkan object pada Player.



Gambar 1.27 Tampilan *Script Enemy Attack*

8. Klik game *object* Player, pergi ke *Inspector* dan ubah nilai nyawa menjadi 3 pada Player(*Script*).



Gambar 1.28 Memberi nyawa player

9. Jalankan maka ketika player jatuh atau terkena musuh akan mati dan respawn di tempat awal.

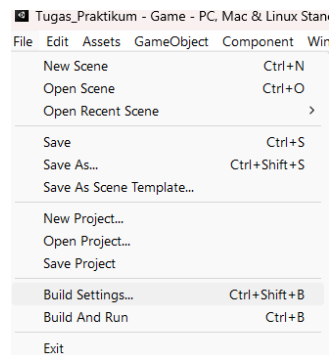


Gambar 1.29 Tampilan respawn & enemy attack



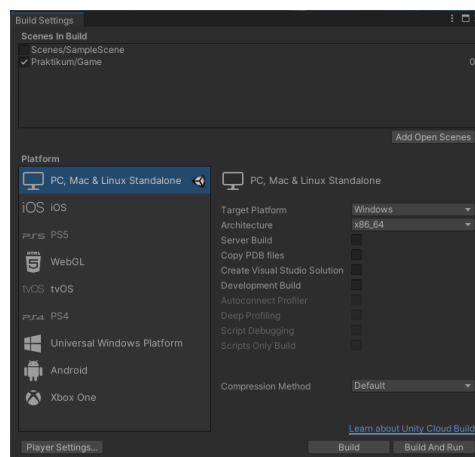
E. Render

1. Pergi ke menu *file* pilih *Build Setting* (Ctrl + Shift + B)



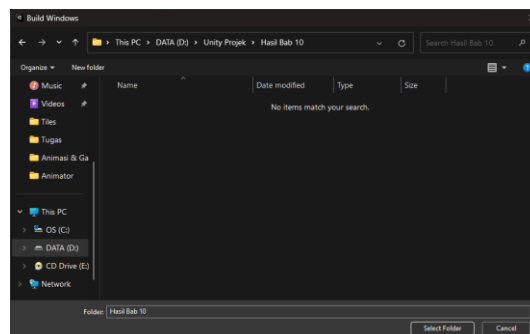
Gambar 1.30 Tampilan Menu *File*

2. Pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian



Gambar 1.31 Tampilan *Setting Build*

3. Pilih dimana *project* disimpan dan tunggu hasilnya



Gambar 1.32 Peletakan folder untuk menyimpan



4. Lalu pilih project yang sudah di *render* klik 2x untuk melihat hasilnya



Gambar 1.33 Tampilan hasil Akhir