**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Grafik Komputer 2

Kelas : 3IA11

Praktikum ke- : 5

Tanggal : 9/05/2025

Materi : Logika Game

NPM : 51422161

Nama : Muhammad Tarmidzi Bariq

Ketua Asisten : Aisyah

Paraf Asisten :

Nama Asisten :

1. Cahyaningrum Respati
2. Muhammad Rafi Ilham
3. Muhammad Insan Kamil
4. Intan Alifia Ramadhan

Jumlah Lembar : 9 Lembar



**LABORATORIUM INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2025**

**Jelaskan scripit c# yang kalian buat pada saat mengerjakan act!**

**Camera Movement Mouse**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraMouvementMouse : MonoBehaviour

{

public Transform target;

public float distance = 5.0f;

public float xSpeed = 700.0f;

public float ySpeed = 700.0f;

float x = 0.0f;

float y = 0.0f;

// Start is called before the first frame update

void Start()

{

Vector3 angles = transform.eulerAngles;

x = angles.y;

y = angles.x;

Cursor.lockState = CursorLockMode.Locked;

}

// Update is called once per frame

void LateUpdate()

{

x += Input.GetAxis("Mouse X") \* xSpeed \* Time.deltaTime;

y -= Input.GetAxis("Mouse Y") \* ySpeed \* Time.deltaTime;

y = Mathf.Clamp(y, -80, 80);

Quaternion rotation = Quaternion.Euler(y, x, 0);

Vector3 position = rotation \* new Vector3(0.0f, 0.0f, -distance) + target.position;

transform.rotation = rotation;

transform.position = position;

}

}

public Transform target;

public float distance = 5.0f;

public float xSpeed = 700.0f;

public float ySpeed = 700.0f;

float x = 0.0f;

float y = 0.0f;

* target: objek yang menjadi pusat rotasi kamera (biasanya karakter).
* distance: jarak kamera dari target.
* xSpeed dan ySpeed: kecepatan rotasi kamera berdasarkan input mouse.
* x dan y: menyimpan sudut rotasi horizontal dan vertikal kamera.

void Start()

{

Vector3 angles = transform.eulerAngles;

x = angles.y;

y = angles.x;

Cursor.lockState = CursorLockMode.Locked;

}

* Mengambil rotasi awal kamera (transform.eulerAngles) untuk disimpan ke x dan y.
* Cursor.lockState = CursorLockMode.Locked; menyembunyikan dan mengunci kursor di tengah layar agar pemain bisa menggerakkan kamera tanpa batasan layar.

void LateUpdate()

{

x += Input.GetAxis("Mouse X") \* xSpeed \* Time.deltaTime;

y -= Input.GetAxis("Mouse Y") \* ySpeed \* Time.deltaTime;

y = Mathf.Clamp(y, -80, 80);

Quaternion rotation = Quaternion.Euler(y, x, 0);

Vector3 position = rotation \* new Vector3(0.0f, 0.0f, -distance) + target.position;

transform.rotation = rotation;

transform.position = position;

}

* Input Mouse:
  + Mouse X menggerakkan kamera horizontal (rotasi Y).
  + Mouse Y menggerakkan kamera vertikal (rotasi X), tetapi arah negatif agar naik ke atas saat mouse ditarik ke atas.
* Clamp Vertikal:
  + y = Mathf.Clamp(y, -80, 80); membatasi sudut vertikal supaya kamera tidak berputar 360° (mencegah kamera terbalik).
* Rotasi dan Posisi Kamera:
  + Quaternion.Euler(y, x, 0): membuat rotasi berdasarkan input.
  + rotation \* new Vector3(0, 0, -distance): memindahkan kamera sejauh distance dari target.
  + + target.position: membuat kamera selalu fokus ke target.
* Penerapan:
  + transform.rotation dan transform.position diterapkan ke kamera.

Script ini memungkinkan kamera untuk berputar mengelilingi karakter berdasarkan input mouse dengan jarak tetap. Cocok digunakan dalam game third-person seperti RPG atau petualangan.

**Player Movement**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class PlayerMovement : MonoBehaviour

{

private Rigidbody rb;

private Animator animator;

private bool isGrounded = true;

private float score = 0f;

public float moveSpeed;

public float jumpForce;

public Text scoreText;

void Start()

{

rb = GetComponent<Rigidbody>();

animator = GetComponent<Animator>();

//Pert 7

scoreText.text = "Score: " + score;

}

void FixedUpdate()

{

//arah belok -1 ke kiri 1 ke kanan

float moveX = 0;

//arah maju -1 mundur 1 maju

float moveZ = 0;

//membuat kamera menyesuaikan arah karakter

Camera cam = FindObjectOfType<Camera>();

Vector3 camForward = cam.transform.forward;

Vector3 camRight = cam.transform.right;

camForward.y = 0f;

camRight.y = 0f;

camForward.Normalize();

camRight.Normalize();

if (Input.GetKey(KeyCode.A))

{

moveX = -1;

}

else if (Input.GetKey(KeyCode.D))

{

moveX = 1;

}

if (Input.GetKey(KeyCode.W))

{

moveZ = 1;

}

else if (Input.GetKey(KeyCode.S))

{

moveZ = -1;

}

if (Input.GetKey(KeyCode.Space) && isGrounded)

{

rb.AddForce(Vector3.up \* jumpForce, ForceMode.Impulse);

animator.SetTrigger("jumpTrigger");

isGrounded = false;

}

//menormalisasi arah kamera dengan arah karakter

Vector3 moveDirection = (camRight \* moveX + camForward \* moveZ).normalized;

if (moveDirection != Vector3.zero)

{

//merotasi karakter ketika berbelok

transform.rotation = Quaternion.LookRotation(moveDirection);

}

//menggerakkan karakter

Vector3 movement = moveDirection \* moveSpeed \* Time.fixedDeltaTime;

rb.MovePosition(rb.position + movement);

//mengubah kondisi isMoving supaya animasi lari bisa auto play

bool isMoving = moveX != 0 || moveZ != 0;

animator.SetBool("isMoving", isMoving);

}

private void OnCollisionEnter(Collision collision)

{

//ketika karakter menyentuh object dengan collider yang mempunyai tag Ground

if (collision.gameObject.CompareTag("Ground"))

{

//ubah kondisi isGrounded menjadi true supaya karakter bisa melompat

isGrounded = true;

}

}

//Pert7

private void OnTriggerEnter(Collider other)

{

//ketika karakter menyentuh object dengan collider yang mempunyai tag Point dan collidernya isTrigger = on

if (other.gameObject.CompareTag("Point"))

{

//menghilangkan object yang ditabrak (object pointnya)

Destroy(other.gameObject);

//menambahkan skor setiap menabrak object point

addScore();

}

}

//Pert 7

void addScore()

{

score++;

scoreText.text = "Score: " + score;

}

}

private Rigidbody rb;

private Animator animator;

private bool isGrounded = true;

private float score = 0f;

public float moveSpeed;

public float jumpForce;

public Text scoreText;

* rb: komponen fisika (Rigidbody) untuk pergerakan.
* animator: mengontrol animasi karakter.
* isGrounded: mengecek apakah karakter sedang menyentuh tanah.
* score: menyimpan nilai poin.
* moveSpeed dan jumpForce: mengatur kecepatan gerak dan kekuatan lompatan.
* scoreText: UI teks untuk menampilkan skor.

rb = GetComponent<Rigidbody>();

animator = GetComponent<Animator>();

scoreText.text = "Score: " + score;

* Mengambil komponen Rigidbody dan Animator.
* Inisialisasi skor ke UI.

if (Input.GetKey(KeyCode.A)) { moveX = -1; }

if (Input.GetKey(KeyCode.D)) { moveX = 1; }

if (Input.GetKey(KeyCode.W)) { moveZ = 1; }

if (Input.GetKey(KeyCode.S)) { moveZ = -1; }

* Input Keyboard

Camera cam = FindObjectOfType<Camera>();

Vector3 camForward = cam.transform.forward;

Vector3 camRight = cam.transform.right;

* Kamera sebagai referensi gerakan
* Supaya arah gerakan menyesuaikan arah kamera (tidak fixed ke world axis).
* Komponen y diset ke 0 agar hanya gerakan horizontal (tanpa mengikut rotasi vertikal kamera).

if (Input.GetKey(KeyCode.Space) && isGrounded)

{

rb.AddForce(Vector3.up \* jumpForce, ForceMode.Impulse);

animator.SetTrigger("jumpTrigger");

isGrounded = false;

}

* Karakter hanya bisa melompat jika sedang di tanah (isGrounded == true).

Vector3 moveDirection = (camRight \* moveX + camForward \* moveZ).normalized;

if (moveDirection != Vector3.zero)

transform.rotation = Quaternion.LookRotation(moveDirection);

* Membuat karakter menghadap ke arah gerak.

Vector3 movement = moveDirection \* moveSpeed \* Time.fixedDeltaTime;

rb.MovePosition(rb.position + movement);

* Gerakan dilakukan dengan MovePosition agar lebih halus dan stabil secara fisika.

bool isMoving = moveX != 0 || moveZ != 0;

animator.SetBool("isMoving", isMoving);

* Jika ada input arah, maka animasi jalan diaktifkan.

if (collision.gameObject.CompareTag("Ground"))

isGrounded = true;

* Karakter bisa lompat lagi setelah menyentuh tanah.

**Jelaskan peran rigidbody dalam script!!**

Rigidbody adalah komponen Unity yang membuat GameObject dapat berinteraksi dengan sistem fisika — seperti gaya gravitasi, tumbukan (collision), dan gaya (force).

rb.MovePosition(rb.position + movement);

* Fungsi ini menggerakkan karakter dengan mempertahankan interaksi fisika (misalnya tetap bisa menabrak benda, kena gravitasi, dsb).
* Lebih stabil dan smooth dibanding langsung ubah posisi (transform.position = ...) karena menghormati sistem Collision.

rb.AddForce(Vector3.up \* jumpForce, ForceMode.Impulse);

* AddForce() menambahkan gaya dorong ke atas ketika pemain menekan space.
* Menggunakan ForceMode.Impulse berarti gaya diberikan seketika (bukan bertahap), cocok untuk lompatan cepat.

private void OnCollisionEnter(Collision collision)

{

if (collision.gameObject.CompareTag("Ground"))

{

isGrounded = true;

}

}

* Rigidbody memungkinkan sistem OnCollisionEnter() mendeteksi tabrakan dengan objek lain.
* Tanpa Rigidbody, metode ini tidak akan dipanggil, karena Unity hanya memproses tabrakan antar objek yang memiliki Collider dan Rigidbody.