



TUGAS PERTEMUAN: 10

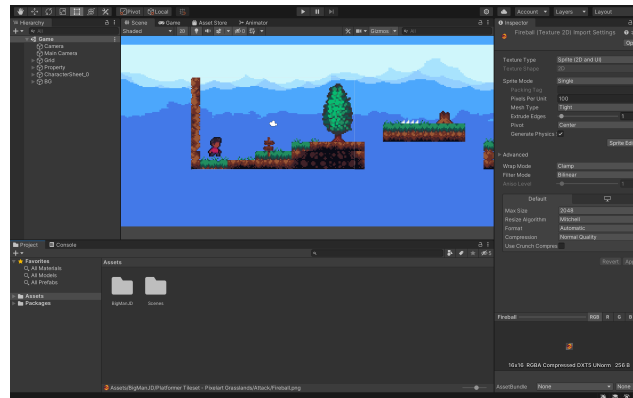
RESPAWN AND AI ENEMY ATTACK

NIM	:	2118008
Nama	:	Agnus Dei Dharma Prawira
Kelas	:	A
Asisten Lab	:	Naufal Dhiaurrafif (2218059)

10.1 Tugas 1 : Membuat Respawn & AI Enemy Attack

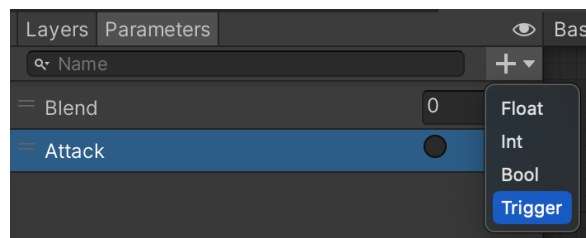
A. Membuat Mekanisme Attack

1. *Buka project Unity sebelumnya untuk melanjutkan tahap pembuatan AI enemy attack dan respawn.*



Gambar 10.1 Tampilan *Project*

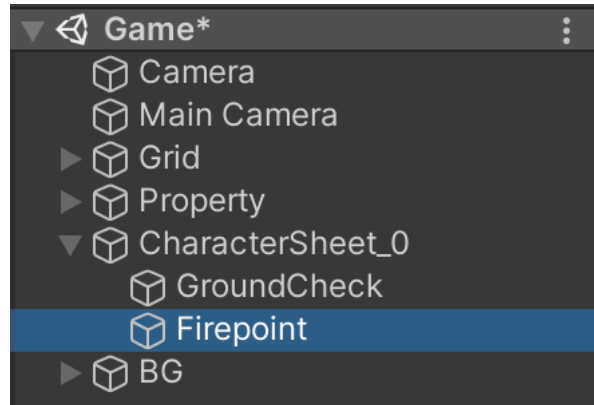
2. Kemudian pada menu Tab Animator, tambahkan parameter *trigger* dan rename menjadi *Attack*.



Gambar 10.2 Tampilan Menambahkan Parameter

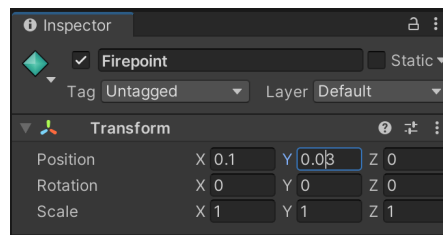


3. Selanjutnya, buat Layer *Game object* baru didalam *CharacterSheet_0* dengan cara klik kanan, lalu pilih *Create Empty* dan *rename* menjadi *Firepoint*.



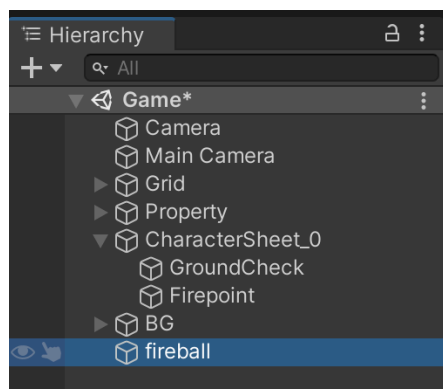
Gambar 10.3 Tampilan *Add Firepoint*

4. Pada menu *Hierarchy*, klik *Firepoint* untuk *setting* pada *Inspector*, lalu ubah *Icon* menjadi titik dan atur letak titik didepan *player*.



Gambar 10.4 Tampilan Mengatur *Icon*

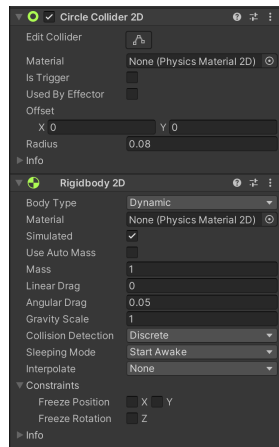
5. Pada menu *Hierarchy*, tambahkan Fireball dari folder *Attack*, kemudian Fireball dengan *drag and drop* dan *rename* menjadi *fireball*.



Gambar 10.5 Menambahkan *Fireball*



6. Klik *fireball* untuk menambahkan komponen *Circle Collider 2D* dan *Rigidbody 2D*, lalu *setting* sesuai gambar dibawah ini.



Gambar 10.6 Add Circle Collider dan Rigidbody

7. Buat Folder baru *Resources* di dalam folder BAB7 pada menu *Project*, kemudian *drag and drop fireball* ke dalam folder *Resources* dan hapus *fireball* pada *Hierarchy*



Gambar 10.7 Buat Folder Resource

8. Pada *script Player* tambahkan *Script* berikut.

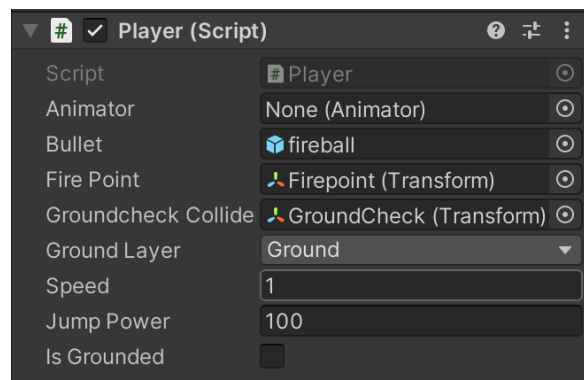
```
//Pada class Player
// Deklarasi variable
Public Animator animator;
Public GameObject bullet;
Public Transform Firepoint;

//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);
}
```



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

9. Pada Inspector Player, ubah seperti dibawah ini, dimana Bullet berisi objek yang akan ditembak, sedangkan *firepoint* adalah titik tembak pertama.



Gambar 10. 8 Add Bullet Objek

10. Buat *script attack* pada folder *Script*.



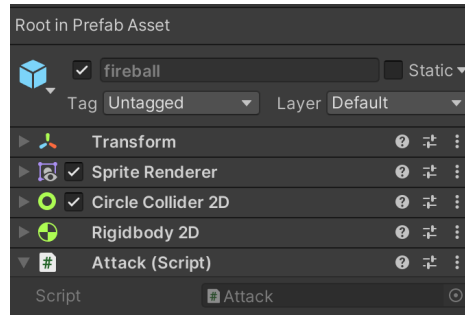
Gambar 10.9 Membuat *Script Attack*

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);
        }
    }
}
```

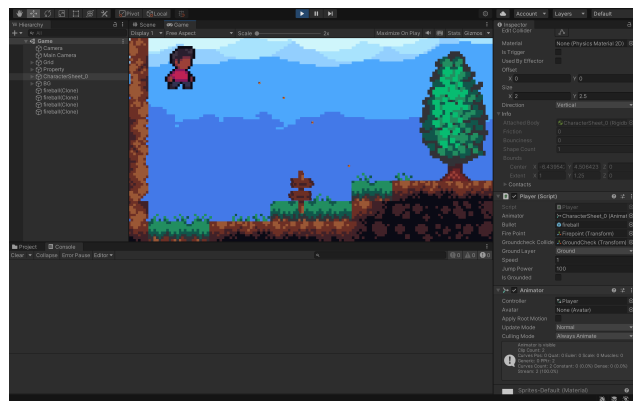


12. Didalam folder *resources*, tambahkan *Script Attack* di *Prefab fireball*, dengan cara klik *fireball*, kemudian pada menu Inspector arahkan *Script Attack* kedalam Inspector.



Gambar 10.10 Memindahkan *GroundCheck*

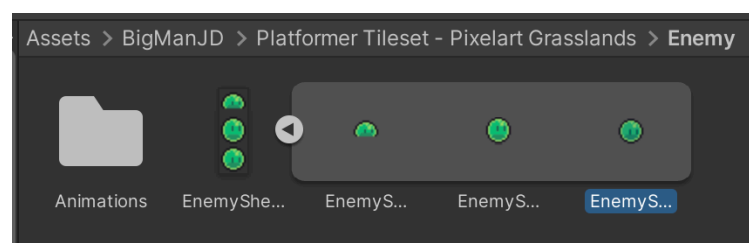
13. Tembak dengan menekan tombol C untuk menembak.



Gambar 10.11 Tampilan Menembak

B. Membuat Enemy Behavior NPC

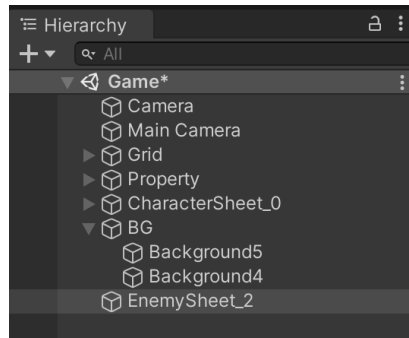
1. Cari sebuah *sprite pack* bernama *enemy* dan buka folder bernama “EnemySheet_2”.



Gambar 10.12 Tampilan *Enemy*

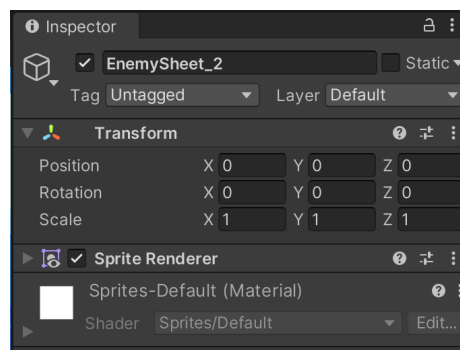


2. Tambahkan “EnemySheet_2” ke *Hierarchy*.



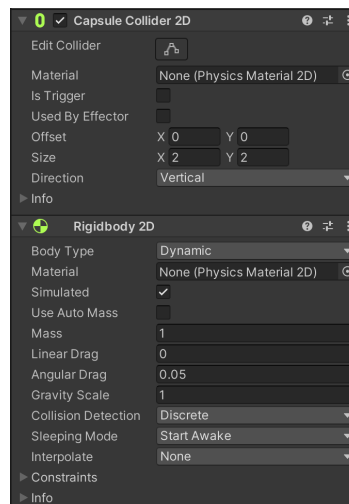
Gambar 10.13 Add EnemySheet_2

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



Gambar 10.14 Setting Transform Scale

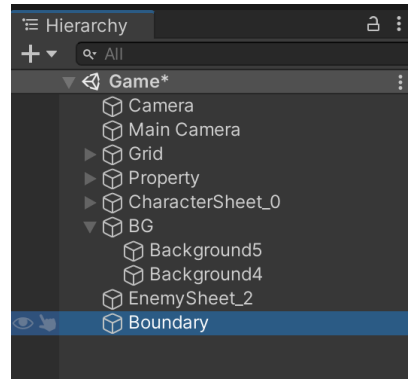
4. Tambahkan sebuah komponen bernama *Capsule Collider 2D* dan *Rigidbody 2D* dalam *inspector game object* EnemySheet_2. Lalu atur seperti gambar berikut.



Gambar 10.15 Add Component

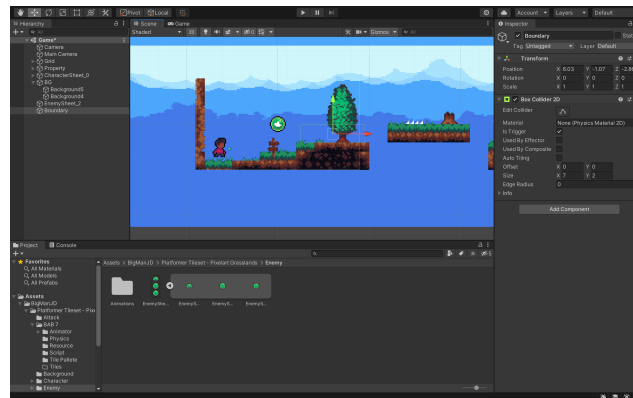


5. *Create Empty object* pada Hierarchy dan *rename* menjadi *Boundary*.



Gambar 10.16 *Create Empty Object*

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



Gambar 10.17 *Add Box Collider*

7. Buat sebuah file *script* didalam folder *Script*, lalu beri nama “*Enemy_Behavior*”, kemudian *drag* dan masukkan ke dalam *game object* “*CharacterSheet_2*”



Gambar 10.21 *Membuat Script Enemy*

8. Tambahkan *Script* berikut di *script* *Enemy_Behavior*.

```
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>();
}

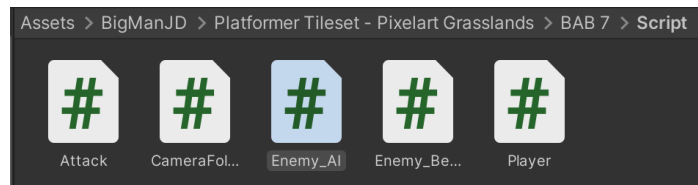
void Update()
{
    if (isFacingRight())
    {
        rb.velocity = new Vector2(moveSpeed, 0f);
    }
    else
    {
        rb.velocity = new Vector2(-moveSpeed, 0f);
    }
}

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. Membuat Enemy AI

1. Buat Script Enemy_AI pada folder *Script*.



Gambar 10.22 Membuat Script AI

2. Tambahkan *script* berikut pada *script* Enemy_AI.

```
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()
{
}
```

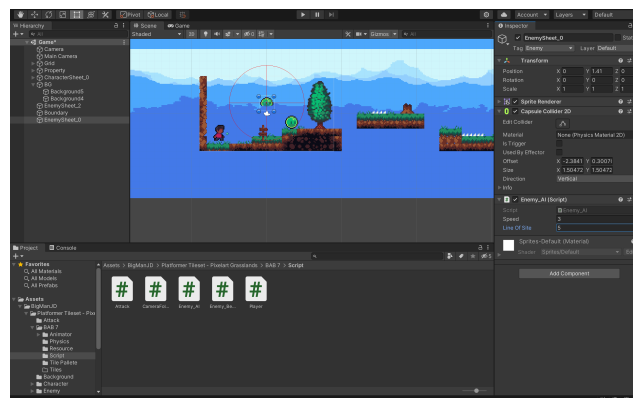


```
        if (initialPosition.x < transform.position.x &&
facingRight)
        {
            Flip();
        }
        else if (initialPosition.x >
transform.position.x && !facingRight)
        {
            Flip();
        }
    }

// 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);
}
}
```

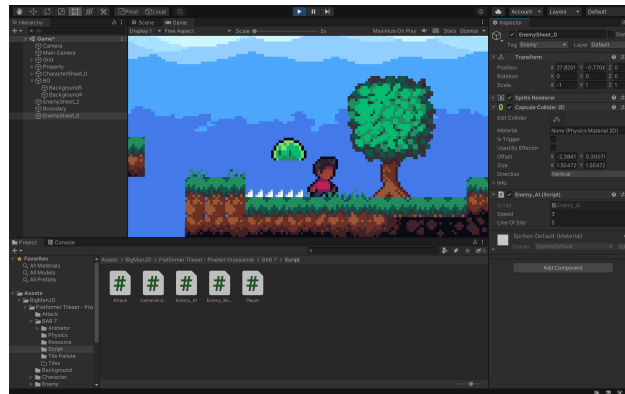
3. *Drag and drop script Enemy_AI ke dalam GameObject EnemySheet_1, lalu pada inspector Enemy_AI, atur speed dan Line of Site untuk menentukan jarak dan speed pada enemy.*



Gambar 10.23 Mengatur Speed Line



4. *Play game*, maka bat akan mengikuti gerakan player.



Gambar 10.24 Hasil Tampilan Game

D. Membuat 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 kemudian tambahkan pada *CharacterSheet_0*. Klik game *object* *Player*, pergi ke *Inspector* dan ubah nilai *nyawa* menjadi 3 pada *Player(Script)*

```
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;
            }
        }
    }
}
```

E. KUIS

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{

```



```
public float attackRange = 2.0f;      public int
attackDamage = 10; // Mengoreksi kesalahan penulisan
variabel attacDamage menjadi attackDamage

void Update()
{
    if (Input.GetButtonDown("Fire1")) //
Mengoreksi kesalahan penulisan InputGetButtonDown menjadi
Input.GetButtonDown
    {
        PerformMeleeAttack();
    }
}

void PerformMeleeAttack()
{
    RaycastHit hit;      if
(Physics.Raycast(transform.position,
transform.forward, out hit, attackRange))
    {
        // Lengkapi kode di sini untuk mengenai musuh
dan mengurangi health mereka
        EnemyHealth enemyHealth =
hit.transform.GetComponent<EnemyHealth>();
        if (enemyHealth != null)
        {
            enemyHealth.TakeDamage(attackDamage);
        }
    }
}

7. }
```

Analisa :

Dalam kode `PlayerAttack` terdapat beberapa kesalahan dan kekurangan yang perlu diperbaiki. Pertama, tipe data untuk variabel `attackRange` harus diubah dari `int` menjadi `float`, dan penulisannya harus diperbaiki dari `atackRange` menjadi `attackRange` untuk konsistensi. Selanjutnya, variabel `attacDamage` harus diperbaiki penulisannya menjadi `attackDamage`. Kemudian, pada fungsi `update` terdapat metode `InputGetButtonDown` yang harus diperbaiki menjadi `Input.GetButtonDown` untuk mengikuti konvensi penamaan metode di Unity. Terakhir, di dalam fungsi `PerformMeleeAttack`, perlu ditambahkan kode untuk mengecek apakah objek yang terkena serangan memiliki komponen `EnemyHealth`. Jika ya, maka panggil metode `TakeDamage` pada komponen tersebut untuk mengurangi *health* musuh.



F. Link Github

https://github.com/Agnusddp/2118008_PRAK_ANIGAME