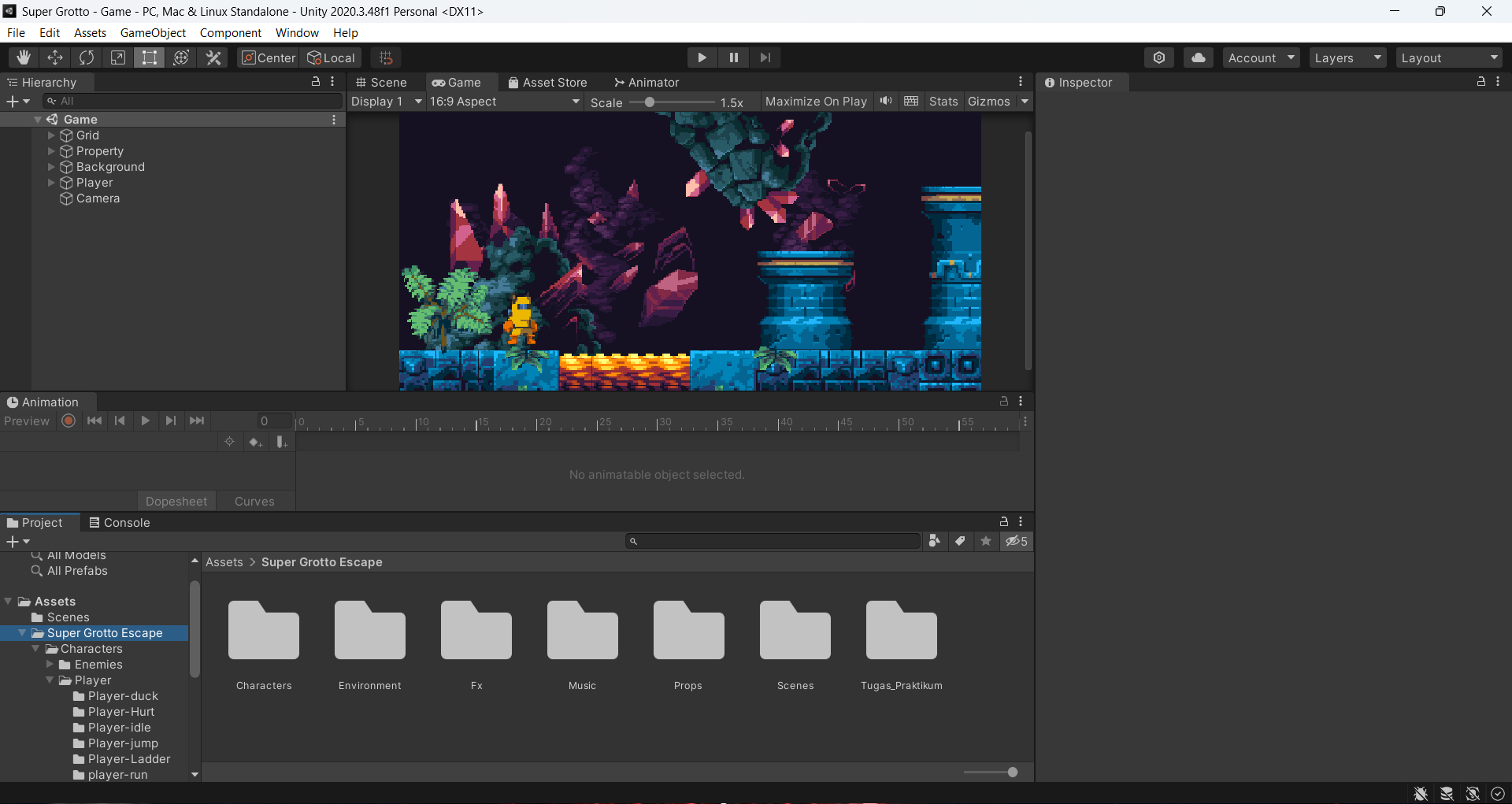
# 10 RESPAWN dan AI ENEMY ATTACK

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118107 |
| **Nama** | : | Rangga Aditia Abiyova |
| **Kelas** | : | D |
| **Asisten Lab** | : | Zain Aryanta (2118051) |

## Tugas 1 : Membuat Respawn dan AI Enemy Attack

1. Mekanisme Attack
2. Untuk membuat *respawn* dan *enemy attack* yang pertama buka projek sebelumnya.



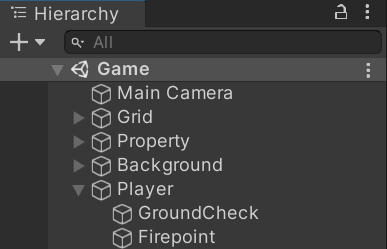
### 10.1 Membuka Projek Sebelumnya

1. Lalu menuju tab *Animator* kemudian buat *parameters* baru pada *tab Animator* dengan nama *Attack* dan tipe data *Trigger*.



### 10.2 Membuat Parameter Baru

1. Kemudian pada *hierarchy* klik kanan pada *Player* kemudian pilih *Create Empty* dan beri nama *Firepoint*.



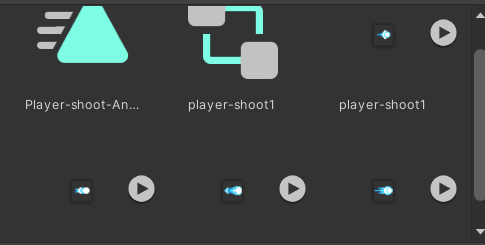
### 10.3 Membuat Hierarchy Baru

1. Kemudian klik *firepoint* dan ubah *icon* pada *inspector* menjadi seperti berikut.



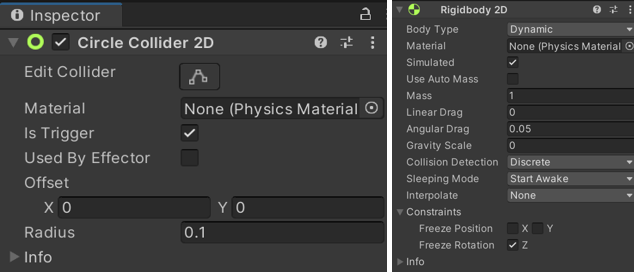
### 10.4 Mengubah Icon Firepoint

1. Masukkan *asset* pada *hierarchy* sebagai pelurunya kemudian ubah nama menjadi *Shoot*.



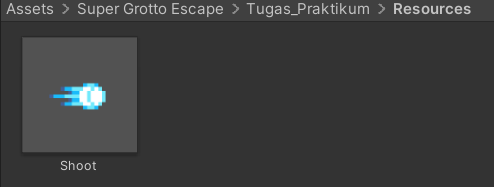
### 10.5 Menambahkan Asset

1. Klik *hierarchy Shoot*, kemudian tambahkan *component* seperti berikut.



### 10.6 Menambahkan Component

1. Lalu buatlah *folder* baru pada *folder* praktikum dengan nama *Resources* dan masukkan *hierarchy Sword* pada *folder*.



### 10.7 Memasukkan Hierarchy Sword

1. Kemudian buka *script* *Player* lalu masukkan *code* dibawah ini.

|  |
| --- |
| public GameObject bullet;  public Transform firePoint; |

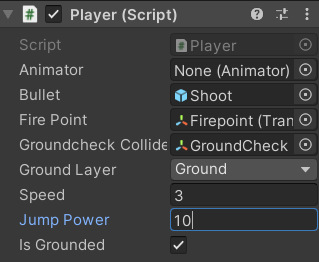
1. Setelah menambahkan *variable* diatas kemudian masukkan *code* berikut.

|  |
| --- |
| 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);  } |

1. Lalu tambahkan code berikut pada fungsi Update.

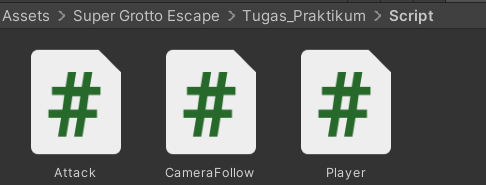
|  |
| --- |
| if (Input.GetKeyDown(KeyCode.C))  {  StartCoroutine(Attack());  } |

1. Kemudian klik *hierarchy Player* pada *Inspector component script* atur pada bagian *Bullet* dan *Fire Point*.



### 10.8 Mengatur Component Script

1. Lalu pada *folder Script* buatlah *script* baru dengan nama *Attack*.

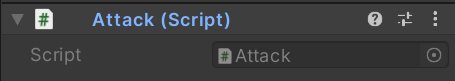


### 10.9 Menambahkan Script Attack

1. Buka script Attack yang telah dibuat, kemudian tambahkan code dibawah ini.

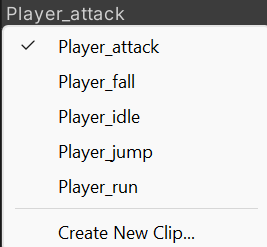
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Attack : MonoBehaviour  {  private void OnTriggerEnter2D(Collider2D collision)  {  if (collision.gameObject.CompareTag("Enemy"))  {  Destroy(gameObject);  Destroy(collision.gameObject);  }  }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

1. Klik *Sword* pada *folder resources* kemudian tambahkan *script* *Attack* pada *inspector*.



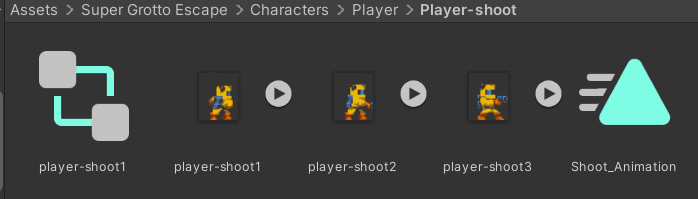
### 10.10 Menambahkan Script

1. Kemudian buat *animation* baru dengan cara klik karakter dan pilih *Create New Clip* dan buat *Player\_attack* untuk memberi animasi saat *shoot*.



### 10.11 Menambahkan Clip Baru

1. Kemudian tambahkan *asset* animasi *shoot* pada *folder asset*.



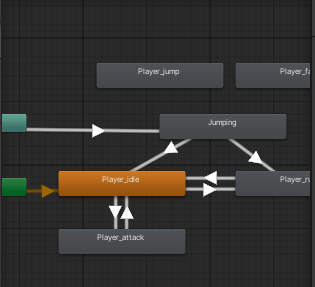
### 10.12 Menambahkan Animasi *Shoot*

1. Buat parameter baru pada *tab animator* beri nama Serang dengan tipe data *Bool*.



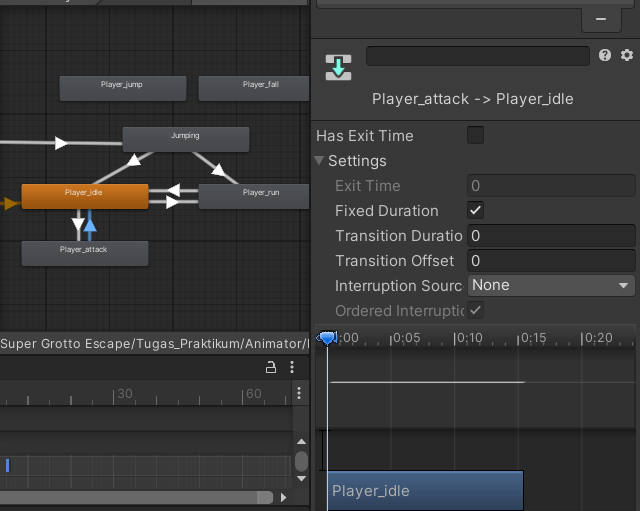
### 10.13 Menambahkan Parameter Baru

1. Buat transisi pada *player\_idle* ke *player\_attack* dan klik arah panah beri kondisi *Shoot* dengan kondisi *true* dan atur juga *setting* nya.



### 10.14 Membuat Transisi

1. Buat transisi lagi pada *player\_attack* ke *player\_idle* dan bari kondisi *Shoot* dengan kondisi *false*.



### 10.15 Membuat Transisi Lagi

1. Lalu buka *script Player* kemudian tambahkan *variable* berikut.

|  |
| --- |
| bool *shoot*; |

1. Selanjutnya rubah kondisi berikut pada *void Update*.

|  |
| --- |
| if (Input.GetKeyDown(KeyCode.C) && !*shoot*)  {  StartCoroutine(Attack());  } |

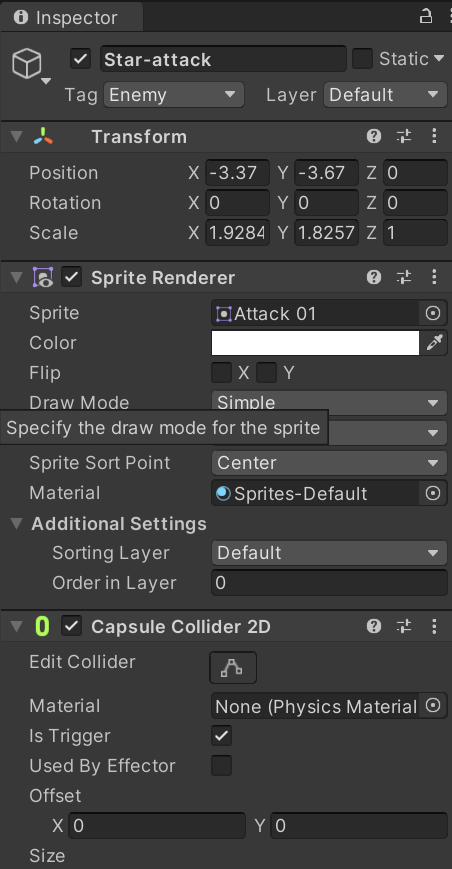
1. Lalu rubah fungsi attack seperti berikut.

|  |
| --- |
| IEnumerator Attack()  {    animator.SetTrigger("Attack");  *shoot* = true;  animator.SetBool("*Shoot*", true);  yield return new WaitForSeconds(0.25f);    float direction = facingRight ? 1f : -1f;    GameObject fireball = Instantiate(bullet, firePoint.position, Quaternion.identity);  fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 10f, 0);  animator.SetBool("*Shoot*", false);  *shoot* = false;    Destroy(fireball, 2f);  } |

1. Rubah juga fungsi pada *void Move*.

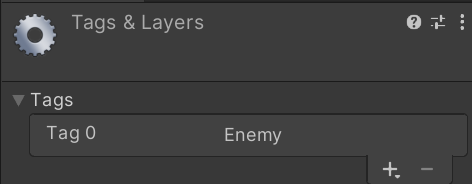
|  |
| --- |
| void Move(float dir, bool jumpflag, bool serangFlag)  {  float xVal = dir \* speed \* 100 \* Time.fixedDeltaTime;  Vector2 targetVelocity = new Vector2(xVal, rb.velocity.y);  rb.velocity = targetVelocity;  *shoot* = serangFlag;  if(isGrounded && serangFlag)  {  serangFlag = true;  }  if(isGrounded && jumpflag)  {  isGrounded = false;  jumpflag = false;  rb.AddForce(new Vector2(0f, jumpPower));  }  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;  }  } |

1. Kemudian tambahkan *asset enemy* pada *Hierarchy*, lalu tambahkan component Capsule Collider 2D.



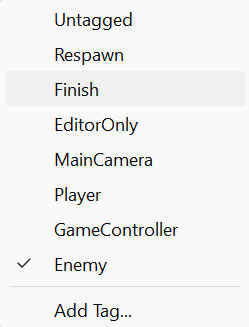
### 10.16 Menambahkan Enemy dan Component

1. Kemudian tambahkan *tag* baru dengan nama *Enemy*.



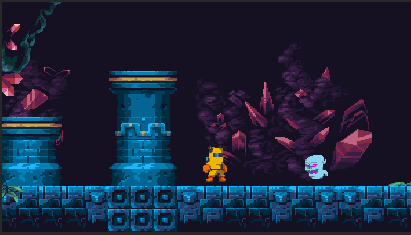
### 10.17 Menambahkan Tag Baru

1. Rubah *tag* pada *Star-attack* menjadi *tag Enemy*.



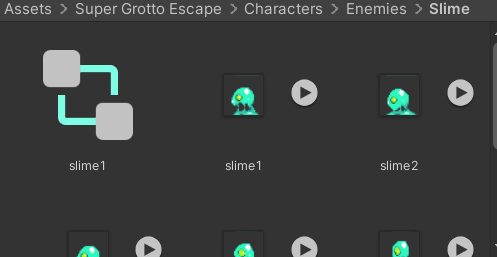
### 10.18 Merubah Tag

1. Kemudian lakukan test dengan menjalankan gamenya.



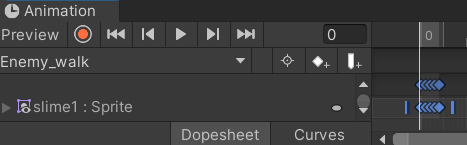
### 10.19 Melakukan Test Enemy

1. Enemy Behavior NPC
2. Untuk melakukan *Enemy Behavior*, pertama tambahkan *asset enemy* kedua pada *Hierarchy*.



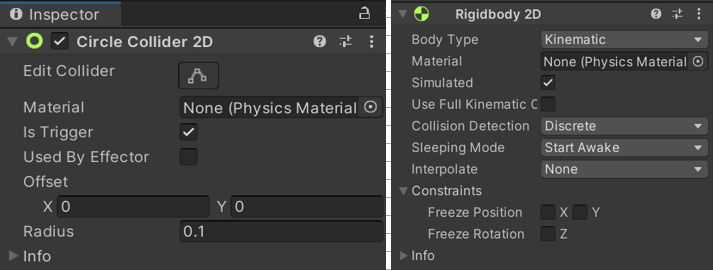
### 10.20 Menambahkan Enemy Kedua

1. Lalu tambahkan clip baru untuk animasi bergerak pada *enemy* kedua.



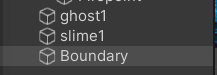
### 10.21Menambahkan Animasi Enemy Kedua

1. Pada *hierarchy* *enemy* kedua tambahkan *component* baru dan juga atur *component*-nya.



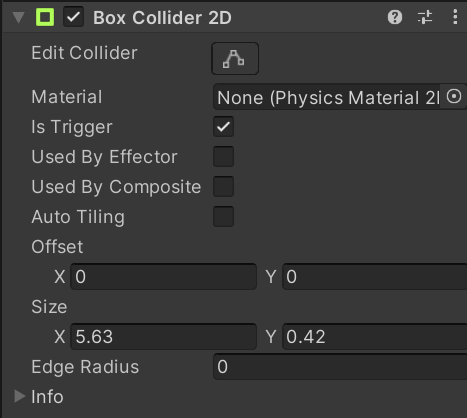
### 10.22 Menambahkan Component

1. Lalu buatlah *hierarchy* baru dengan nama *Boundary*.



### 10.23 Membuat Hierarchy Baru

1. Pada *hierarchy Boundary* tambahkan *component Box Collider* 2D.



### 10.24 Menambahkan Component

1. Lalu buatlah *script* baru dengan nama *Enemy\_Behavior*.

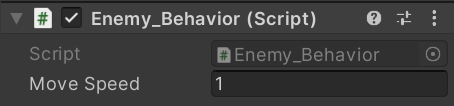


### 10.25 Menambahkan Script Baru

1. Kemudian buka script Enemy\_Behavior lalu tambahkan code berikut ini.

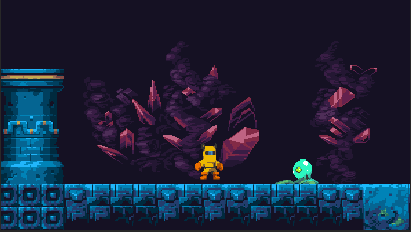
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_Behavior : MonoBehaviour  {  [SerializeField] float moveSpeed = 1f;  Rigidbody2D rb;  void Start()  {  rb = GetComponent<Rigidbody2D>();  }  void Update()  {  if (isFacingRight())  {  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);  }  } |

1. Lalu pada *inspector enemy* kedua tambahkan *script Enemy\_Behavior*.



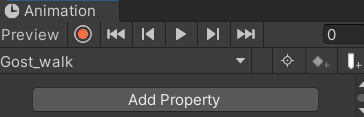
### 10.26 Menambahkan Component

1. Kemudian lakukan test dengan menjalankan gamenya.



### 10.27 Melakukan Test Enemy Behavior

1. Enemy AI
2. Untuk membuat *enemy AI* pertama klik *hierarchy* *enemy* pertama, kemudian tambahkan animasnya.



### 10.28 Menambahkan Animasi Pada Enemy Pertama

1. Lalu pada *folder script* tambahkan *script* baru dengan nama *Enemy\_AI*.

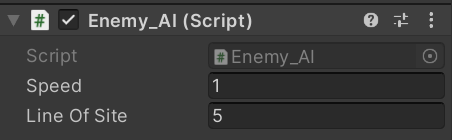


### 10.29 Membuat Script Baru

1. Lalu buka *script* tersebut kemudian tambahkan *code* berikut ini.

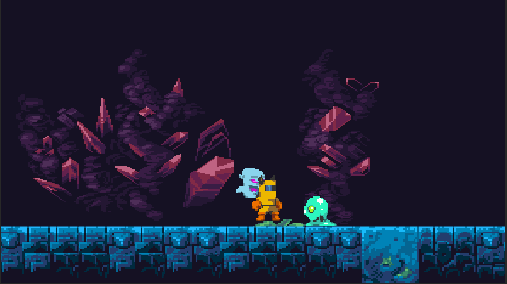
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_AI : MonoBehaviour  {  public float speed; // Kecepatan gerakan musuh  public float lineOfSite; // Jarak penglihatan musuh  private Transform player; // Transform dari pemain  private Vector2 initialPosition; // Posisi awal musuh  private bool facingRight = true; // Menyimpan arah awal musuh (menghadap 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);  // Menghadapkan musuh ke arah pemain  FlipTowardsPlayer();  }  else  {  // Musuh kembali ke posisi awal  transform.position = Vector2.MoveTowards(transform.position, initialPosition, speed \* Time.deltaTime);  // Menghadapkan musuh ke arah awal jika tidak mengejar pemain  FlipTowardsInitialPosition();  }  }  // Menghadapkan musuh ke arah pemain  void FlipTowardsPlayer()  {  if (player.position.x > transform.position.x && !facingRight)  {  Flip();  }  else if (player.position.x < transform.position.x && facingRight)  {  Flip();  }  }  // Menghadapkan musuh ke arah awal  void FlipTowardsInitialPosition()  {  if (initialPosition.x > transform.position.x && !facingRight)  {  Flip();  }  else if (initialPosition.x < transform.position.x && facingRight)  {  Flip();  }  }  // Membalik arah musuh  void Flip()  {  facingRight = !facingRight;  Vector3 localScale = transform.localScale;  localScale.x \*= -1;  transform.localScale = localScale;  }  // Untuk menggambar jarak penglihatan musuh di editor  private void OnDrawGizmosSelected()  {  Gizmos.color = Color.red;  Gizmos.DrawWireSphere(transform.position, lineOfSite);  }  } |

1. Kemudian tambahkan *script* *Enemy\_AI* pada *inspector* *enemy* pertama.



### 10.30 Menambahkan Script

1. Lalu jalankan *game* tersebut untuk melihat hasil dari *Enemy AI*.



### 10.31 Mencoba Enemy AI

1. Respawn
2. Untuk membuat *Respawn* pada *player* tambahkan *code* berikut pada *script Player*.

|  |
| --- |
| public int nyawa;  [SerializeField] Vector3 respawn\_loc;  public bool play\_again; |

1. Kemudian tambahkan perintah berikut pada fungsi Awake.

|  |
| --- |
| respawn\_loc = transform.position; |

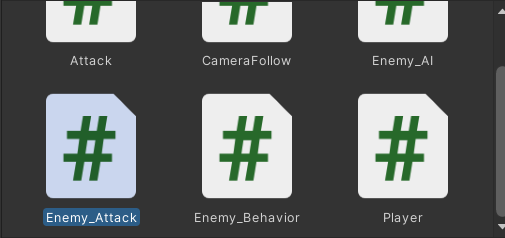
1. Kemudian buat fungsi baru dengan nama playagain.

|  |
| --- |
| void playagain()  {  if(play\_again == true)  {  nyawa = 3;  transform.position = respawn\_loc;  play\_again = false;  }  } |

1. Lalu pada void Update tambahkan kondisi berikut.

|  |
| --- |
| if(nyawa < 0)  {  playagain();  }  if(transform.position.y < -10)  {  play\_again = true;  playagain();  } |

1. Lalu pada *folder script* buat *script* baru dengan nama *Enemy\_Attacked*.

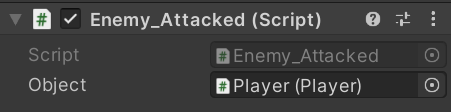


### 10.32 Membuat Script Baru

1. Buka file script tersebut kemudian berikan code berikut ini.

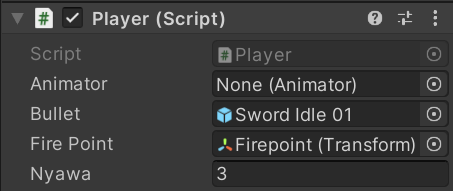
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_Attack : MonoBehaviour  {  [SerializeField] private Player Object;  void Start()  {  if (Object == null)  {  Object = GameObject.FindWithTag("Player").GetComponent<Player>();  }  }  void OnTriggerEnter2D(Collider2D other)  {  if (other.CompareTag("Player"))  {  Object.nyawa--;  if (Object.nyawa < 0)  {  Object.play\_again = true;  }  }  }  } |

1. Pada *inspector enemy* pertama dan *enemy* kedua tambahkan *script Enemy\_Attack* lalu rubah *object* menjadi *Player*.



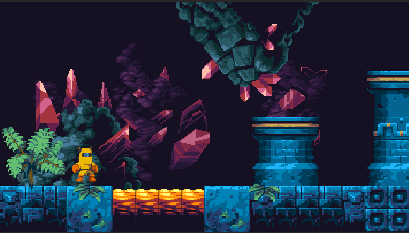
### 10.33 Menambahkan Component Script

1. Klik *Hierarchy Player* dan beri nilai pada nyawa pada *component script player*.



### 10.34 Memberi Nilai Nyawa

1. Lalu lakukan *test* pada *game* dengan menjalankan *game* tersebut.



### 10.35 Melakukan Test Pada Game

## Kuis

Code masih terdapat kesalahan

|  |
| --- |
| using UnityEngine;  public class PlayerAttack : MonoBehaviour  {      public int atackRange = 2.0f;      public int attacDamage = 10;      void Update()      {          if (InputGetButtonDown("Fire1"))          {              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          }      }  } |

Setelah diperbaiki

|  |
| --- |
| 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))  {  if (hit.collider.CompareTag("Enemy"))  {  EnemyHealth enemyHealth = hit.collider.GetComponent<EnemyHealth>();  if (enemyHealth != null)  {  enemyHealth.TakeDamage(attackDamage);  }  }  }  }  } |

Analisa

Pada *source code* PlayerAttack yang diimplementasikan memungkinkan pemain untuk melakukan serangan jarak dekat dengan menekan tombol "*Fire1*". Serangan ini dilakukan dengan menembakkan *ray* dari posisi pemain ke arah depan hingga jarak tertentu yang ditentukan oleh variabel *attackRange*. Jika ray tersebut mengenai objek dengan *tag "Enemy",* komponen *EnemyHealth* pada objek tersebut akan dicari. Jika ditemukan, *method* *TakeDamage* pada komponen tersebut akan dipanggil, mengurangi kesehatan musuh sesuai dengan jumlah kerusakan yang ditentukan oleh variabel *attackDamage*. Kode ini memastikan bahwa serangan hanya berlaku pada musuh yang berada dalam jangkauan serangan, memberikan mekanisme yang efisien untuk mengelola interaksi tempur dalam permainan.