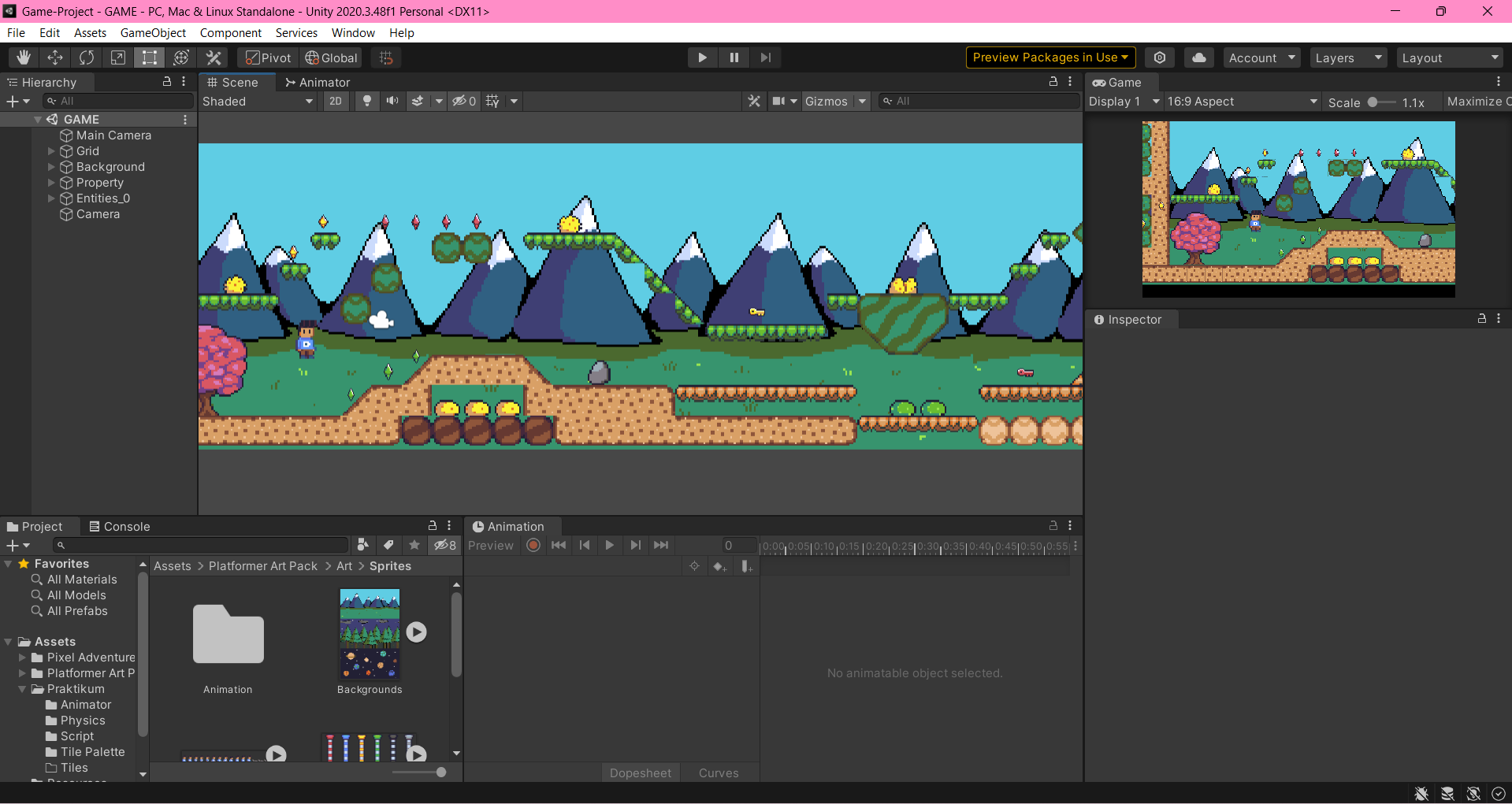
# 10 Respawn and AI Enemy Attack

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118060 |
| **Nama** | : | CORNELIA LUBA TARA BORO |
| **Kelas** | : | B |
| **Asisten Lab** | : | BAGAS ANARDI SURYA WIJAYA (2118004) |

## Tugas 9 : Membuat Respawn and AI Enemy Attack

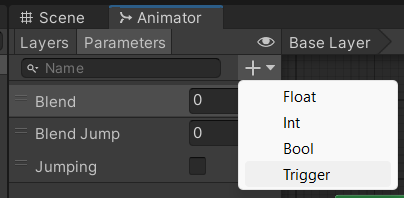
Membuat *Respawn and AI Enemy Attack*

1. **Membuat Respawn and AI Enemy Attack**
2. Buka *project* yang sudah dikerjakan pada bab sebelumnya yang telah terimport *asset*.



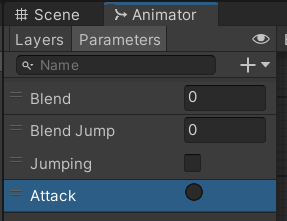
* 1. Buka Project

1. Buka Menu *Animator*, lalu tambahkan parameter baru dengan tipe data Trigger.



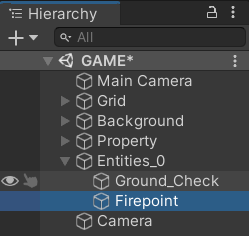
* 1. Menambahkan Parameter

1. Selanjutnya ganti nama parameters menjadi attack.



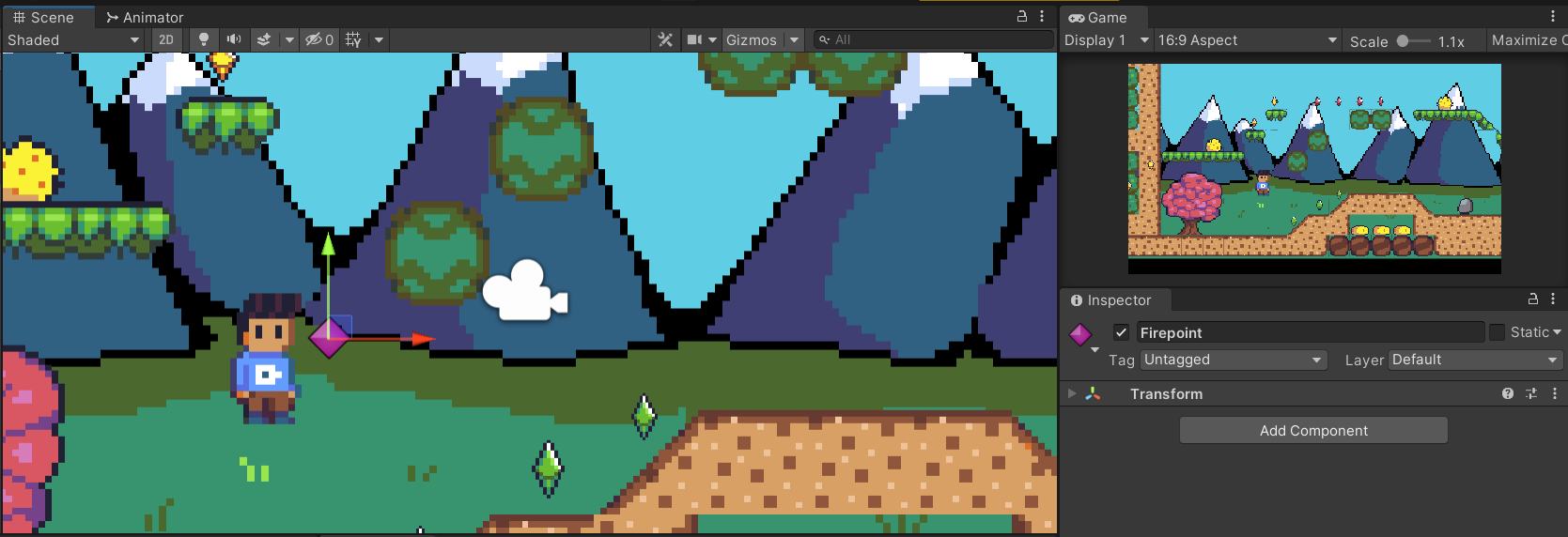
* 1. Menambahkan Parameter

1. Selanjutnya tambahkan create empthy pada fie Entities\_0 lalu ganti nama menjadi Firepoint



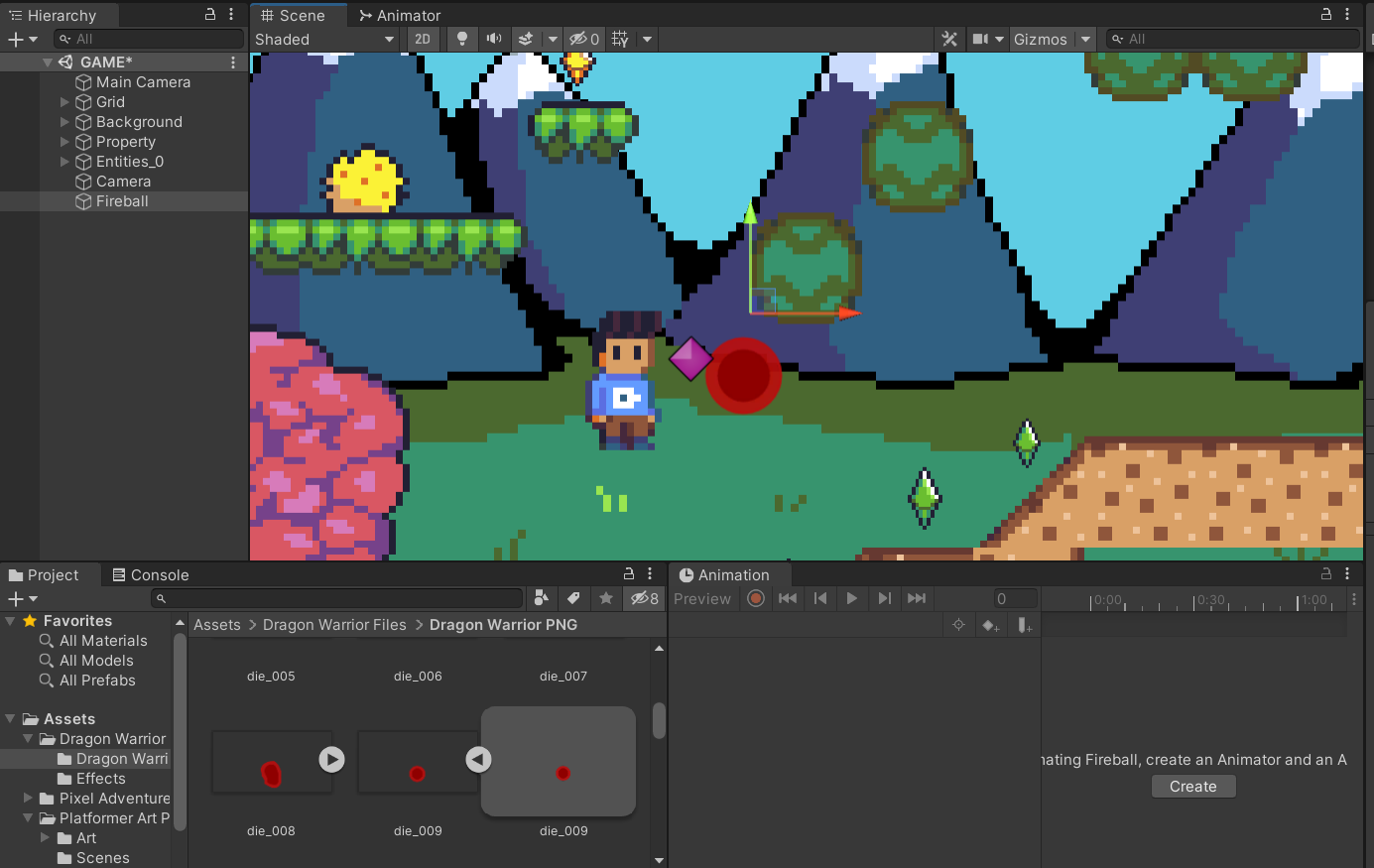
* 1. Menambahkan *File*

1. Lalu ubah icon firepoint seperti pada gambar berikut ini.



* 1. Mengubah Inspector Firepoint

1. Selanjutnya, tambahkan icon die dari asset ke dalam hierarchy untuk menjadi icon penyerangan player. Kemudian ganti nama icon tersebut menjadi fireball.



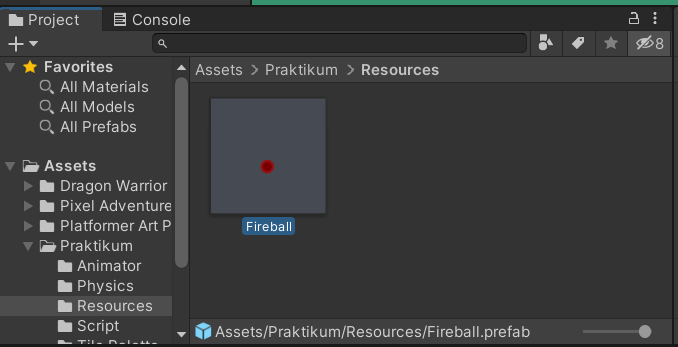
* 1. Mengubah *Icon*

1. Selanjutnya, pada icon fireball tambahkan komponen Corcel Collider 2D dan Rigidbody 2D dan ubah aturan inspector seperti gambar dibawah ini.



* 1. Menambahkan komponen

1. Selanjutya buat folder baru di dalam folder praktikum dengan nama Resources, kemudian pindahkan fireball ke dalam folder tersebut, lalu hapus file fireball pada hierarchy.

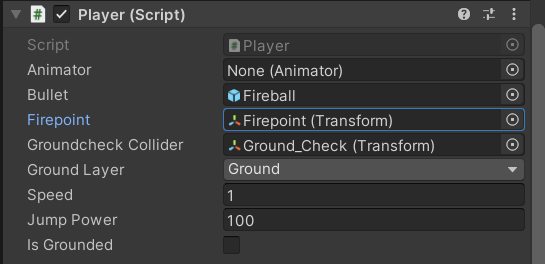


* 1. Membuat Folder Baru

1. Selanjutnya, tambahkan ubah source code Player.cs seperti dibawah ini.

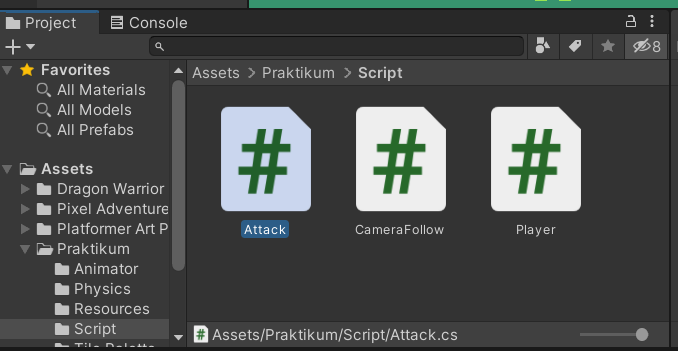
|  |
| --- |
| public Animator animator;  public GameObject bullet;  public Transform Firepoint; |
| #Tambahkan dibawah fungsi fixedUpdate  IEnumerator Attack()  {    animator.SetTrigger("Attack");  yield return new WaitForSeconds(0.25f);    float direction = 1f;    GameObject fireball = Instantiate(bullet, firePoint.position, Quaternion.identity);  fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 10f, 0);    Destroy(fireball, 2f);  } |
| #Tambakan pada Function Void Update  if (Input.GetKeyDown(KeyCode.C)){  StartCoroutine(Attack());} |

1. Selanjutnya buka file Entities\_0 lalu ubah firepoint dan bullet pada component Player (Script) seperti dibawah ini.



* 1. Mengubah Inspector

1. Tambahkan Script baru dan beri nama Attack.

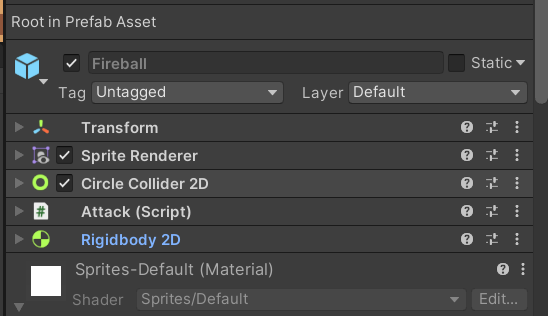


* 1. Menambahkan Script

1. Selanjutnya tambahkan script seperti dibawah ini.

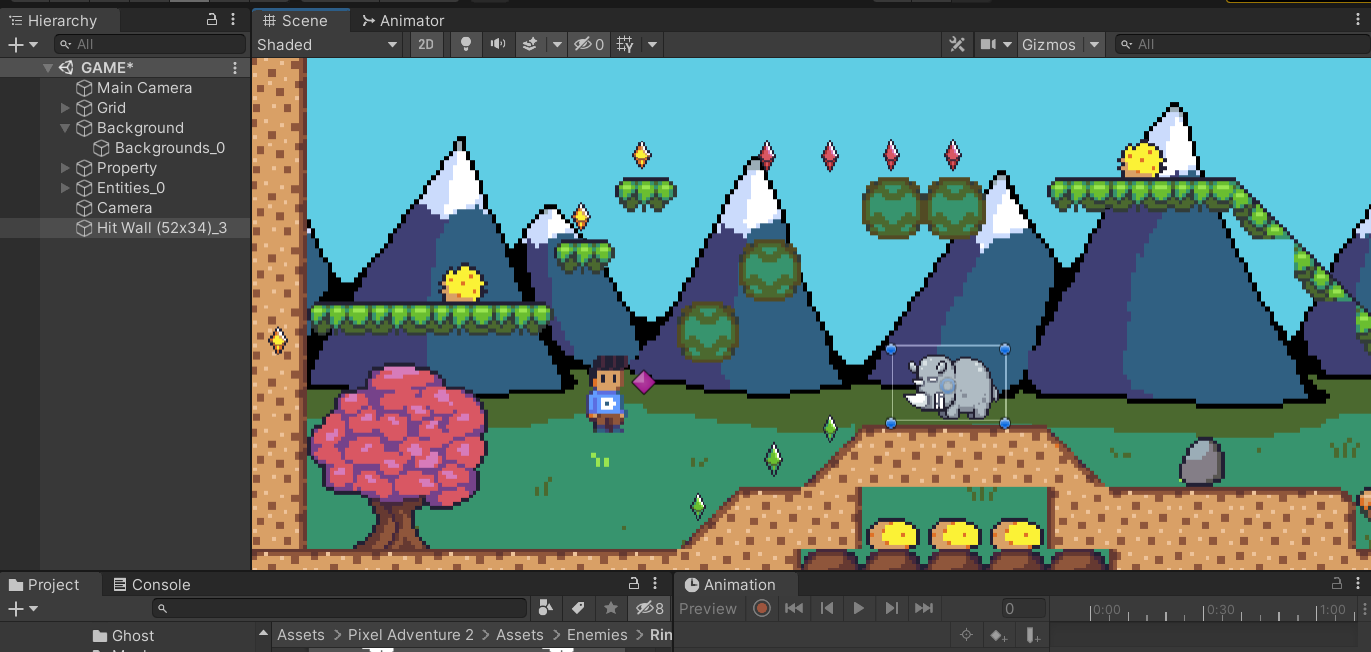
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Attack : MonoBehaviour  {  private void OnTriggerEnter2D(Collider2D collision)  {  if (collision.gameObject.CompareTag("Enemy"))  {  Destroy(gameObject);  Destroy(collision.gameObject);  }  }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

1. Didalam folder resource tambahkan Script Attack di Prefab fireball, dengan cara Klik fireball kemudian pada menu Inspector arahkan Script Attack kedalam Inspector.



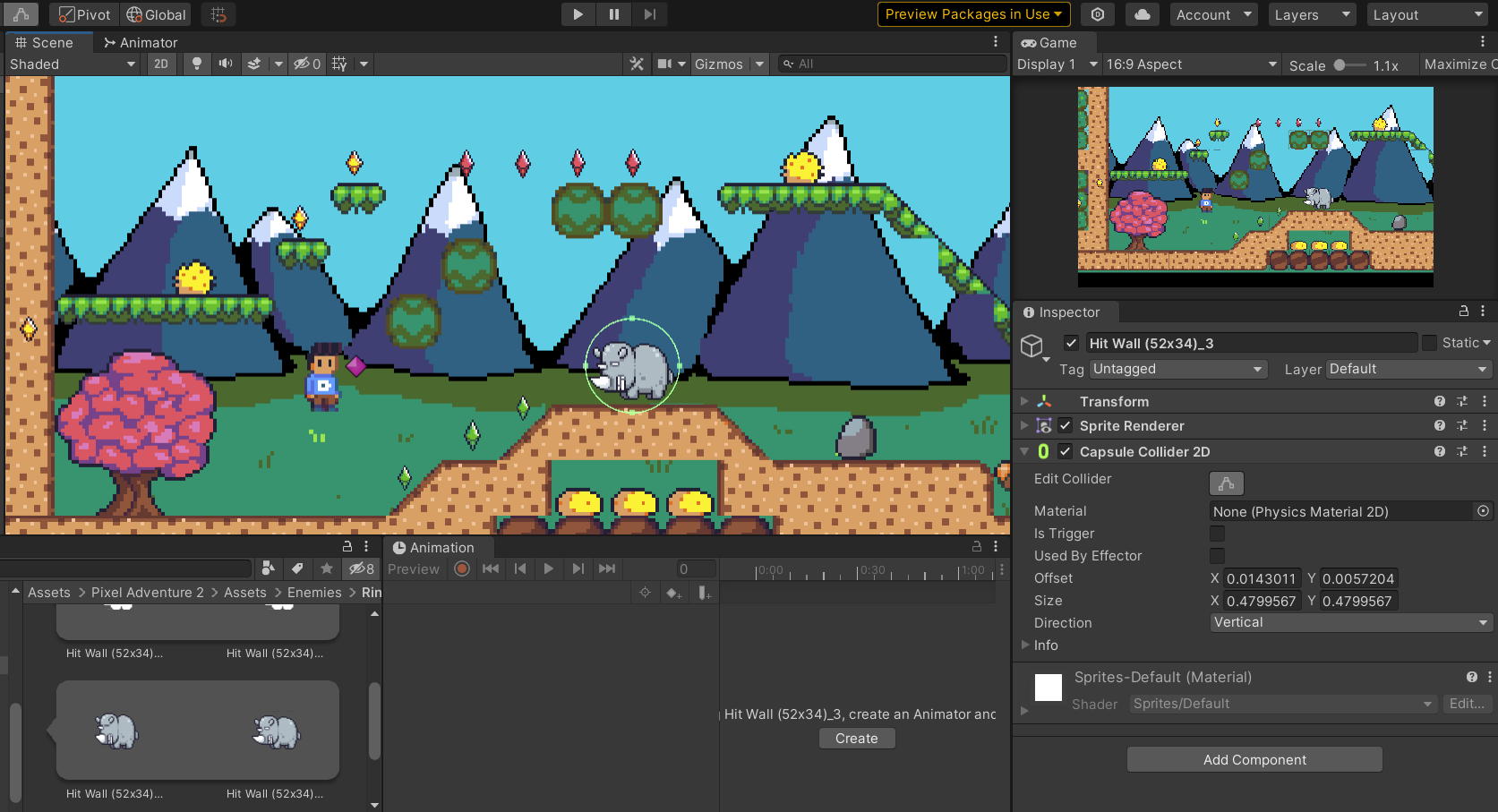
* 1. Menambahkan Script

1. Selanjutnya tambahkan enemy.



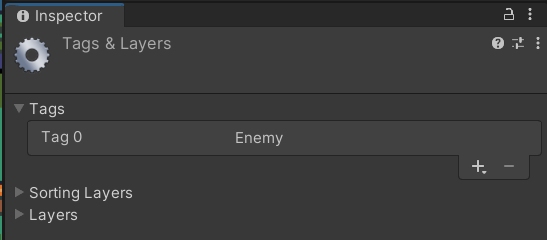
* 1. Menambahkan *Enemy*

1. Kemudian tambahkan komponen seperti gambar dibawah ini.



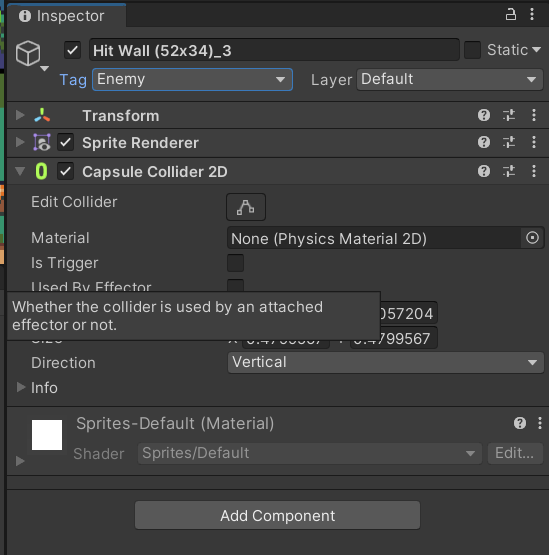
* 1. Menambahkan *Component*

1. Pada inspector Rino, tambahkan tag baru dan beri nama tag tersebut dengan nama enemy.



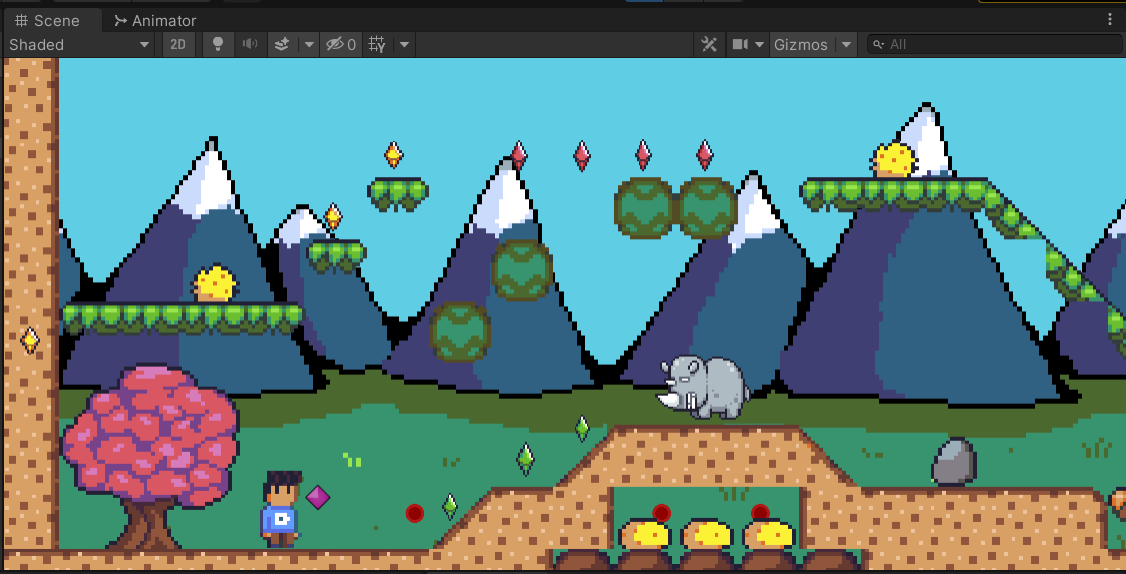
* 1. Menambahkan *Tag*

1. Kemudian ganti tag menjadi enemy.



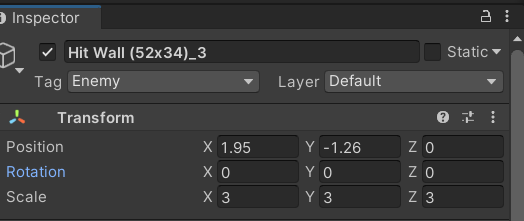
* 1. Mengganti Tag

1. Selanjutnya, ketika game dijalankan player sudah bisa melakukan penyerangan dan jika berhasil mengenai enemy, maka enemy tersebut akan hilang.



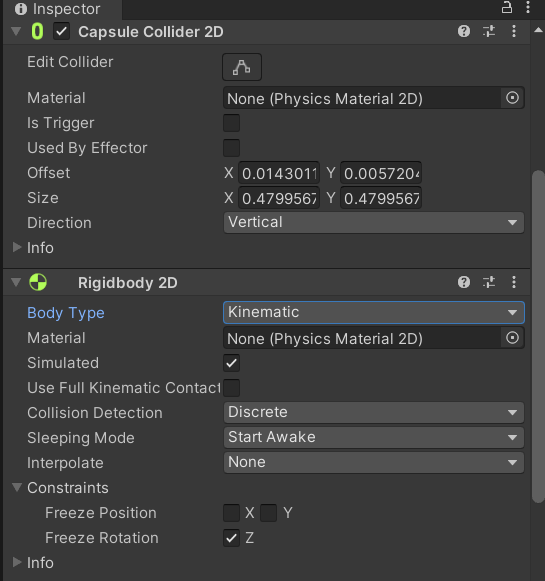
* 1. Hasil Play Game

1. Atur Inspector Rino menjadi seperti dibawah ini.



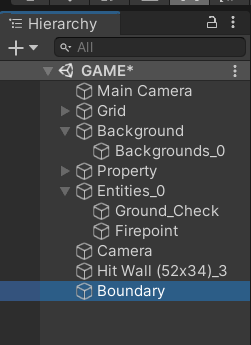
* 1. Mengubah Pengaturan Inspector

1. Selanjutnya, tambahkan komponen seperti dibawah ini dan ubah aturan menjadi seperti digambar.



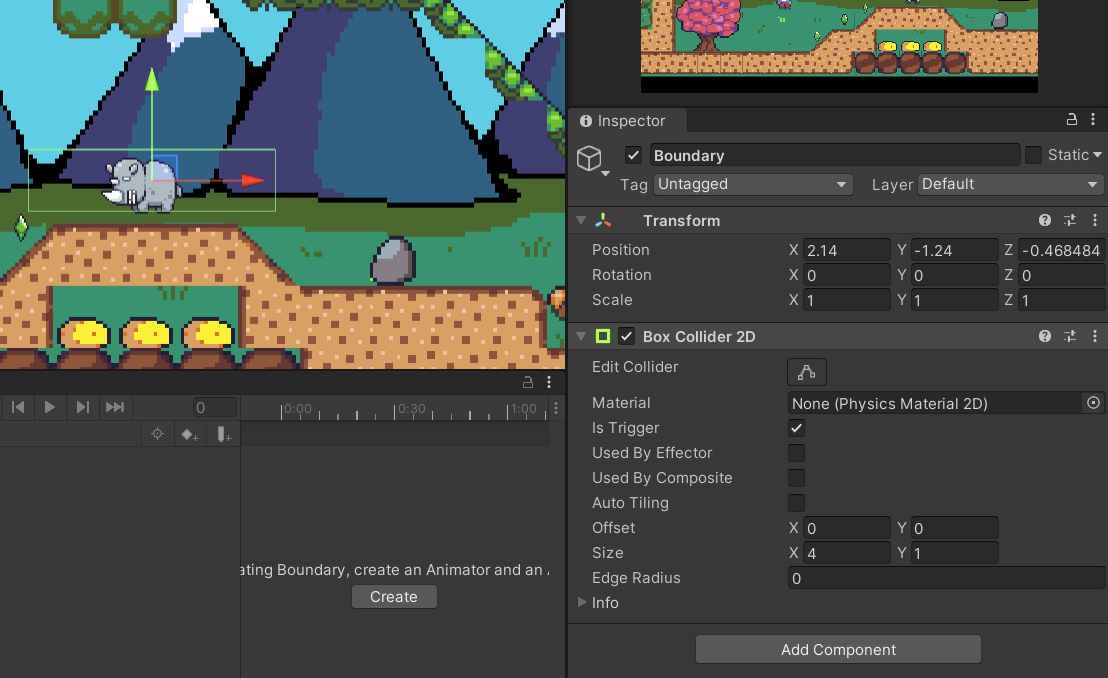
* 1. Mengubah Pengaturan Inspector

1. Kemudian tambahkan layer emphty baru dan beri nama boundary.



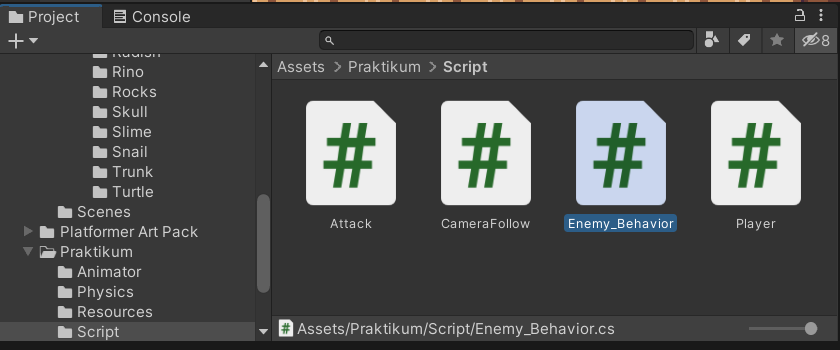
* 1. Menambahkan Layer

1. Lalu tambahkan component box collider pada layer boundary.



* 1. Menambahkan Component

1. Selanjutnya tambahkan script baru dan beri nama Enemy\_behavior. Kemudian, drag and drop ke layer enemy

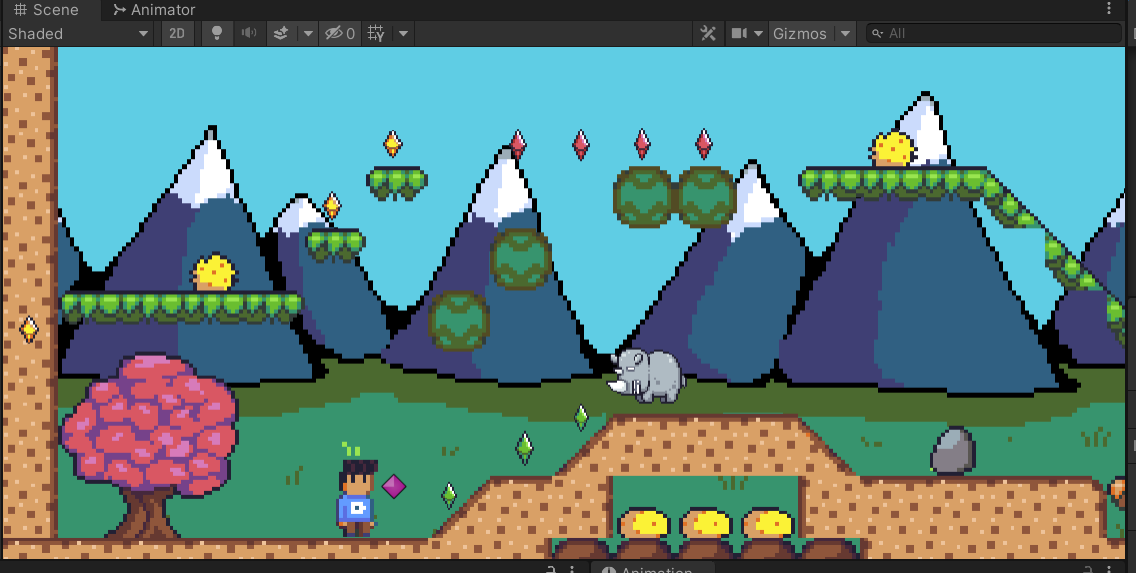


* 1. Menambahkan Script

1. Tambahkan *source code* berikut ini.

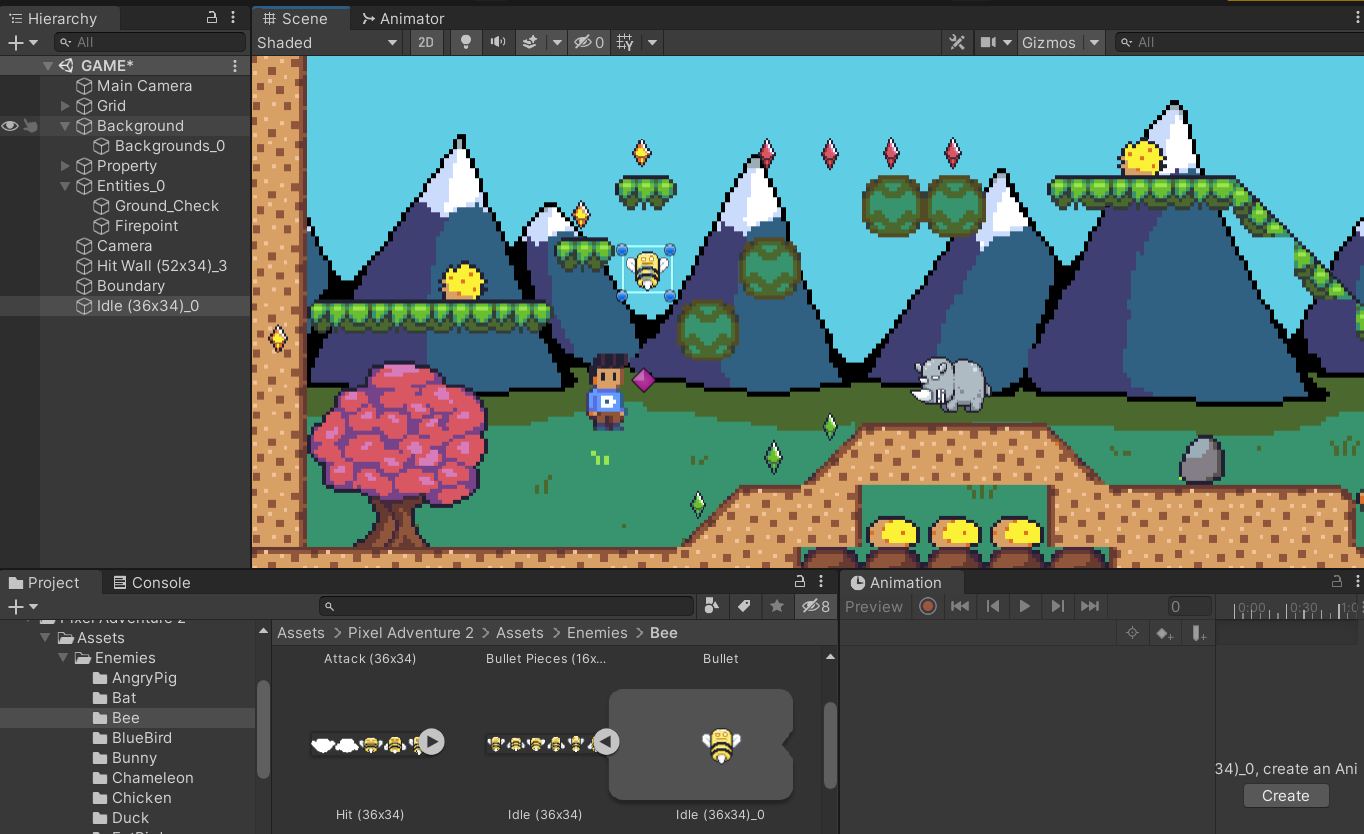
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_Behavior : MonoBehaviour  {  [SerializeField] float moveSpeed = 1f;  Rigidbody2D rb;  void Start()  {  rb = GetComponent<Rigidbody2D>();  }  void Update()  {  if (isFacingRight())  {  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. Maka, jika project dijalankan saat ini enemy akan bergerak ke kiri dan kanan didalam box atau area yang sudah ditentukan sebelumnya.



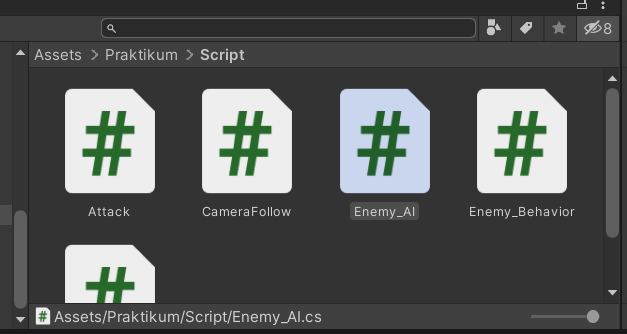
* 1. Hasil Play Game

1. Selanjutna, tambahkan enemy baru seperti gambar dibawah ini.



* 1. Menambahkan Enemy Bee

1. Buat Script baru dan beri nama Enemy\_AI.

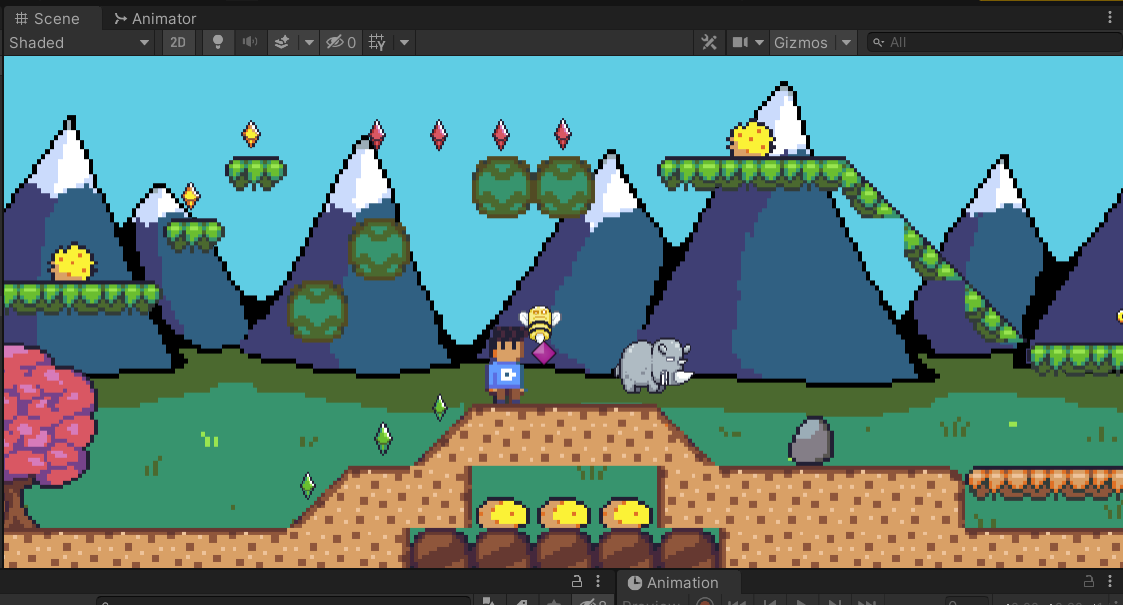


* 1. Menambahkan script

1. Selanjutnya tambahkan code seperti dibawah ini.

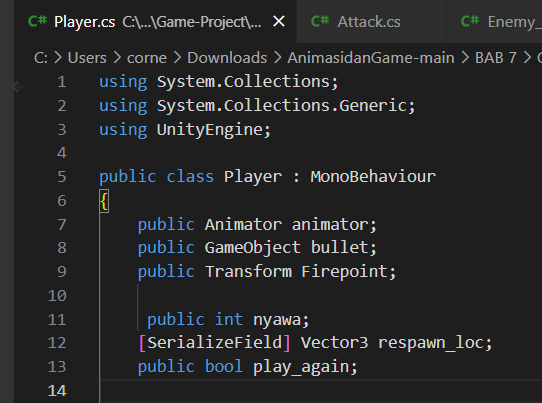
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_AI : MonoBehaviour  {  public float speed; // Kecepatan gerakan musuh  public float lineOfSite; // Jarak penglihatan musuh  private Transform player; // Transform dari pemain  private 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);  }  } |

1. Selanjutnya ketika game di jalankan maka enemy bee akan mengikuti player ketika telah berada di jangkauan tertentu.



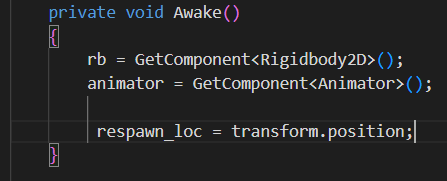
* 1. Hasil Play Game

1. Selanjutnya tambahkan source code seperti dibawah ini.



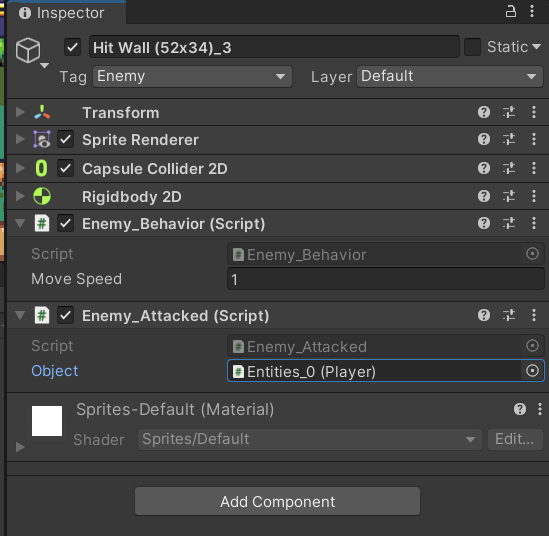
* 1. Menambahkan *Source Code*

1. Lalu tambahkan juga seperti dibawah ini.



* 1. Menambahkan *Source Code*

1. Ubah Object pada script enemy Rino.



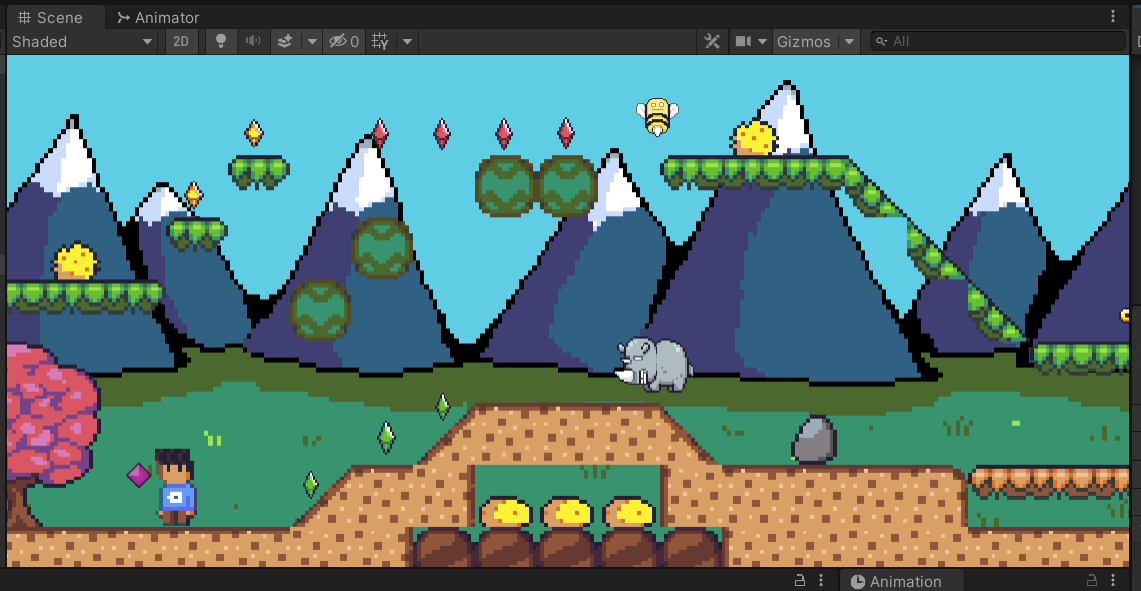
* 1. Mengatur Enemy\_Attacked (Script)

1. Ubah pengaturan inspector pada layer Entities\_0 atur nyawa menjadi 2.



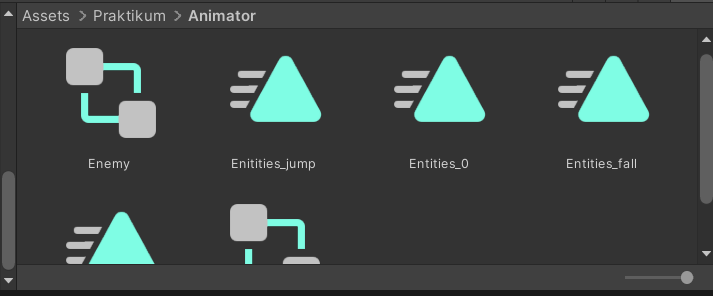
* 1. Mengatur Nilai Nyawa

1. Sehingga ketika player mengenai enemy sebanyak 2 kali, akan game over dan player akan kembali ke posisi awal.



* 1. Hasil Play Game

1. Langkah selanjutnya adalah menambahkan animasi pada enemy.



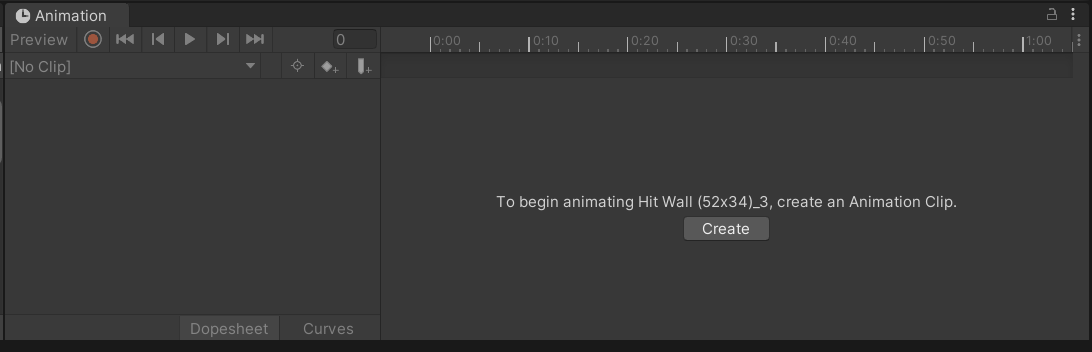
* 1. Membuat Animasi

1. Tambah animator pada layer enemy.



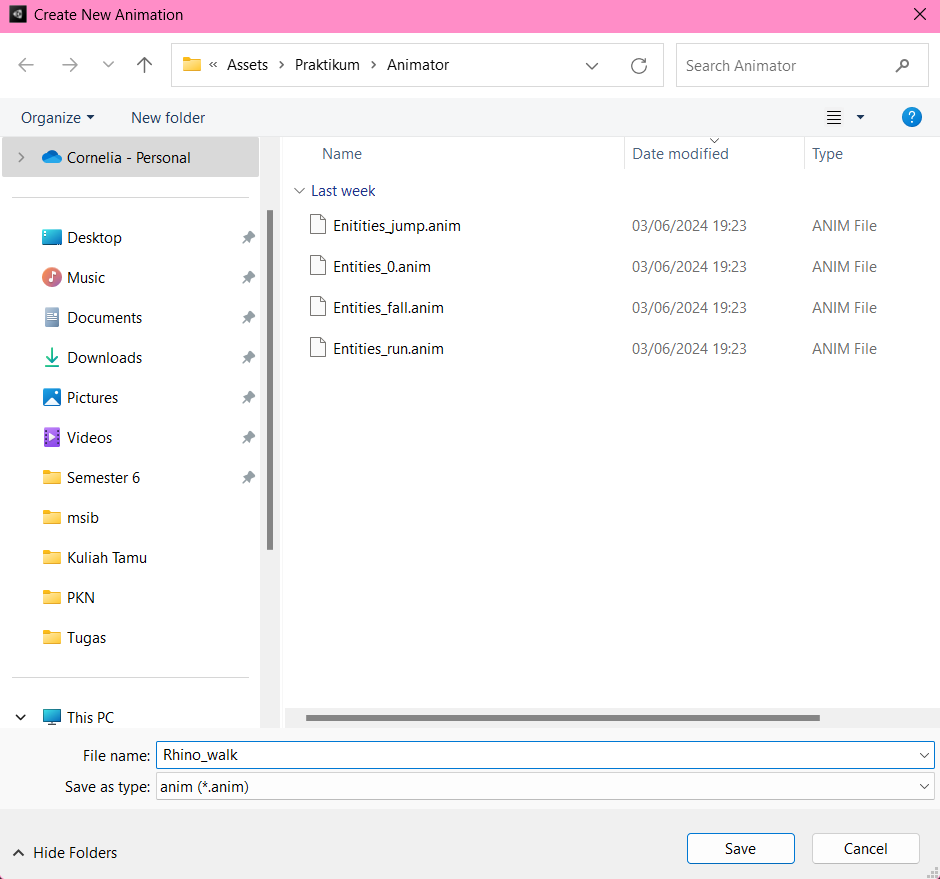
* 1. Membuat Animasi

1. Create Animasi baru.



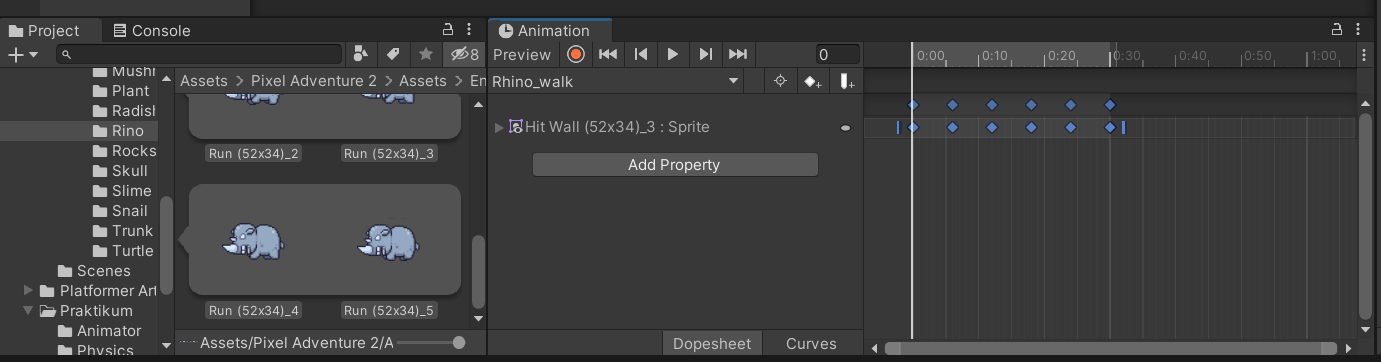
* 1. Membuat Animasi

1. Beri nama Rhino-walk.



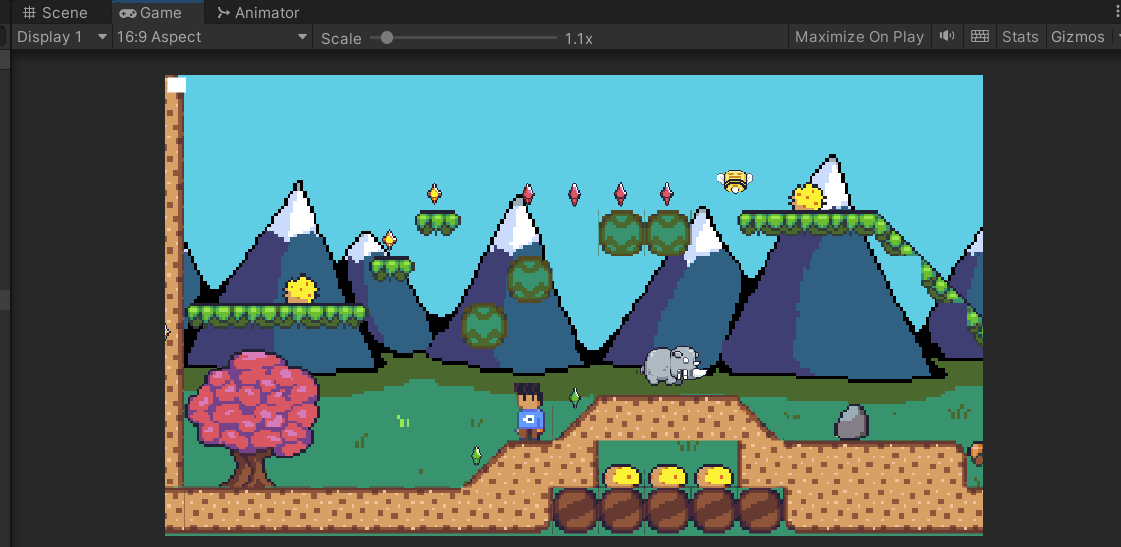
* 1. Membuat Animasi

1. Lalu tambahkan enemy rino dari asset untuk animasi berjalan .



* 1. Membuat Animasi

1. Lakukan hal yang sama untuk enemy Bee.



* 1. Membuat Animasi

1. **KUIS**
2. *Source Code*

|  |
| --- |
| using UnityEngine;  public class PlayerAttack : MonoBehaviour  {  public float attackRange = 2.0f; // Jarak serang  public int attackDamage = 10; // Kerusakan serangan  void Update()  {  if (Input.GetButtonDown("Fire1"))  {  PerformMeleeAttack();  }  }  void PerformMeleeAttack()  {  RaycastHit hit;  if (Physics.Raycast(transform.position, transform.forward, out hit, attackRange))  {  // Cek apakah objek yang terkena adalah musuh  Enemy enemy = hit.transform.GetComponent<Enemy>();  if (enemy != null)  {  enemy.TakeDamage(attackDamage);  }  }  }  } |

Penjelasan

Source code diatas adalah untuk mengatur kemampuan serangan player. Langkah pertama adalah mendefinisika variabel, yaitu variabel attackrange untuk mengatur jarak serang dan attackdamage untuk menentukan jumlah kerusakan. Fungsi update digunakan untuk mengecek apakah ada inputan oleh user atau tidak, jika iya makan akan memanggil fungsi PerformMEleeAttack. Fungsi PerformMeleeAttack untuk proses penyerangan player ke enemy, sehingga setiap kali player menekan tombol “Fire1”, game akan melakukan pengecekan untuk melihat apakah ada musuh di depan pemain dalam jarak yang bisa diserang. Jika ada musuh di depan dalam jarak serangan, game akan memberikan damage kepada musuh tersebut, mengurangi health mereka sesuai dengan kerusakan yang ditetapkan.

1. **Link Github Pengumpulan**

[GitHub - CorneliaBoro/AnimasidanGame](https://github.com/CorneliaBoro/AnimasidanGame)