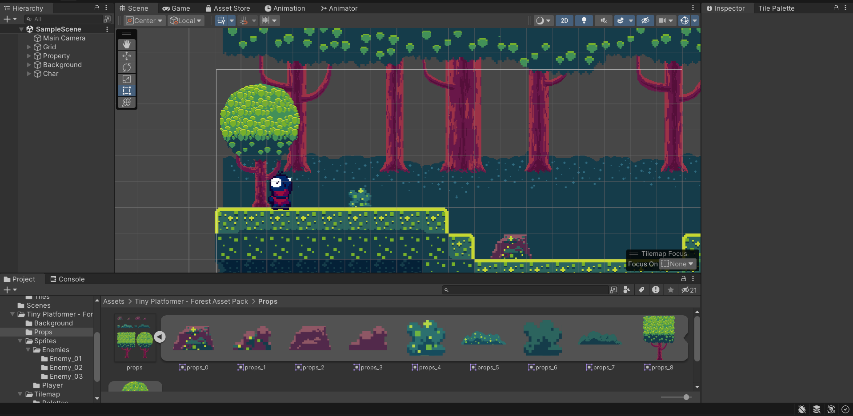
# 10 Respawn and AI Enemy Attack

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118120 |
| **Nama** | : | Rizky Iman Virgiawan |
| **Kelas** | : | D |
| **Asisten Lab** | : | 2218034 - Berchmans Bayu Bin Jaya |
| **Baju Adat** | : | Baju Jas Tutup |
| **Referensi** | : | https://berita.99.co/wp-content/uploads/2023/08/banjar-babaju-kubaya-panjang.jpg |

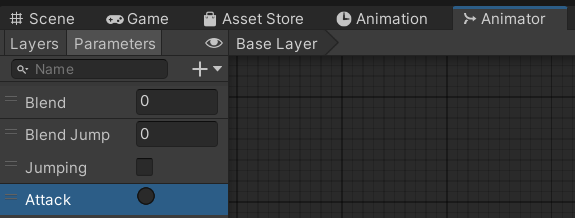
## Tugas 10: Respawn, Menyerang dan Diserang Musuh

1. **Menyerang Musuh**
2. Buka proyek sebelumnya



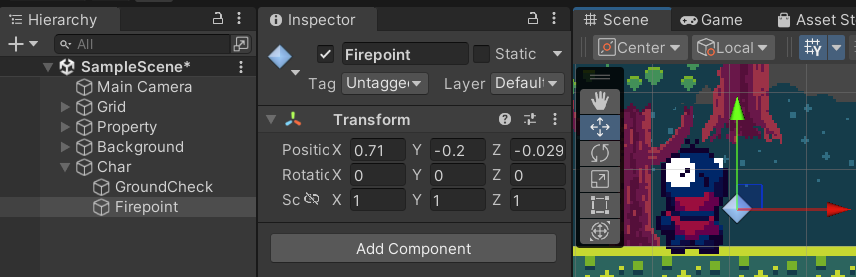
### 10.1 Proyek Sebelumnya

1. Tambahkan parameter *Attack* tipe data trigger di menu *Animator*



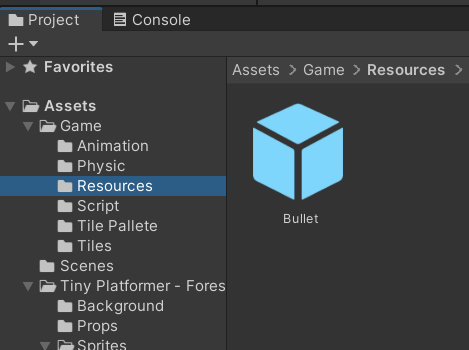
### 10.2 Menambah Parameter *Attack*

1. Buat hierarki ‘Firepoint’ dan masukkan ke hierarki Char. Ubah *icon* menjadi kristal biru dan posisikan di depan karakter



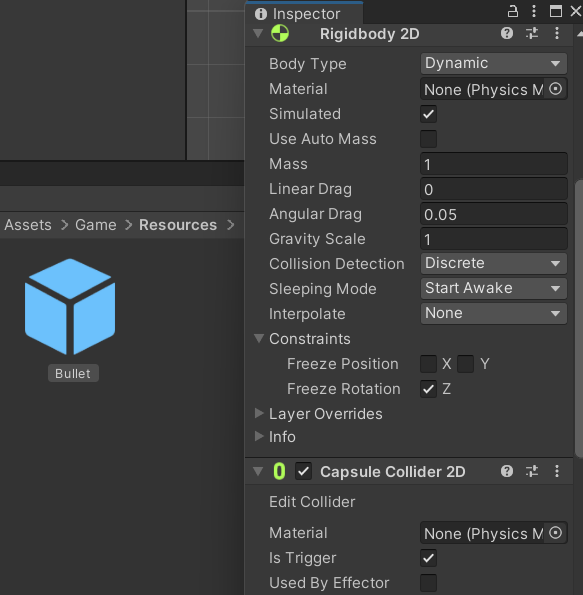
### 10.3 Menambah Firepoint pada Char

1. Salin aset *Bullet* di *Tiny Platformer > Props* ke direktori *Resources*



### 10.4 Menyalin Aset *Bullet*

1. Tambahkan komponen *Rigidbody 2D* dan *Collider* pada *Bullet*. Atur konfigurasinya seperti gambar 10.5

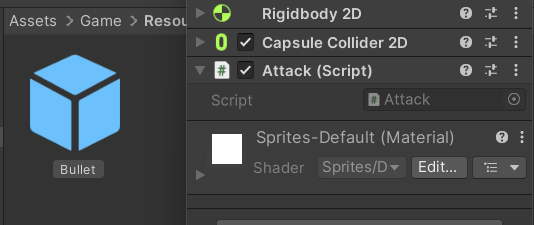


### 10.5 Komponen Bullet

1. Buat kodek *Attack* di dalam direktori Script dan tuliskan kodeknya

|  |
| --- |
| 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. Tambahkan kodek *Attack* di komponen *Bullet*

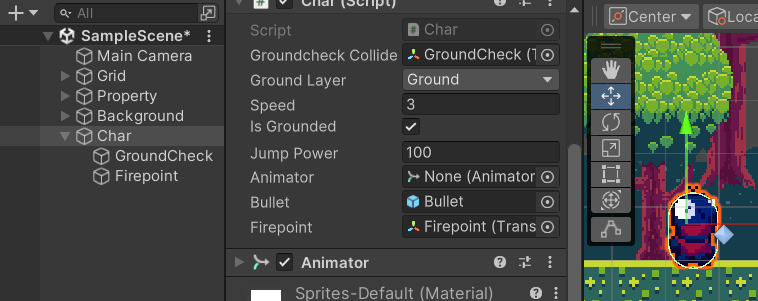


### 10.6 Menambahkan Kodek *Attack* di Komponen Bullet

1. Tambahkan beberapa kodek pada berkas Char.cs agar *Char* dapat melontarkan *Bullet*

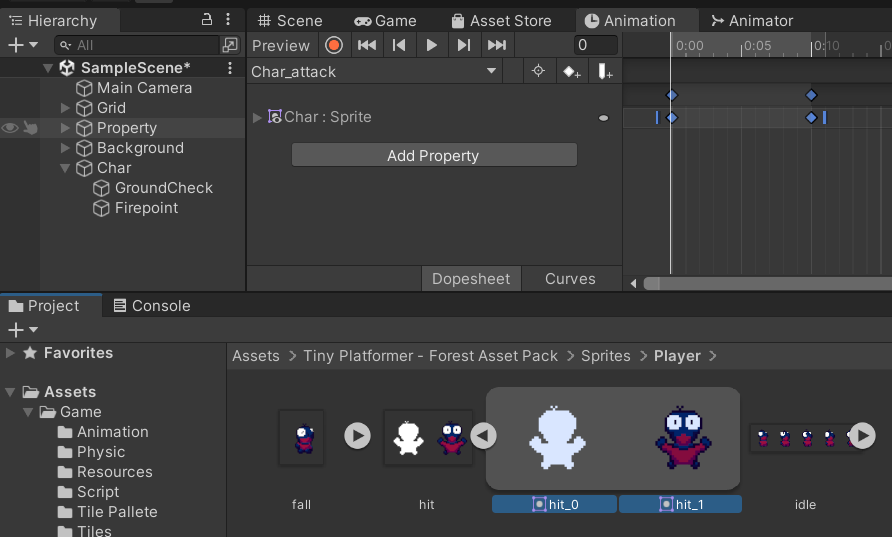
|  |
| --- |
| public GameObject bullet; // di dalam class Char  public Transform Firepoint;  IEnumerator Attack(){          animator.SetTrigger("Attack");          yield return new WaitForSeconds(0.25f);          float direction = facingRight ? 1f : -1f;         float rotationAngle = facingRight ? 0f : 180f;          Quaternion rotation = Quaternion.Euler(0, 0, rotationAngle);          GameObject Rocket = Instantiate(bullet, Firepoint.position, rotation);          Rocket.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 5f, 0);          Destroy(Rocket, 2f);      }  // di dalam void Update()  if (Input.GetKeyDown(KeyCode.C)){              StartCoroutine(Attack());              animator.SetBool("AttackChar", true);          } else if (Input.GetKeyUp(KeyCode.C)) {              animator.SetBool("AttackChar", false);          } |

1. Atur konfigurasi kodek Char di komponen Char dan pilih *Bullet* yang ada di direktori Resources sebagai *Bullet* dan *Firepoint* di dalam hierarki Char sebagai *Firepoint*



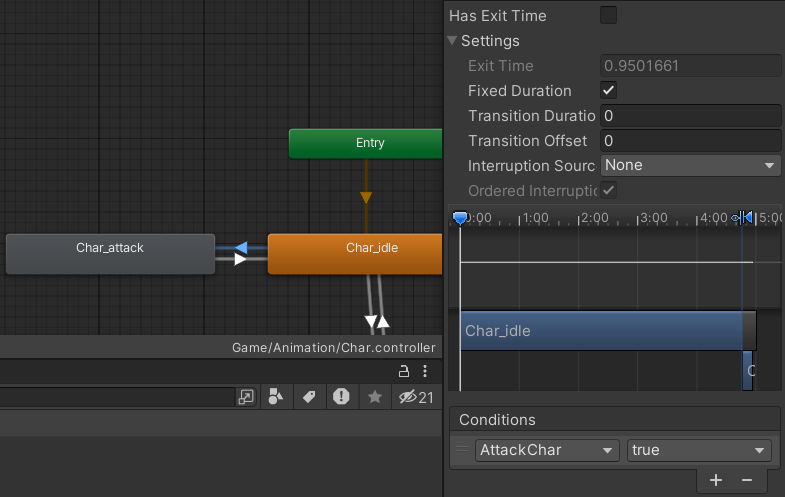
### 10.7 Menambahkan *Bullet* dan *Firepoint* di Char

1. Buat *clip* baru dengan nama ‘Char\_attack’ dan *drag drop* seluruh aset hit ke halaman kerja *Animation.* Beri rentang hingga 0:10



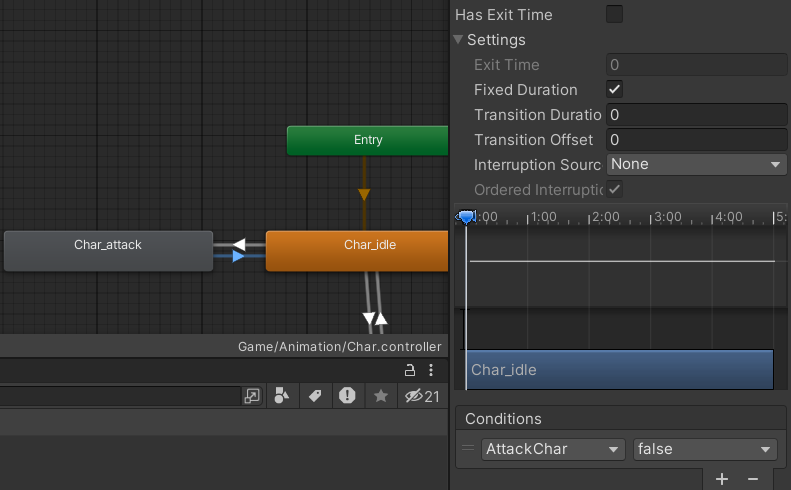
### 10.8 Membuat Animasi *Char* Menyerang

1. Buat parameter *AttackChar* tipe data *bool* dan hubungkan char\_idle ke char\_attack dengan konfigurasi seperti gambar 10.9



### 10.9 Menghubungkan char\_idle ke char\_attack

1. Hubungkan char\_attack ke char\_idle dengan konfigurasi seperti gambar 10.10



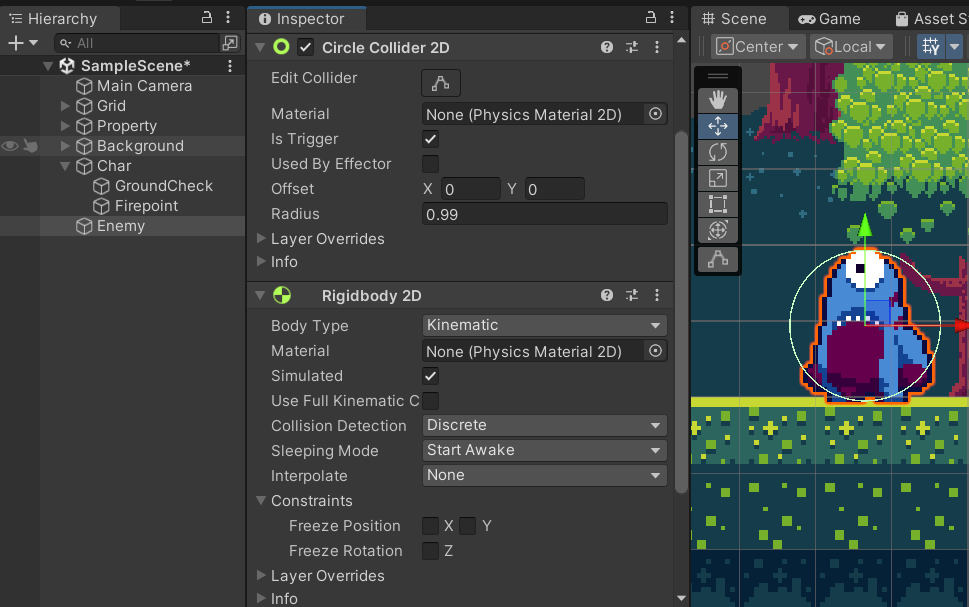
### 10.10 Menghubungkan char\_attack ke char\_idle

1. **Enemy Behavior**
2. Tambahkan musuh dari aset Enemy\_01 dan beri nama *Enemy*



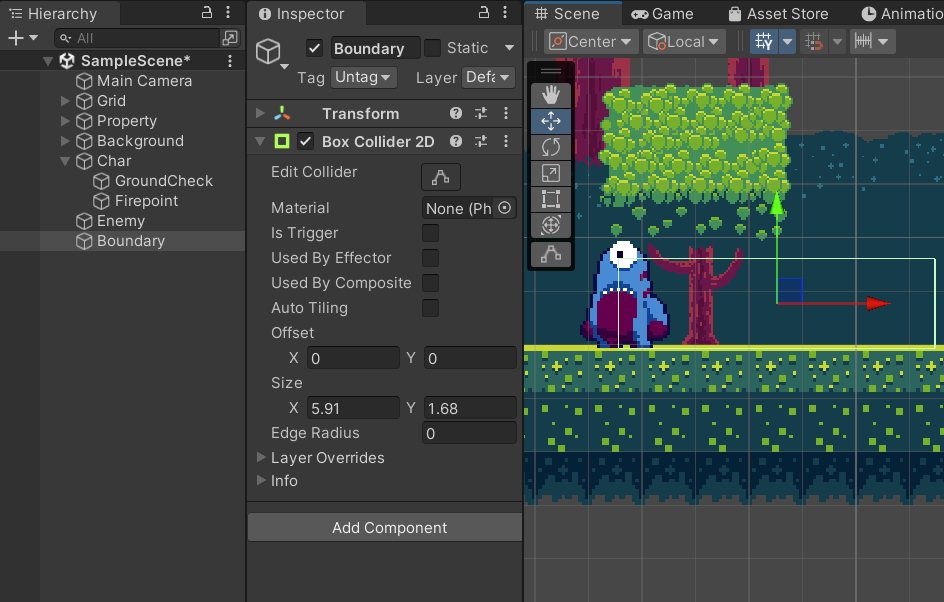
### 10.11 Menambahkan Musuh

1. Tambahkan komponen *Rigidbody 2D* dan *Collider* dengan konfigurasi seperti gambar 10.12



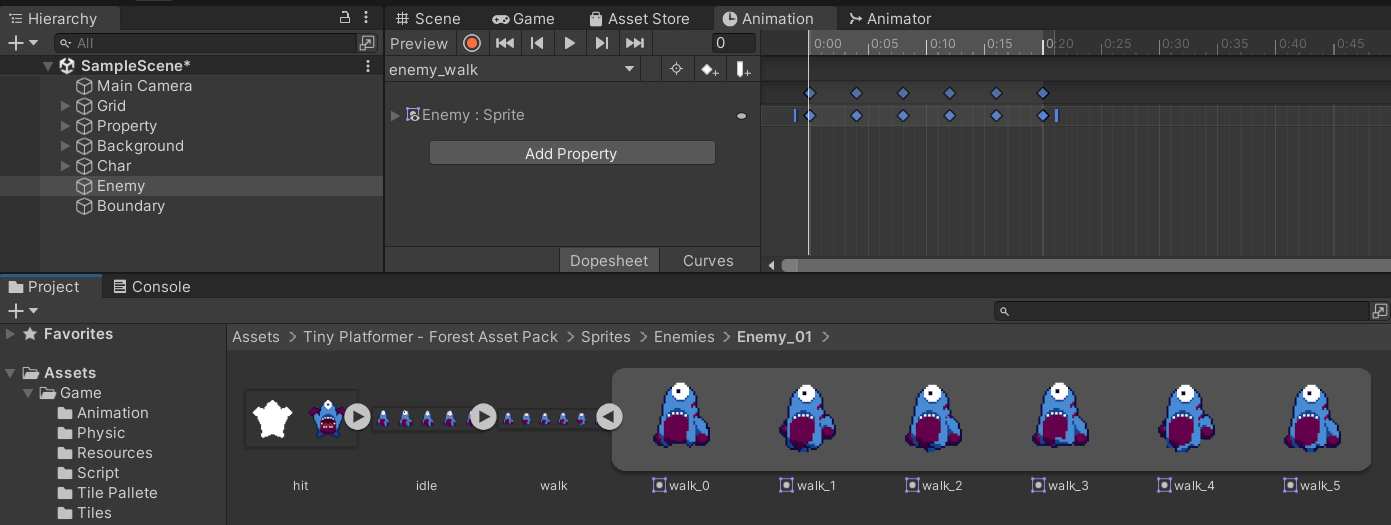
### 10.12 Komponen *Enemy*

1. Buat hierarki *Boundary* dan tambahkan komponen *Box Collider* 2D. Atur konfigurasinya seperti gambar 10.13



### 10.13 Menambah Hierarki *Boundary*

1. Pada hierarki *Enemy,* tambahkan animasi enemy\_walk. *Drag drop* aset *walking*. Atur rentang animasi hingga 0:20

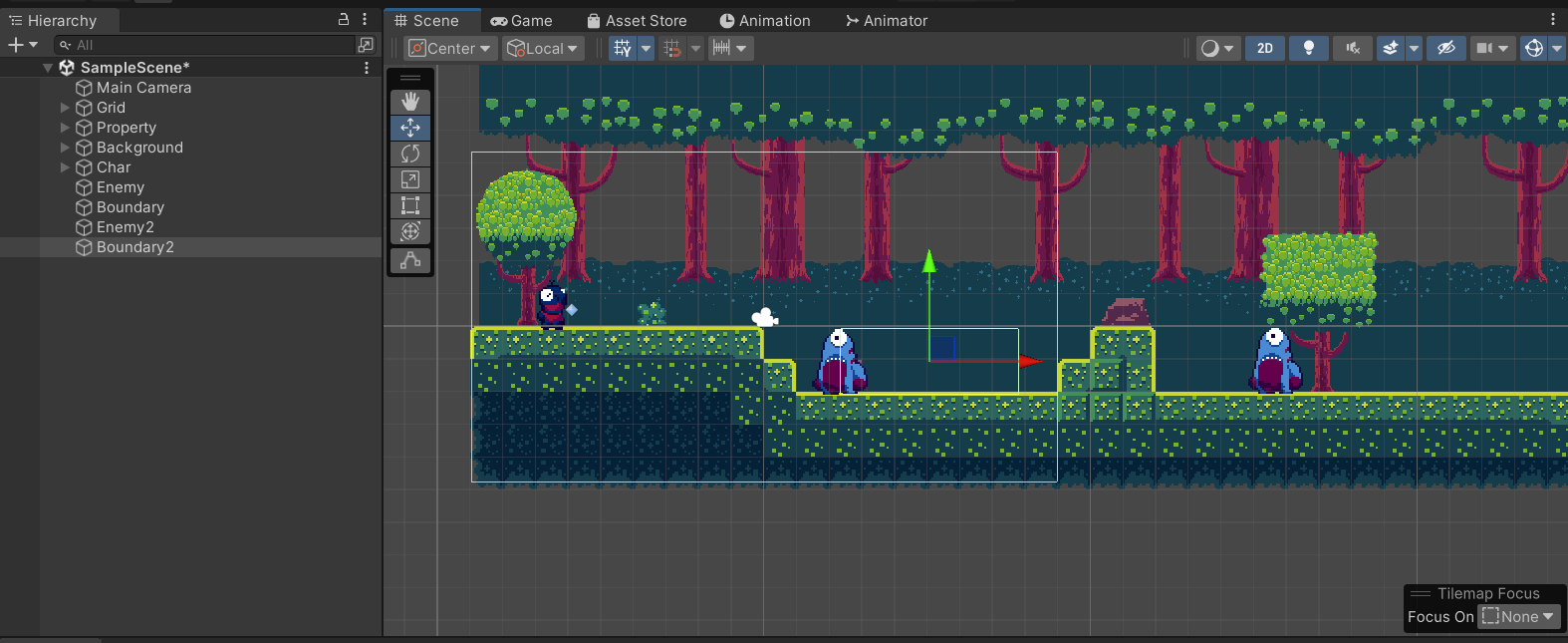


### 10.14 Menambahkan Animasi Musuh

1. Buat kodek *Enemy\_behavior* di dalam direktori Script dan tuliskan kodeknya. Kemudian tambahkan ke komponen *Enemy*

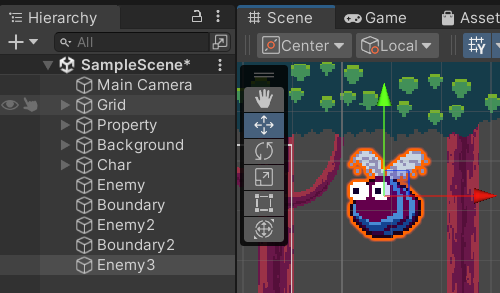
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_behavior : MonoBehaviour{      [SerializeField] float moveSpeed = 1f;      Rigidbody2D rb;      void Start(){          rb = GetComponent<Rigidbody2D>();}      void Update(){          if (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. Duplikat hierarki *Enemy* dan *Boundary*. Posisikan agak berjauhan



### 10.15 Menduplikat *Enemy* dan *Boundary*

1. **Enemy AI**
2. Tambahkan musuh dari aset Enemy\_03 dan beri nama *Enemy3*

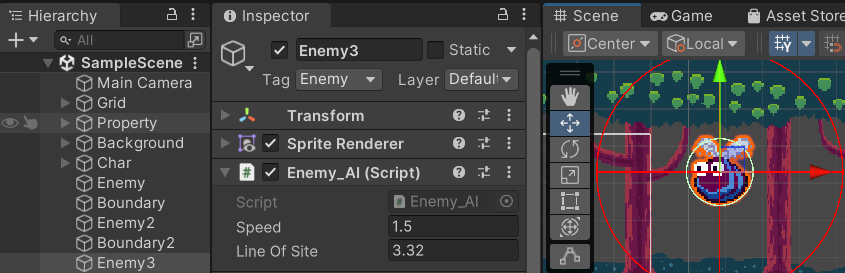


### 10.16 Menambahkan *Enemy3*

1. Buat kodek Enemy\_AIdi dalam direktori Script dan tuliskan kodeknya

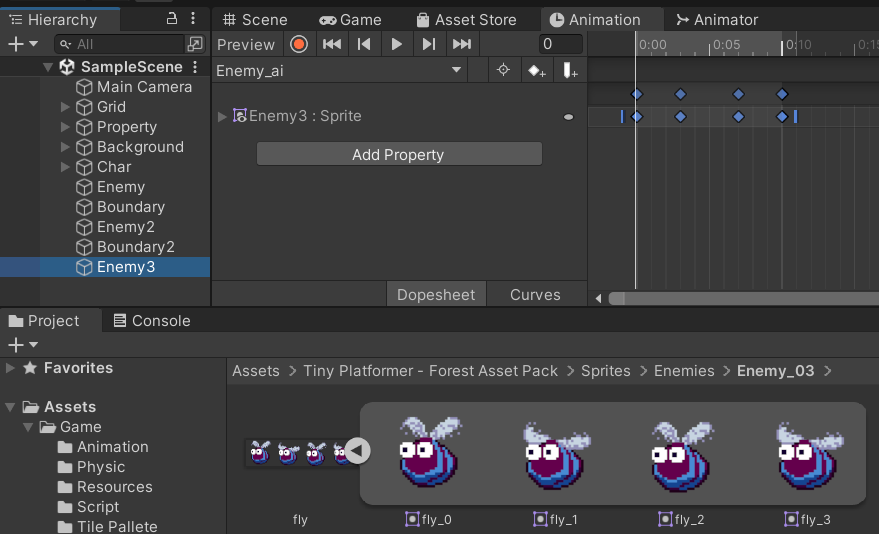
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Enemy\_AI : MonoBehaviour{      public float speed;      public float lineOfSite;      private Transform player;      private Vector2 initialPosition;      private bool facingRight = true;      void Start(){  player = GameObject.FindGameObjectWithTag("Player").transform;  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);              FacePlayer();          }else{              transform.position = Vector2.MoveTowards(transform.position, initialPosition, speed \* Time.deltaTime);              FaceInitialPosition();}}      private void FacePlayer(){          if (player.position.x < transform.position.x && !facingRight){              Flip();          }else if (player.position.x > transform.position.x && facingRight){              Flip();}}      private void FaceInitialPosition(){          if (initialPosition.x > transform.position.x && !facingRight){              Flip();          }else if (initialPosition.x < transform.position.x && facingRight){              Flip();}}      private void Flip(){          facingRight = !facingRight;          Vector3 theScale = transform.localScale;          theScale.x \*= -1;          transform.localScale = theScale;}      private void OnDrawGizmosSelected(){          Gizmos.color = Color.red;          Gizmos.DrawWireSphere(transform.position, lineOfSite);}} |

1. Tambahkan komponen kodek Enemy\_AI dan *Circle Collider*. Atur konfigurasi seperti gambar 10.17



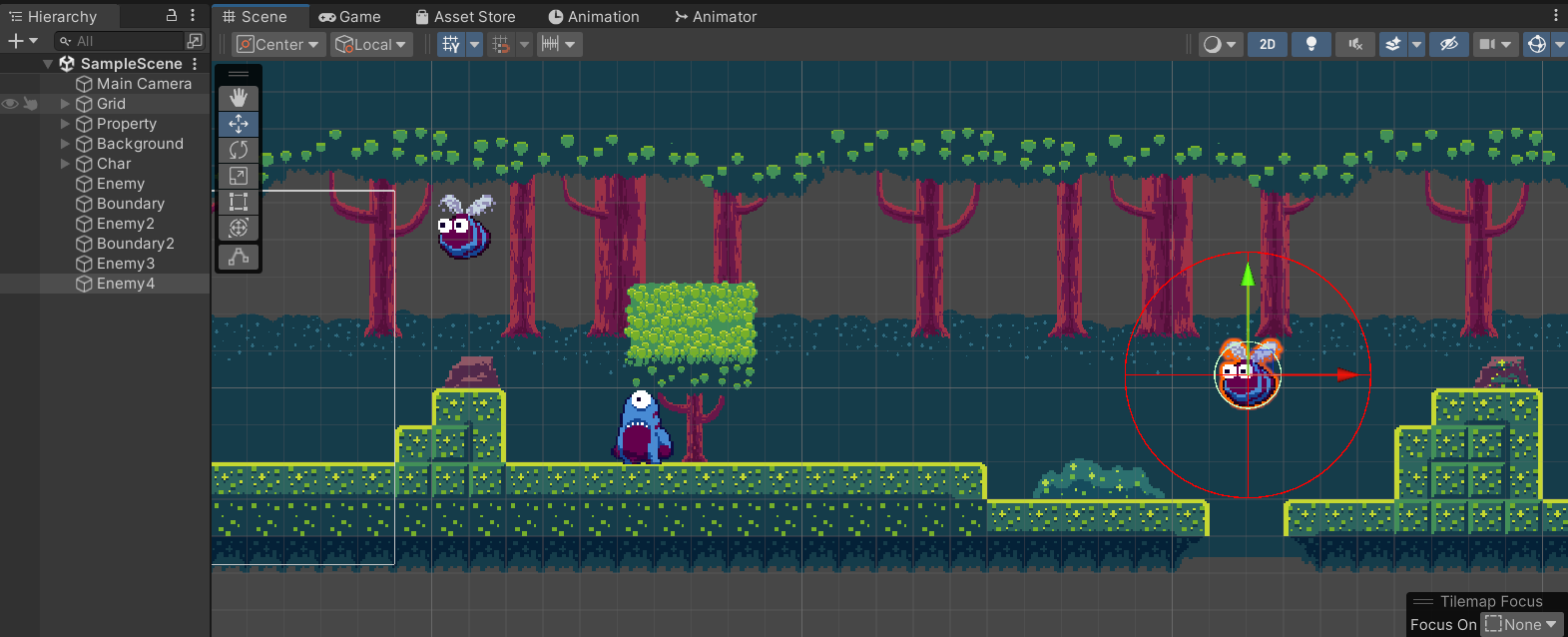
### 10.17 Menambahkan Komponen *Enemy3*

1. Pada hierarki *Enemy3,* tambahkan animasi enemy\_ai. *Drag drop* aset *fly*. Atur rentang animasi hingga 0:10



### 10.18 Menambahkan Animasi *Enemy3*

1. Duplikat *Enemy3* dan beri jarak agak berjauhan



### 10.19 Menduplikat *Enemy3*

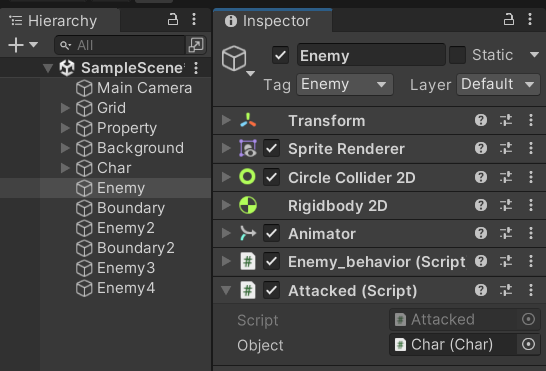
1. **Respawn**
2. Tambahkan beberapa kodek pada berkas Char.cs agar Char dapat kembali ke titik awal apabila *health point* habis

|  |
| --- |
| public int HP; // di dalam class Char  public bool play\_again;  [SerializeField] Vector3 respawn\_loc;  void playagain(){          if(play\_again == true){              HP = 3;              transform.position = respawn\_loc;              play\_again = false;}}  // di dalam void awake()  respawn\_loc = transform.position;  // di dalam void Update()  if(HP < 0){              playagain();}  if(transform.position.y < -10){  play\_again = true;  playagain();  } |

1. Buat kodek *Attacked* di dalam direktori Script dan tuliskan kodeknya

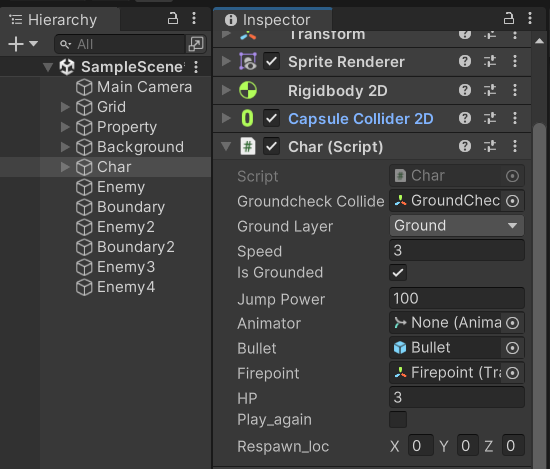
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Attacked : MonoBehaviour{      [SerializeField] private Char Object;      void Start(){          if (Object == null){              Object = GameObject.FindWithTag("Player").GetComponent<Char>();}}      void OnTriggerEnter2D(Collider2D other){          if (other.CompareTag("Player")){              Object.HP--;  Object.animator.SetBool("Attacked", true);              if (Object.HP <= 0){                  Object.play\_again = true;}          } else {         Object.animator.SetBool("Attacked", false);}      }} |

1. Tambahkan komponen *Attacked* ke semua musuh dan atur *Object* ke Char agar HP Char berkurang jika menyentuh musuh



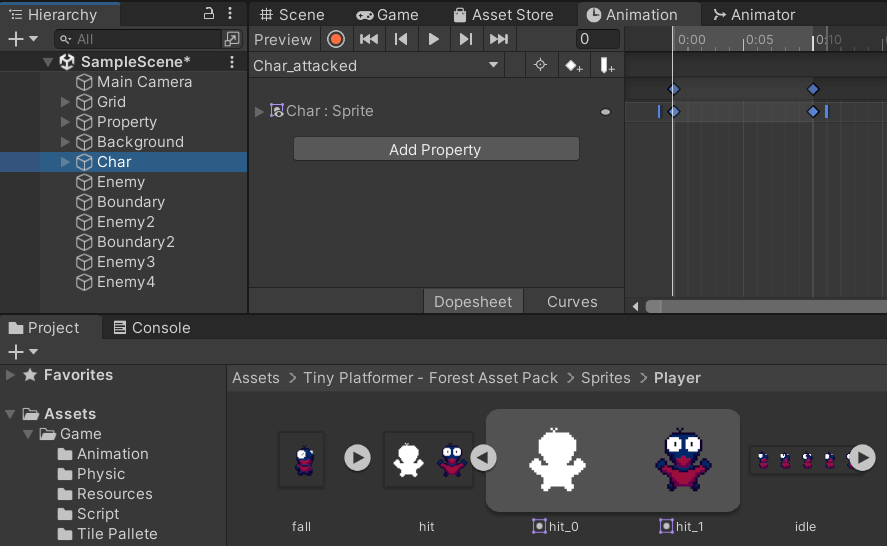
### 10.20 Menambah Komponen Attacked ke Semua Enemy

1. Atur *health point* Char menjadi 3



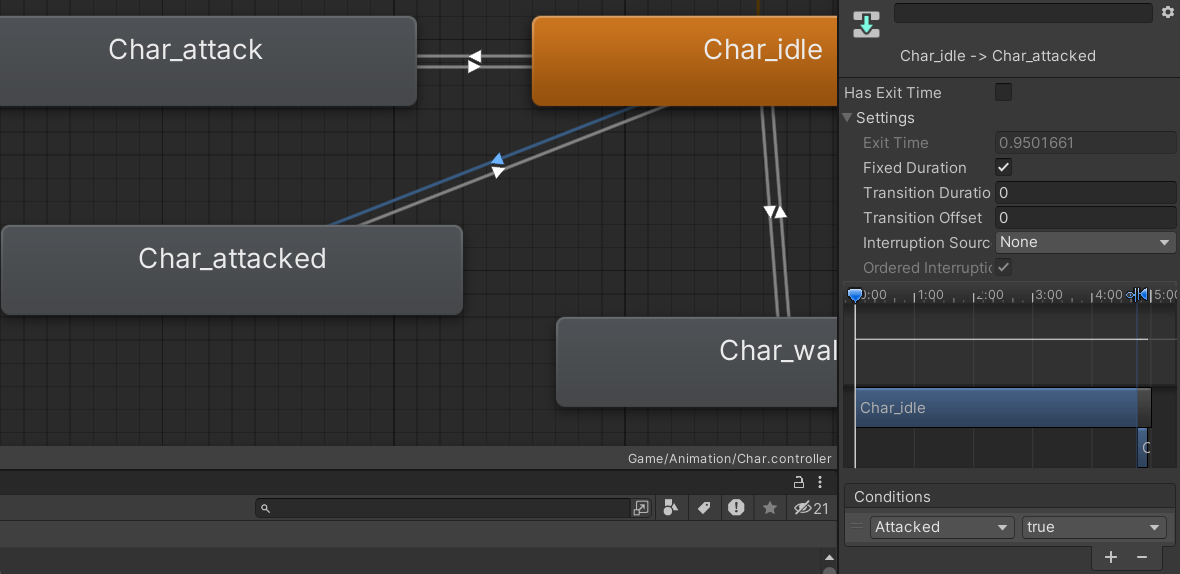
### 10.21 Mengatur HP Char

1. Buat *clip* baru dengan nama ‘char\_attacked’ dan *drag drop* aset *hit* ke halaman kerja *Animation.* Beri rentang hingga 0:10



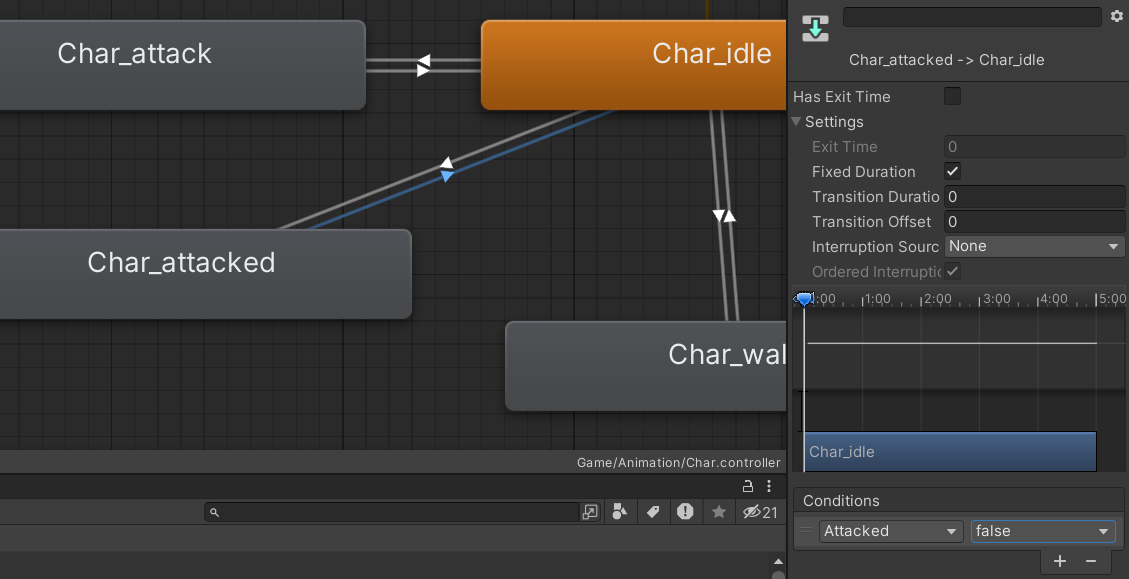
### 10.22 Membuat Animasi Char Diserang

1. Buka halaman kerja *Animator* dan buat variabel ‘Attacked’ dengan tipe data *bool* di menu *Parameter.* Kemudian hubungkan char\_idle ke char\_attacked dengan konfigurasi seperti gambar 10.23



### 10.23 Menghubungkan char\_idle ke char\_attacked

1. Hubungkan char\_attacked ke char\_idle dengan konfigurasi seperti gambar 10.24



### 10.24 Menghubungkan char\_attacked ke char\_idle

1. **Link Github Pengumpulan**

https://github.com/rizkyiman/praktikumanimasigame/tree/main/Bab%2010

1. **Kuis**

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

Analisis

Kodek di atas diperbaiki pada *void* *PerformMeleeAttack*(). Pertama, tipe variabel *attackRange* diubah dari *int* menjadi *float* untuk mencerminkan penggunaannya sebagai nilai jarak serangan. Kedua, terdapat kesalahan ketik pada *InputGetButtonDown* diperbaiki menjadi *Input.GetButtonDown*, dan *attacDamage* diperbaiki menjadi *attackDamage*. Kemudian, penambahan tag *enemyTag* memungkinkan identifikasi musuh melalui *tak* dan memastikan hanya musuh yang terkena serangan. Dalam *void* *PerformMeleeAttack*, ditambahkan pemeriksaan untuk memastikan bahwa obyek yang terkena *raycast* memiliki komponen *Health* yang berfungsi mengurangi *health* musuh. Dengan perubahan ini, kodek mampu mengurangi *health* musuh tanpa memerlukan kodek tambahan terpisah untuk *EnemyHealth.*