TUGAS PERTEMUAN: 10

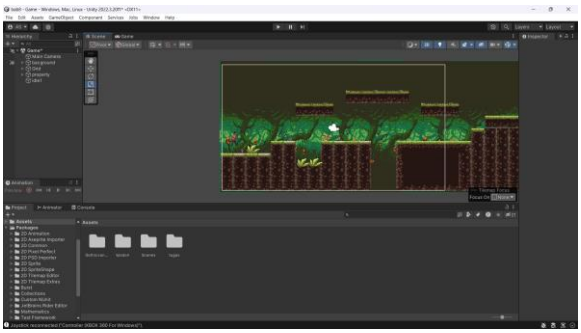
GAME ANIMATION

NIM	:	2118121
Nama	:	Acmad saiful udin
Kelas	:	D
Asisten Lab	:	Berchmans Bayu Bin Jaya (2218034)

10.1 Tugas 10 :Membuat Enemy AI dan Attack

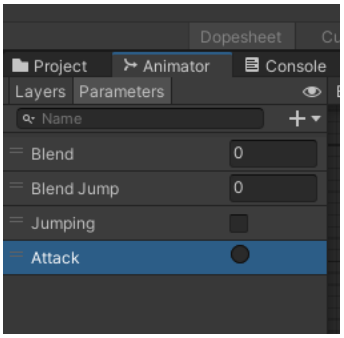
A. Membuat Mekanisme Attack

1. Buka project Unity yang telah di buat sebelumnya.



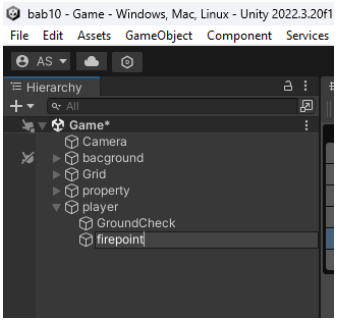
Gambar Tampilan project

2. Buka Animator Controller "Player", tambahkan parameter trigger bernama "Attacking".



Gambar Add parameter Animator

3. Buat empty object di bawah player, beri nama "*Firepoint*". Ini akan menjadi titik awal peluru .



Gambar *Firepoint*

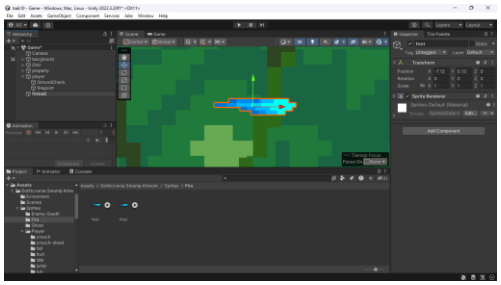
4. Ubah warna dan ikon " *Firepoint* ".

Gambar warna dan icon

5. Atur posisi "*Firepoint*" di ujung senjata atau tempat Anda ingin peluru keluar.

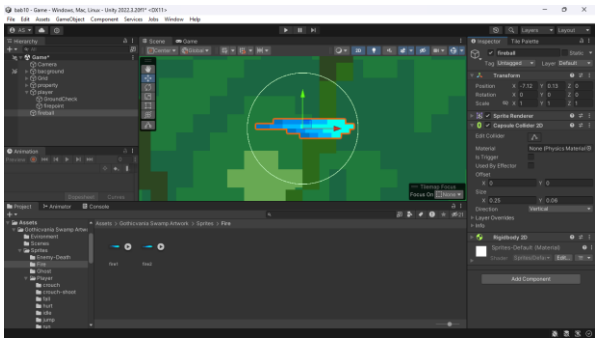
Gambar Posisi *Firepoint*

6. *Drag and drop sprite* peluru dari *Assets* ke Hierarchy, rename menjadi "fireball".



Gambar *fireball*

7. Tambahkan Collider dan Rigidbody: Pada "fireball", tambahkan komponen Circle Collider 2D dan Rigidbody 2D.

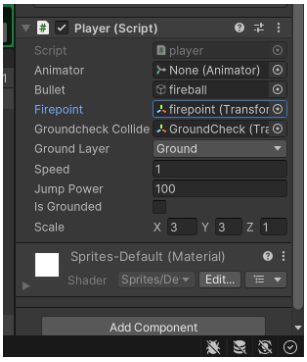


Gambar *fireball*

8. Tambahkan kode untuk serangan ke script `Player.cs`.

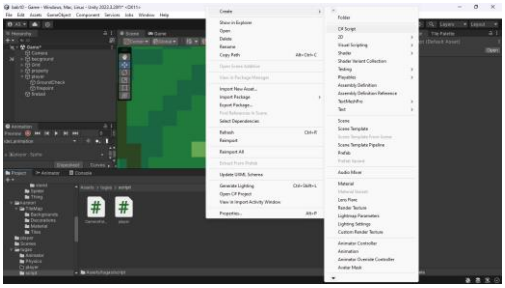
```
IEnumerator Attack()
{
    animator.SetTrigger("Attack");
    yield return new WaitForSeconds(0.25f);

    float direction = facingRight ? 1f : -1f;
    float rotationAngle = facingRight ? 0f : 180f;
    Quaternion rotation = Quaternion.Euler(0, 0, rotationAngle);

    GameObject fireball = Instantiate(bullet, Firepoint.position, rotation);
    fireball.transform.localScale = new Vector3(0.5f, 0.5f, 1f);
    fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction * 5f, 0);

    Destroy(fireball, 2f);
}
```

9. Pada komponen *Player Script* di *Inspector*, atur nilai-nilai yang dibutuhkan.



Gambar Inspector

10. Buat script C# baru bernama "Attack“.

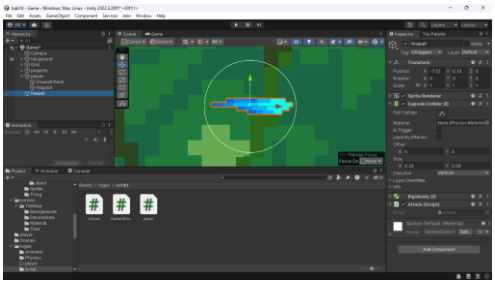


Gambar Attack

11. Tambahkan kode untuk mengontrol peluru ke script Attack.cs.

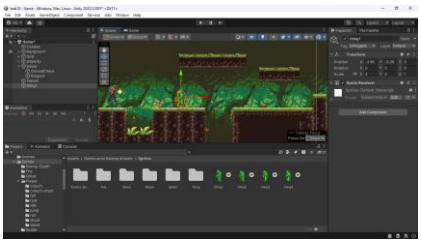
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Attack : MonoBehaviour
{
    private      void      OnTriggerEnter2D(Collider2D
collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
          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

12. Drag and drop script `Attack` ke objek "fireball" di Hierarchy.



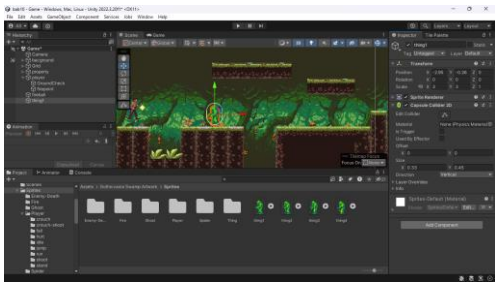
Gambar *Drag and drop script*

13. Drag and drop sprite musuh ke scene.



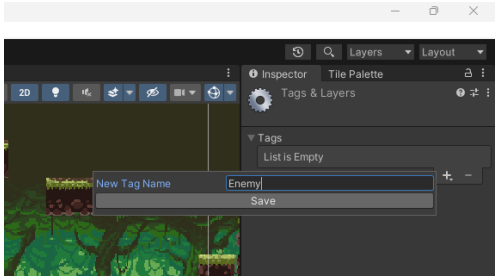
Gambar Musuh

14. Pada musuh, tambahkan komponen *Capsule Collider 2D*.



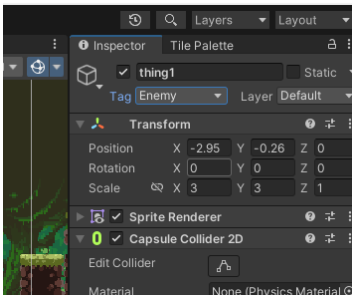
Gambar *Capsule Collider 2D*

15. Buat tag baru bernama "Enemy" di menu tag.



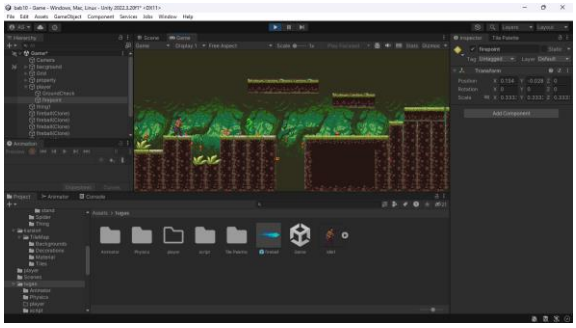
Gambar *Capsule Collider 2D*

16. Pada musuh, pilih tag "Enemy".



Gambar *Capsule Collider 2D*

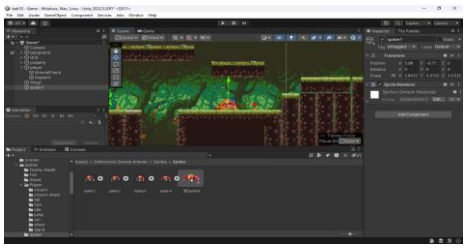
17. Jalankan game dan pastikan mekanisme serangan berfungsi.



Gambar Jalankan game

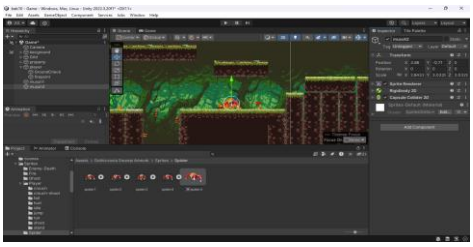
B. Membuat Enemy Behavior NPC

1. Drag and drop sprite musuh ke scene.



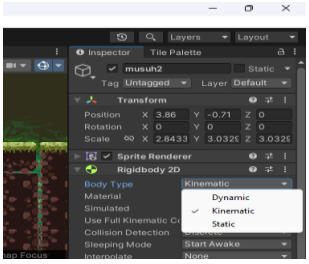
Gambar Musuh

2. Pada musuh kedua, tambahkan *Capsule Collider 2D dan Rigidbody 2D*.



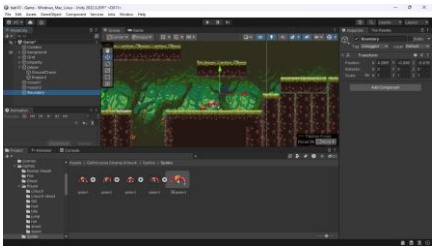
Gambar *Capsule Collider 2D dan Rigidbody 2D*

3. Pada *Rigidbody 2D* musuh kedua, ubah "*Body Type*" menjadi "*Kinematic*".



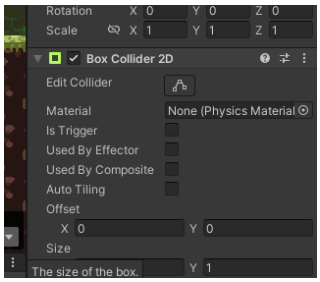
Gambar *Kinematic*

4. Buat empty object, beri nama "*Boundary*". Ini akan menjadi batas pergerakan musuh.



Gambar *Boundary*

5. Pada "*Boundary*", tambahkan komponen Box Collider 2D.



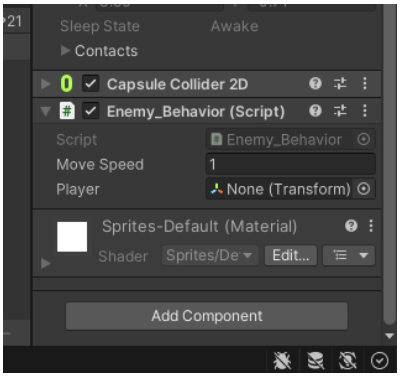
Gambar Tampilan project

6. Tambahkan kode untuk mengontrol musuh ke script Enemy_Behavior.cs.

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

public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 1f;
```

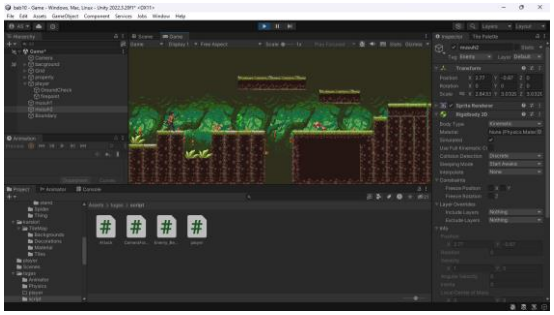
```
    [SerializeField] Transform player; // Referensi ke
pemain
    Rigidbody2D rb;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        // Perbarui posisi musuh untuk mengikuti pemain
        FollowPlayer();
    }

    void FollowPlayer()
    {
        if (player != null)
        {
            Vector2 direction = (player.position -
transform.position).normalized;
            rb.velocity = direction * moveSpeed;
        }
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        // Tambahkan logika jika diperlukan
    }
}
```

7. Drag and drop script *Enemy_Behavior* ke kedua objek musuh.



Gambar Tampilan project

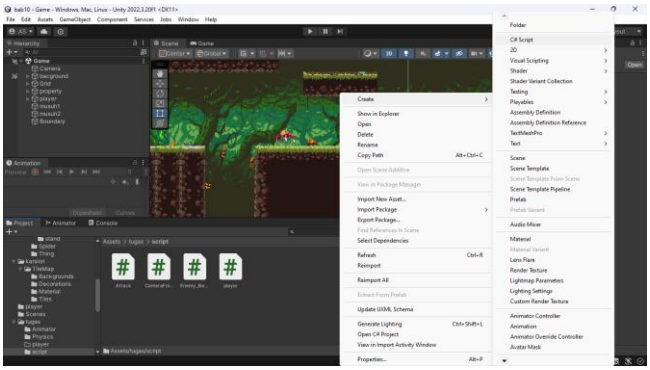
8. jalankan game dan pastikan musuh bergerak sesuai yang diinginkan.



Gambar jalankan game

C. Membuat Enemy A

1. Buat script C# baru bernama "Enemy_AI".



Gambar *Enemy AI*

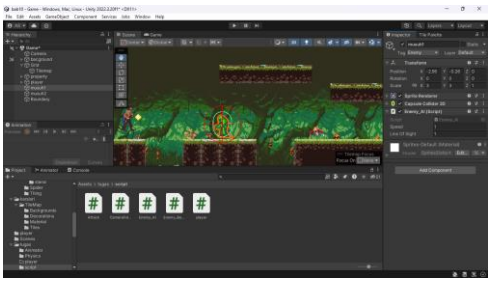
2. Tambahkan kode berikut ke script Enemy_AI.cs.

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

public class Enemy_AI : MonoBehaviour
{
```

```
    public float speed; // Kecepatan gerakan musuh
    public float lineOfSight; // Jarak penglihatan musuh
    private Transform player; // Transform dari pemain
    private Vector2 initialPosition; // Posisi awal musuh

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

    void Update()
    {
        float distanceToPlayer = Vector2.Distance(player.position, transform.position);
        if (distanceToPlayer < lineOfSight)
        {
            // Gerakan musuh menuju pemain
            transform.position = Vector2.MoveTowards(transform.position, player.position, speed * Time.deltaTime);
        }
        else
        {
            // Gerakan musuh kembali ke posisi awal
            transform.position = Vector2.MoveTowards(transform.position, initialPosition, speed * Time.deltaTime);
        }
    }

    private void OnDrawGizmosSelected()
    {
        Gizmos.color = Color.red;
```

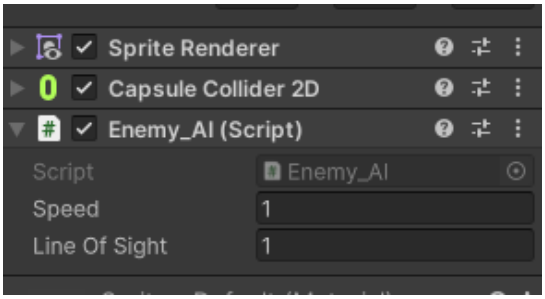
```
        Gizmos.DrawWireSphere(transform.position,
lineOfSight);
    }
}
```

3. *Drag and drop script* *Enemy_AI* ke objek musuh_1.



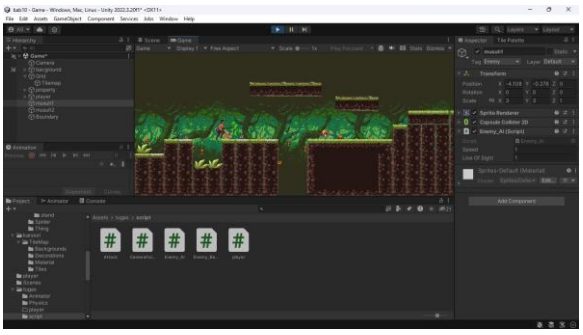
Gambar Drag and drop script

4. Atur nilai lineOfSite (jarak penglihatan) dan `speed` (kecepatan) di Inspector musuh.



Gambar Tampilan project

5. Jalankan game dan pastikan musuh mengikuti player saat berada dalam jangkauan penglihatannya.



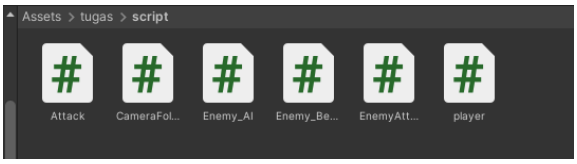
Gambar Jalankan game

D. Respawn

1. Tambahkan kode berikut ke script Player.cs.

```
using UnityEngine;
public class PlayerHealth : MonoBehaviour // More descriptive class name
{
    public int nyawa = 3;
    [SerializeField] private Vector3 respawnLocation;
    public bool playAgain;
    private void Awake ()
    {
        respawnLocation = transform.position; // Initialize in Start
    }
    public void PlayAgain()
    {
        if (playAgain) {
            nyawa = 3;
            transform.position = respawnLocation;
            playAgain = false;
        }
    }
}
```

2. Buat script C# baru bernama "EnemyAttacked".



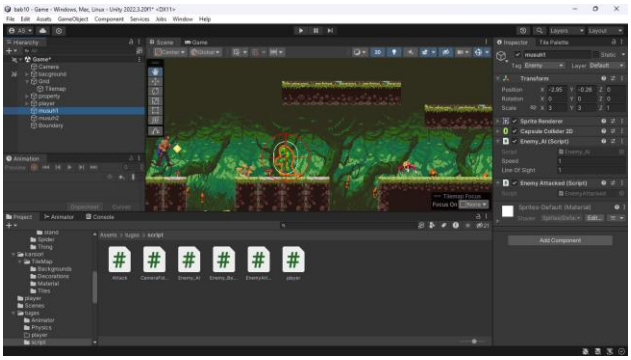
Gambar EnemyAttacked

3. Tambahkan kode berikut ke script EnemyAttacked.cs.

```
using UnityEngine;

public class EnemyAttacked : MonoBehaviour
{
    [SerializeField] private PlayerHealth playerHealth; // Use PlayerHealth reference
```

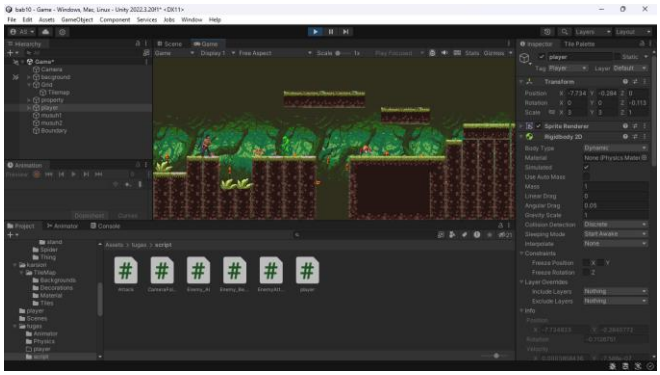
```
    private void Start()
    {
        // Find the PlayerHealth component if not assigned in the inspector
        if (playerHealth == null)
        {
            playerHealth = GameObject.FindWithTag("Player")?.GetComponent<PlayerHealth>();
            if (playerHealth == null)
            {
                Debug.LogError("EnemyAttacked: Player with tag 'Player' and PlayerHealth component not found!");
                enabled = false; // Disable the script to avoid errors
            }
        }
    }

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            playerHealth.nyawa--;
            Debug.Log("Player health: " + playerHealth.nyawa); // Added logging for debugging

            if (playerHealth.nyawa <= 0)
            {
                // Handle player death
                playerHealth.PlayAgain(); // Trigger PlayAgain logic
            }
        }
    }
}
```

4. Drag and drop script EnemyAttacked ke objek musuh1.



Gambar EnemyAttacked

5. Jalankan game, biarkan player mati, dan pastikan player respawn di lokasi yang ditentukan.



Gambar Jalankan game

E. Link Github Pengumpulan

https://github.com/acmadsaiful090/2118121_PRAK_ANIGAME/tree/main/BAB%2010

KUIS

Kuis bab 10

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    public int attackDamage = 10;
    public string enemyTag = "Enemy";

    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))
            {
                Health                  healthComponent                 =
hit.collider.GetComponent<Health>();
                if (healthComponent != null)
                {
                    healthComponent.TakeDamage(attackDamage);
                }
            }
        }
    }
}
```

Analisa :

Kode sumber di atas telah diperbaiki pada metode PerformMeleeAttack(). Pertama, tipe variabel attackRange diubah dari int menjadi float untuk mencerminkan penggunaannya sebagai nilai jarak serangan. Kedua, kesalahan penulisan pada InputGetButtonDown diperbaiki menjadi Input.GetButtonDown, dan attacDamage diperbaiki menjadi attackDamage. Penambahan tag enemyTag memungkinkan identifikasi musuh melalui tag, memastikan hanya musuh yang terkena serangan. Dalam metode PerformMeleeAttack, ditambahkan pemeriksaan untuk memastikan bahwa objek yang terkena raycast memiliki komponen Health, yang bertanggung jawab untuk mengurangi kesehatan musuh dan menangani kematian mereka.