



TUGAS PERTEMUAN: 10

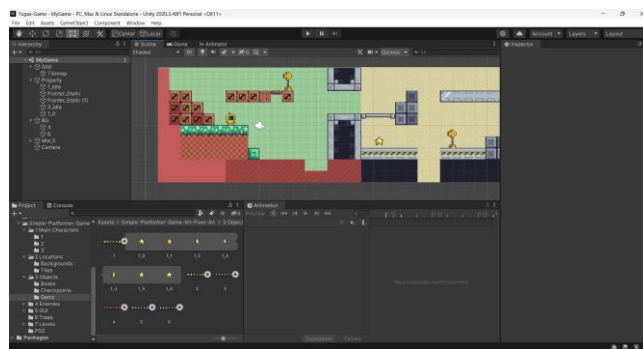
RESPAWN AND AI ENEMY ATTACK

NIM	:	2118028
Nama	:	Novianto Aldo Wibisono
Kelas	:	A
Asisten Lab	:	Aprillia Dwi Dyah S. (2118143)

10.1 Tugas 1 : Membuat Respawn And AI Enemy Attack

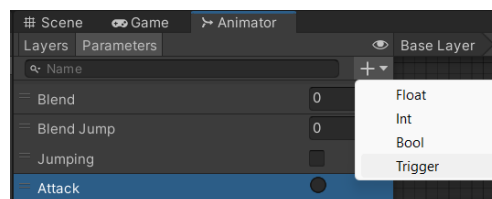
A. Mempersiapkan Mekanisme Menyerang

1. Buka *project* Unity sebelumnya yang telah ditambahkan *game animation*.



Gambar 10.1 Membuka File Project

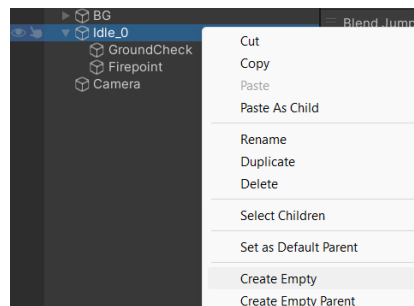
2. Buka tab Animator, tambahkan parameter *Trigger* dengan nama *Attack*.



Gambar 10.2 Menambah Parameter

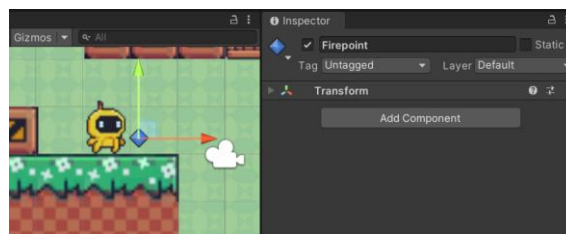


3. Selanjutnya buat *layer game object* baru di dalam *Idle_0*, klik kanan pilih *Create Empty* dan ubah nama menjadi *Firepoint*.



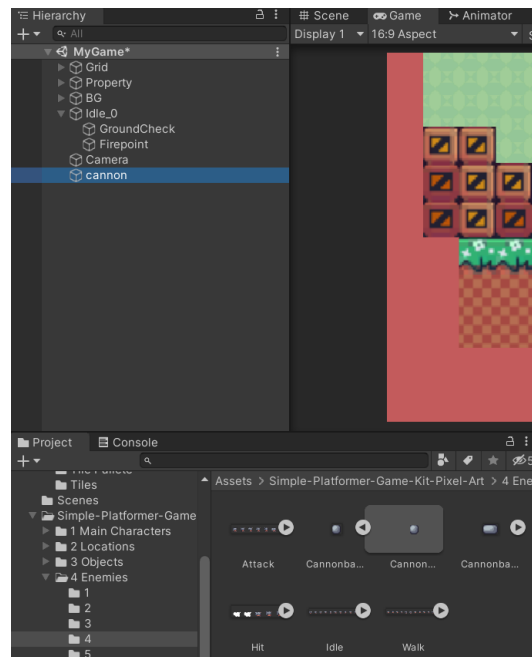
Gambar 10.3 Membuat Game Object Baru

4. Kemudian atur *icon Firepoint* menjadi titik melalui *tab Inspector* dan letakkan di depan *player*.



Gambar 10.4 Memindahkan Firepoint

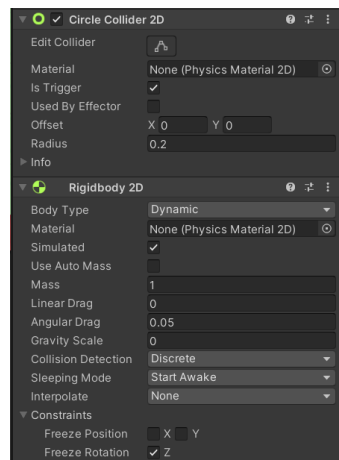
5. Lalu tambahkan satu *item* sebagai peluru pada *folder Enemies* lalu *folder* 4 pilih *Cannonball 1* dan ubah namanya menjadi *cannon*.



Gambar 10.5 Menambahkan Peluru

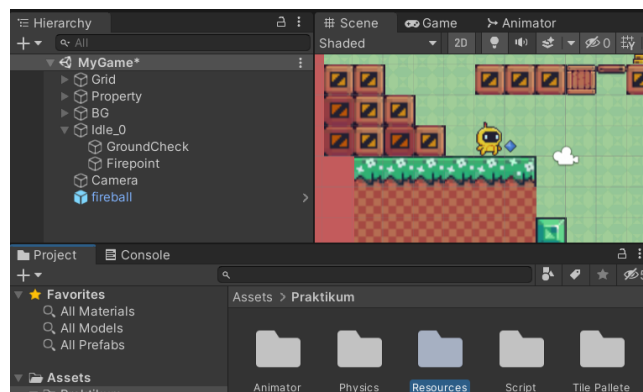


6. Kemudian tambahkan komponen *Circle Collider 2D* dan *Rigidbody 2D* serta atur seperti berikut.



Gambar 10.6 Menambahkan Komponen

7. Buat *folder* baru *Resources* pada *folder project*, kemudian *drag and drop* *fireball* ke dalam *folder Resources* dan hapus *fireball* pada *Hierarchy*.



Gambar 10.7 Membuat Folder Resources

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

```
public Animator animator;
// Deklarasi objek bullet dan firePoint serta
variabel direction
public GameObject bullet;
public Transform firePoint;
float direction;

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

    if (Input.GetKeyDown(KeyCode.C))
    {
        StartCoroutine(Attack());
    }
}
```



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

    // Letakkan setelah fungsi FixedUpdate
    IEnumerator Attack()
    {
        animator.SetTrigger("Attack");
        yield return new WaitForSeconds(0.25f);

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

        Destroy(cannon, 2f);
    }

    void Move(float dir, bool jumpflag)
    {
        if(isGrounded && jumpflag)
        {
            isGrounded = false;
            jumpflag = false;
            rb.AddForce(new Vector2(0f, jumpPower));
        }

        #region bergerak kanan kiri
        float xVal = dir * speed * 100 *
        Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
        rb.velocity.y);
        rb.velocity = targetVelocity;

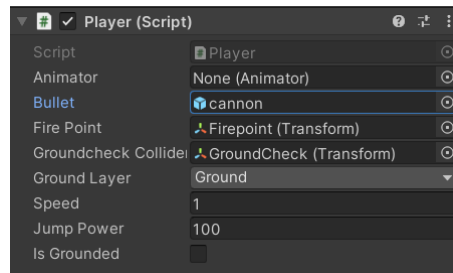
        if (facingRight && dir < 0)
        {
            transform.localScale = new Vector3(-4, 4, 4);
            facingRight = false;
            direction = -1f;
        }

        else if (!facingRight && dir > 0)
        {
            transform.localScale = new Vector3(4, 4, 4);
            facingRight = true;
            direction = 1f;
        }

        #endregion
    }
}
```



9. Pada *tab Inspector Player*, ubah *Bullet* dengan objek yang akan ditembakkan sedangkan *Fire Point* sebagai titik tembak pertama.



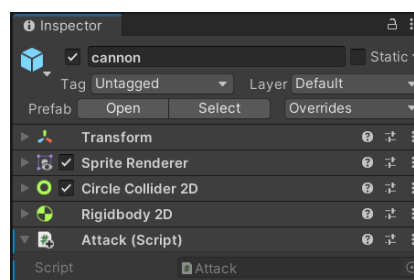
Gambar 10.8 Mengatur Player

10. Buat *script Attack* pada *folder Script* dan tambahkan *script* 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);
        }
    }
}
```

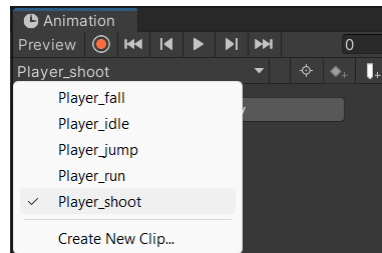
11. Pada *folder Resources* tambahkan *script Attack* ke dalam *Prefab fireball*, dengan cara klik *fireball* kemudian pada *tab Inspector* arahkan *script Attack* ke dalam *Inspector*.



Gambar 10.9 Menambahkan Script Attack

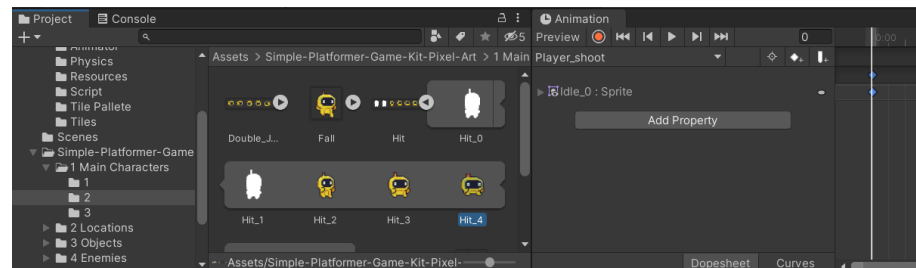


12. Tambahkan animasi ketika menembak, klik *Player* dan buat *clip* baru dengan nama *Player_shoot*.



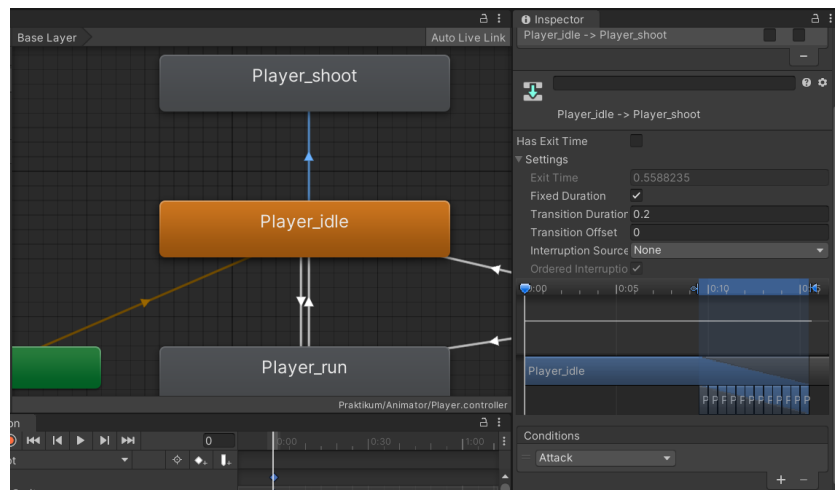
Gambar 10.10 Menambahkan Clip Baru

13. Lalu *drag and drop* animasi yang ingin digunakan ke dalam *timeline* animasi.



Gambar 10.11 Menambahkan Animasi

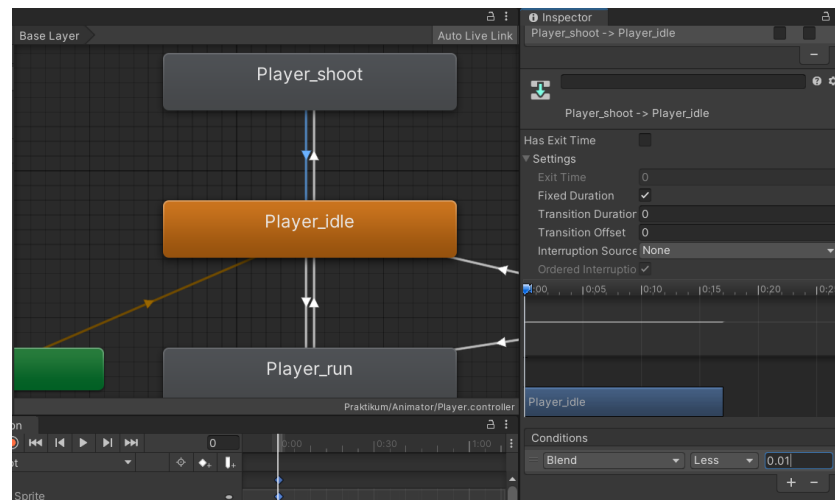
14. Kemudian buat transisi dari *Player_idle* ke *Player_shoot*, buat kondisi menjadi *Attack* dan beri durasi transisi 0.2 serta atur seperti berikut.



Gambar 10.12 Transisi Idle Ke Shoot

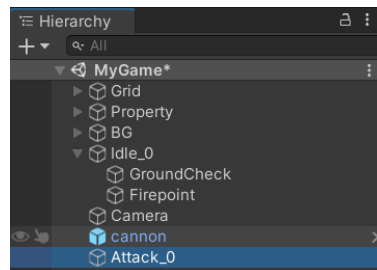


15. Selanjutnya buat transisi dari Player_shoot ke Player_idle, buat kondisi *Blend* dan *Less* dengan *value* 0.01 serta atur seperti berikut.



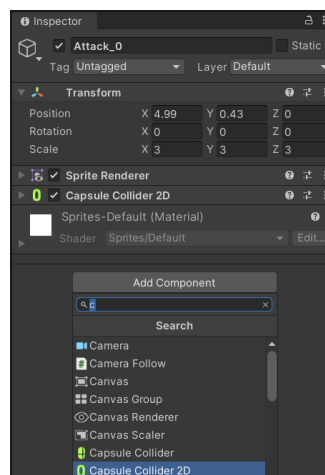
Gambar 10.13 Transisi Shoot Ke Idle

16. Tambahkan *Enemy* yang ada pada *folder Enemies* lalu *folder* 5, pilih *Attack_0*.



Gambar 10.14 Menambahkan Enemy

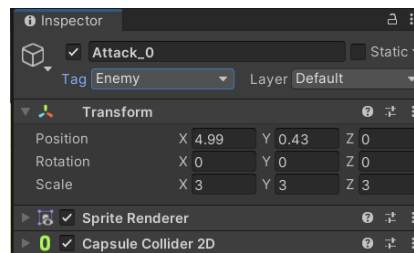
17. Klik pada *Attack_0*, lalu pada *tab Inspector* tambahkan *Capsule Collider 2D*.



Gambar 10.15 Menambahkan Capsule Collider 2D



18. Pilih *Tag* dan tambahkan *tag* baru dengan nama *Enemy*, kemudian tambahkan *tag Enemy* pada *Attack_0*.



Gambar 10.16 Mengubah Tag

19. Coba jalankan *game* dan tembak dengan menekan tombol C untuk menghancurkan musuh.



Gambar 10.17 Menjalankan Game

B. Membuat Enemy Behavior

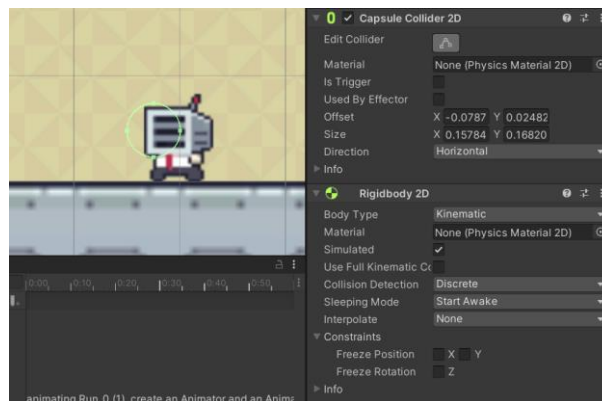
1. Tambahkan *sprite Run* ke dalam hirarki dari *folder Enemies* pada *folder 1* dan 2.



Gambar 10.18 Menambahkan Sprite

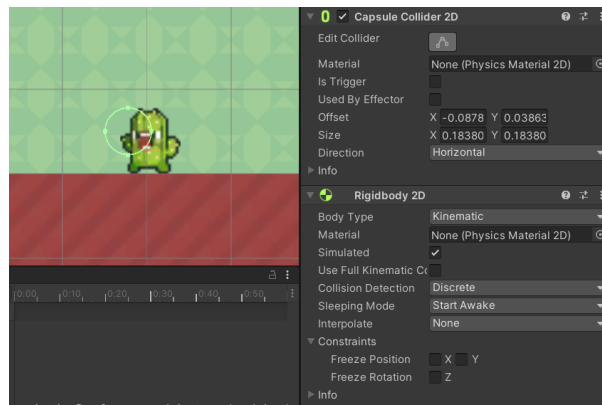


2. Pada kedua *enemy* tersebut tambahkan komponen *Capsule Collider 2D* dan *Rigidbody 2D*. Lalu atur pada *enemy 1* sebagai berikut.



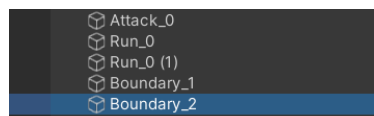
Gambar 10.19 Pengaturan Enemy 1

3. Kemudian untuk *enemy 2* sebagai berikut.



Gambar 10.20 Pengaturan Enemy 2

4. Buat 2 *empty object* pada hirarki dengan nama *Boundary_1* dan *Boundary_2*.



Gambar 10.21 Membuat Empty Object



5. Tambahkan *Box Collider 2D* pada tiap *Boundary*, centang *Is Trigger* dan sesuaikan *size* dan *offset* sebagai batas pergerakan *enemy* serta rotasi sumbu Y pada *enemy* sebesar 180.



Gambar 10.22 Menambahkan Box Collider

6. Lalu buat sebuah *script* baru dengan nama *Enemy_Behavior* dan tambahkan *script* berikut.

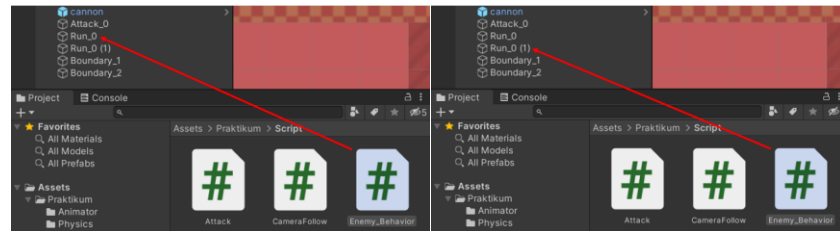
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 1f;
    Rigidbody2D rb;

    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);
    }
}
```

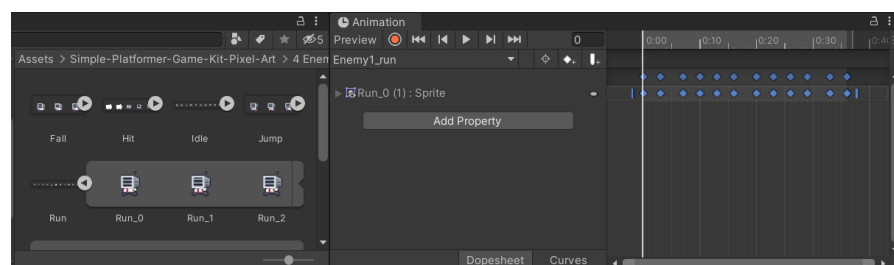


7. Kemudian *drag and drop* script pada *enemy 1* dan *enemy 2*.



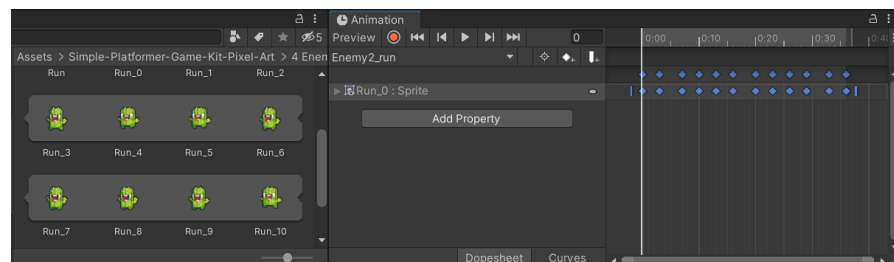
Gambar 10.23 Menambahkan Script

8. Lalu tambahkan animasi berjalan pada *enemy 1* dengan membuat *clip* baru dan tambahkan animasi *sprite* berjalan ke dalam *timeline* serta atur waktunya.



Gambar 10.24 Menambah Animasi Enemy 1

9. Selanjutnya tambahkan animasi berjalan pada *enemy 2* dengan membuat *clip* baru dan tambahkan animasi *sprite* berjalan ke dalam *timeline* serta atur waktunya.



Gambar 10.25 Menambah Animasi Enemy 2

10. Kemudian jalankan *game* dan pastikan *enemy 1* dan *2* berjalan.

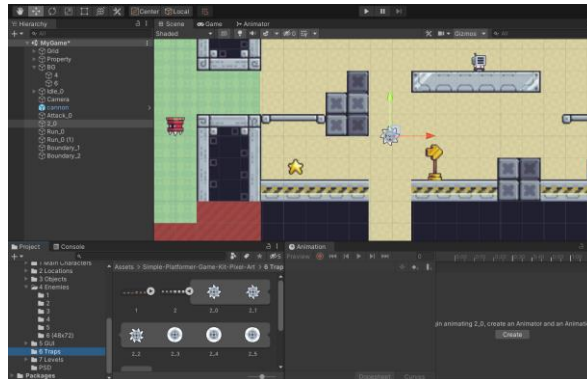


Gambar 10.26 Menjalankan Game



C. Membuat Enemy AI

1. Cari sebuah *sprite* bernama 2_0 dan tambahkan ke dalam hirarki.



Gambar 10.27 Menambahkan Sprite

2. Buat *script* Enemy_AI dan tambahkan *script* 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

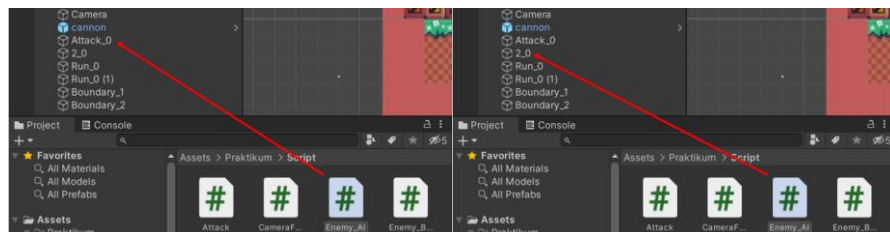
    // Use this for initialization
    void Start()
    {
        // Mencari pemain berdasarkan tag
        player =
        GameObject.FindGameObjectWithTag("Player").transform;
        // Menyimpan posisi awal musuh
        initialPosition =
        GetComponent<Transform>().position;
    }
    // Update 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
        }
    }
}
```



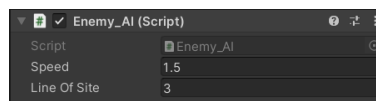
```
        transform.position =  
Vector2.MoveTowards(this.transform.position,  
player.position, speed * Time.deltaTime);  
    }  
    else  
    {  
        // Musuh kembali ke posisi awal  
        transform.position =  
Vector2.MoveTowards(transform.position,  
initialPosition, speed * Time.deltaTime);  
    }  
}  
// Untuk menggambar jarak penglihatan musuh di  
editor  
private void OnDrawGizmosSelected()  
{  
    Gizmos.color = Color.red;  
    Gizmos.DrawWireSphere(transform.position,  
lineOfSite);  
}  
}
```

3. Kemudian *drag and drop script Enemy_AI* ke *enemy 3* dan *4*.



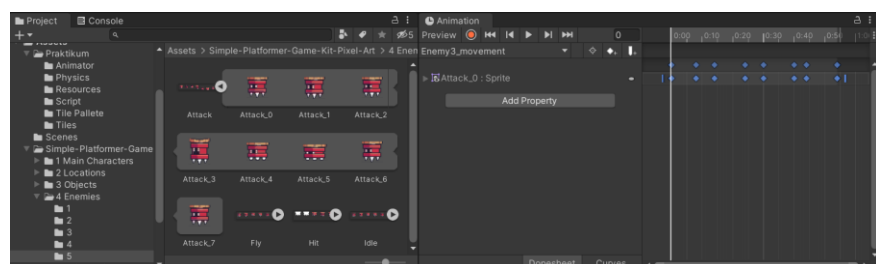
Gambar 10.28 Menambahkan Script

4. Lalu atur *Speed* dan *Line of Site* untuk menentukan jarak terhadap *player* dan kecepatan.



Gambar 10.29 Mengatur Kecepatan Dan Jarak

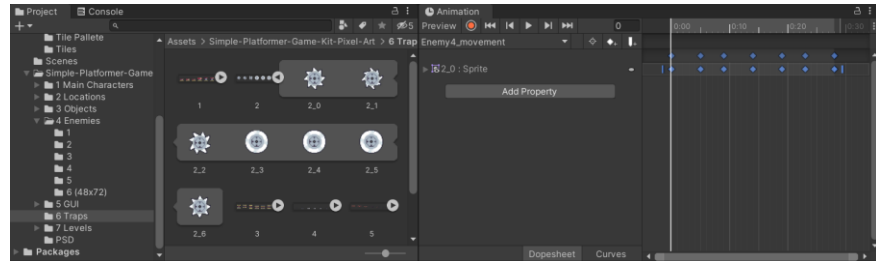
5. Lalu tambahkan animasi bergerak pada *enemy 3* dengan membuat *clip* baru dan tambahkan animasi *sprite* bergerak ke dalam *timeline* serta atur waktunya.



Gambar 10.30 Menambah Animasi Enemy 3



6. Selanjutnya tambahkan animasi bergerak pada *enemy 4* dengan membuat *clip* baru dan tambahkan animasi *sprite* bergerak ke dalam *timeline* serta atur waktunya.



Gambar 10.31 Menambah Animasi Enemy 4

D. Respawn

1. Tambahkan *script* berikut ke dalam *script Player* untuk mengembalikan *player* ke tempat awal ketika jatuh atau nyawa kurang dari 0.

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

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

    // Menentukan tempat respawn
    respawn_loc = transform.position;
}

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

    if (Input.GetKeyDown(KeyCode.C))
    {
        StartCoroutine(Attack());
    }

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

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



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

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

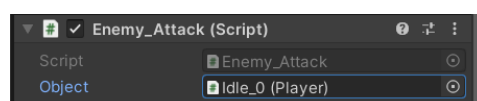
2. Kemudian buat *script* baru dengan nama `Enemy_Attack` dan tambahkan *script* berikut.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_Attack : 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;
            }
        }
    }
}
```

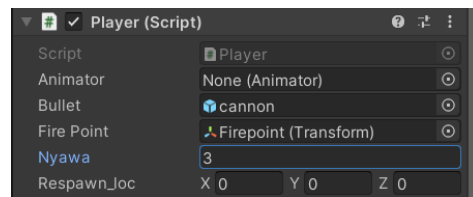
3. Lalu *drag and drop script* `Enemy_Attack` ke semua musuh dan arahkan *object* pada `Idle_0`.



Gambar 10.32 Menambahkan Script

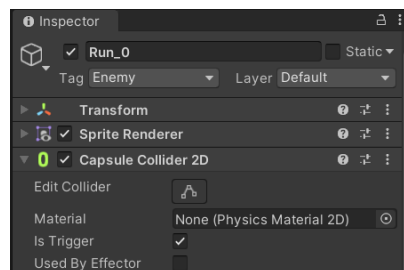


4. Pergi ke *tap Inspector game object Player* dan ubah nilai Nyawa menjadi 3 pada *Player (Script)*.



Gambar 10.33 Mengubah Nilai Nyawa

5. Kemudian pastikan bahwa setiap musuh menggunakan *Tag Enemy* dan memiliki komponen *Capsule Collider 2D* dengan *Is Trigger* check.



Gambar 10.34 Tab Inspector Enemy

6. Lalu coba jalankan *game*, maka ketika *player* menyentuh musuh, nyawa akan berkurang 1 dan jika nyawa kurang dari 0 atau *player* jatuh maka akan *respawn* ke titik awal.

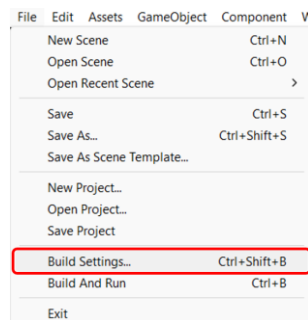


Gambar 10.35 Menjalankan Game



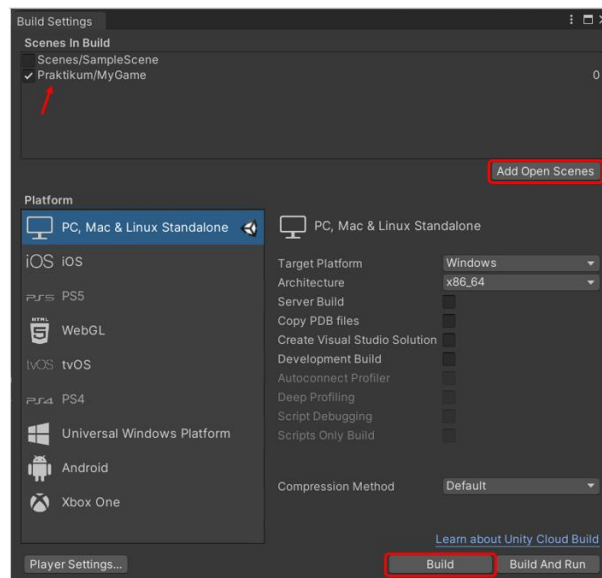
E. Render Game

1. Lalu untuk merender *game*, pilih *Build Settings* pada menu *File*.



Gambar 10.36 Menu File

2. Selanjutnya *Add Open Scenes* dan pilih *Scene* pada *project* yang ingin dirender, lalu pilih *Build* dan simpan kedalam *folder* yang diinginkan.



Gambar 10.37 Build Settings

F. Link Github Pengumpulan

https://github.com/Nziaxi/2118028_PRAK_ANIGAME



10.2 Kuis : Melengkapi Source Code

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public int 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))
        {
            Health enemyHealth =
                hit.transform.GetComponent<Health>();
            if (enemyHealth != null)
            {
                enemyHealth.TakeDamage(attackDamage);
            }
        }
    }
}
```

Penjelasan:

Mula-mula akan diperiksa apakah objek yang terkena serangan memiliki komponen *Health*. Jika objek tersebut memiliki komponen *Health* dengan nilai tidak *null*, maka fungsi *TakeDamage* dipanggil pada objek tersebut untuk mengurangi *Health* dengan jumlah *attackDamage*. Sehingga dapat dipastikan hanya objek dengan komponen *Health* yang akan menerima serangan.