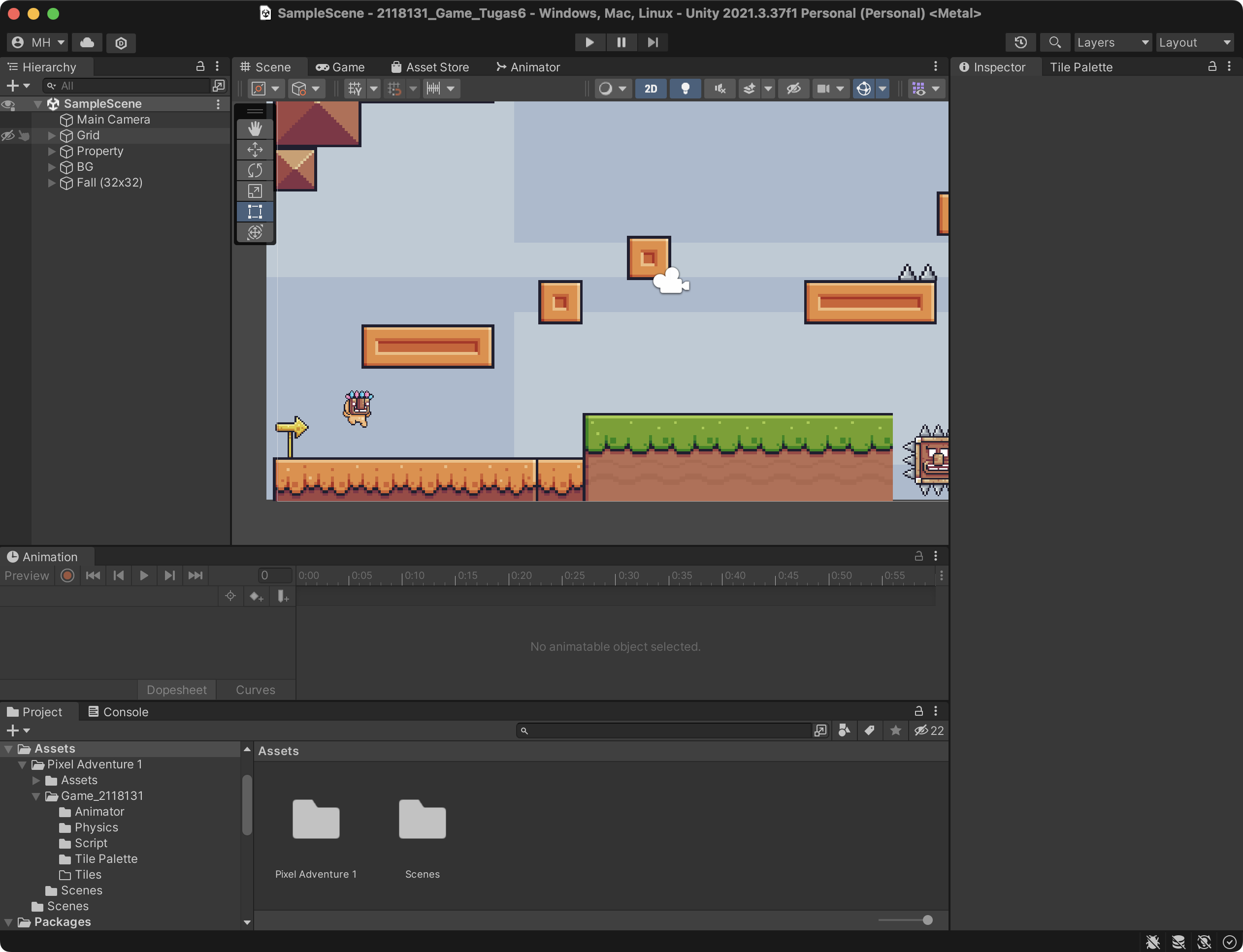
# 10 MEMBUAT KARAKTER

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118131 |
| **Nama** | : | Mohammad Harifin |
| **Kelas** | : | D |
| **Asisten Lab** | : | Wisando Berlian Pandensolang (2218095) |

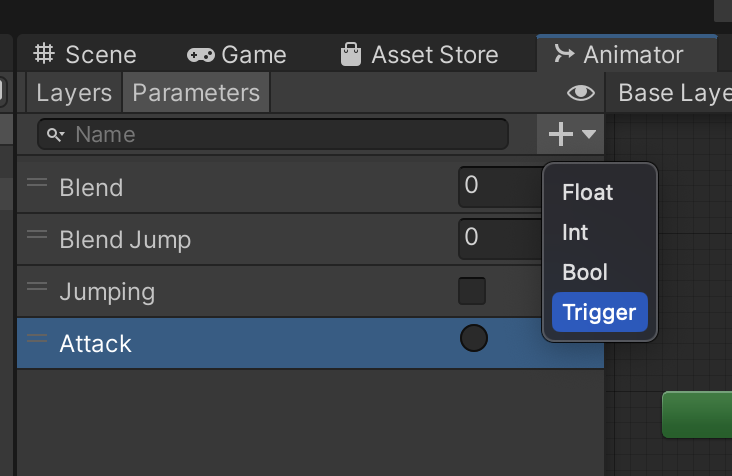
## Tugas 1 : Judul Tugas 1

1. **Membuat mekanisme Attack**
2. Buka Project Bab 9 Untuk melanjutkannya.



### Membuka project bab9

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



### menambahan parameter attack

1. Setelah menambahkan 2 parameter Attack, Langkah selanjutnya adalah membuat Layer *Game object* baru didalam *player-idle-1*, Klik kanan pilih *Create Empty* lalu Rename menjadi *Firepoint dan Firepoint2*

A screenshot of a computer program

Description automatically generated

### menambahakan 2 firepoint

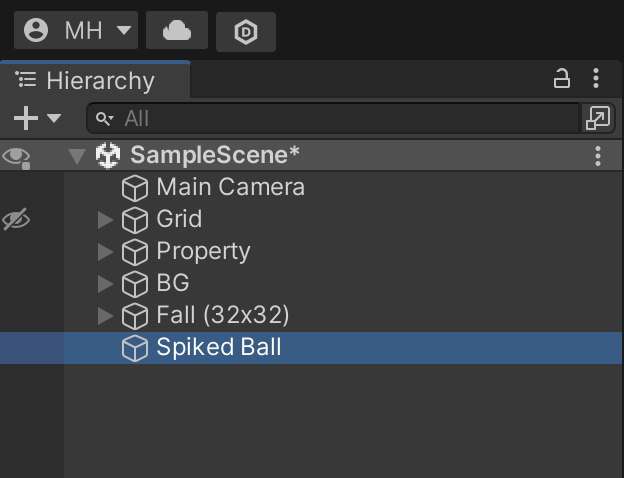
1. Pada menu *Hierarchy* klik Firepoint untuk setting pada Inspector, Ubah *Icon* Menjadi titik, atur letak titik didepan dan belakang player

A screenshot of a computer

Description automatically generated

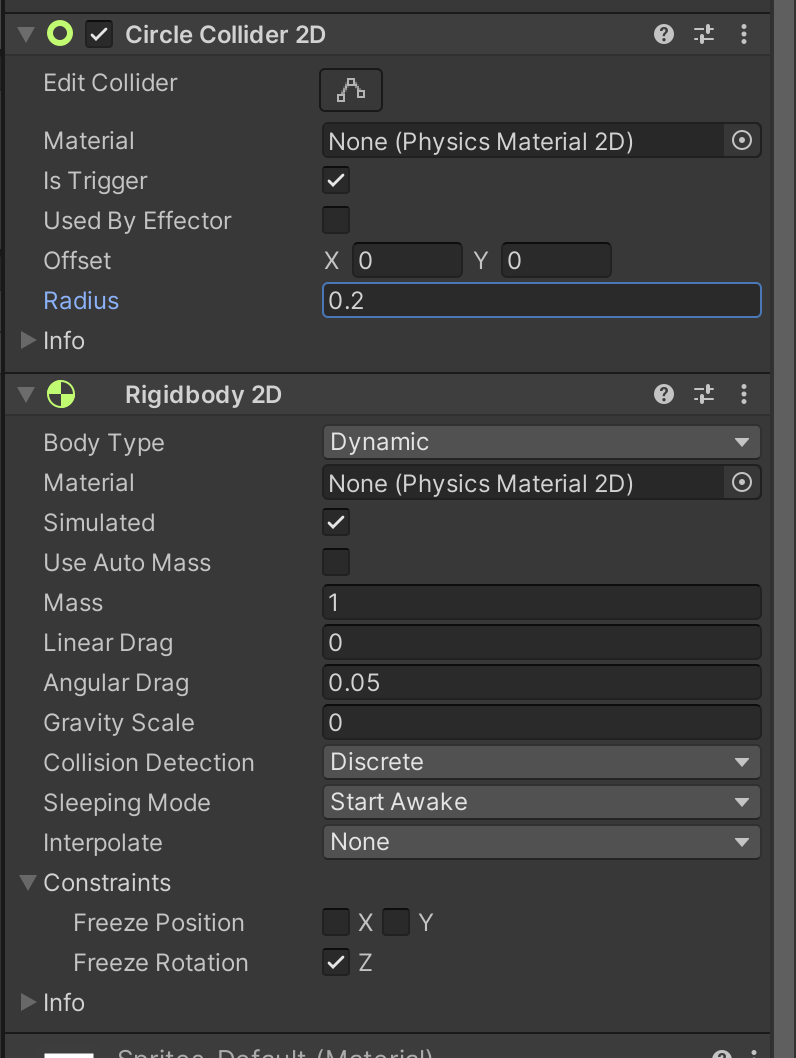
### Mengatur titik Firepoint

1. Pada menu Hierarchy Tambahkan spikedball ball, di folder assets>, *rename* menjadi *fireball*



### menambahkan fireball

1. Klik fireball(spikedball) untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



### menambahkan colide dan ridig body

1. Buat Folder baru *Resources* di menu Project, kemudian drag and drop fireball kedalam folder Resources, dan sembunyikan *fireball* pada *Hierarchy*

A screenshot of a computer

Description automatically generated

### memindahkan lokasi fireball

1. Pada Script Player Tambahkan Script dibawah ini

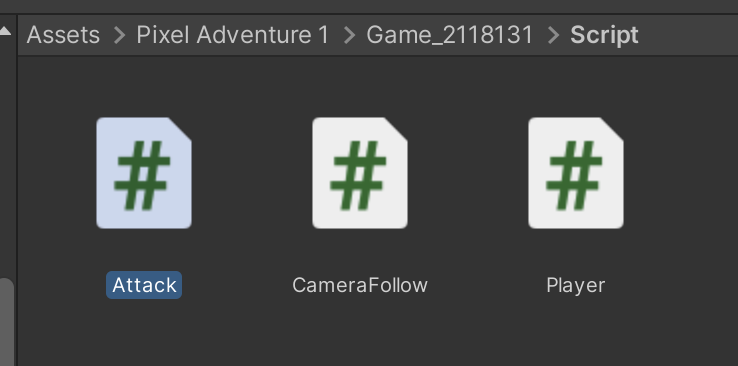
|  |
| --- |
| #Buat variabel baru  public Animator animator;  public GameObject bullet;  public Transform firePoint;  #Tambahkan dibawah fungsi fixedUpdate  IEnumerator Attack()  {  animator.SetTrigger("Attack");  yield return new WaitForSeconds(0.25f);  float direction = Mathf.Sign(transform.localScale.x);  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. Pada Inspector Player, Ubah seperti dibawah ini, Dimana Bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak pertama



### import firepoint

1. Buat Script Attack pada folder Script

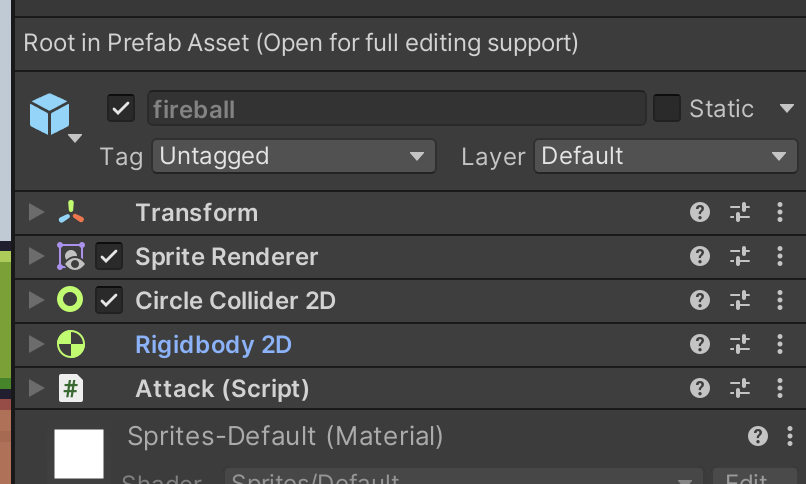


### menambhakan file script

1. Tambahkan Script Attack dibawah ini

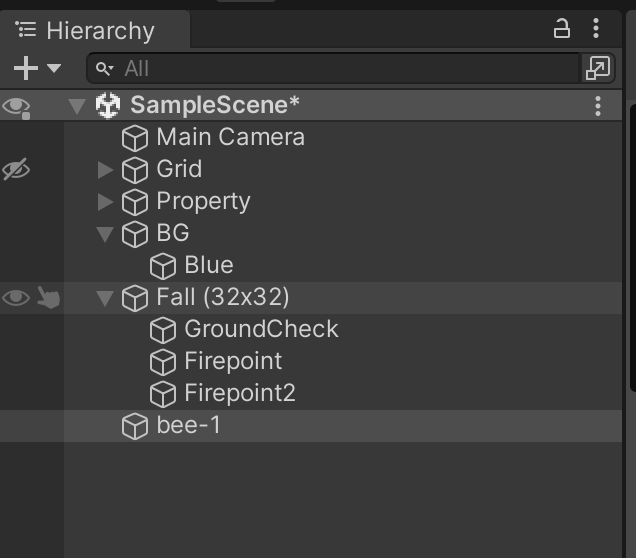
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Attack : MonoBehaviour  {  private void OnTriggerEnter2D(Collider2D collision)  {  if (collision.gameObject.CompareTag("Enemy"))  {  Destroy(gameObject);  Destroy(collision.gameObject);  }  }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

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



### tambahkan attack pada inspector fireball

1. Tambahkan Enemy pada hierarchy di folder Assests> enemies>bee



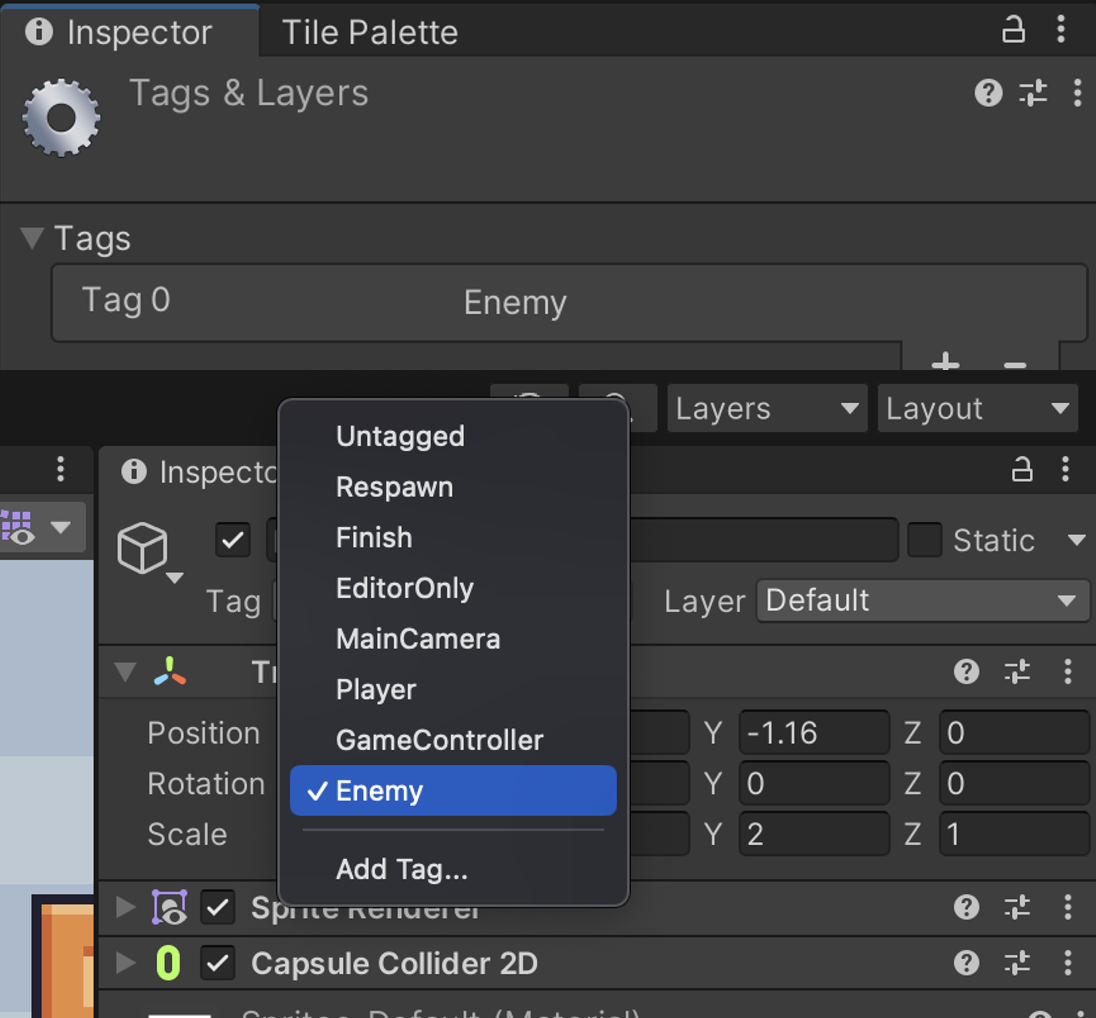
### menambahkan enemy

1. Kemudian klik pada bee-1, lalu pada menu tab inspector tambahkan collider 2D untuk mendeteksinya



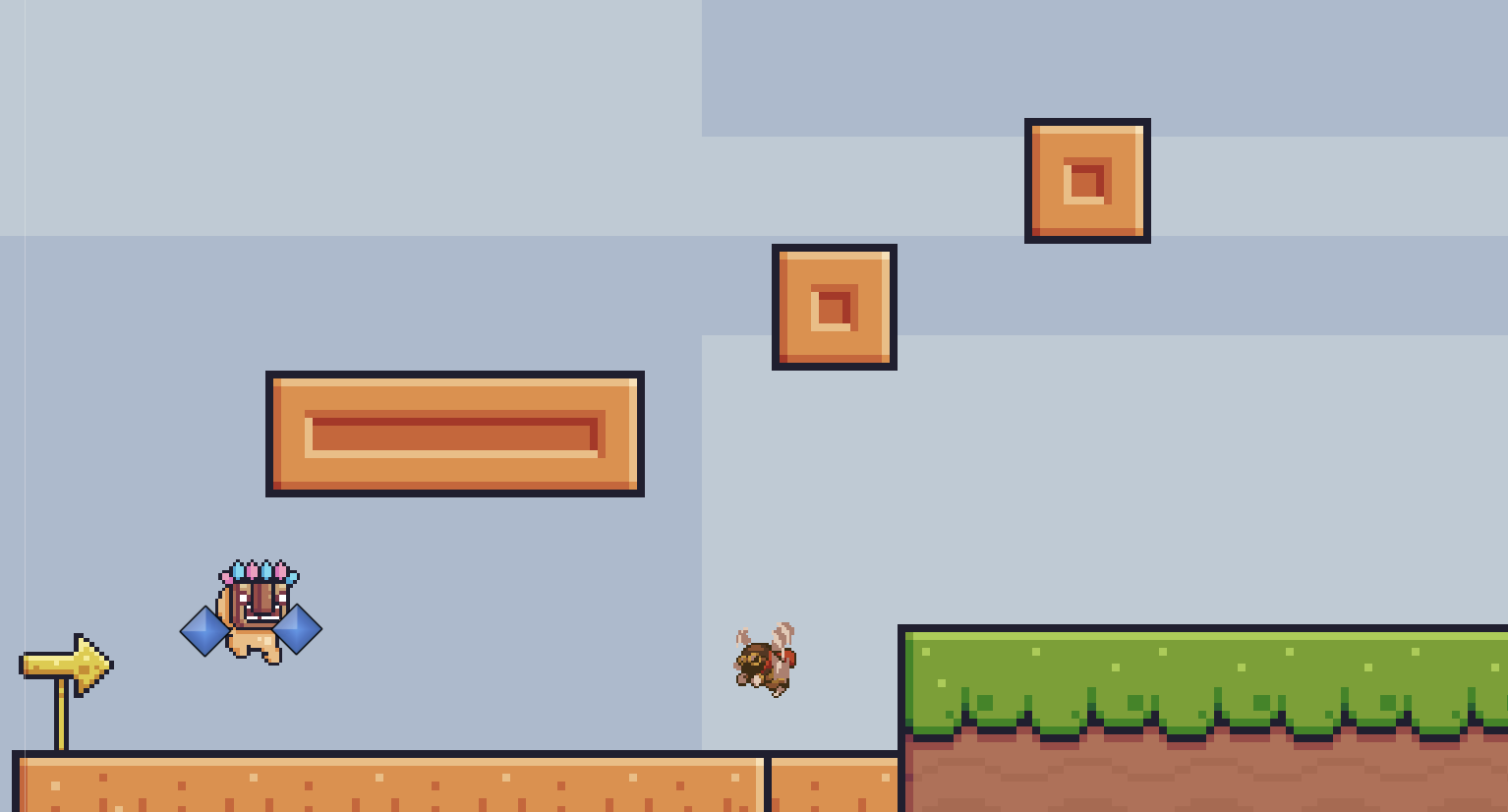
### menambhakan capsule colide pada bee-2

1. Tambahkan Tag Enemy dengan cara Pilih Add Tag, kemudian add tag to the list, Tuliskan Enemy



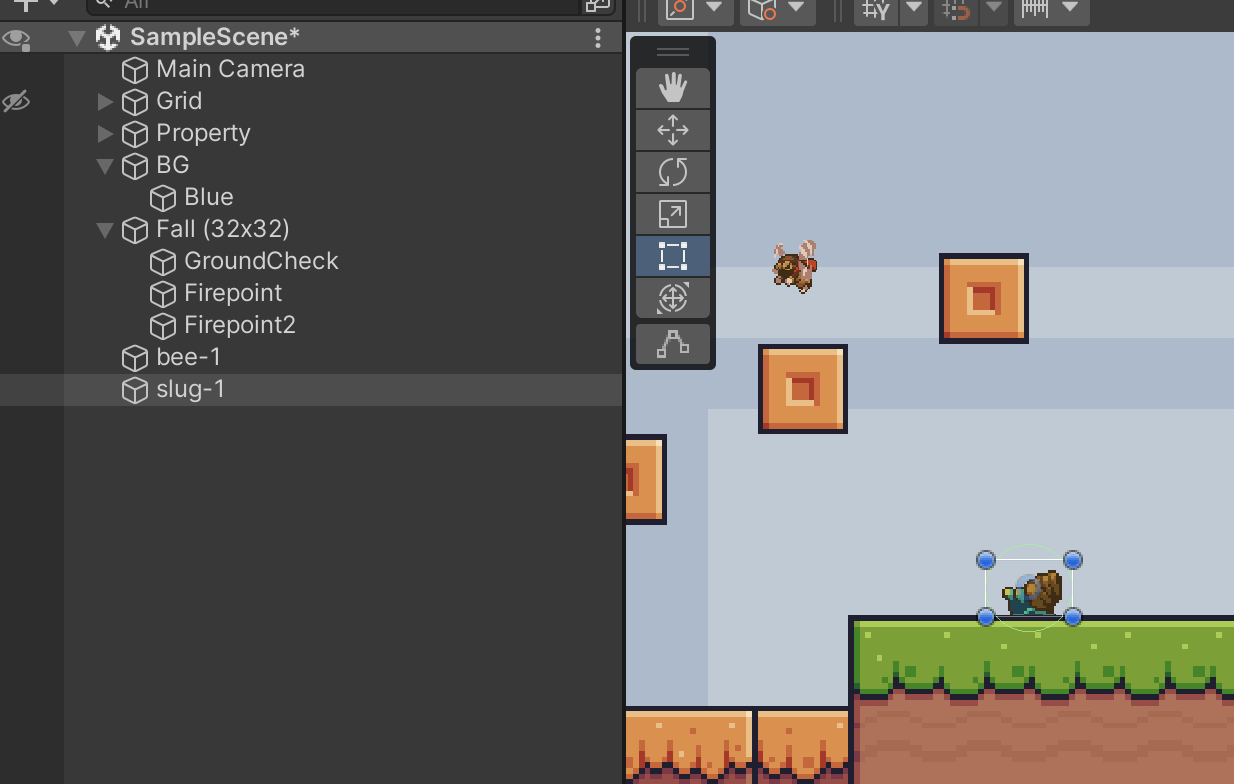
### menambhakan tak enemy

1. Tembak Enemey dengan menekan Tombol C untuk menghancurkan musuh



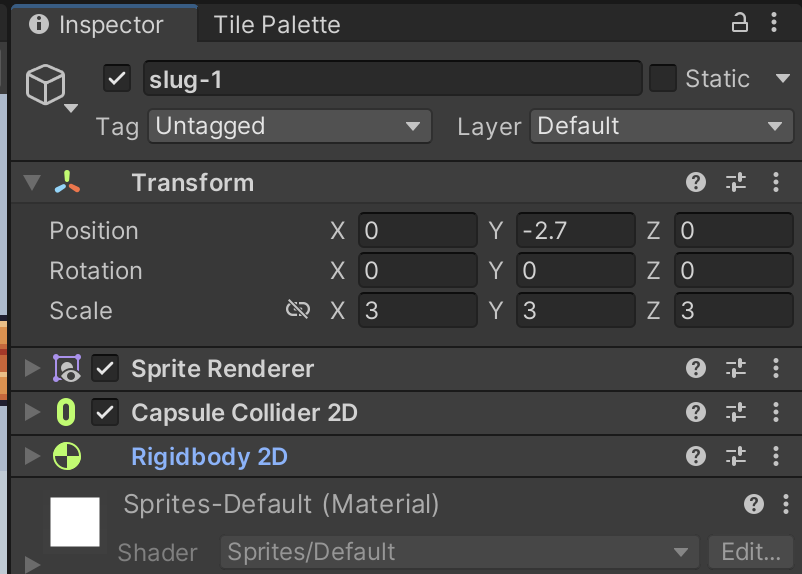
### hasil enemy

1. **Membuat Enemy Behavior**
2. Cari sebuah sprite pack Bernama enemies dan buka folder bernama “slug” ,Tambahkan “slug-1” ke Hierarchy.



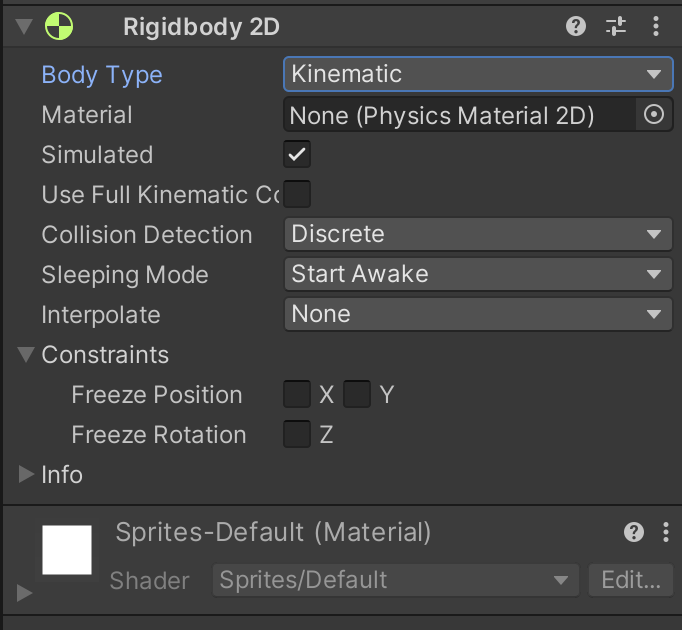
### Gambar Slug

1. Tambahkan sebuah komponen bernama Capsule Colider 2D dan Rigidbody dalam inspector game objek slug-1

****

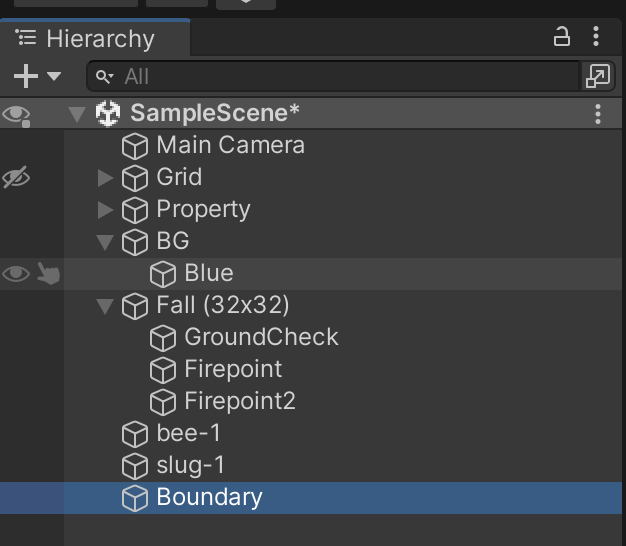
### Capsule Colider 2D

1. Atur sedikit rigitbody 2d tersebut seperti ukurannya diubah jika terlalu besar, dan pada *Body Type* Ubah menjadi *Kinematic*



### Menambahkan Body “kinematics”

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

****

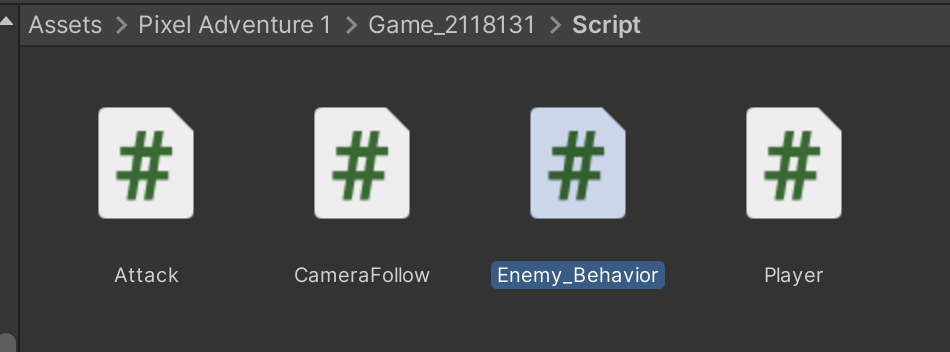
### Menambahkan boundary

1. Tambahkan Box Collider 2d pada Boundary, centang pada Is Trigger lalu atur sesuai keinginan pada size dan offside

****

### Box collider 2D

1. Buat sebuah file script didalam folder Script beri nama “Enemy\_Behavior”, kemudian drag dan masukkan ke dalam game object “slug-1”

****

### Memuat Enemy\_behaviour

1. Tambahkan Script dibawah 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);  }  } |

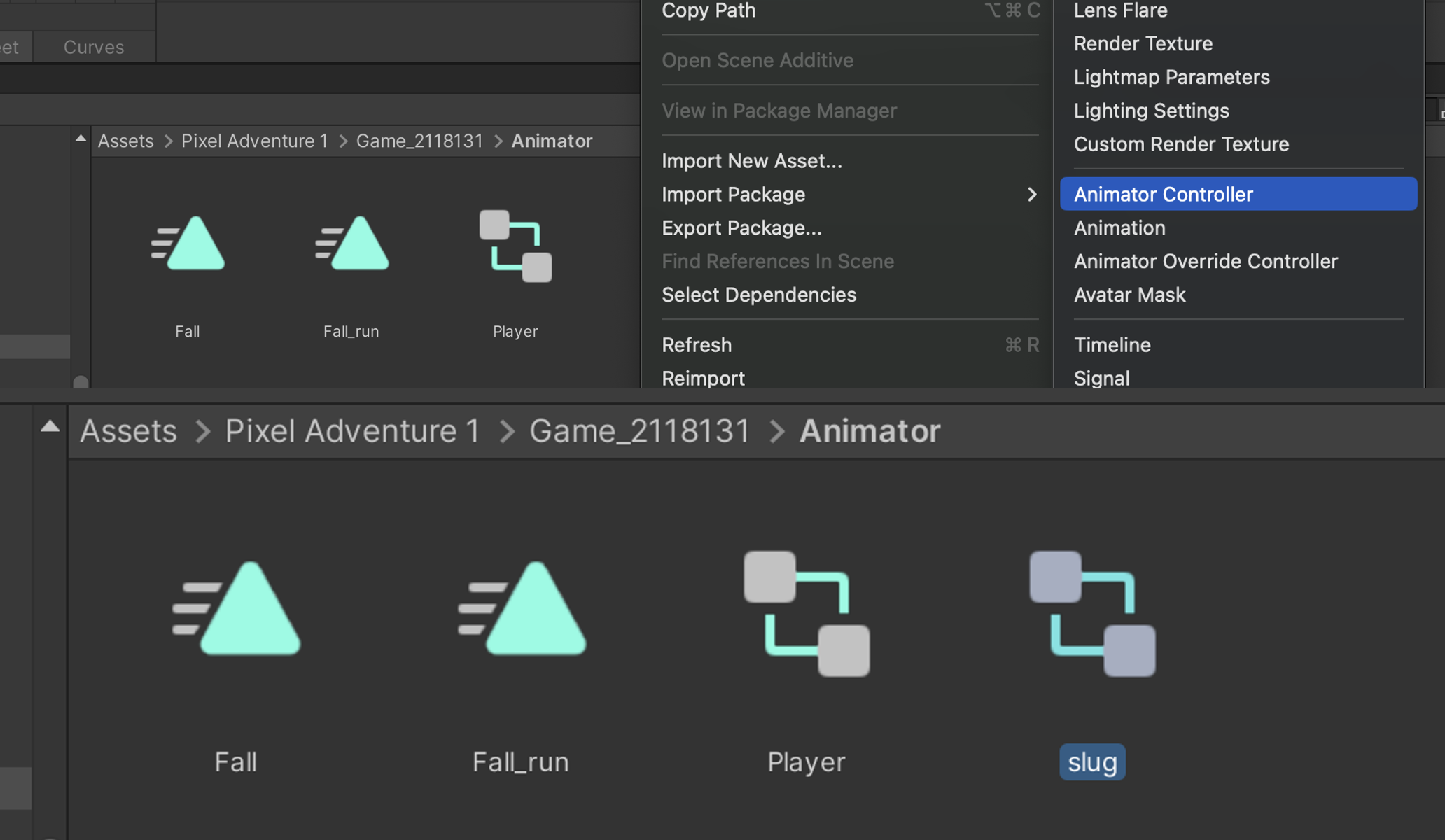
1. Buat animasi untuk slug-1, klik slug-1, tambahkan component Animator.

A screenshot of a video game

Description automatically generated

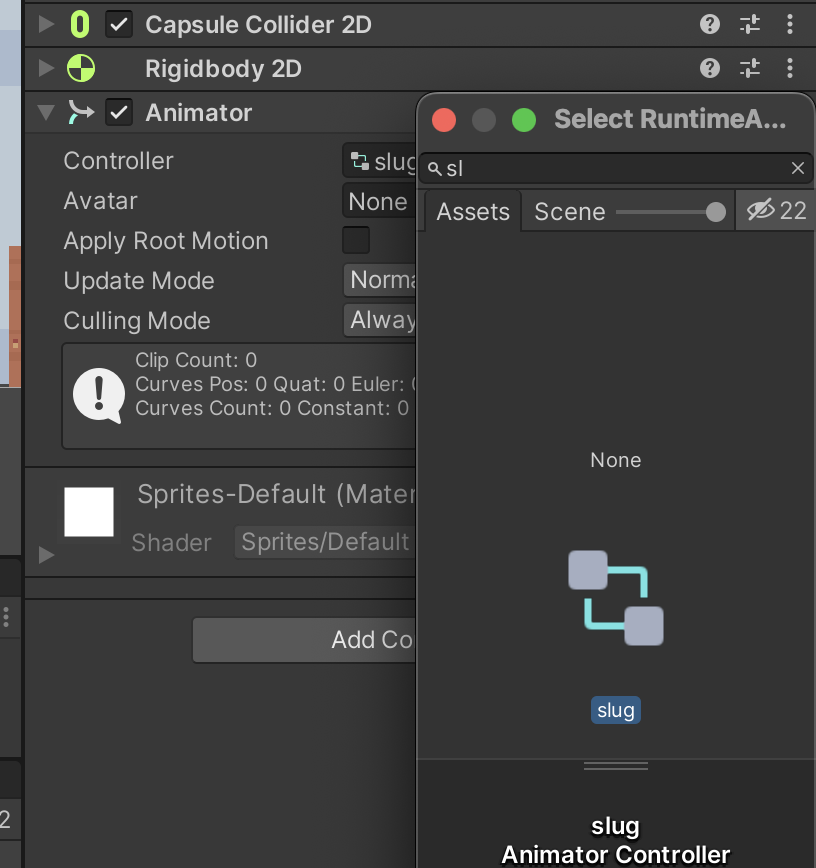
### Componen Animator

1. Buat animator di bagian animator, klik kanan pada folder animator > animator controller , berinama slug



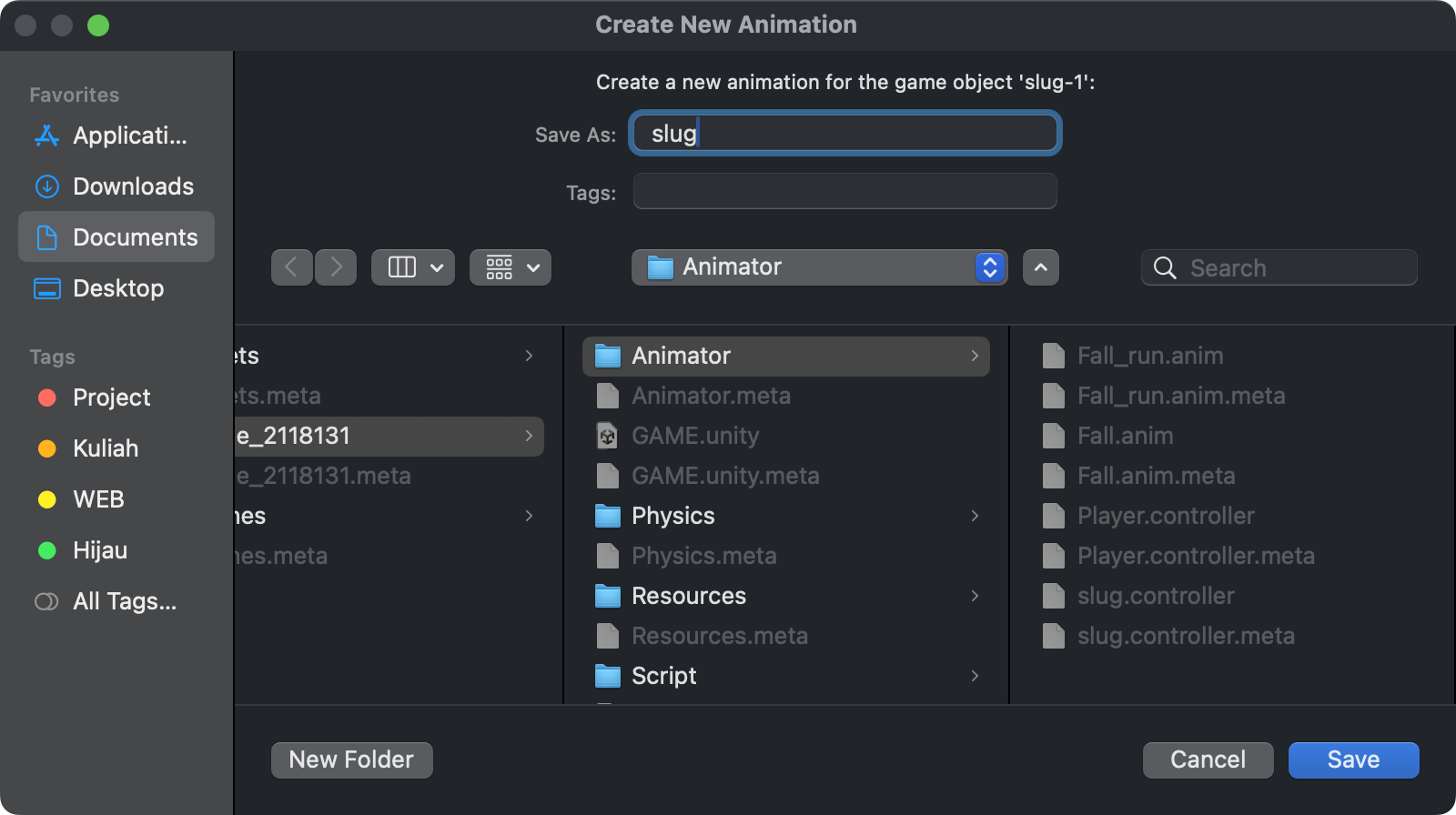
### Membuat animator Controller

1. Buka inspector slug-1, pada animator inport controller slug yang sudah dibuat.



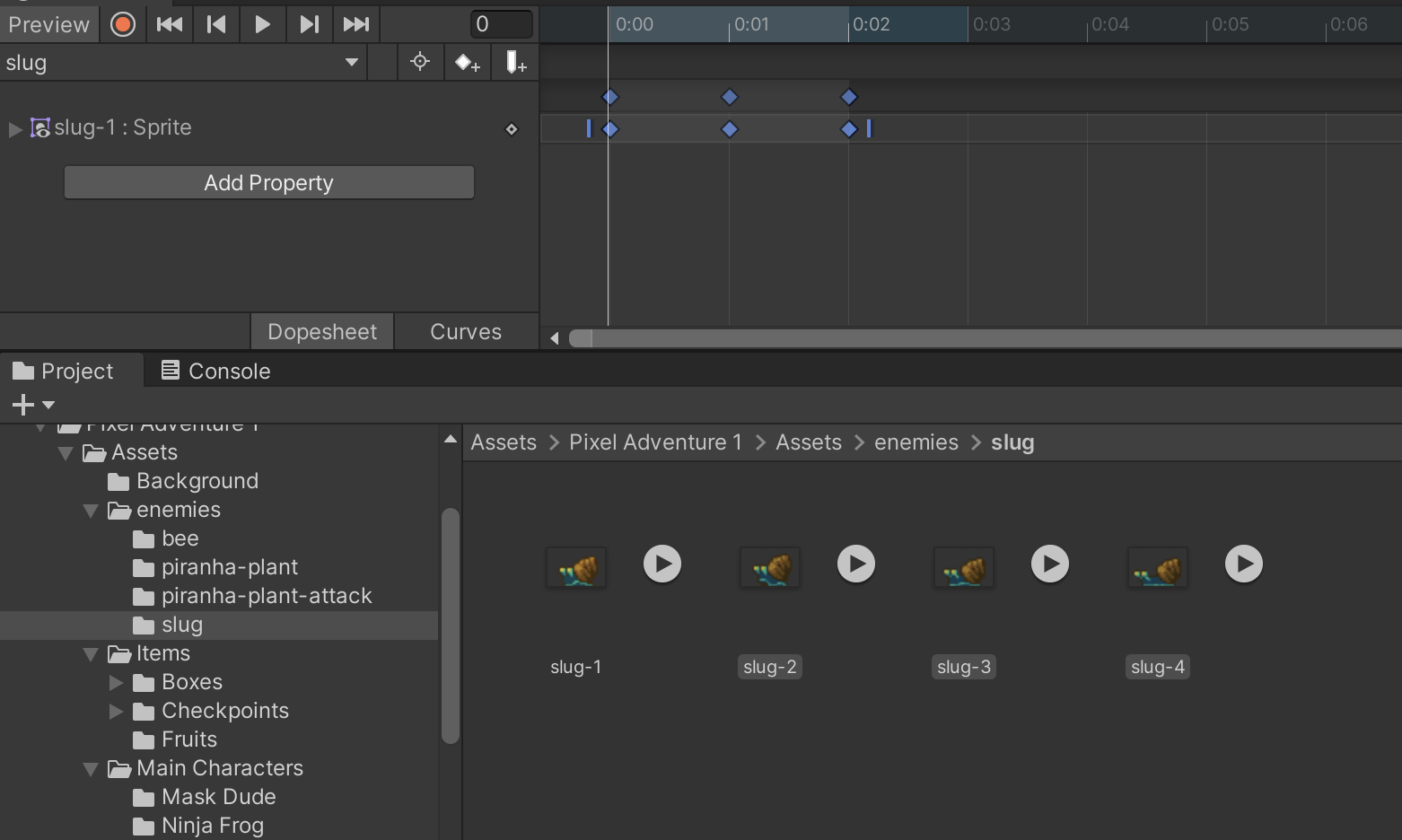
### add controller slug

1. Kemudian tambah animasi dengan klik create pada animation, kemudian save pada folder animator dengan nama slug.



### Create animation

1. Kemudian drag and drop animation slug-2 hingga slug-4 ke dalam add property



### Import slug ke property

1. Maka ketika di game di play akan terdapat animasi pada property slug.



### Animasi slug

1. Duplikat slug ,dan boundary dan posisikan ke tempat yang diinginkan.



### Duplikat slug dan boundary

1. **Membuat Enemy AI**
2. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama 'bee-1'. Tambahkan 'bee-1' pada Hierarchy



### Menambahkan property bee

1. Buat Script Enemy\_AI pada folder Praktikum - Script

A screenshot of a computer

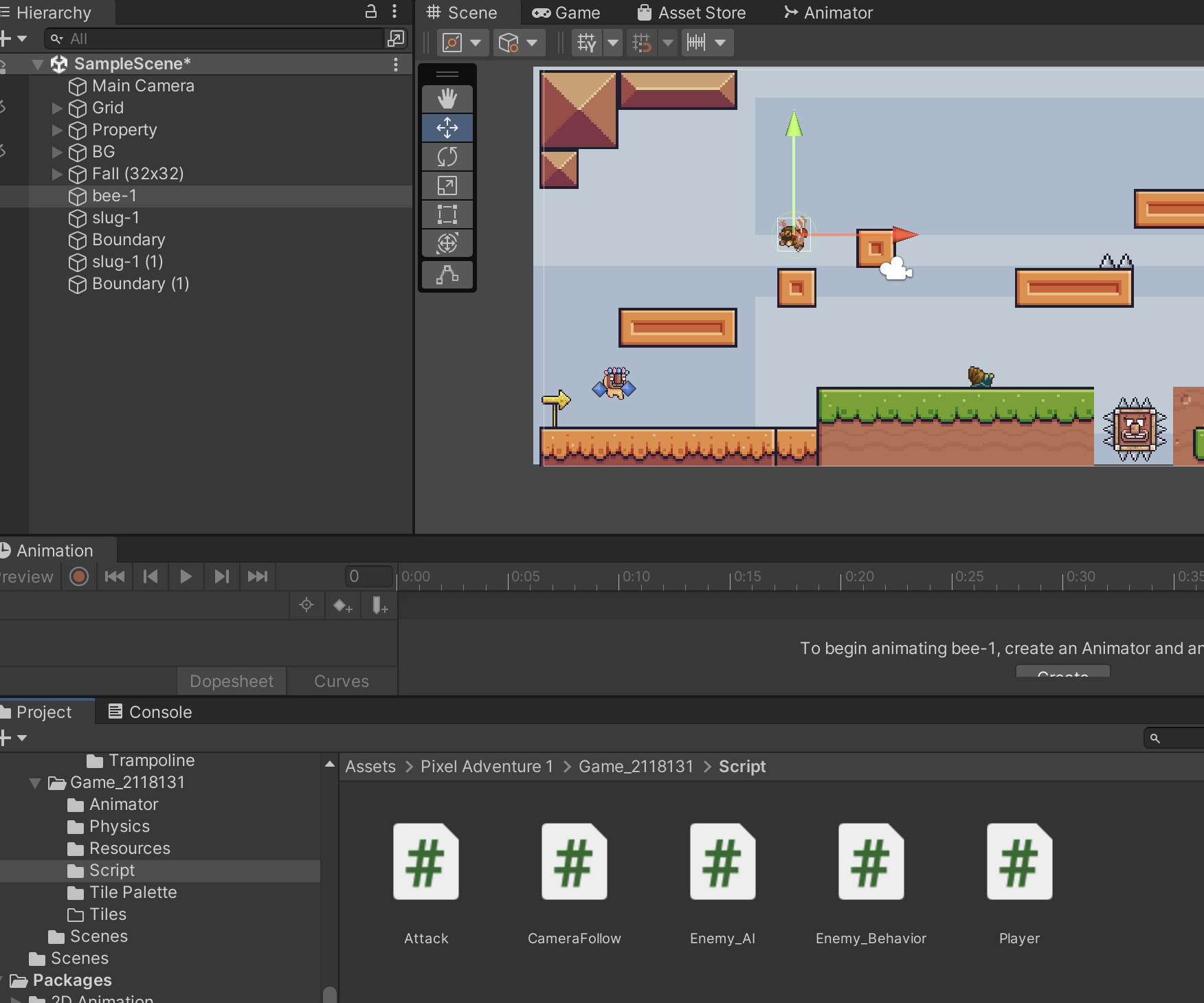
Description automatically generated

### Script Enemy\_Ai

1. Tambahkan Script 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. Kemudian drag and drop script pada bee-1



### Drag and drop script ke bee

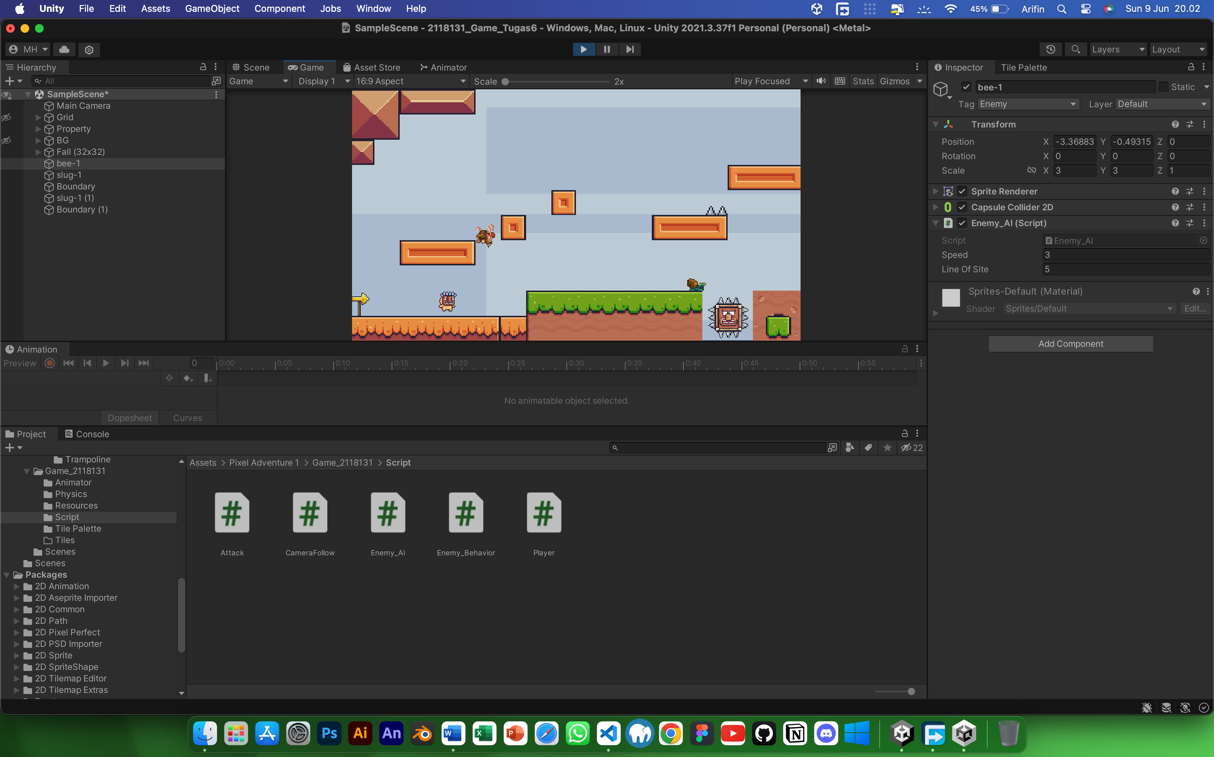
1. Pada Inspector Enemy\_Ai, Atur Speed juga Line of Site untuk menentukan jarak dan speed pada enemy

A screenshot of a video game

Description automatically generated

### Mengatur speed dan line of site

1. Running Game, maka bee akan mengikuti Gerakan Player



### Running game

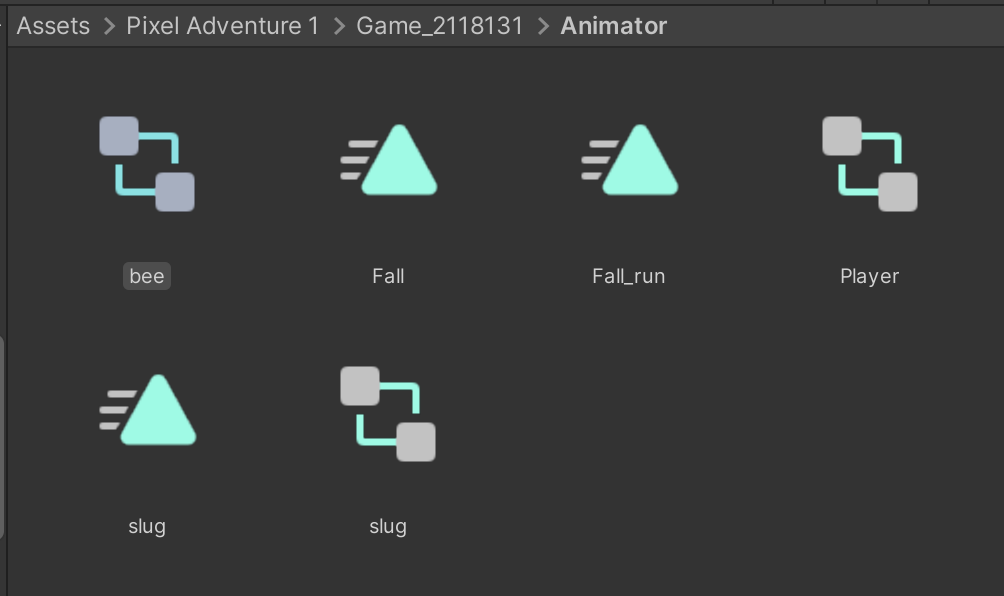
1. Buat animasi untuk bee, masuk ke dalam inspector bee > tambahkan component animator.

A screenshot of a computer program

Description automatically generated

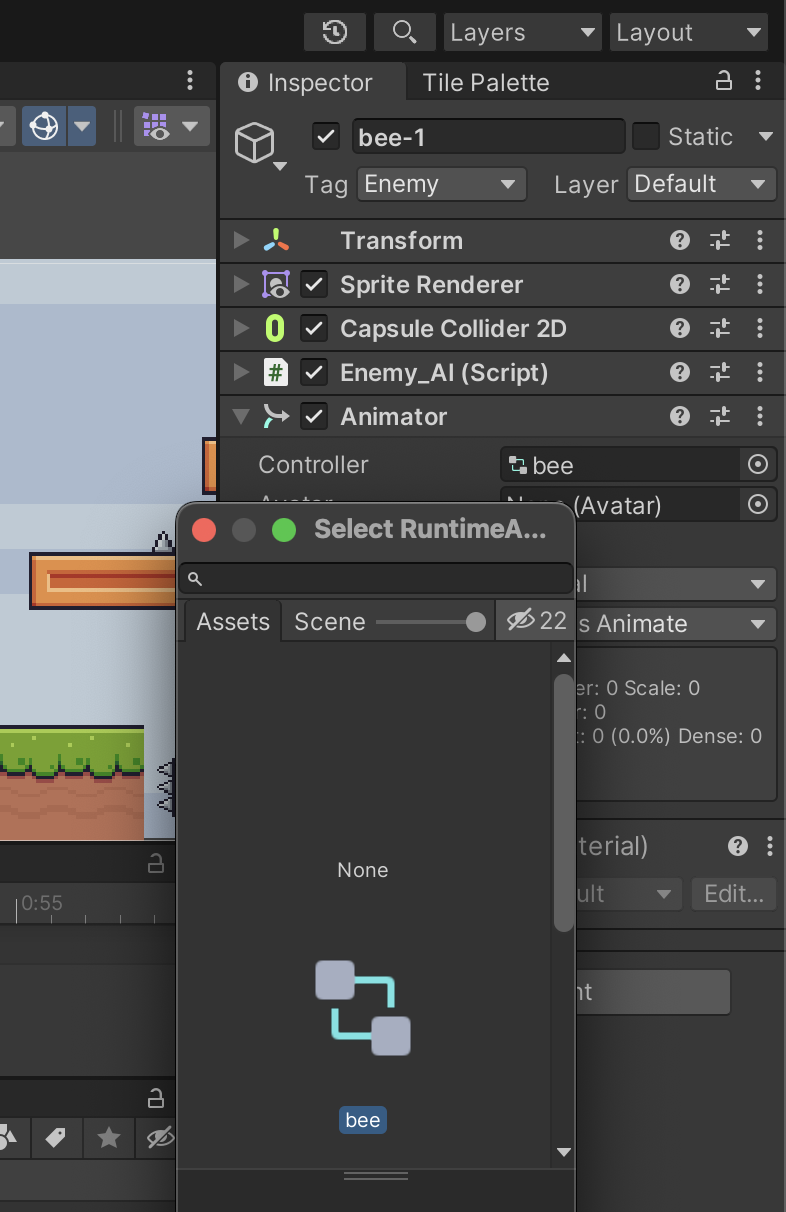
### component animator bee

1. Tambahkan animator bee pada folder animator.



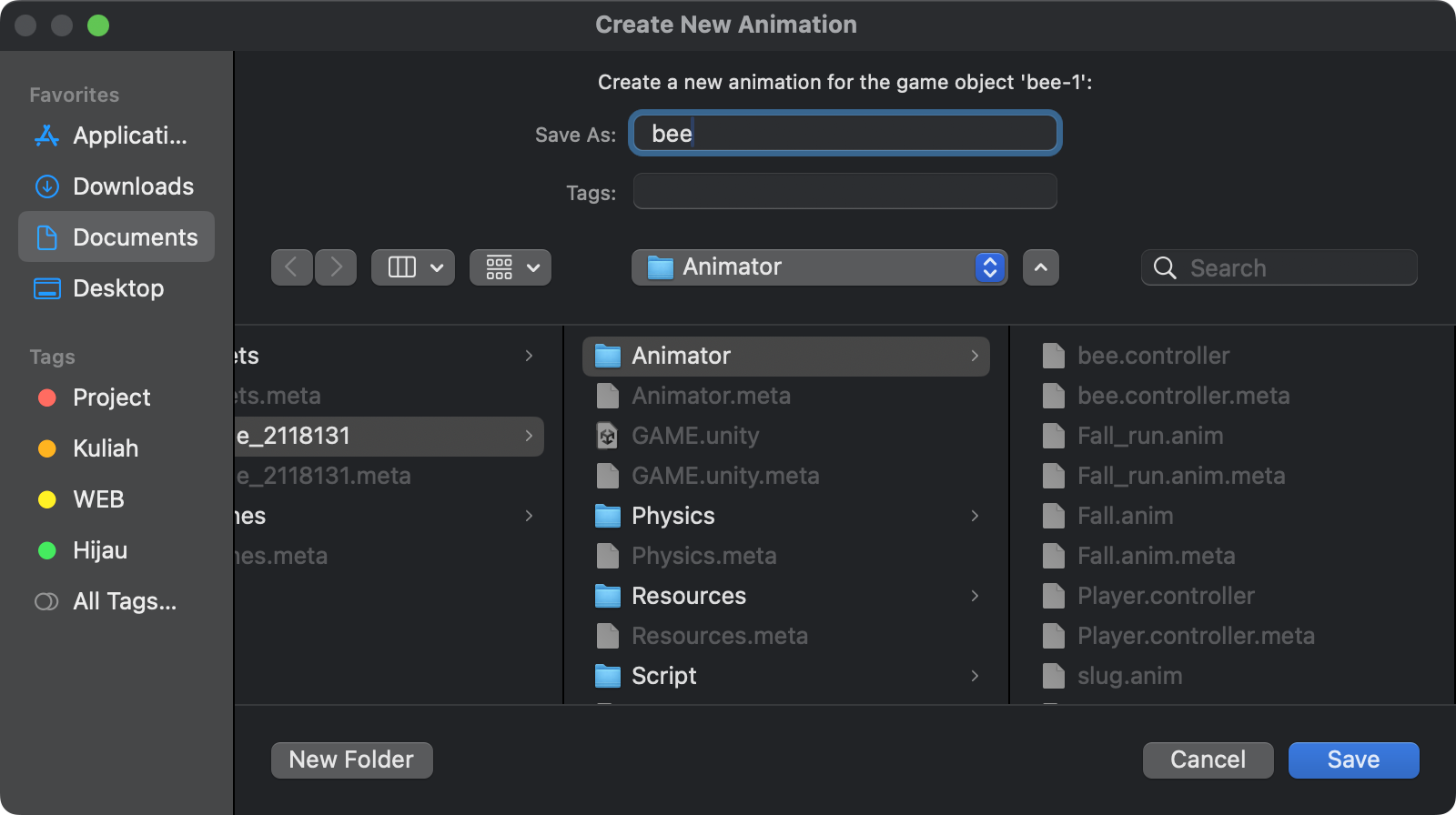
### Menambahkan animator bee

1. Kemudian import animator yang sudah di baut ke dalam animator inspector bee, di bagian controller.



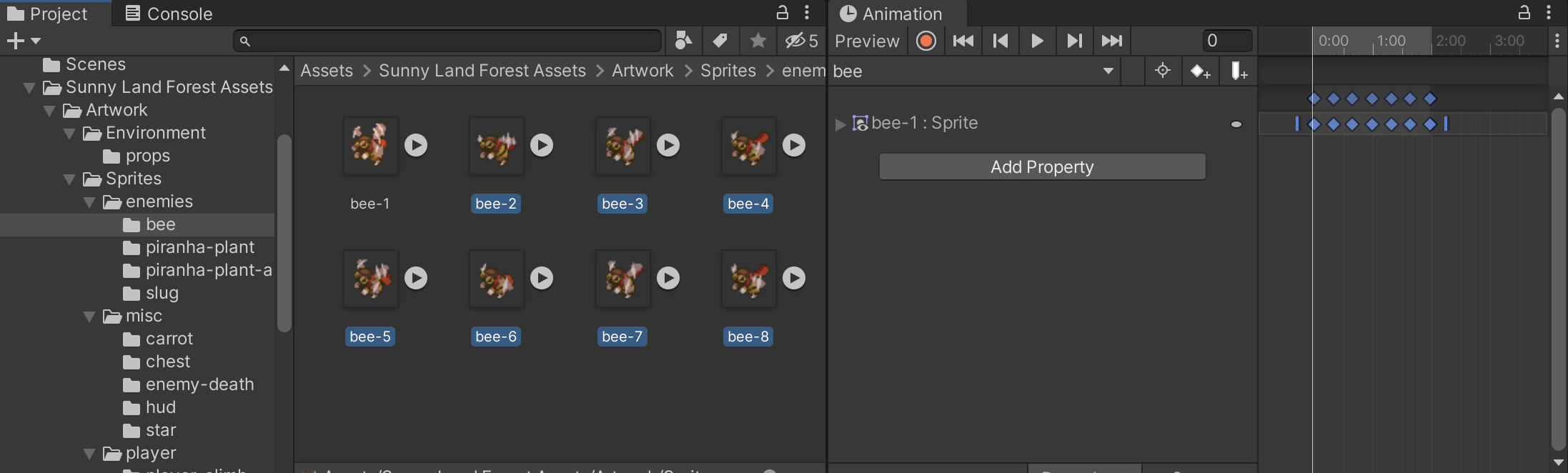
### Import animator controller

1. Kemudian tambah animasi dengan klik create pada animation, kemudian save pada folder animator dengan nama bee.



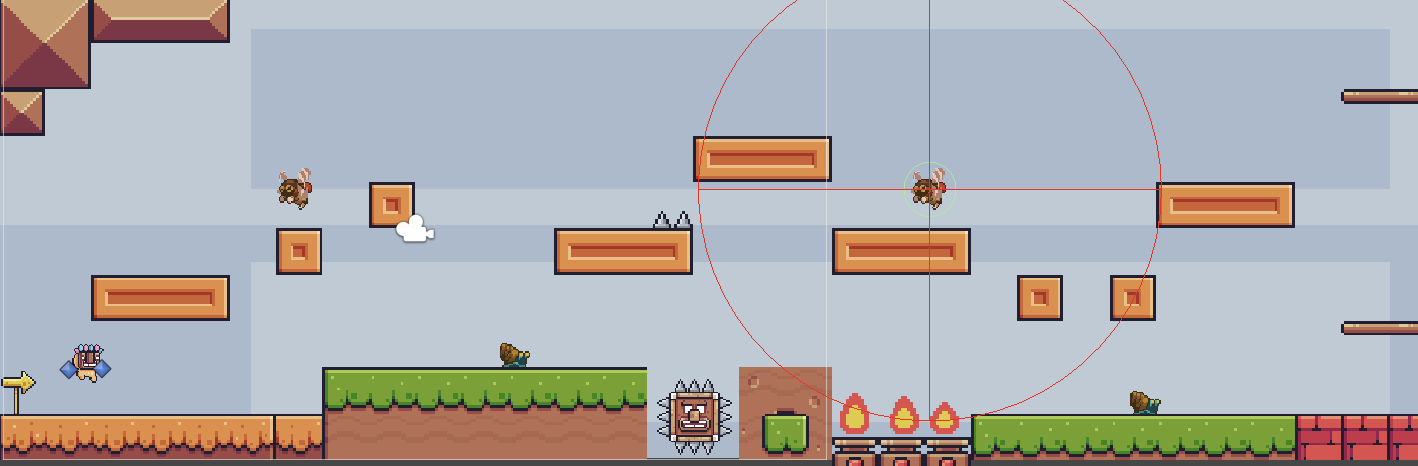
### Create new animator bee

1. Kemudian drag and drop animation slug-2 hingga slug-4 ke dalam add property.



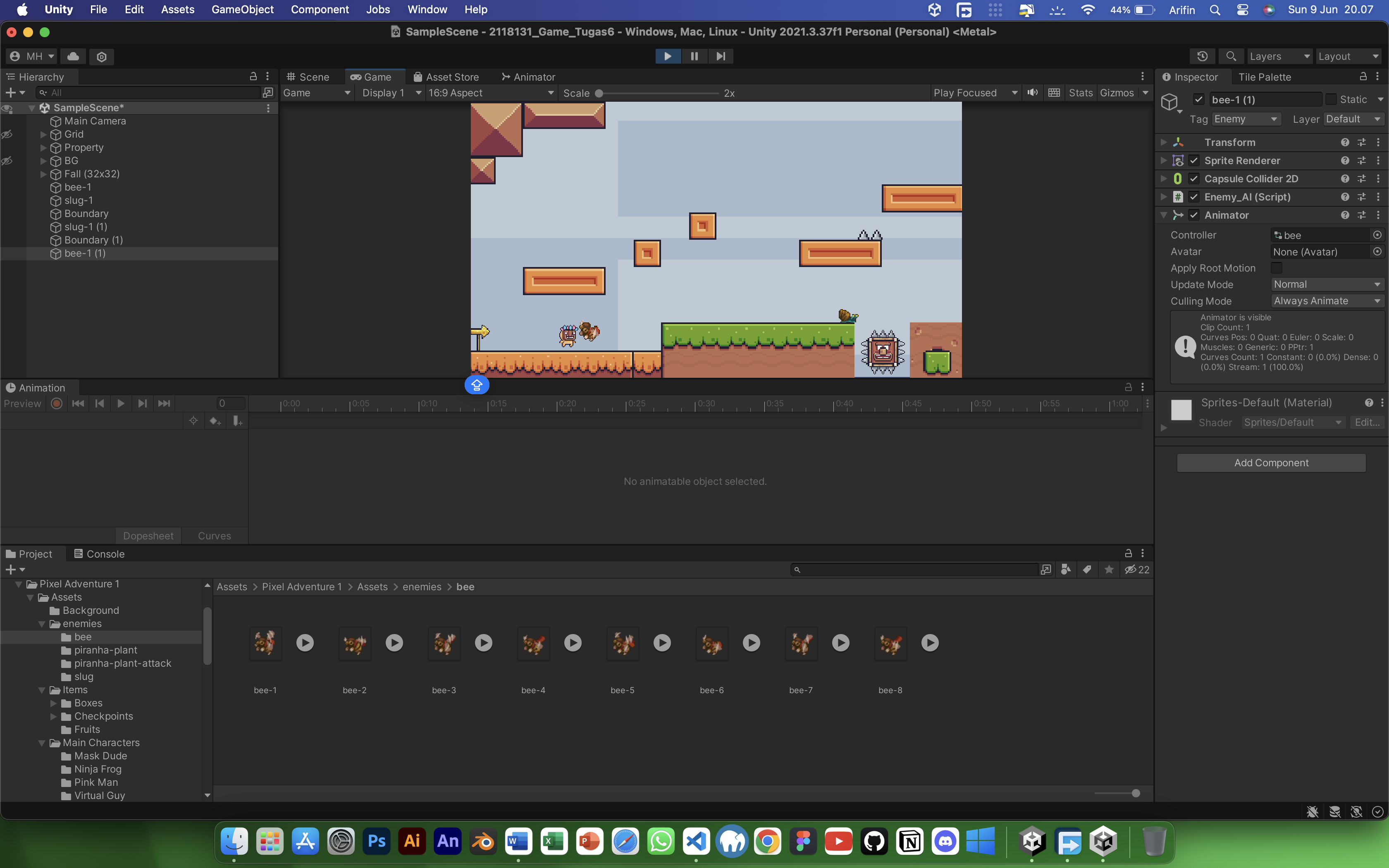
### Drag and drop amination bee

1. Duplikat property bee, dan atur posisi sesuai yang diinginkan.



### Duplikat property bee

1. Running game, maka bee akan mengikuti Gerakan player.



### Jalankan game

1. **Respawnbt**
2. Buka file script (Player.cs) tambahkan variabel nyawa seperti dibawah ini

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

1. Tambahkan kode dibawah ini untuk mengatur posisi respawn sesuai dengan posisi awal permainan dimulai, tambahkan source code di function Awake()

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

1. Tambahkan kode dibawah ini di dalam void update Player.cs agar ketika nyawa player dibawah 0 maka akan melakukan respawn

|  |
| --- |
| if(nyawa < 0){  playagain();  } |

1. Tambahkan juga kode berikut dibawah code sebelumnya agar ketika player jatuh dibawah platform akan melakukan respawn

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

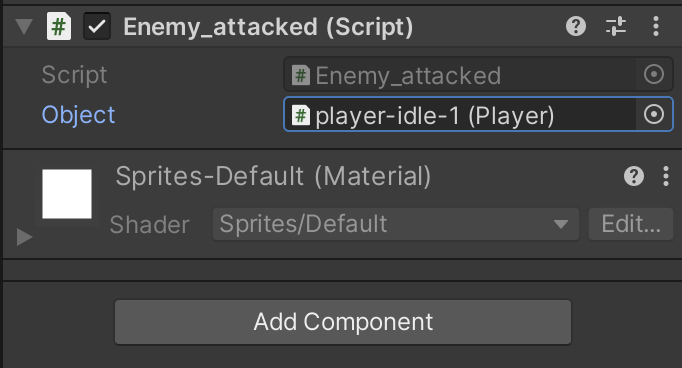
1. Tambahkan fungsi playagain() dalam script Player.cs

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

1. Tambahkan file script (Enemy\_Attacked.cs) dan isikan source code dibawah ini

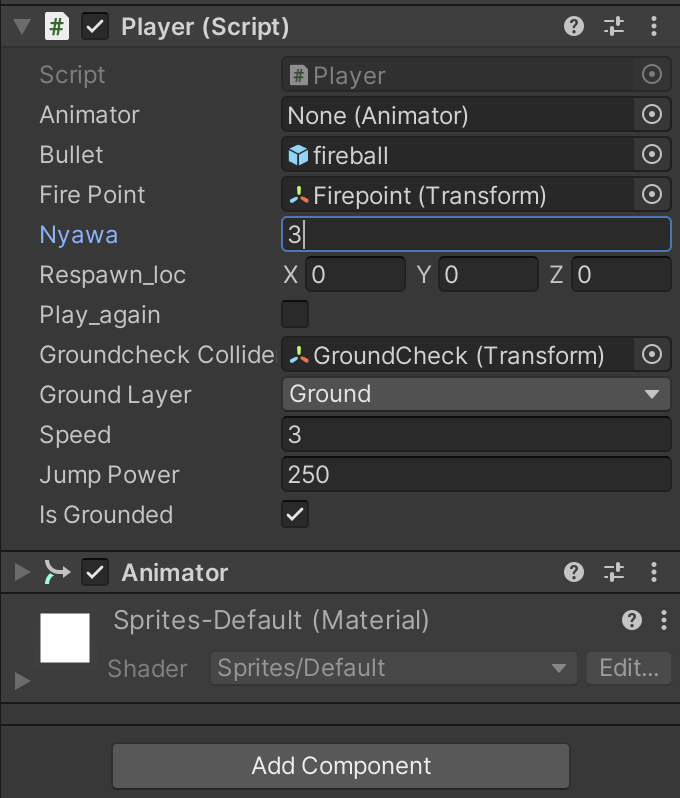
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_attacked : MonoBehaviour  {  [SerializeField] private Player Object;  void Start()  {  if (Object == null)  {  Object = GameObject.FindWithTag("Player").GetComponent<Player>();  }  } 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

1. Pada hierarchy bee-1 Tambahkan Script enemy\_attack, arahkan object pada player-idle-1



### Enemy attack

1. Klik game object Player, pergi ke Inspector dan ubah nilai Nyawa menjadi 3 pada Player(Script)



### Healt player

1. Jika di play, Player mengenai atau menyentuh opposum-1 sebanyak 3 kali maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan reswpawn ke titik awal.



### Play games

1. **Link Github Pengumpulan**

https://github.com/Arifin-MRX/2118131\_Animasi\_dan\_Game.git

## Tugas 2 : QUIS

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

Analisa

Pada source code di tasa yaitu melakukan perbaikan dan penambahanuntuk memastikan fungsionalitas yang benar. Pertama, tipe data attackRange diubah dari int menjadi float untuk mendukung nilai desimal. Kedua, typo pada variabel attacDamage diperbaiki menjadi attackDamage. Ketiga, metode Update diperbaiki dengan menambahkan titik di antara Input dan GetButtonDown sehingga menjadi Input.GetButtonDown("Fire1"). Terakhir, pada metode PerformMeleeAttack, setelah mendeteksi hit dengan Raycast, dilakukan pengecekan apakah objek yang terkena memiliki komponen Health, dan jika iya, metode TakeDamage dipanggil untuk mengurangi health musuh. Dengan perbaikan ini, skrip PlayerAttack akan berfungsi untuk mendeteksi serangan jarak dekat dan mengurangi health musuh yang terkena serangan, sesuai dengan mekanisme game yang diharapkan.