



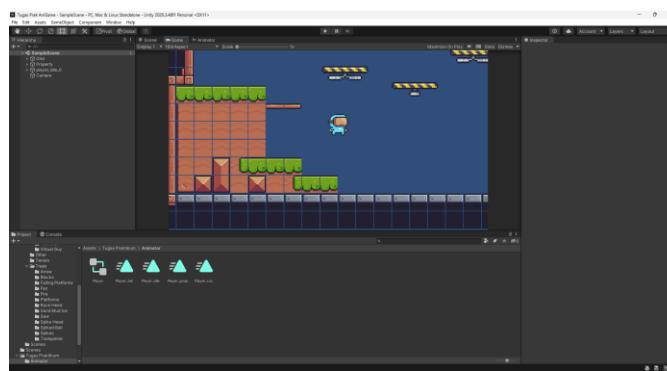
TUGAS PERTEMUAN: 10

Respawn Dan AI Enemy Attack

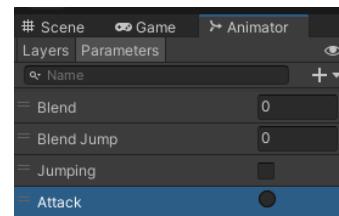
NIM	:	2118014
Nama	:	Muhammad Ridho Putra Syalabi
Kelas	:	D
Asisten Lab	:	NAUFAL DHIAURRAFIF (2218059)

10.1 Tugas 1 : Membuat Mekanisme Attack

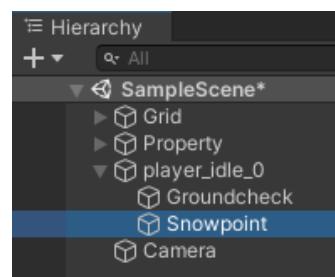
1. Buka Project Sebelumnya



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



3. Buat Layer Game object baru didalam *player-idle-0*, Klik kanan pilih Create Empty lalu Rename menjadi Snowpoint.

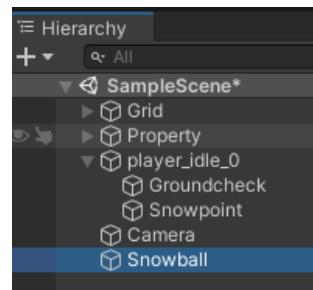




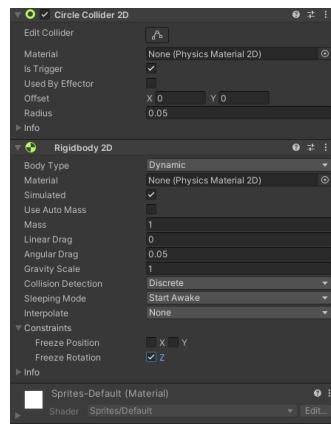
4. Pada menu *Hierarchy* klik *Snowpoint* untuk setting pada *Inspector*, Ubah *Icon* Menjadi titik, atur letak titik didepan player



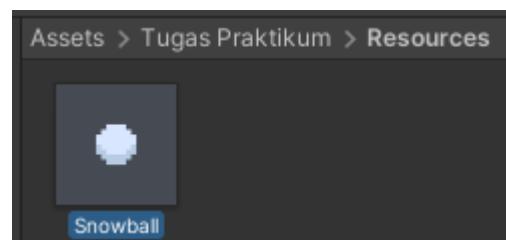
5. Pada menu *Hierarchy* Tambahkan item *Snowball* untuk menjadi object yang dikeluarkan oleh player, *rename* menjadi *snowball*.



6. Klik *Snowball* untuk menambahkan *Component Circle Collider 2D*, dan *Rigidbody 2D*, Setting sesuai gambar dibawah ini



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





8. Tambahkan *code* berikut pada *script player*

```
//dibawah class player
public GameObject bullet;
public Transform firepoint;

//dibawah void fixupdate
IEnumerator Attack() {

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

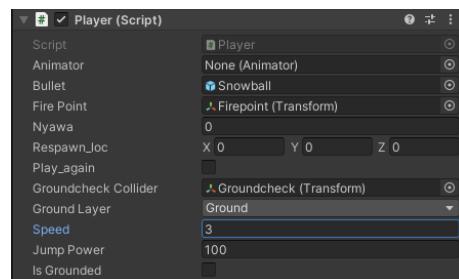
    float direction = 1f;

    GameObject     fireball      =   Instantiate(bullet,
firePoint.position, Quaternion.identity);
    fireball.GetComponent<Rigidbody2D>().velocity
= new Vector2(direction * 10f, 0);

    Destroy(fireball, 2f);
}

//dibawah void update
if (Input.GetKeyDown(KeyCode.C)) {
    StartCoroutine(Attack());
}
```

9. Setting *script player* pada menu inspector seperti berikut



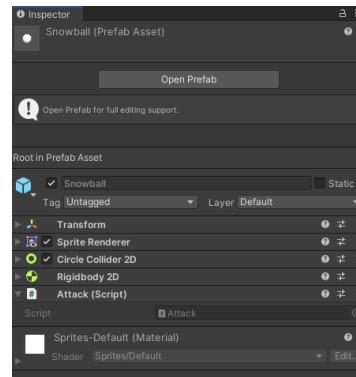
10. Buat *script* baru “attack” kemudian gunakan *code* berikut

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

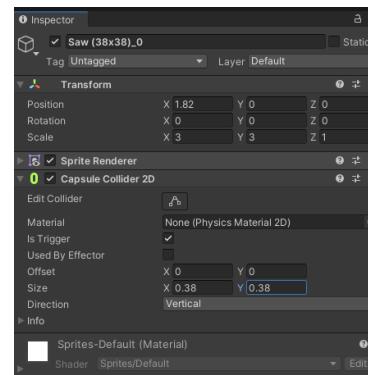
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```



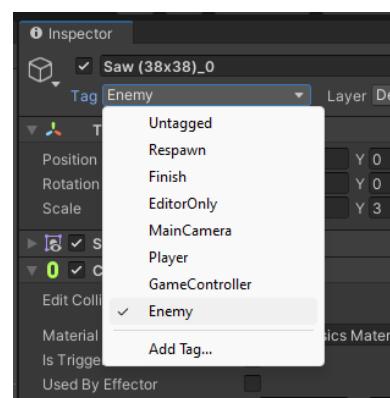
11. Didalam *folder resource* Tambahkan *Script Attack* di *Prefab Snowball*, dengan cara Klik *Snowball* kemudian pada menu *Inspector* arahkan *Script Attack* kedalam *Inspector*.



12. Tambahkan “dog1” sebagai enemy di hirarki, Kemudian klik pada *dog1*, lalu pada menu tab inspector tambahkan collider 2D untuk mendeteksinya



13. Tambahkan Tag baru dengan nama “enemy”, kemudian ubah tag untuk saw_0 menjadi enemy.

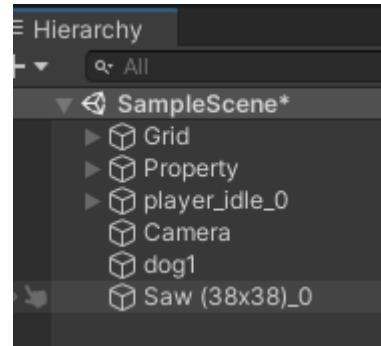




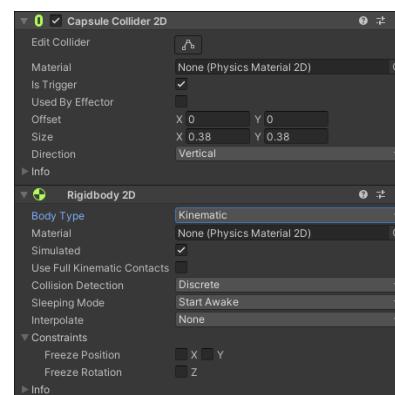
10.2 Tugas 2 : Membuat Enemy AI

a. Enemy Behavior NPC

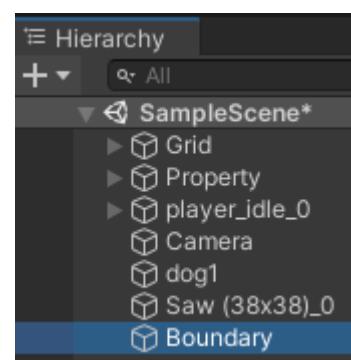
1. Tambahkan Property “Saw” untuk dijadikan NPC di hirarki



2. Kemudian tambahkan *Capsule Colider 2D* dan *Rigidbody* dalam *inspector* game objek Saw. dan pada *Body Type* Ubah menjadi *Kinematic*

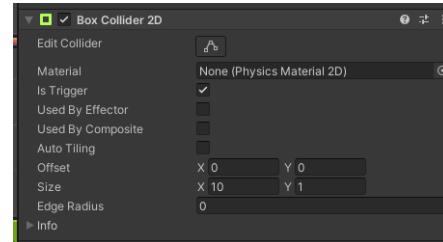


3. Create Empty object pada Hierarchy, Rename Menjadi *Boundary*





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



5. Buat sebuah file script didalam folder Script beri nama “Enemy_Behavior”, kemudian drag dan masukkan ke dalam game object “Saw”. Kemudian gunakan code berikut

```
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);
    }
}
```



b. Enemy AI

1. Buat script baru “Enemy_AI” pada folder script gunakan code berikut

```
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);

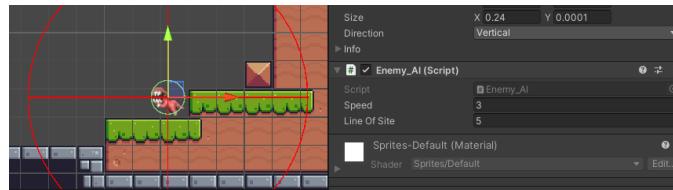
        // Jika pemain berada dalam jarak penglihatan
        // musuh
        if (distanceToPlayer < lineOfSite)
        {
            // Musuh bergerak menuju pemain
            transform.position = Vector2.MoveTowards(this.transform.position,
                player.position, speed * Time.deltaTime);
        }
        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);
    }
}
```

2. Pada inspector dog1 tambahkan script sebelumnya, kemudian konfigurasi sesuai kebutuhan



3. Asd
4. Asd

10.3 Respawn

1. Bujka Player.cs tambahkan code berikut

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

2. Gunakan code berikut untuk mengatur posisi respawn sesuai dengan posisi awal permainan dimulai, pada void awake

```
respawn_loc = transform.position;
```

3. Tambahkan code berikut untuk menggunakan logika kapan untuk player diwajibkan respawn, di void update

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

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

4. Buat void playagain() dibawah void awake()

```
void playagain(){
    if(play_again == true) {
        nyawa = 3;
        transform.position = respawn_loc;
        play_again = false;
    }
}
```



5. Buat script baru “Enemy_attacked.cs”, dan gunakan code berikut

```
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.FindGameObjectWithTag("Player").GetComponent<Player>();
        }
    }

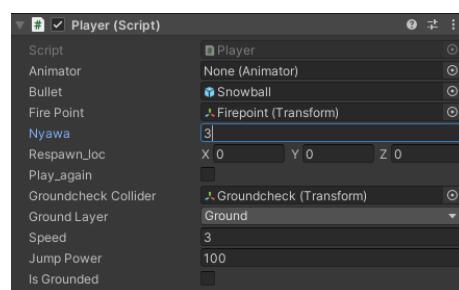
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            Object.nyawa--;

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

6. Pada hirarki dog1 tambahkan script enemy_attack dan arahkan object pada player



7. Kemudian pada objek player dibagian inspector ubah parameter nyawa menjadi sesuka hati





QUIZ

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    public int attackDamage = 10;

    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PerformMeleeAttack();
        }
    }

    void PerformMeleeAttack()
    {
        RaycastHit hit;
        if (Physics.Raycast(transform.position,
        transform.forward, out hit, attackRange))
        {
            component
            EnemyHealth enemyHealth =
            hit.collider.GetComponent<EnemyHealth>();
            if (enemyHealth != null)
            {
                enemyHealth.TakeDamage(attackDamage);
            }
        }
    }
}
```



Perbaikan code

```
void HandleJumpInput()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        animator.SetBool("isJumping", true); // ✗
        Hilang nilai boolean (true atau false)
        rb.AddForce(Vector2.up * jumpForce,
        ForceMode2D.Impulse);
    }
    else if (Input.GetKey(KeyCode.Space))
    {
        animator.SetBool("isJumping", true); // ✗
        Hilang nilai boolean (true atau false)
    }
}

void HandleMovementInput()
{
    float move = Input.GetAxis("Horizontal");

    if (move == 0) // ✗ Salah kondisi pengecekan,
    seharusnya == 0 untuk idle
    {
        animator.SetBool("isIdle", true);
        transform.Translate(Vector3.left * move *
    Time.deltaTime);
    }
    else
    {
        animator.SetBool("isWalking", true); // ✗
        Seharusnya true untuk menandakan sedang berjalan
    }

    if (move < 0) // ✗ Kesalahan logika, seharusnya
    mengecek arah negatif terlebih dahulu
    {
```



```
        transform.localScale = new Vector3(-1, 1, 1); //  
X Perbaiki skala untuk arah kiri  
    }  
    else if (move > 0)  
    {  
        transform.localScale = new Vector3(1, 1, 1); //  
X Perbaiki skala untuk arah kanan  
    }  
}
```

Analisa :

Kesalahan pada kode melibatkan beberapa aspek: nilai boolean yang hilang pada `animator.SetBool` untuk parameter "isJumping", logika pengecekan yang salah pada `if (move != 1)` yang seharusnya memeriksa apakah `move` adalah nol untuk mengatur status idle, serta kesalahan dalam mengatur animasi berjalan yang seharusnya `true` saat karakter bergerak. Selain itu, logika pengecekan arah pada `if (move != 0)` dan `else if (move > 0)` perlu diubah, dengan pengecekan arah negatif (kiri) terlebih dahulu dan arah positif (kanan) kemudian, serta memperbaiki skala transformasi untuk arah yang tepat. Dengan perbaikan ini, kode menjadi lebih logis dan tidak menghasilkan error.