LAPORAN PRAKTIKUM PEMOGRAMAN VISUAL
JUNIOR PROGRAMMER PATHWAY

Dosen Pengampu :

Evianita Dwi Fajrianti, S.Tr.T., M.Tr.T., Ph.D

Disusun Oleh :

1. Ridho Maulana Mochtar (5124521001)
2. Linatul Fatimah A. K (5124521005)
3. Sabda Bintang Ramadhani (5124521008)
4. Moch. Ihtizam B Pratama (5124521017)
5. Paulus Fadlee Lanan P (5124521023)

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA DEPARTEMEN TEKNOLOGI MULTIMEDIA KREATIF TEKNOLOGI MULTIMEDIA BROADCASTING KAMPUS LAMONGAN 2025

BAB I
PENDAHULUAN

1.1 LATAR BELAKANG

Pemrograman visual dalam pengembangan game telah menjadi keterampilan fundamental dalam industri multimedia kreatif. Unity adalah salah satu game engine yang paling populer dan banyak digunakan untuk menciptakan pengalaman interaktif, mulai dari game sederhana hingga simulasi kompleks. Untuk menguasai dasar-dasar pengembangan game dan memahami logika di balik interaksi objek, praktikum ini berfokus pada Junior Programmer Pathway yang disediakan oleh Unity Learn. Mission pertama dari pathway ini secara spesifik berfokus pada penguasaan konsep pergerakan objek (Player movement), implementasi fisika melalui komponen Prototype dan cara menangani masukan dari pengguna (Input system).

1.2 TUJUAN PRAKTIKUM

Pada praktikum ini memiliki tujuan:

1. Untuk memahami dasar pemrograman di Unity.
2. Menerapkan logika dasar seperti variabel, perulangan dan interaksi objek dalam permainan.
3. Melatih kemampuan membuat game sederhana yang memiliki sistem kontrol, skor dan tampilan interaktif.

BAB II
DASAR TEORI

Dalam konteks praktikum atau laporan, dasar teori berfungsi untuk menjelaskan konsep-konsep penting yang digunakan dalam pembuatan proyek.
Kalimatmu sudah mencakup tiga teori utama yang memang menjadi pondasi dari prototype Unity tingkat pemula:

2.1 Player Movement (Pergerakan Pemain)

Membahas logika dan skrip untuk menggerakkan karakter di dunia 3D/2D menggunakan Transform, Rigidbody atau CharacterController.

2.2 Physics Components (Komponen Fisika)

Berhubungan dengan collider, rigidbody dan physics materials yang digunakan agar objek bisa berinteraksi secara realistis (jatuh, bertabrakan, memantul, dll).

2.3 Input System (Sistem Masukan)

Mengatur interaksi pemain dengan game melalui keyboard, mouse, atau joystick.
Versi baru Unity menggunakan Input System Package (bukan Input.GetAxis() lagi).

BAB III
METODE PRAKTIKUM

3.1 ALAT DAN BAHAN

a. Laptop/PC (sebagai hardware utama untuk pengembangan).
b. Software Unity (Game Engine tempat pengembangan dilakukan).
c. Modul Unity Learn: Junior Programmer Pathway (sebagai panduan langkah kerja dan materi).

3.2 PROSEDUR KERJA

Langkah-langkah yang dilakukan dalam praktikum, yang berfokus pada Mission 1: Player Movement, Rigidbody, Input system , adalah sebagai berikut:

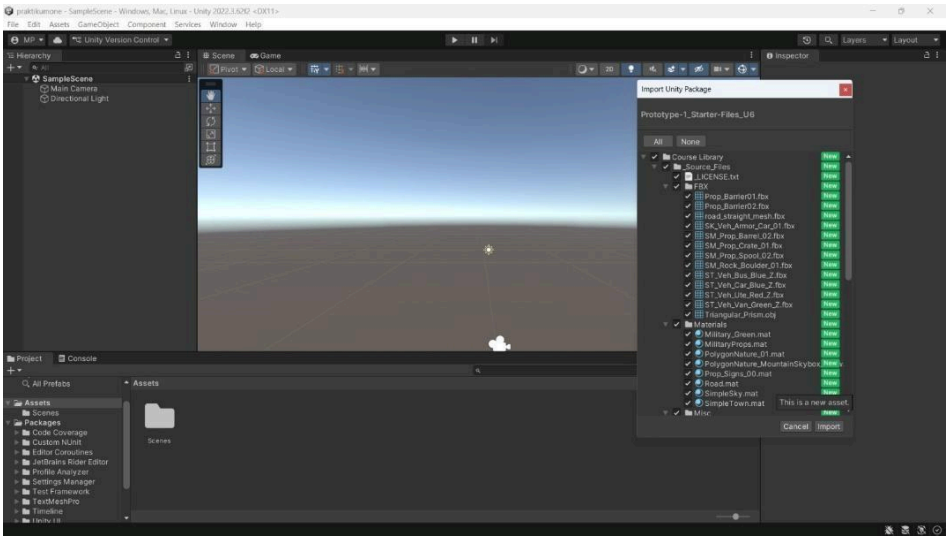
a. Persiapan Proyek: Membuka proyek Unity baru dengan template 3D.
b. Setup Player: Menambahkan GameObject dasar (misalnya kubus atau kapsul) yang akan berfungsi sebagai Player.
c. Implementasi Fisika: Menambahkan komponen Rigidbody ke GameObject Player untuk mengaktifkan kontrol fisika.
d. Pembuatan Skrip: Membuat skrip C# baru.
e. Pengkodean: Menulis kode dalam skrip P untuk menggerakkan player berdasarkan input keyboard. Kode ini akan mendefinisikan variabel kecepatan speed dan menggunakan fungsi Update atau FixedUpdate untuk mengaplikasikan gaya dorong.
f. Pengujian: Menjalankan game mode Play dan menguji hasilnya, memastikan player dapat bergerak sesuai dengan tombol keyboard yang ditekan.

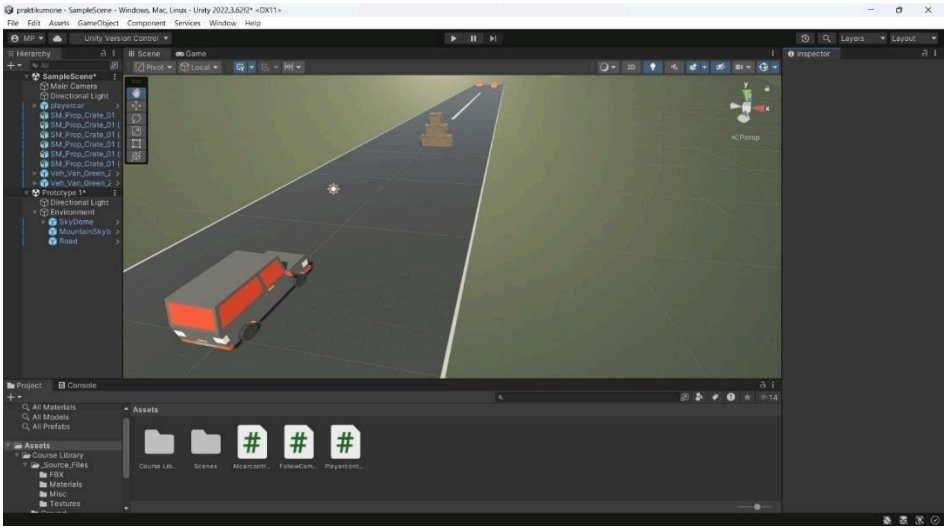
BAB IV
HASIL DAN PEMBAHASAN

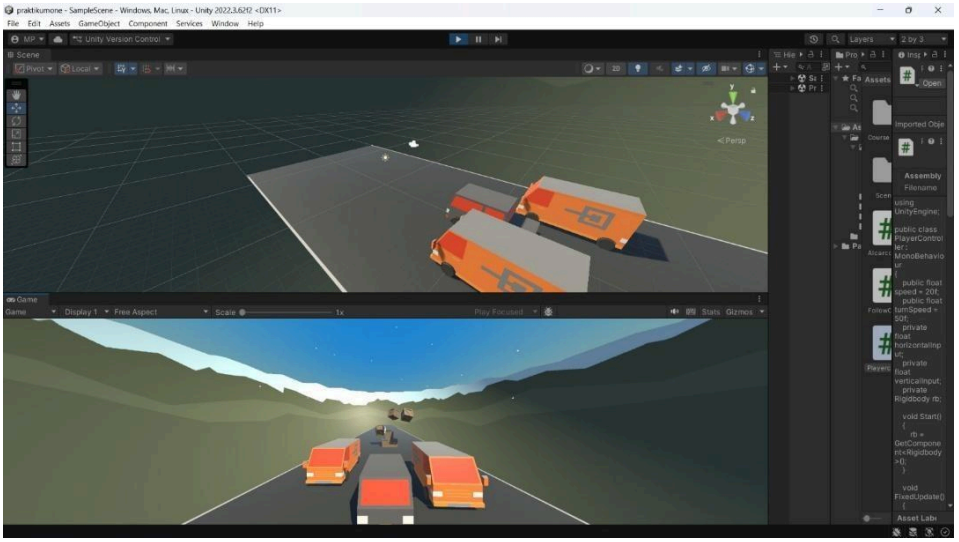
4.1 HASIL PRAKTIKUM

Di praktikum ini, kita membuat proyek Unity dengan template 3D, lalu bikin satu objek utama sebagai player. Objeknya bisa berupa kubus atau kapsul, dan kita kasih komponen Prototype biar bisa bergerak pakai sistem fisika bawaan Unity.

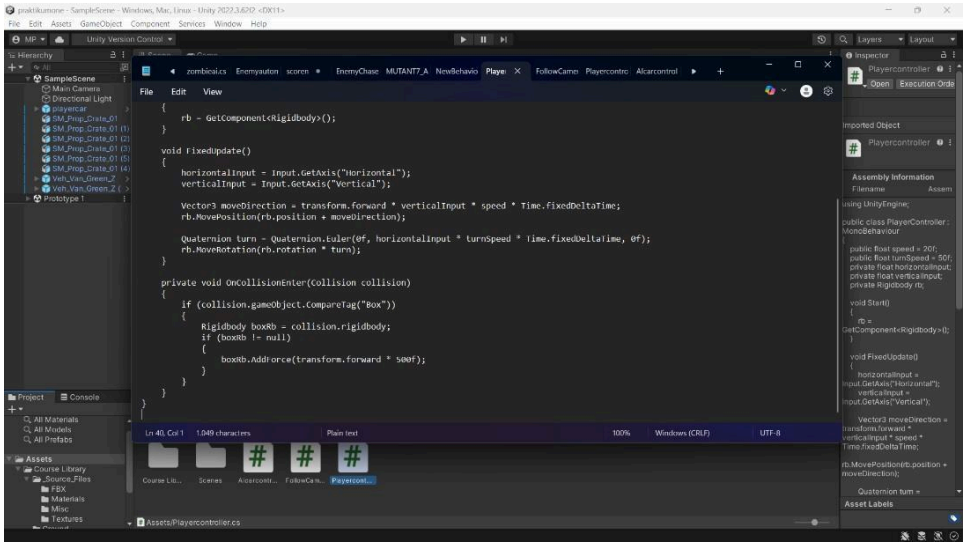
Berikut empelmentasi pengerjaan project prototype:



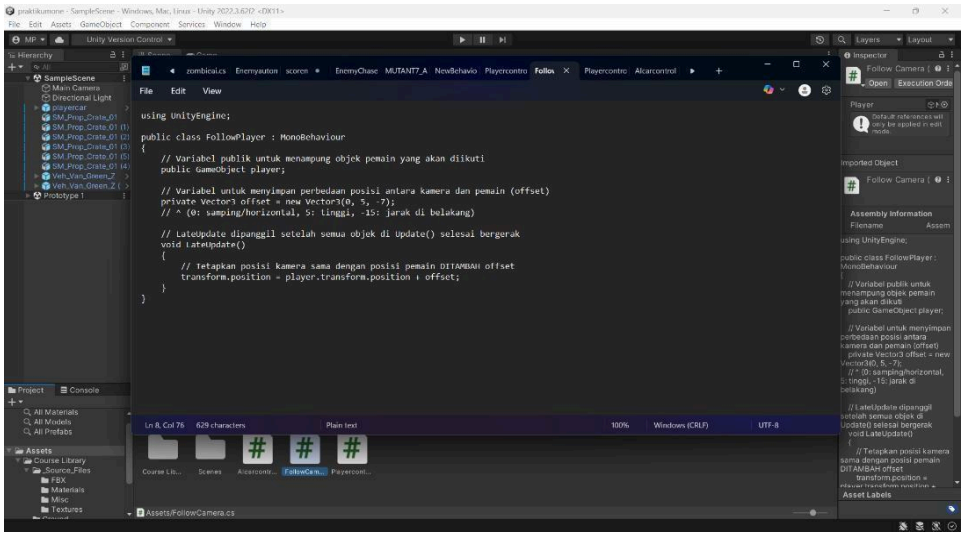




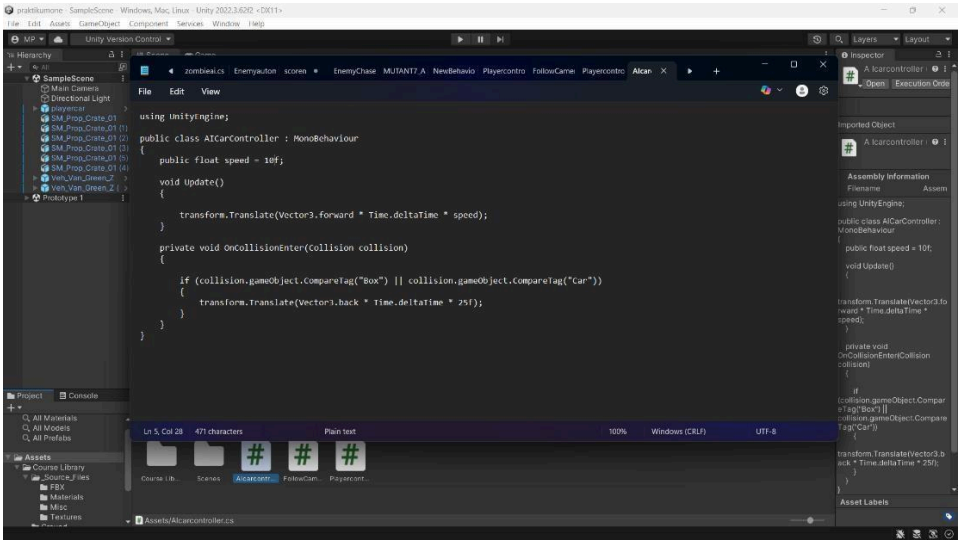
Berikut script dari prototype:



```
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        verticalInput = Input.GetAxis("Vertical");

        Vector3 moveDirection = transform.forward * verticalInput * speed * Time.fixedDeltaTime;
        rb.MovePosition(rb.position + moveDirection);

        Quaternion turn = Quaternion.Euler(0f, horizontalInput * turnSpeed * Time.fixedDeltaTime, 0f);
        rb.MoveRotation(rb.rotation * turn);
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Box"))
        {
            Rigidbody boxRb = collision.rigidbody;
            if (boxRb != null)
            {
                boxRb.AddForce(transform.forward * 500f);
            }
        }
    }
}
```

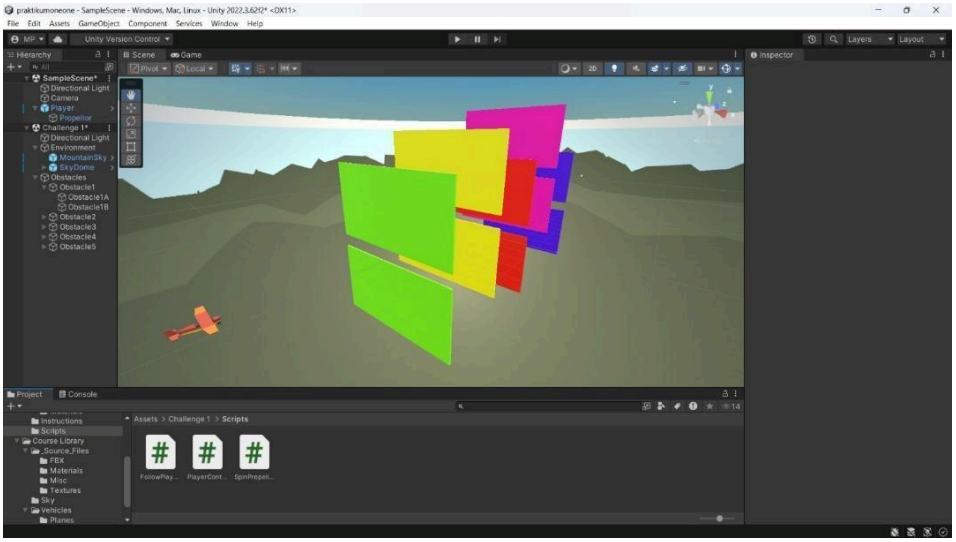


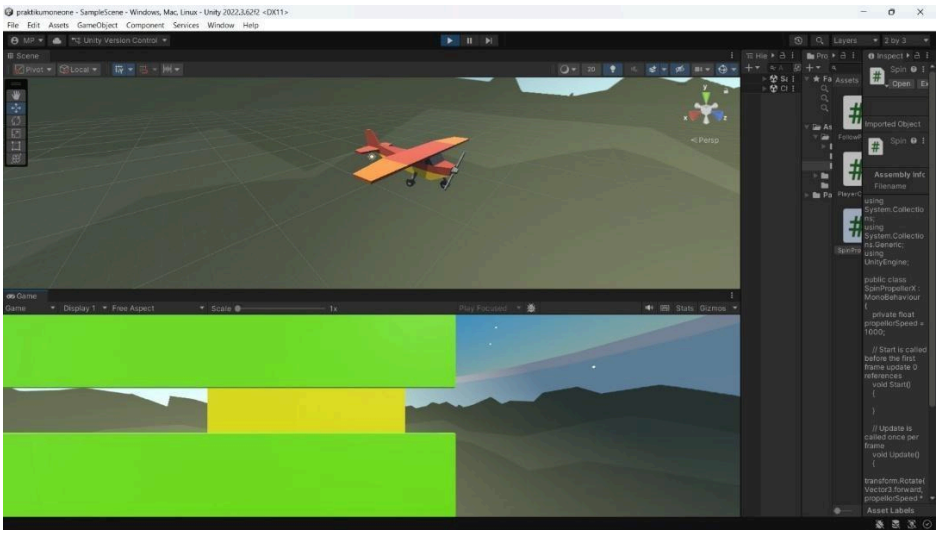
```
using UnityEngine;

public class FollowPlayer : MonoBehaviour
{
    // Variabel publik untuk menampung objek pemain yang akan diikuti
    public GameObject player;

    // Variabel untuk menyimpan perbedaan posisi antara kamera dan pemain (offset)
    private Vector3 offset = new Vector3(0, 5, -7);
    // ^ (0: samping/horizontal, 5: tinggi, -15: jarak di belakang)

    // LateUpdate dipanggil setelah semua objek di Update() selesai bergerak
    void LateUpdate()
    {
        // Tetapkan posisi kamera sama dengan posisi pemain DITAMBAH offset
        transform.position = player.transform.position + offset;
    }
}
```



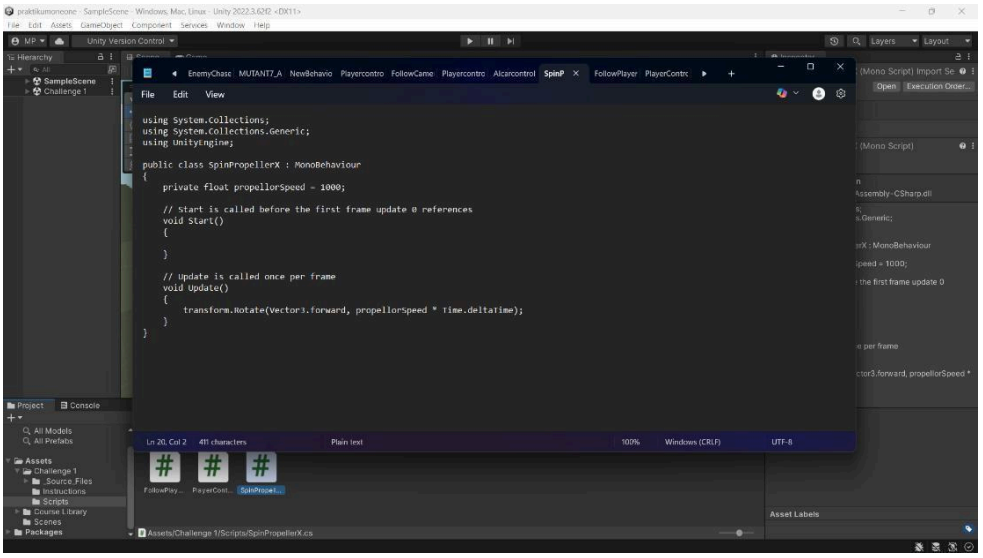
```
using UnityEngine;

public class AICarController : MonoBehaviour
{
    public float speed = 10f;

    void Update()
    {
        transform.Translate(Vector3.forward * Time.deltaTime * speed);
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Box") || collision.gameObject.CompareTag("Car"))
        {
            transform.Translate(Vector3.back * Time.deltaTime * 25f);
        }
    }
}
```

Berikut Implementasi Project challenge 1 (Plane):

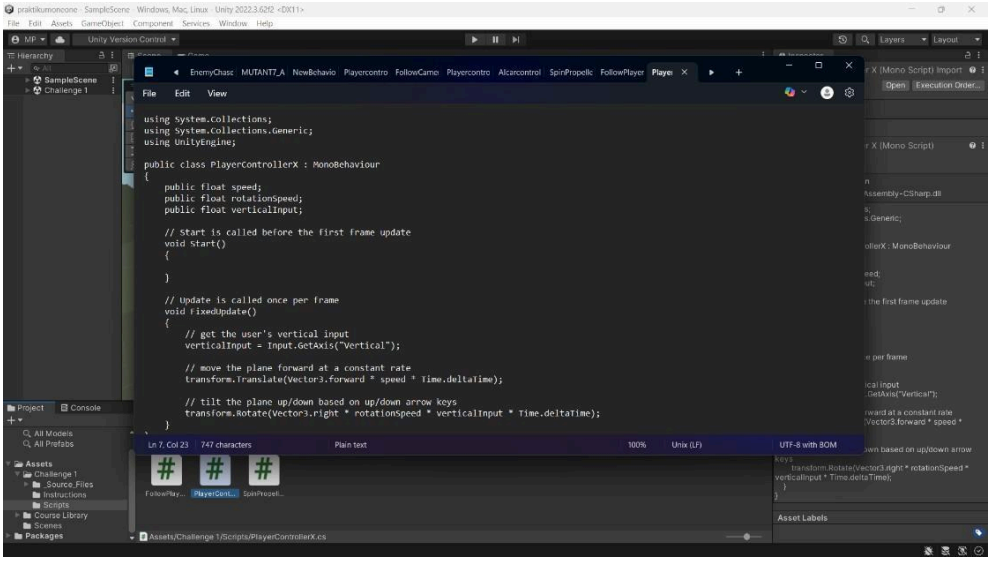




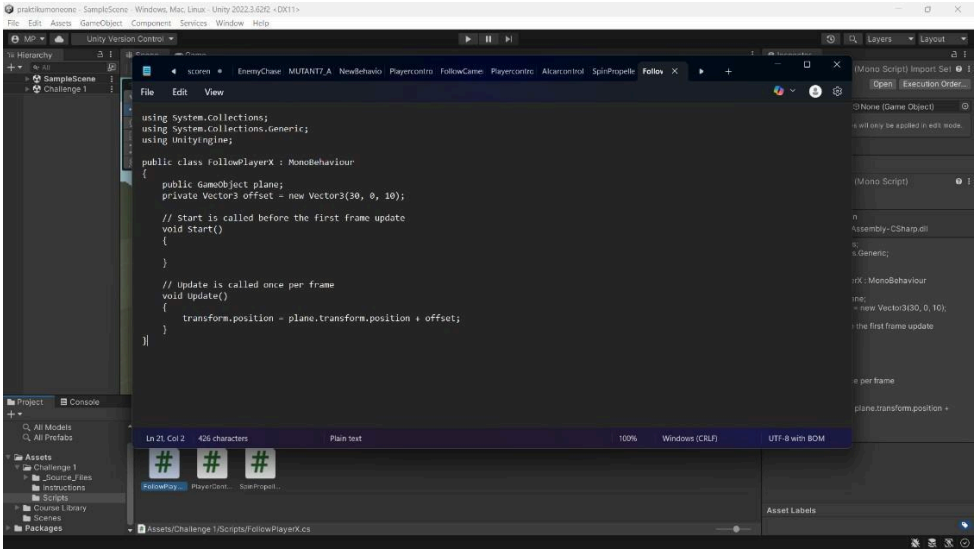
Berikut Script dari challenge 1(plane):



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SpinPropellerX : MonoBehaviour
{
    private float propellorSpeed = 1000;

    // Start is called before the first frame update 0 references
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        transform.Rotate(Vector3.forward, propellorSpeed * Time.deltaTime);
    }
}
```



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerControllerX : MonoBehaviour
{
    public float speed;
    public float rotationSpeed;
    public float verticalInput;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void FixedUpdate()
    {
        // get the user's vertical input
        verticalInput = Input.GetAxis("Vertical");

        // move the plane forward at a constant rate
        transform.Translate(Vector3.forward * speed * Time.deltaTime);

        // tilt the plane up/down based on up/down arrow keys
        transform.Rotate(Vector3.right * rotationSpeed * verticalInput * Time.deltaTime);
    }
```



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FollowPlayerX : MonoBehaviour
{
    public GameObject plane;
    private Vector3 offset = new Vector3(30, 0, 10);

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        transform.position = plane.transform.position + offset;
    }
}
```

4.2 PEMBAHASAN

A. Analisis Implementasi prototype

Pada praktikum ini, kita menambahkan komponen Rigidbody ke objek player. Tujuan dari penambahan komponen ini adalah supaya objek bisa bergerak dan bereaksi secara realistis terhadap fisika di dunia game, seperti gaya dorong, gravitasi, atau tabrakan, Kalau Rigidbody diaktifkan, otomatis Unity akan memperlakukan objek seolah-olah punya berat dan gaya. Misalnya: Kalau tidak ada lantai di bawahnya, objek akan jatuh karena gravitasi. Kalau diberi gaya dorong dari skrip, objek bisa meluncur atau bergeser. Kalau menabrak sesuatu, akan ada reaksi seperti mantul atau berhenti tergantung kekuatan gaya dan massa objeknya.

B. Analisis Skrip C# prototype

Skrip ini mengatur supaya player bisa bergerak ke berbagai arah menggunakan tombol keyboard:

1) AIcarcontrol : Script ini digunakan untuk mengontrol pergerakan mobil AI di dalam game, khususnya agar mobil AI Bergerak maju secara otomatis danMundur atau menghindar jika menabrak objek tertentu (seperti Box atau Car).
2) Camera mobil : Script ini digunakan untuk mengatur kamera agar mengikuti posisi pemain (player) secara otomatis.Bagaimana nilai input tersebut diterjemahkan menjadi vektor gaya yang diterapkan pada Rigidbody.
3) Player control mobil : Script ini digunakan untuk mengendalikan pergerakan player mobil dengan fisika realistis menggunakan Rigidbody, dan mendorong objek Box saat terjadi tabrakan.

C. Analisis Implementasi Plane

Proyek ini adalah simulasi pesawat terbang sederhana di Unity, Proyek ini menggunakan beberapa komponen utama Unity, seperti:

1) GameObject (Plane/pesawat) sebagai objek utama.
2) Rigidbody untuk mengaktifkan sistem fisika agar pesawat bisa bergerak dan bereaksi secara realistis.

D. Skrip C# Plane

Skrip ini mengatur supaya player bisa bergerak ke berbagai arah menggunakan tombol keyboard:

1) Camera pesawat: Script ini membuat kamera mengikuti pesawat (plane) secara otomatis, dengan jarak tertentu agar posisi kamera tidak menempel langsung ke pesawat.
2) Player control pesawat : Script ini membuat pesawat bergerak maju otomatis, dan pemain bisa mengatur naik atau turun menggunakan tombol panah atas dan bawah.
3) Spiner pesawat: Script ini digunakan untuk memutar baling-baling (propeller) secara terus menerus, seperti pada pesawat atau helicopter.

BAB V
PENUTU
P

1. 1 KESIMPULAN

Dari hasil praktikum Junior Programmer Pathway di Unity, dapat disimpulkan bahwa kegiatan ini membantu memahami konsep dasar pemrograman visual dan sistem fisika dalam pengembangan game 3D.
Melalui latihan ini, kita bisa belajar bagaimana sebuah objek bisa bergerak, berinteraksi, dan merespons gaya fisika seperti gravitasi dan tabrakan dengan menggunakan komponen Rigidbody dan Plane di Unity.

Komponen Rigidbody berperan penting dalam memberikan efek fisika yang realistis, sedangkan Plane berfungsi sebagai permukaan tempat objek berpijak. Selain itu, penggunaan skrip C# seperti PlayerController, CameraController, AI Car Control, dan SpinPropeller juga melatih peserta untuk menggabungkan logika pemrograman dengan komponen visual di Unity.

Dari hasil pengujian, semua komponen bekerja dengan baik. Player, mobil, maupun pesawat dapat bergerak sesuai perintah dan menunjukkan reaksi fisika yang sesuai. Kamera juga dapat mengikuti objek utama dengan lancar, dan animasi seperti baling-baling pesawat dapat berputar secara otomatis.