# PRAKTIKUM ANIMASI DAN GAME PERTEMUAN :10 Enemy AI & Attack

|  |  |  |
| --- | --- | --- |
| **NIM** | **:** | 1918039 |
| **NAMA** | **:** | M.Sofian Attasauri |
| **MATERI** | **:** | Enemy AI & Attack |
| **KELAS** | **:** | D |
| **TUGAS** | **:** | Enemy AI & Attack |

## Tujuan

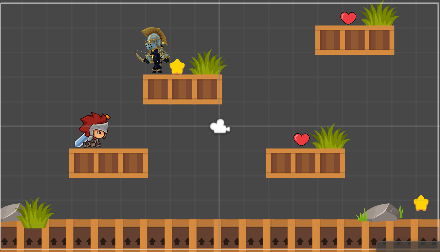
* + 1. Praktikan dapat mengetahui *Game* 2D dan *Game* 3D.
    2. Praktikan dapat mengetahui jenis *Game.*
    3. Praktikan dapat menerapkan *Game* pada Unity.

## Alat dan Bahan

* + 1. Laptop/pc.
    2. Modul Praktikum Animasi dan *Game* 2024.
    3. Unity 2017.

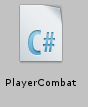
## Langkah-Langkah Membuat Tugas

1. **Membuat Mekanisme Serang**
2. Buat file *projek Unity* Tugas 9



### Buka projek tugas 9

1. Buat File *Script* dalam folder *Script* dengan nama” *PlayerCombat*.cs”

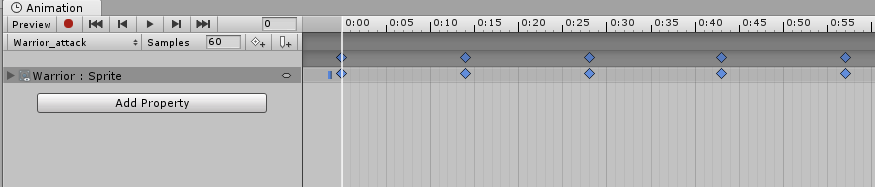


### Membuat *Script* *PlayerCombat*

1. Tambahkan *Source code* berikut dalam folder script *PlayerCombat*.

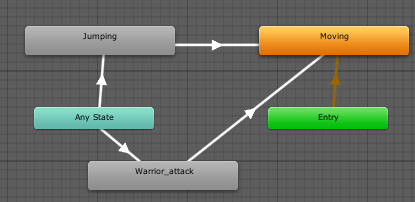
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class PlayerCombat : MonoBehaviour  {  public Animator animator;  public Transform attackPoint;  public LayerMask enemyLayers;  public float attackRange = 0.5f;  public int attackDamage = 35;  // Update is called once per frame  void Update()  {  if (Input.GetKeyDown(KeyCode.C))  {  Attack();  }  }  void Attack()  {  //play attck animation  animator.SetTrigger("Attack");  // detect enemy in range of attck  Collider2D[] hitEnemies =  Physics2D.OverlapCircleAll(attackPoint.position, attackRange,  enemyLayers);  //damage ke musuh  foreach (Collider2D enemy in hitEnemies)  {  enemy.GetComponent<EnemyHealth>().TakeDamage(attackDamage);  }  }  private void OnDrawGizmosSelected()  {  if (attackPoint == null)  return;  Gizmos.DrawWireSphere(attackPoint.position, attackRange);  }  } |

1. Buat sebuah *clip* animasi baru di tab *Animation* dan isikan nama menjadi “*Warrior*\_*attack*.anim” lalu tambahkan *Astroattack* dan atur timenya sampai 1:10



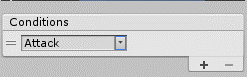
### Menambahkan *Warrior* *Attack*

1. Buatlah sebuah *transtition* untuk *warrior* *attack*



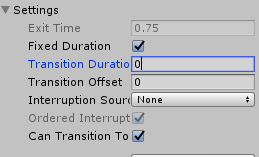
### Membuat *Transition* *Warrior* *Attack*

1. Klik panah yang mengarah ke karaket\_*attack*, pergi ke *Inspector* dan tambahkan *condition* dan pilih *condition* Attack



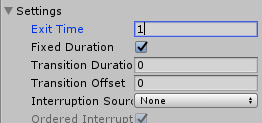
### Mengubah *Condition* *Attack*

1. Klik *setting* dan ubah *Transition Duration* menjadi 0, hilangkan centang *Has Exit Time*, untuk yang lain sesuaikan saja



### Mengatur *Transition Duration*

1. Klik panah yang mengarah dari *Warrior*\_*attack* ke *Moving*, klik *setting*, ubah *Exit Time* menjadi 1 dan *Transition Duration* menjadi 0



### Mensetting *Warrior* *Attack*

1. Pergi ke *Hierarchy*, klik kanan *warrior* dan pilih *Create Empty*, dan ubah Namanya menjadi *AttackPoint*



### Membuat *Attack* *Point*

1. Klik game object *warrior*, pergi ke Inspector dan ubah komponen Player Combat seperti berikut



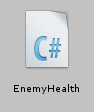
### Mengubah komponen *player combat*

1. klik *game* *object* *Enemy*, klik *Layer* *Default* dan pilih *Add* *Layer*, isikan “*Enemy*” pada user Layer 9



### Membuat layer *Enemy*

1. Buat sebuah *file Script* beri nama *EnemyHealth*.cs, simpan didalam folder *script*

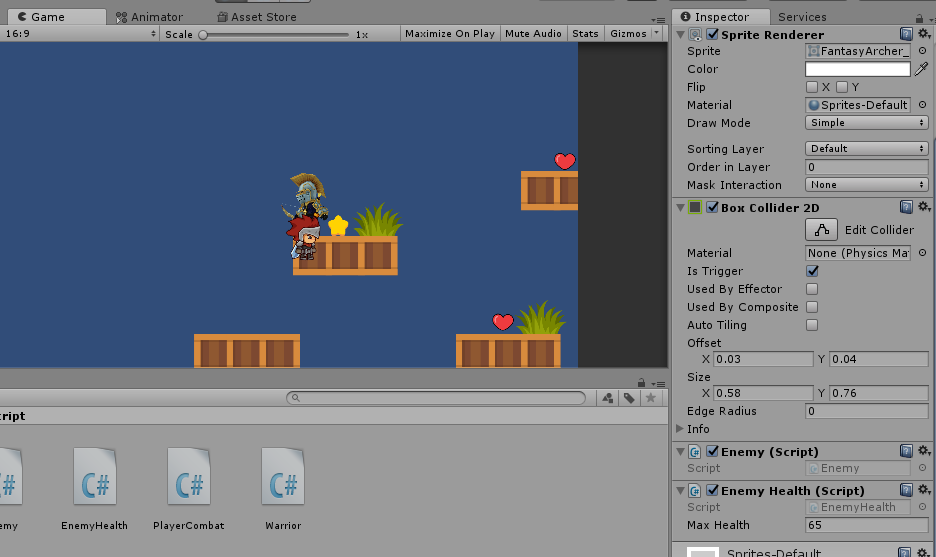


### Membuat Script *EnemyHealt*

1. Tambahkan *Source Code* seperti dibawah ini

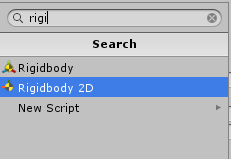
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class EnemyHealth : MonoBehaviour  {  public int maxHealth = 100;  // Use this for initialization  void Start()  {  }  // Update is called once per frame  void Update()  {  }  public void TakeDamage(int damage)  {  maxHealth -= damage;  //hurt anim  if (maxHealth <= 0)  {  Wafat();  }  }  void Wafat()  {  Destroy(gameObject);  }  } |

1. Jika *diplay* lalu menekan *keybord* C maka *warrior* akan menyerang dan apibila serangan terkena ke *enemy* maka *maxhealtnya* akan berkurang



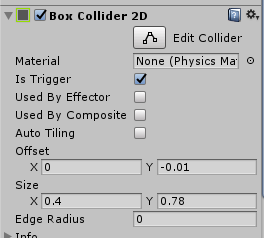
### Tampilan Serang

1. **Membuat Enemy AI**
2. Klik *game object Enemy*, kemudian *Add Componenet Rigidbody* *2D*



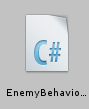
### Menambahkan *Rigibody 2D*

1. Atur sedikit ukuran *box collider* Enemy agar sedikit menyentuh dengan *tilemap* dan buat agak sedikit ramping



### Mengatur *Box Collider*

1. Buat sebuah *file script* baru di dalam folder *script* dan namakan *EnemyBehaviour*.*cs*.



### Membuat *script* *EnemyBehaviour*

1. Isikan *source code* berikut di dalam *file script* *EnemyBehaviour*,*cs*

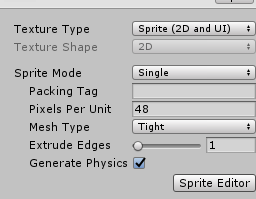
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class EnemyBehaviour : MonoBehaviour  {  [SerializeField]  float moveSpeed = 1f;  Rigidbody2D rb;  // Use this for initialization  void Start()  {  rb = GetComponent<Rigidbody2D>();  }  // Update is called once per frame  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(-(Mathf.Sign(rb.velocity.x)),  transform.localScale.y);  }  } |

1. Klik game *object Enemy* dan ubah *Move Speed* menjadi 2 atau bisa disesuaikan



### Mengatur *Move Speed*

1. Cari *Flying eye* di folder *Assets* *Monster* Ubah nilai *Pixels* Per Unit menjadi 48



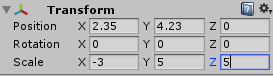
### Mengatur *Pixels* per unit

1. Ubah Namanya di *Hierarchy* menjadi *Enemy\_Eye*



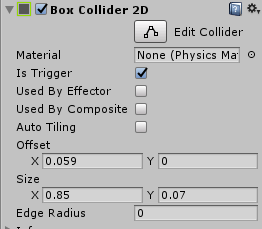
### Mengubah nama *Enemy\_Eye*

1. Klik *game object Enemy\_Eye*, ubah *Layernya* menjadi *Enemy* kemudian sesuaikan nilai *scalenya*



### Mengatur nilai *scale*

1. Tetap di *Inspector Enemy\_Eye*, tambahkan sebuah komponen bernama *Box collider 2D* dan centang *Is Trigger* dan Atur juga ukuran *box collidernya*



### Mengatur *Box Collider 2D*

1. Buat sebuah *file script* bernama *FlyingEye*.*cs* dan simpan dalam folder *Script*

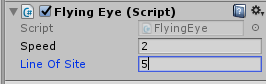


### Membuat *script* *FlyingEye*

1. Isikan *source code* dibawah ini

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class FlyingEye : MonoBehaviour {  public float speed;  public float lineOfSite;  private Transform player;  private Vector2 currentpos;  Warrior wr;  // Use this for initialization  void Start()  {  player = GameObject.FindGameObjectWithTag("Player").transform;  currentpos = GetComponent<Transform>().position;  wr = GameObject.Find("Warrior").GetComponent<Warrior>();  }  // Update is called once per frame  void Update()  {  float jarakdariplayer = Vector2.Distance(player.position,  transform.position);  if (jarakdariplayer < lineOfSite)  {  transform.position =  Vector2.MoveTowards(this.transform.position, player.position, speed \*  Time.deltaTime);  }  else  {  transform.position = Vector2.MoveTowards(transform.position,  currentpos, speed \* Time.deltaTime);  }  }  void OnTriggerEnter2D(Collider2D other)  {  if (other.transform.tag == "Player")  {  wr.nyawa--;  }  if (wr.nyawa < 0)  {  wr.play\_again = true;  }  }  private void OnDrawGizmosSelected()  {  Gizmos.color = Color.red;  Gizmos.DrawWireSphere(transform.position, lineOfSite);  }  } |

1. klik game *object Enemy\_Eye* dan ubah nilai *speed* dan *Line of Site* sesuai keinginan



### Mengatur *speed flying eye*

1. Jika di *play* apabila *warrior* memasuki wilayah *Line of Site Enemy*, maka *Flying eye* akan mengikutinya,



### Tampilan *Running*

## Kesimpulan

* + 1. Dalam Video *Game*, kecerdasan buatan digunakan untuk membuat perilaku cerdas yang biasanya terletak pada *non*-player *characters* ( *NPCs*), dan seringnya mensimulasikan seperti kecerdasan manusia.

|  |  |
| --- | --- |
| **Nilai** | **Asisten Laboratorium** |
|  | (M.Rafi Faddilani)  2118144 |

* + 1. *layer-experience* *modelling*: memahami kemampuan dan kondisi emosional pemain, agar menyesuaikan *game* dengan benar
    2. *Procedural-content* *generation*: membuat elemen dari sebuah lingkungan game seperti kondisi lingkungan, tingkatan, dan bahkan music dengan secara otomatis.