TUGAS PERTEMUAN: 10

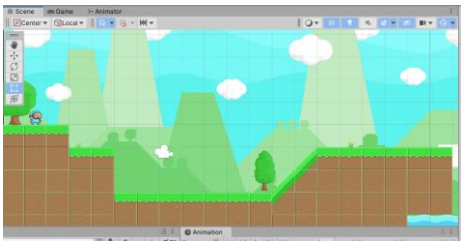
RESPAWN & AI ENEMY ATTACK

NIM	:	2118115
Nama	:	Ellok Ananda Madya Pratiwi
Kelas	:	C
Asisten Lab	:	Nayaka Apta Nayottama (2218102)

1.1 Tugas 1 : AI ENEMY ATTACK

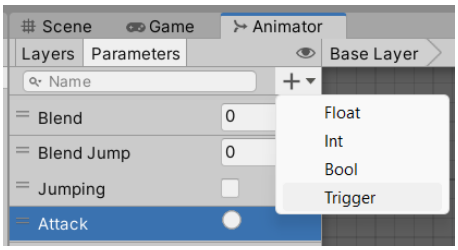
A. Membuat Mekanisme Attack

1. Buka project bab9 sebelumnya



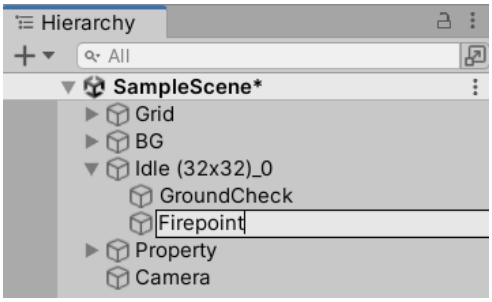
Gambar 10.1 Membuka Project

2. Pada tab animator tambahkan parameter bertipe data trigger dan beri nama Attack



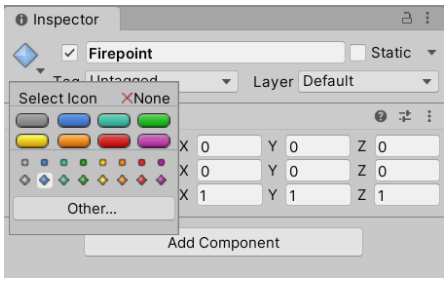
Gambar 10.2 Membuat Parameter

3. Klik idle pada hirarki lalu buat GameObject Baru beri nama Firepoint



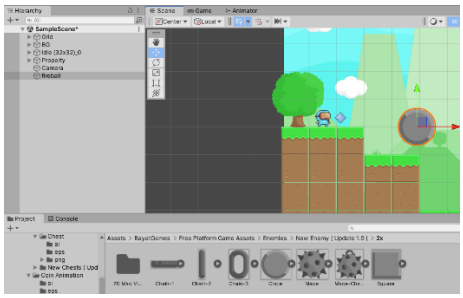
Gambar 10.3 Menambahkan GameObject

4. Pada insepector Firepoint, ubah icon menjadi diamond lalu posisikan didepan player



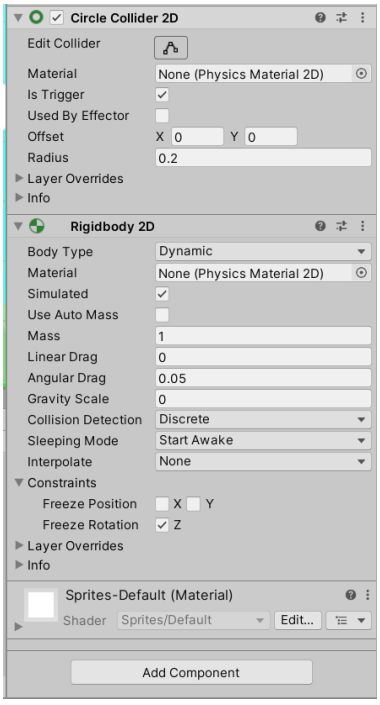
Gambar 10.4 Mengubah Icon

5. Tambahkan circle yang terdapat pada folder Enemies pada hirarki dan ubah nama menjadi fireball



Gambar 10.5 Menambahkan Peluru

6. Pada inspector fireball tambahkan component circle collider 2d dan rigidbody 2d



Gambar 10.6 Mengatur Circle Collider Dan Rigidbody

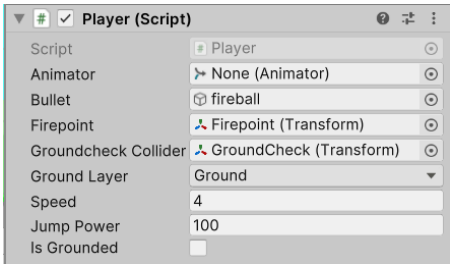
7. Ubah source code player menjadi seperti berikut

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

public class Player : MonoBehaviour
{
    public Animator animator;
    public GameObject bullet;
    public Transform firePoint;
    Rigidbody2D rb;
    private float lastDirection;
    [SerializeField] Transform groundcheckCollider;
    [SerializeField] LayerMask groundLayer;

    const float groundCheckRadius = 0.2f; // +

  [SerializeField] float speed = 1;
    [SerializeField] float jumpPower = 100;
  float horizontalValue;

  [SerializeField] bool isGrounded; // +

  bool facingRight;
  bool jump;

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

  void Update ()
  {
    horizontalValue = Input.GetAxisRaw("Horizontal");

        // Menyimpan arah terakhir berdasarkan input
horizontal
        if (horizontalValue < 0)
        {
            lastDirection = -1f; // Kiri
        }
        else if (horizontalValue > 0)
        {
            lastDirection = 1f; // Kanan
        }

        // Memeriksa jika tombol serangan ditekan
        if (Input.GetKeyDown(KeyCode.C))
        {
            StartCoroutine(Attack(lastDirection));
        }

    if (Input.GetButtonDown("Jump"))
      {
        animator.SetBool("Jumping", true);
        jump = true;
      }
```

```
    else if (Input.GetButtonUp("Jump"))
        jump = false;
  }

 void FixedUpdate()
  {
    GroundCheck();
    Move(horizontalValue, jump);

    animator.SetFloat("Blend",
Mathf.Abs(rb.velocity.x));
    animator.SetFloat("Blend Jump", rb.velocity.y);
  }

  IEnumerator Attack(float direction)
  {
      // Memicu animasi serangan
        animator.SetTrigger("Attack");
        // Menunggu selama 0,25 detik
        yield return new WaitForSeconds(0.25f);

        // Membuat peluru di posisi firePoint dengan
rotasi default
        GameObject fireball = Instantiate(bullet,
firePoint.position, Quaternion.identity);
        // Menetapkan kecepatan peluru sesuai dengan
arah yang diberikan
        fireball.GetComponent<Rigidbody2D>().velocity
= new Vector2(direction * 10f, 0);

        // Menghancurkan peluru setelah 2 detik
        Destroy(fireball, 2f);
  }

 // Fungsi untuk memulai serangan ke kanan
    // public void AttackRight()
    // {
    //                         horizontalValue        =
Input.GetAxisRaw("Horizontal");
    // if (Input.GetKeyDown(KeyCode.C))
    // {
    //     StartCoroutine(Attack(1f));
    // }
    //}

    // Fungsi untuk memulai serangan ke kiri
    // public void AttackLeft()
    // {
    //                             horizontalValue     =
Input.GetAxisRaw("Horizontal");
    // if (Input.GetKeyDown(KeyCode.C))
    // {
    //     StartCoroutine(Attack(-1f));
    // }
    // }

  void GroundCheck()
  {
    isGrounded = false;
```

```
    Collider2D[]              colliders              =
Physics2D.OverlapCircleAll(groundcheckCollider.positio
n, groundCheckRadius, groundLayer);
    if (colliders.Length > 0){
        isGrounded = true;
    }
    animator.SetBool("Jumping", !isGrounded);
  }

  void Move(float dir, bool jumpflag)
  {
    if(isGrounded && jumpflag)
 {
  isGrounded = false;
  jumpflag = false;
  rb.AddForce(new Vector2(0f, jumpPower));
 }
    #region gerak kanan kiri
    float   xVal   =   dir   *   speed   *   100   *
Time.fixedDeltaTime;
    Vector2   targetVelocity   =   new   Vector2(xVal,
rb.velocity.y);
    rb.velocity = targetVelocity;

    if (facingRight && dir < 0)
    {
      // ukuran player
      transform.localScale = new Vector3(-0.5f, 0.5f,
0.5f);
      facingRight = false;
    }

    else if (!facingRight && dir > 0)
    {
      // ukuran player
      transform.localScale = new Vector3(0.5f, 0.5f,
0.5f);
      facingRight = true;
    }

    #endregion
  }
}
```

8. Pada inspector player sesuaikan pengaturan berikut pada komponen script player



Gambar 10.7 Mengatur Script Pada INspector

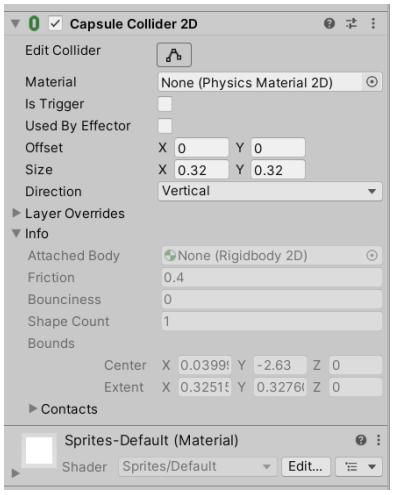
9. Buat script baru Bernama Attack dan isi Source code berikut pada Attack

```
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);
        }
    }
}
```

10. Tambahkan player ninja pada hirarki

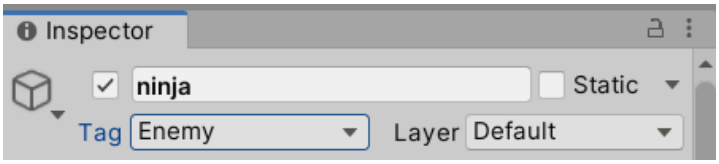
Gambar 10.8 Menambahkan Enemy

11. Kemudian pada inspector ninja, tambahkan komponen capsule collider 2d



Gambar 10.9 MEngatur Capsulle Collider

12. Pada tag ninja, tambahkan tag Enemy



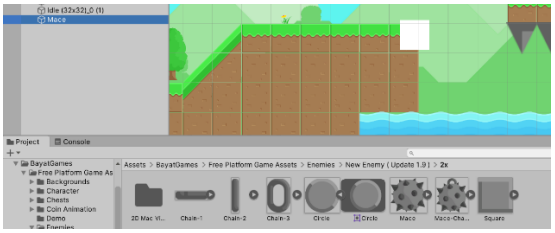
Gambar 10.10 Mengubah Tag

13. Hasil tampilan

Gambar 10.11 Hasil Tampilan

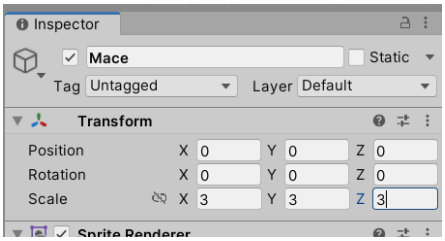
B. Membuat Enemy Behaviour

1. Cari sebuah file Bernama Mace folder Enemies lalu drag ke hirarki



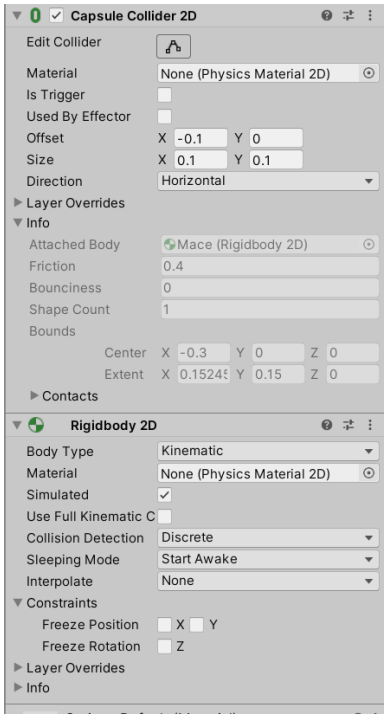
Gambar 10.12 File Enemies

2. Pada inspector Mace atur transform scale menjadi seperti berikut



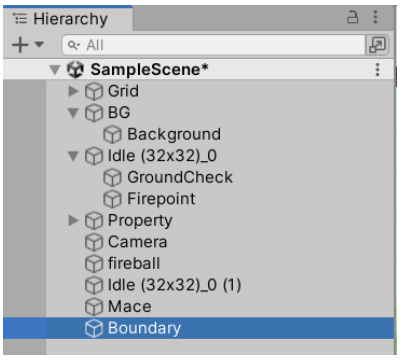
Gambar 10.13 Mengatur Scale Pada Inspector

3. Tambahkan komponen capsule collider 2d dan rigidbody lalu sesuaikan aturannya seperti gambar



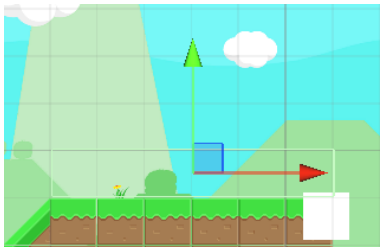
Gambar 10.14 Mengatur Capsulle Collider Dan RigidBody

4. Create empty object pada hirarki ubah nama menjadi Boundary



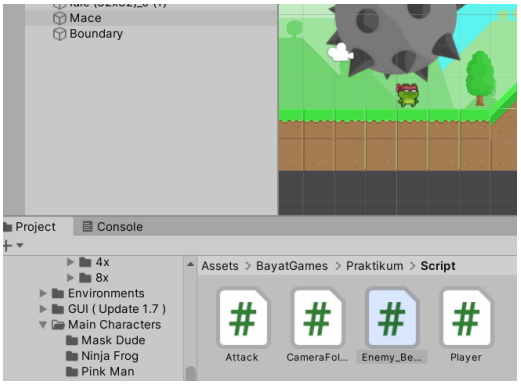
Gambar 10.15 Membuat GameObject

5. Tambahkan box collider 2d pada Boundary, centang is trigger lalu atur size dan offset



Gambar 10.16 Boundary

6. Buat script baru dengan nama Enemy_Behavior lalu drag ke Mace

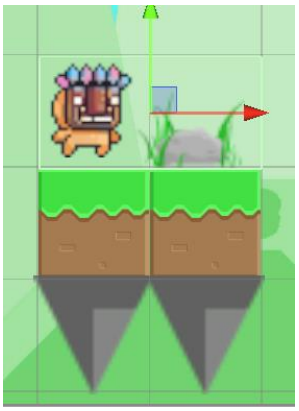


Gambar 10.17 Membuat Script Enemy Behavior

7. Tambahkan script dibawah ini pada Enemy_Behavior

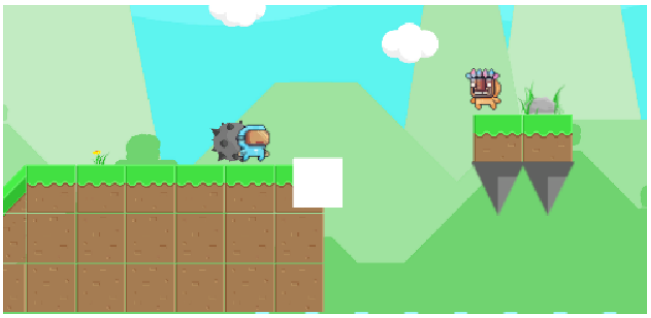
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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);
    }
}
```

8. Tambahkan satu lagi Enemy player topeng dan lakukan mulai Langkah ke 2



Gambar 10.18 Enemy Behavior

9. Jalankan program



Gambar 10.19 Hasil Tampilan Enemy Behavior

C. Membuat Enemy AI

1. Buat script baru dengan nama Enemy_AI

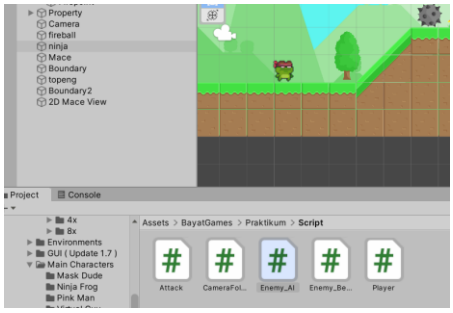
Gambar 10.20 Membuat File C# Enemy_AI

2. 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
```

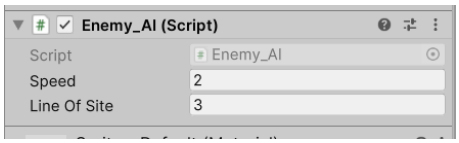
```
    private Transform player; // Transform dari pemain
    private Vector2 initialPosition; // Posisi awal
musuh

    // Use this for initialization
    void Start()
    {
        // Mencari pemain berdasarkan tag
        player =
GameObject.FindGameObjectWithTag("Player").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);
        }
        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);
    }
}
```

3. Drag script Enemy_AI kedalam ninja



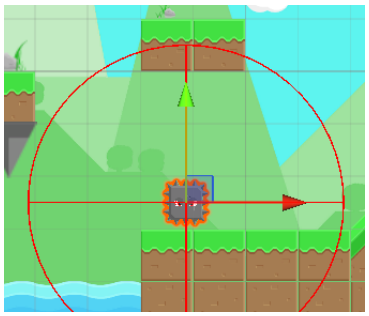
Gambar 10.21 Drag File C# Enemy_AI

4. Pada inspector ninja, atur Speed dan Line of site untuk menentukan jarak dan speed pada enemy



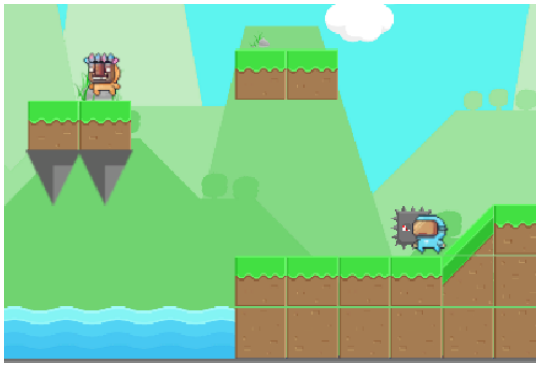
Gambar 10.22 Mengatur Scrip Pada Inspector

5. Tambah 1 lagi enemy 2d Mace View dan lakukan seperti Langkah sebelumnya



Gambar 10.23 Enemy AI

6. Jalankan program



Gambar 10.24 Hasil Tampilan Enemy AI