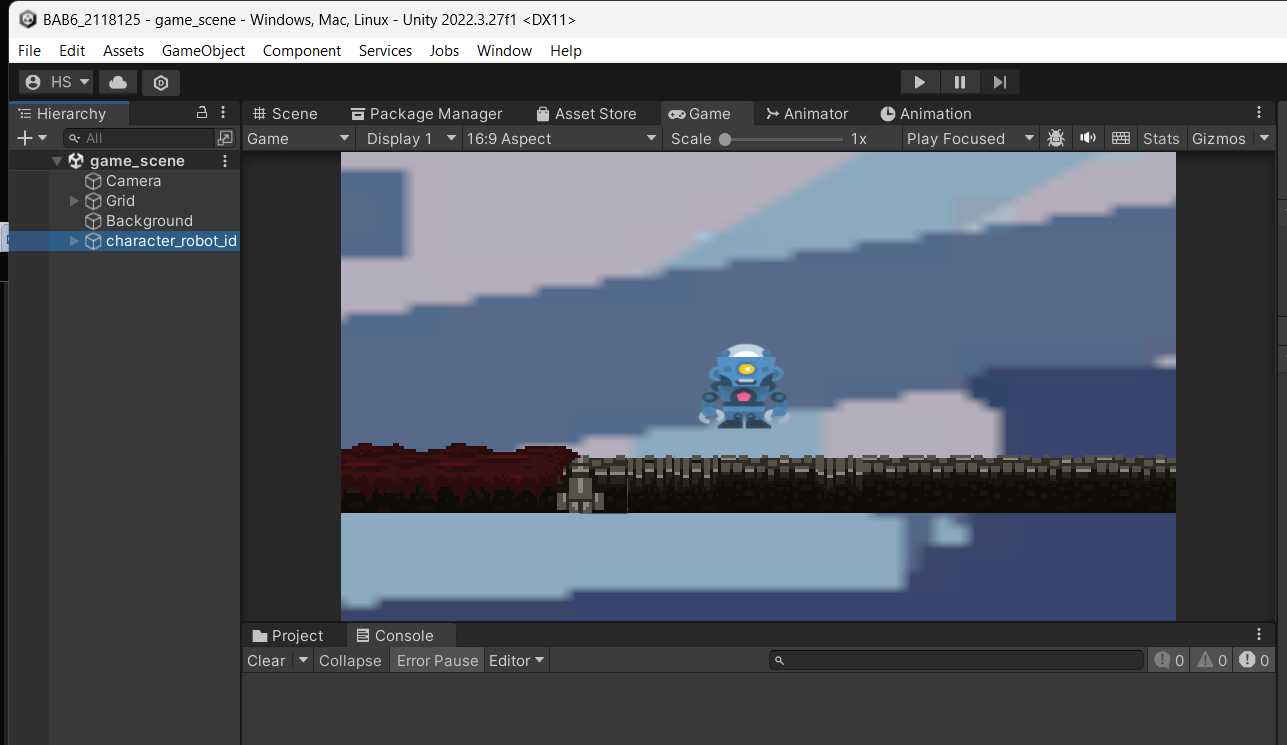
# 10 ENEMY AI AND ATTACK

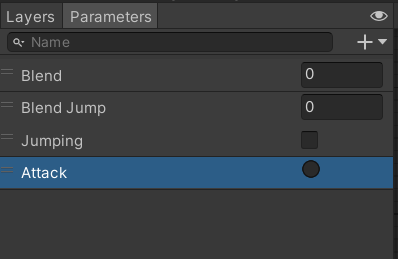
|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118125 |
| **Nama** | : | Hannes Tigor Hamonangan Sinaga |
| **Kelas** | : | D |
| **Asisten Lab** | : | WISANDO BERLIAN PANDENSOLANG (2218095) |

## 1.1 Tugas 1 : Membuat Enemy AI dan Attack

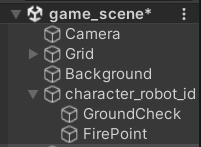
1. **Membuat Mekanisme Attack**
2. Bukalah Project unity sebelumnya



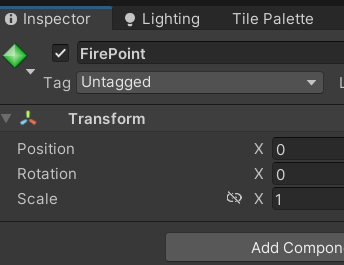
1. Tambahkan parameter baru seperti berikut ini.



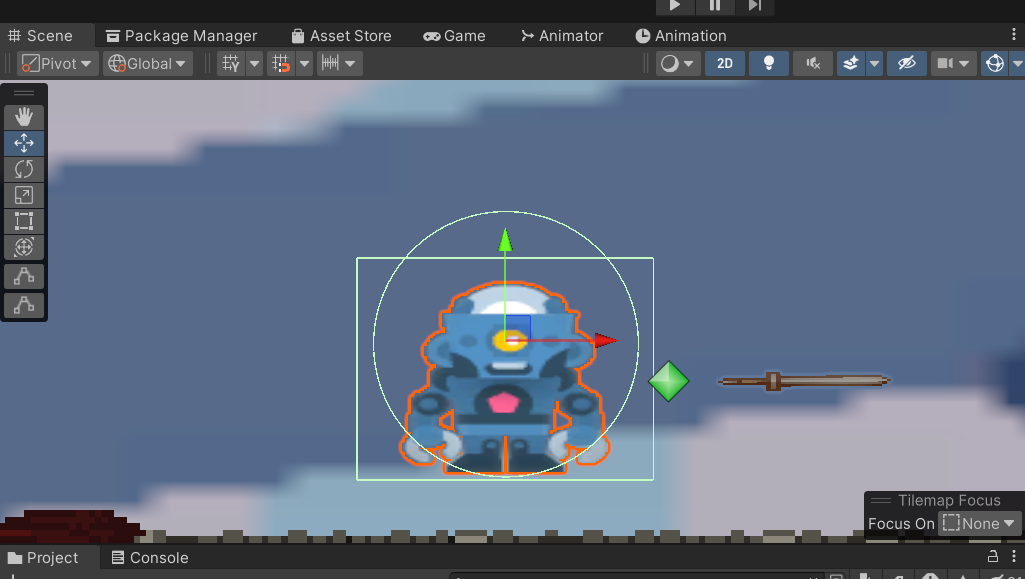
1. Buat objek baru bernama Firepoint di dalam objek player



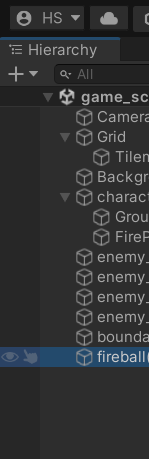
1. Masuk pada inspector dari Firepoint , lalu rubah warna dari icon berikut.



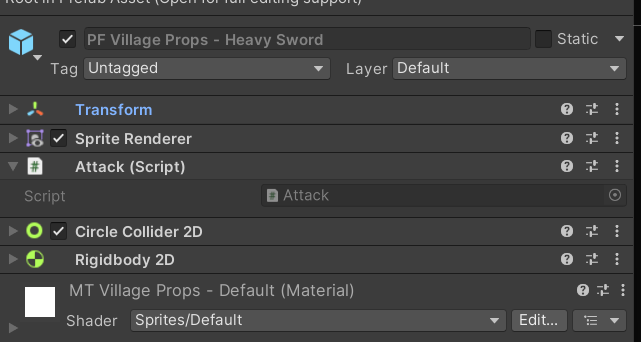
1. Atur posisi bullet seperti gambar berikut ini.



1. Drag and drop asset pedang lalu kita rename menjadi fireball



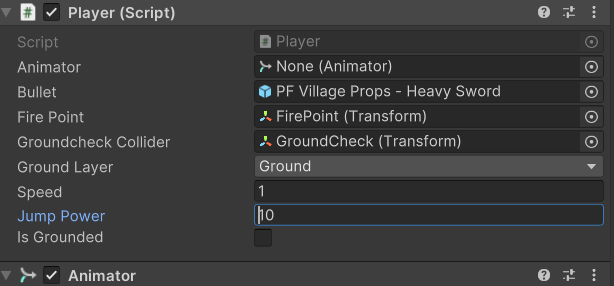
1. Pada objek fireball Add Component circle collider 2D dan rigidbody 2D..



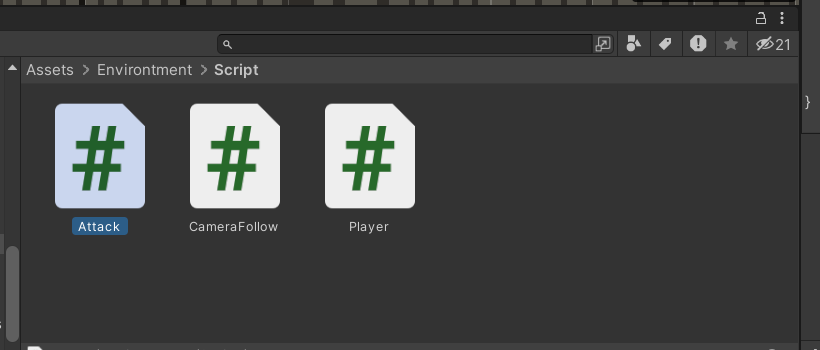
1. Tambahkan source code Pada script Player.cs



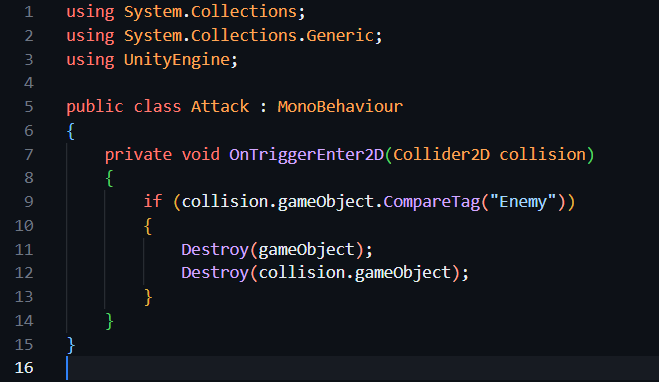
1. Atur inpector pada player bagian player Script seperti berikut ini.



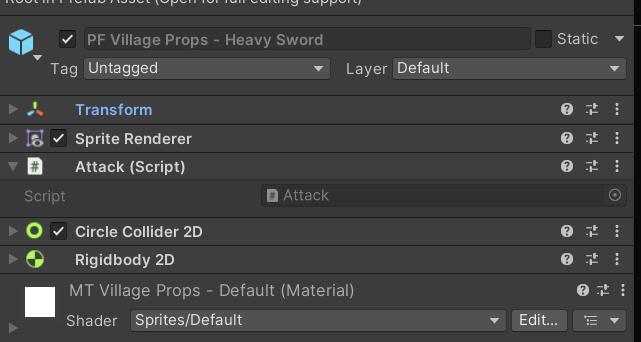
1. buat script baru dengan nama Attack



1. Tambahkan source code berikut ke dalam file Attack.cs



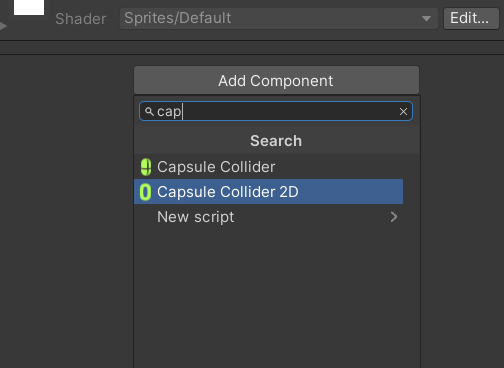
1. Drag & drop file script Attack ke objek fireball



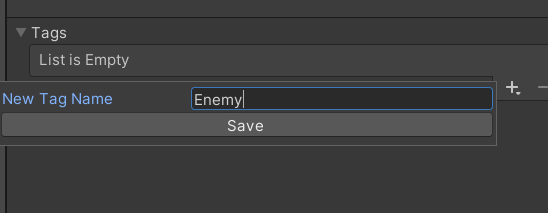
1. Tambahkan asset musuh 1 pada scene game



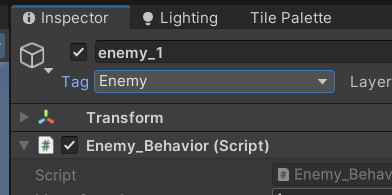
1. Pada objek musuh 1 Add Component capsule collider 2D



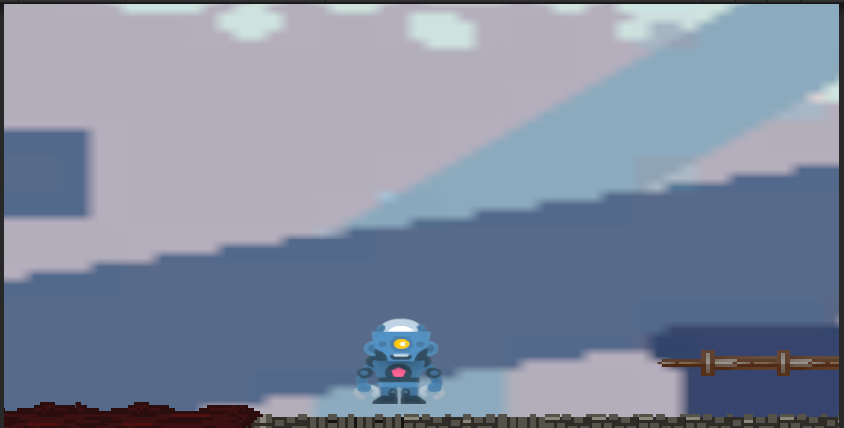
1. Tambahkan tag bernama enemy



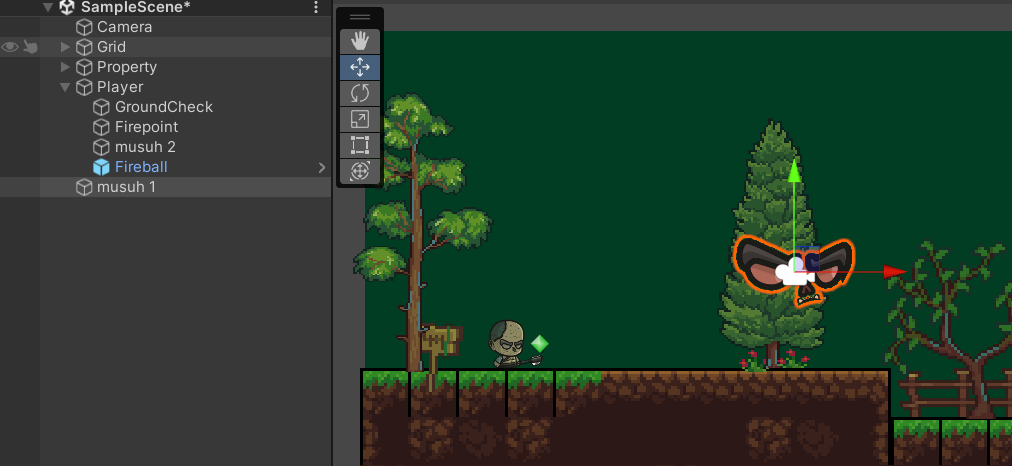
1. Pada objek musuh 1 pilih tag enemy



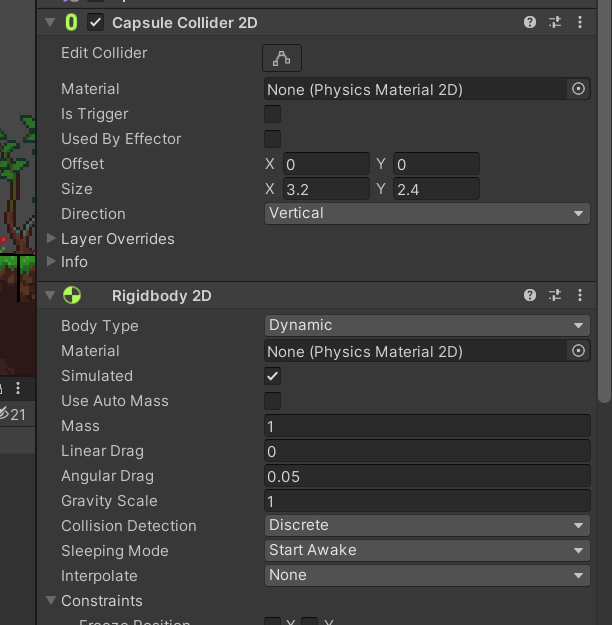
1. Berikut tampilan mekanisme attack



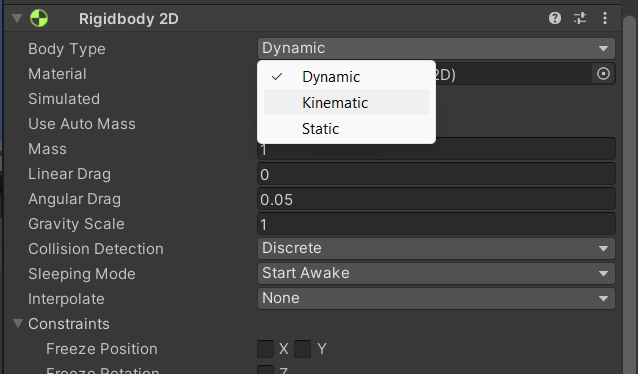
1. **Membuat Enemy Behavior NPC**
2. Tambahkan objek musuh 1 dengan cara drag and drop ke jendela hierarchy.



1. Pada objek musuh 1 tambahkan komponen Capsule Collider 2D dan RigidBody 2D



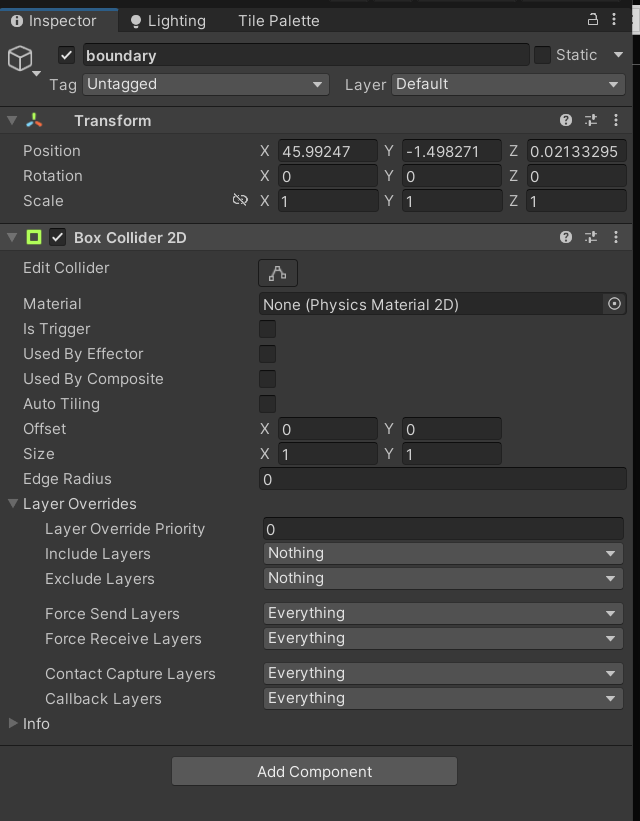
1. Pada komponen RigidBody 2D ubah tipe body menjadi Kinematic.

****

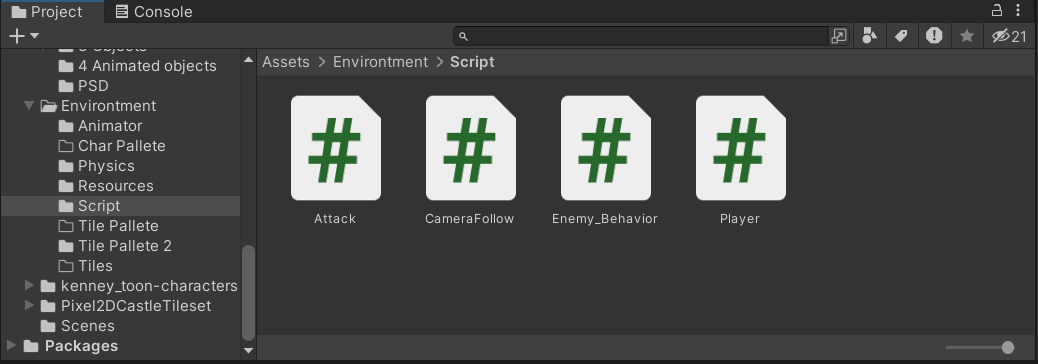
1. Tambahkan objek empty, lalu ubah nama menjadi Boundary

****

1. Pada objek baru tersebut, tambahkan komponen Box Collider 2D

****

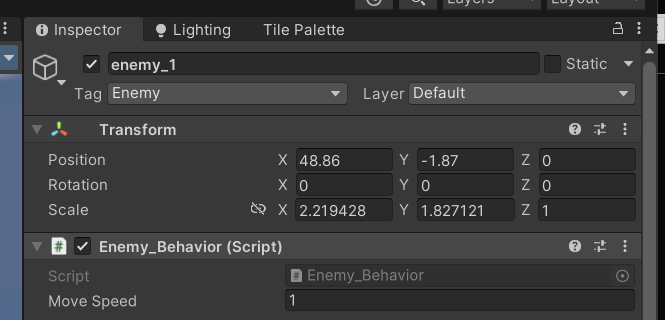
1. Buat file script baru bernama Enemy\_Behavior

****

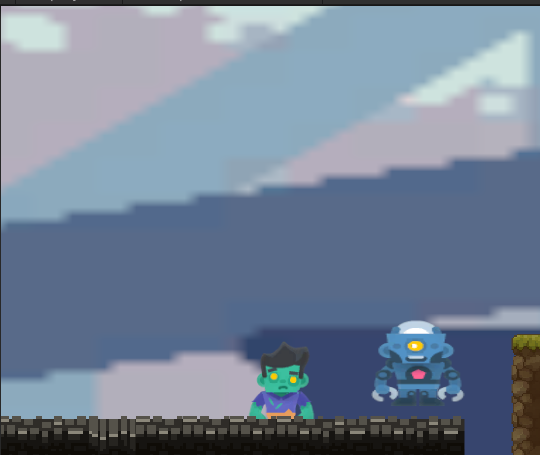
1. Tambahkan source code berikut pada file script Enemy\_Behavior.cs



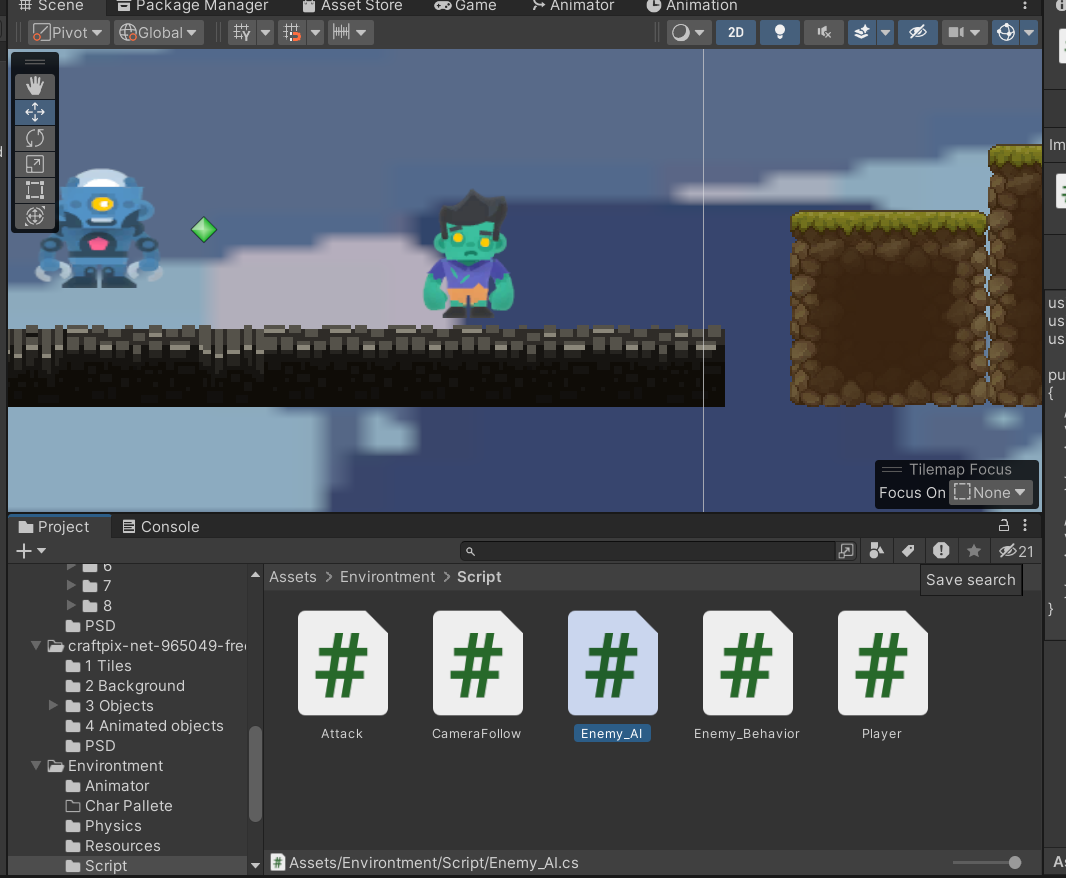
1. Drag and drop file script Enemy\_Behavior ke enemy\_1.

****

1. Jalankan Gamenya.

****

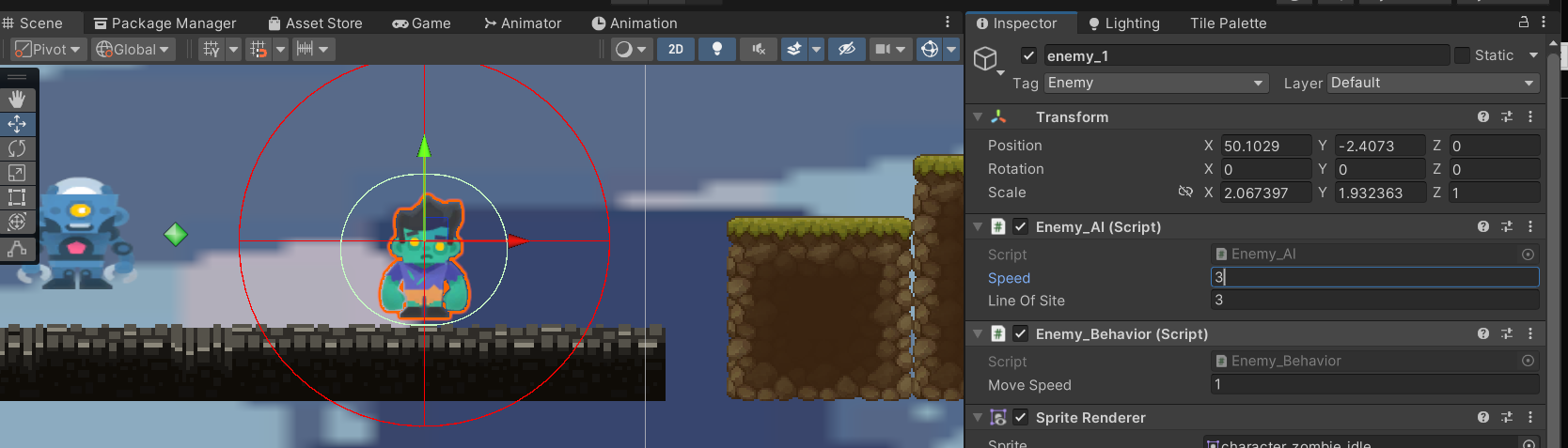
1. **Membuat Enemy AI**
2. tambahkan script baru,dengan nama Enemy\_AI



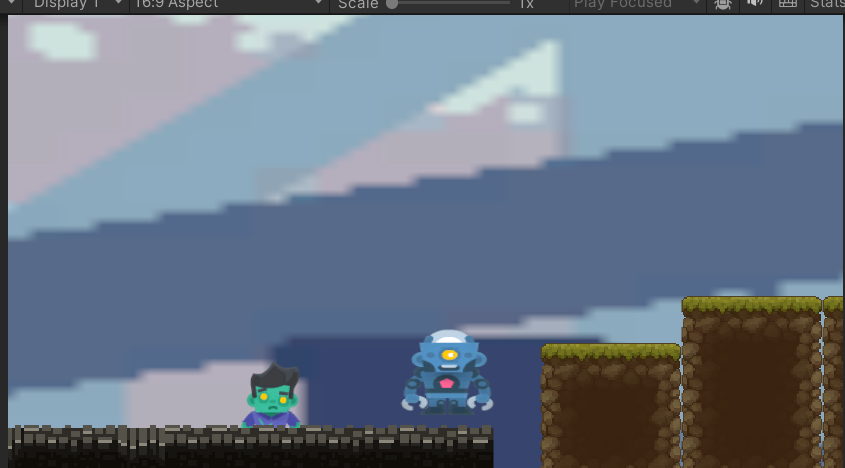
1. Masukkan source code brikut pada file script Enemy\_AI.cs

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_AI : MonoBehaviour  {  public float speed; // Kecepatan gerakan musuh  public float lineOfSite; // Jarak penglihatan musuh  private Transform player; // Transform dari pemain  private Vector2 initialPosition; // Posisi awal musuh  void Start(){  player = GameObject.FindGameObjectWithTag("Player").transform;  // Menyimpan posisi awal musuh  initialPosition = GetComponent<Transform>().position;  }  void Update(){  float distanceToPlayer = Vector2.Distance(player.position, transform.position);  if (distanceToPlayer < lineOfSite){  transform.position = Vector2.MoveTowards(this.transform.position, player.position, speed \* Time.deltaTime);  }  else{  transform.position = Vector2.MoveTowards(transform.position, initialPosition, speed \* Time.deltaTime);  }  }  private void OnDrawGizmosSelected()  {  Gizmos.color = Color.red;  Gizmos.DrawWireSphere(transform.position, lineOfSite);  }  } |

1. Drag & drop file script ke objek enemy\_1, setting line of site dan speed-nya.



1. Run Gamenya,maka enemy\_1 akan mengikuti pergerakan player.



1. **Respawn**
2. Tambahkan source code berikut pada file script Player.cs

|  |
| --- |
| public int nyawa = 3; // Inisialisasi nyawa dengan 3  [SerializeField] Vector3 respawn\_loc;  public bool play\_again; |

1. Tambahkan source code berikut pada void Awake()

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

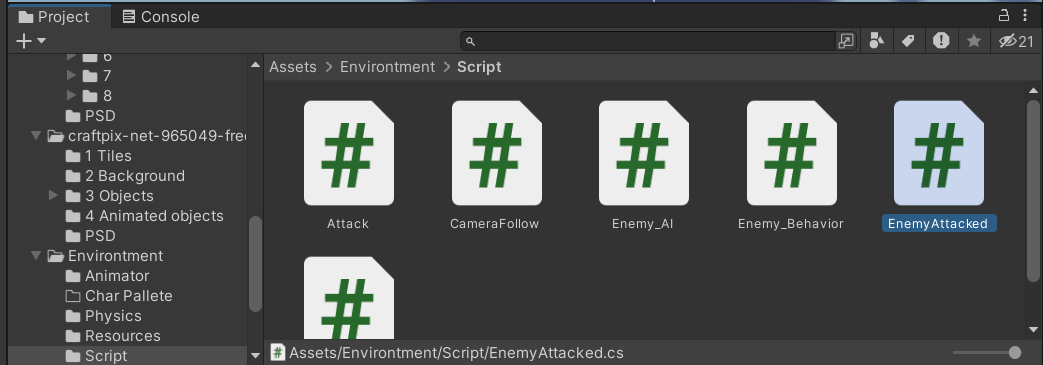
1. Buat void playagain() dan letakkan di bawah void Awake()

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

1. Tambahkan source code berikut pada void Update()

|  |
| --- |
| if (nyawa <= 0)  {play\_again = true;  playagain();  } |

1. Buat file script baru bernama EnemyAttacked

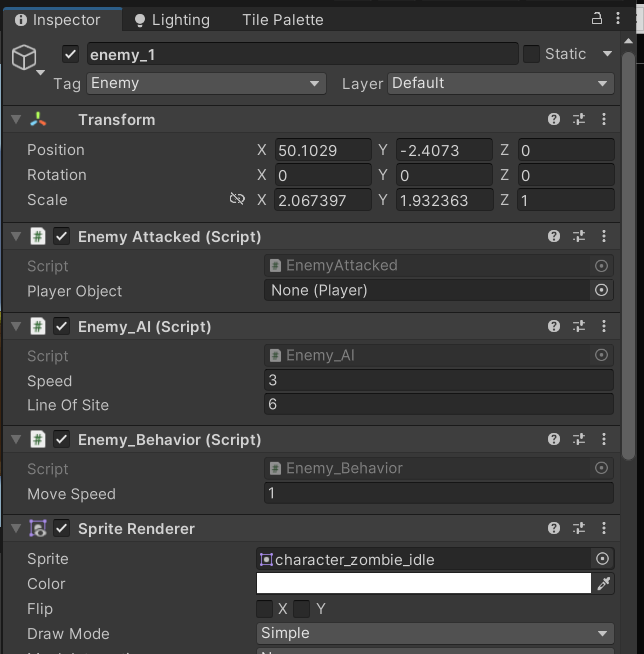


### 10.30 Script EnemyAttacked

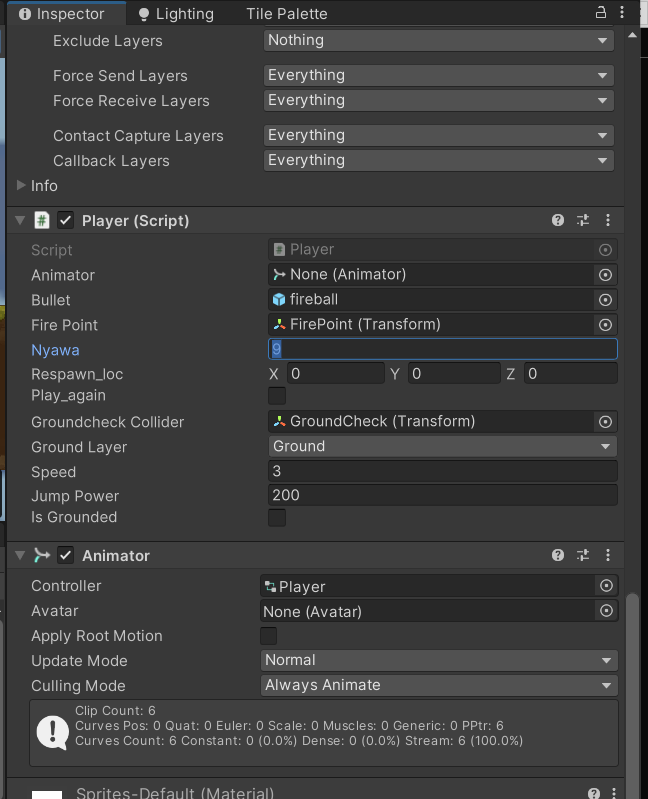
1. Tambahkan source code berikut pada script EnemyAttacked.cs

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class EnemyAttacked : MonoBehaviour  {  [SerializeField] private Player playerObject; // Mengubah nama variabel untuk lebih deskriptif  void Start()  {  if (playerObject == null)  {  playerObject = GameObject.FindWithTag("Player").GetComponent<Player>();  }  }  void OnTriggerEnter2D(Collider2D other)  {  if (other.CompareTag("Player"))  {  playerObject.nyawa--;  if (playerObject.nyawa <= 0)  {  playerObject.Respawn(); // Call the Respawn method instead of destroying the player  }  }  }  } |

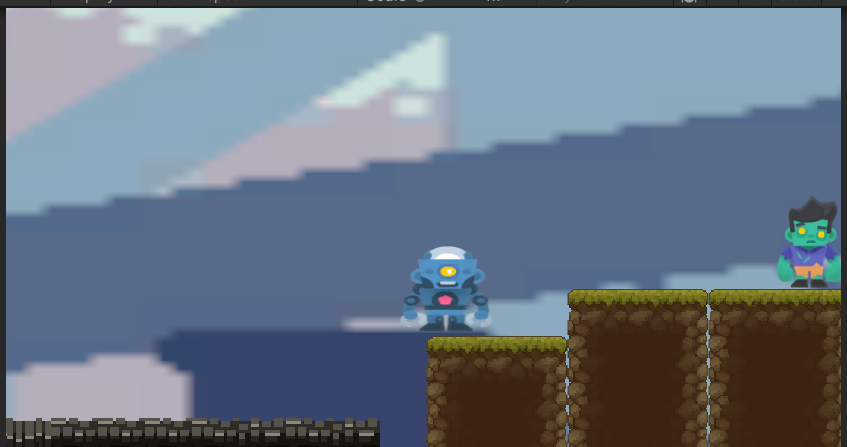
1. Drag & drop pada objek enemy\_1 dan ubah objek ke player



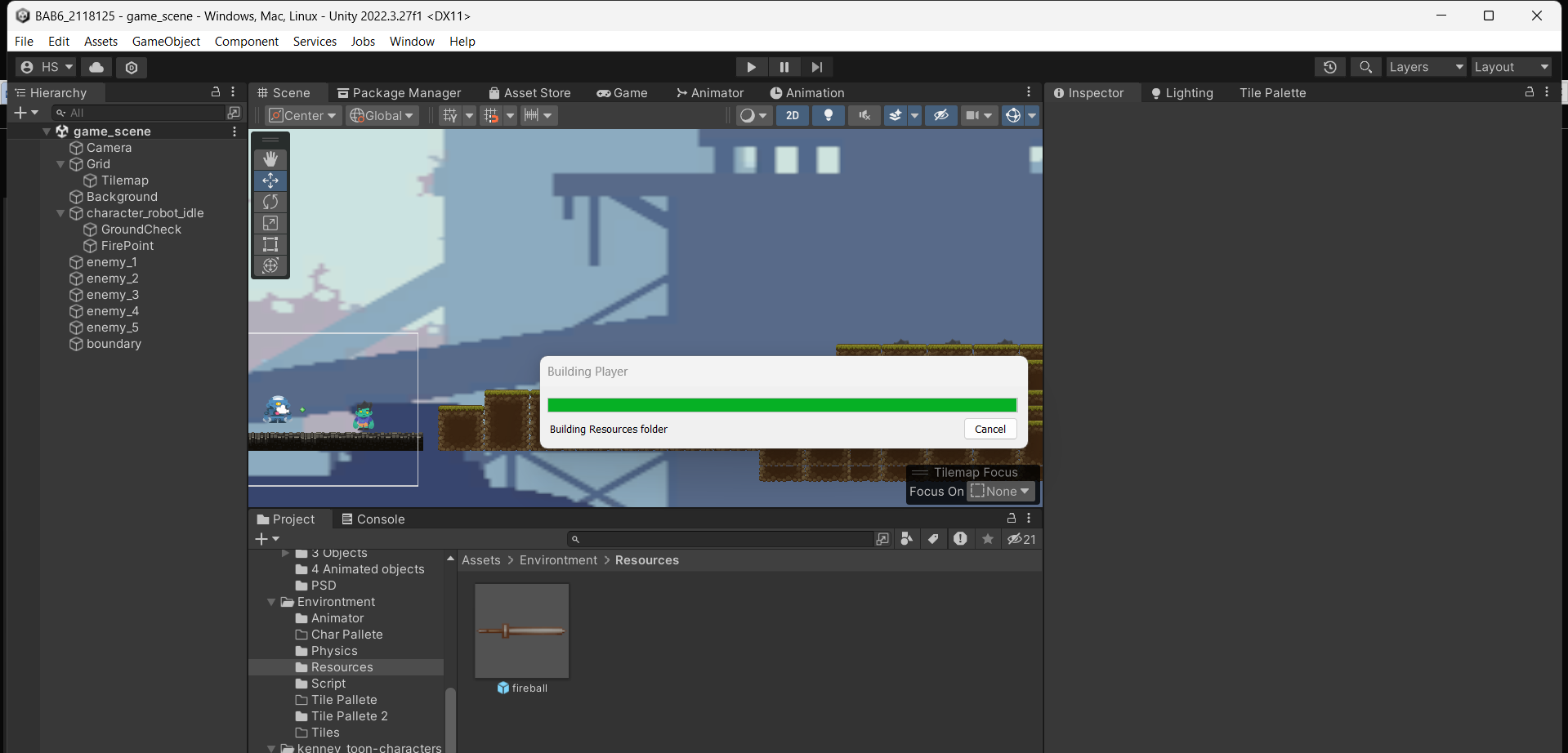
1. Pada Player ubah nyawa menjadi 3



1. Run Gamenya maka akan seperti berikut ini.



1. Proses berikutnya yaitu kita rendering , masuk pada file lalu bulid and run , lalu setting seperti berikut ini.



1. **Melengkapi Script**

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

Penjelasan :

Source code 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 ketik 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 health musuh dan menangani kematian mereka.