

**LAPORAN
PEMROGRAMAN VISUAL**



Dosen Pengampu :

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

Disusun oleh :

1. Anaya Rizka Putri Novianti (5124521002)
2. Muhammad Irsadul Ibad (5124521007)
3. Muhammad Afif Habiburrahman (5124521010)
4. Mayva Sofiana (5124521018)
5. Devi Hidayah (5124521022)

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

LAPORAN UNITY PATHWAY
JUNIOR PROGRAMMING AS SOON AS POSSIBLE

GAMBARAN UMUM UNITY PATHWAY JUNIOR PROGRAMER.

Pathway Junior Programming dibagi menjadi beberapa unit / mission. Di halaman pathway, unit yang terlihat adalah :

1. Player Control
2. Basic Gameplay
3. Sound and Effects
4. Gameplay Mechanics
5. User Interface
6. Manage Scene Flow and Data
7. Apply Object-Oriented Principles

Pada laporan ini, pembahasan difokuskan pada **unit kedua**, yaitu **Basic Gameplay**. Bagian ini menekankan pentingnya pengaturan posisi dan pergerakan karakter pemain agar dapat berinteraksi secara tepat di dalam lingkungan permainan. Melalui tahap ini, dikembangkan pemahaman tentang cara menerima input dari pengguna, membatasi ruang gerak karakter dalam area tertentu, serta menjaga konsistensi logika permainan melalui penggunaan variabel dan kontrol alur program. Materi ini menjadi dasar penting sebelum melanjutkan ke tahap pembuatan fitur gameplay yang lebih kompleks pada misi berikutnya.

DAFTAR ISI

GAMBARAN UMUM UNITY PATHWAY	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Peralatan	2
BAB II PEMBAHASAN	3
2.1 Basic Gameplay.....	3
2.1.1 Create a new Project	3
2.1.2 Add the Player, Animals, and Food.....	4
2.1.3 Player Movement Control	4
2.1.4 Keep The Player Inbounds.....	5
2.1.5 Clean Up Code and Variables.....	5
2.1.6 Creating and Controlling Projectiles in Unity	5
2.1.7 Create a Spawn Manager.....	6
2.1.8 Spawn an Animal if S is Pressed	7
2.1.9 Spawn Random Animals From Array	8
2.1.10 Randomize the Spawn Location	9
2.1.11 Challenge 2 – Play Fetch	10
BAB III PENUTUP	12
3.1 Kesimpulan.....	12

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam proses pengembangan permainan digital, tahap awal yang paling penting adalah memahami bagaimana karakter utama dapat bergerak dan berinteraksi secara efektif di dalam dunia permainan. Pergerakan karakter menjadi inti dari sebagian besar jenis game, baik itu permainan aksi, simulasi, maupun petualangan. Oleh karena itu, kemampuan untuk mengatur posisi, arah, serta batas pergerakan karakter merupakan dasar yang harus dikuasai sebelum melangkah ke tahap desain gameplay yang lebih kompleks.

Melalui **Mission 2: Basic Gameplay pada Unity Junior Programmer Pathway**, kita akan diajak untuk mempelajari bagaimana sebuah permainan dibangun dari fondasi logika sederhana seperti membaca input dari pemain, menggerakkan objek di sepanjang sumbu tertentu, serta membatasi ruang gerak agar tetap berada dalam area permainan yang diinginkan. Tahap ini juga menjadi latihan penting untuk memahami hubungan antara logika pemrograman dan pengalaman bermain (player experience), di mana setiap baris kode berperan langsung dalam mengontrol respons sistem terhadap tindakan pemain.

Selain itu, mission ini juga melatih untuk berpikir secara terstruktur dalam menyusun skrip, menggunakan variabel untuk menyimpan nilai penting, dan menerapkan pernyataan bersyarat untuk menjaga keseimbangan serta kestabilan permainan. Dengan memahami konsep-konsep dasar ini, pengembang pemula akan memiliki landasan kuat untuk menciptakan permainan yang tidak hanya berfungsi dengan baik secara teknis, tetapi juga terasa intuitif dan menyenangkan bagi pengguna.

1.2 Tujuan

1. Memahami konsep dasar gameplay dalam Unity
2. Mempelajari cara membaca dan mengelola input dari pemain
3. Menerapkan logika pemrograman dasar
4. Melatih keterampilan dalam penggunaan skrip C# di Unity

5. Menumbuhkan kemampuan analitis dan problem solving
6. Membangun dasar pemahaman terhadap hubungan antara kode dan pengalaman bermain.

1.3 Peralatan

- Laptop.
- Unity hub
- Unity Editor (Versi 2022)
- Notepad ++ / Visual Studio (Code Editor)

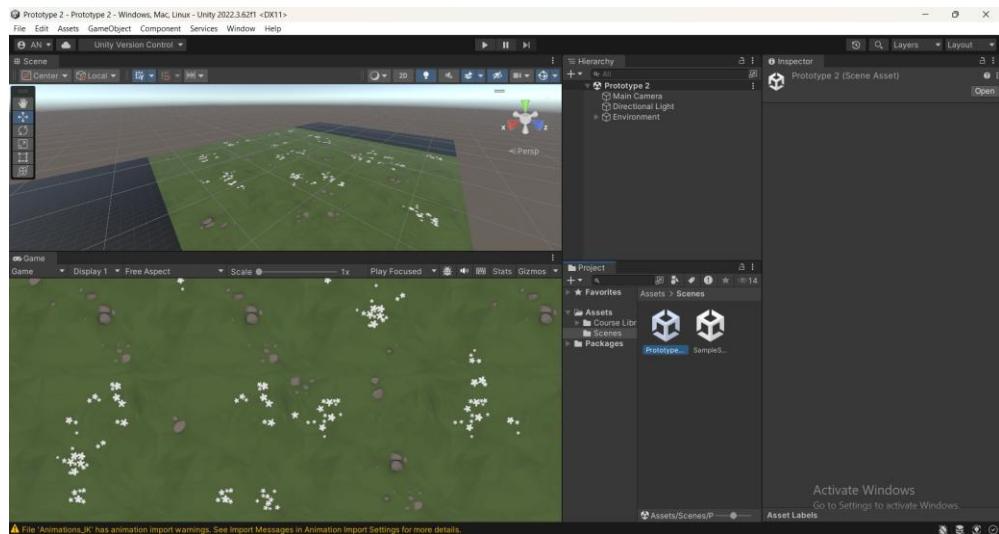
BAB II

PEMBAHASAN

2.1 Basic Gameplay

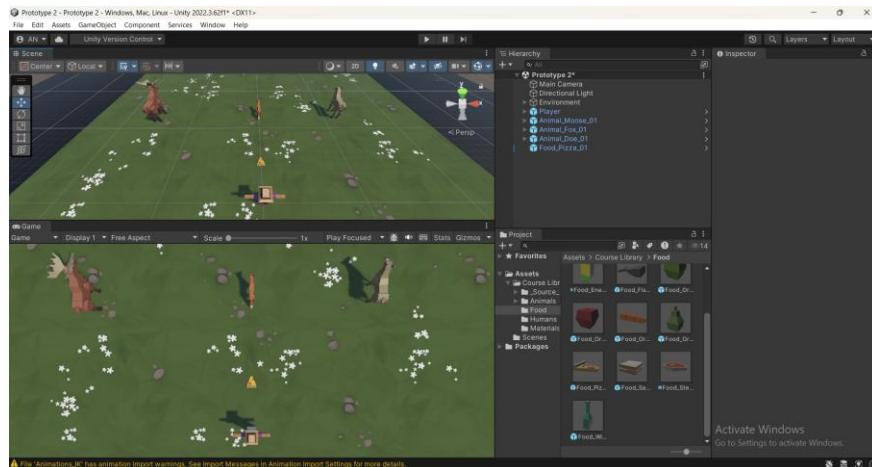
2.1.1 Create a New Project

Pada tahap awal tutorial Player Positioning, dilakukan proses pembuatan proyek baru di Unity Hub dengan nama **Prototype 2**, Unduh Prototype 2 Starter Files kemudian diekstrak, dan diimpor ke dalam proyek sebagai bahan utama untuk latihan.



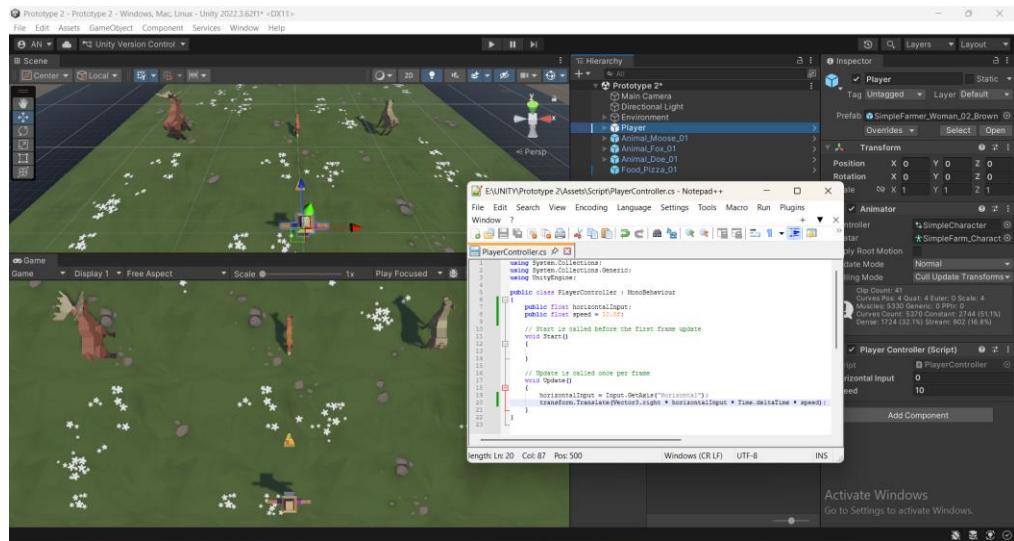
2.1.2 Add the Player, Animals, and Food

Pada tahap kedua dilakukan penambahan objek utama ke dalam scene, yaitu karakter pemain, beberapa hewan, dan makanan. Tampilan Ground dapat diberi material dari Course Library > Materials untuk memperindah visual.



2.1.3 Player Movement Control

Tahap ketiga mengatur pergerakan horizontal pemain dengan membuat script PlayerController yang menangkap input dari keyboard menggunakan Input.GetAxis("Horizontal"), lalu menerapkannya untuk menggerakkan objek pemain ke kiri dan kanan menggunakan transform.Translate() dengan kecepatan yang diatur melalui variabel speed.



2.1.4 Keep The Player Inbounds

Menambahkan pernyataan if di dalam fungsi Update(). Kondisi tersebut memeriksa posisi sumbu X pemain, dan jika melebihi batas yang ditentukan, posisi pemain akan dikembalikan ke titik batas tersebut menggunakan transform.position.

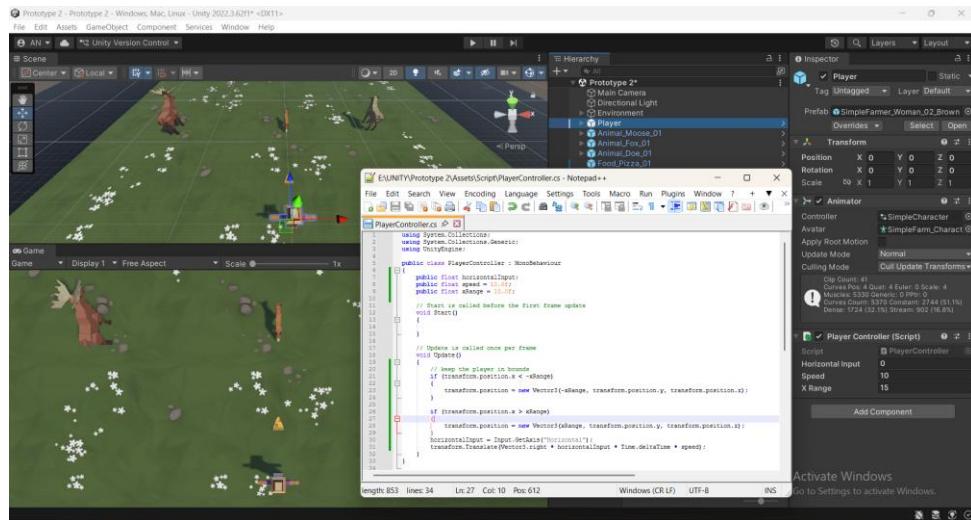
A screenshot of the Unity Editor showing the "PlayerController.cs" script in the code editor. The code is as follows:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      public float horizontalInput;
8      public float speed = 10.0f;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18         if (transform.position.x < -10)
19         {
20             transform.position = new Vector3(-10, transform.position.y, transform.position.z);
21         }
22         horizontalInput = Input.GetAxis("Horizontal");
23         transform.Translate(Vector3.right * horizontalInput * Time.deltaTime * speed);
24     }
25 }
26
27 }
```

The code defines a PlayerController class that inherits from MonoBehaviour. It has two public float variables: horizontalInput and speed (set to 10.0f). The Start() method is empty. The Update() method checks if the player's position on the x-axis is less than -10, and if so, sets it to -10. It then retrieves the horizontal input from the keyboard and moves the player using transform.Translate().

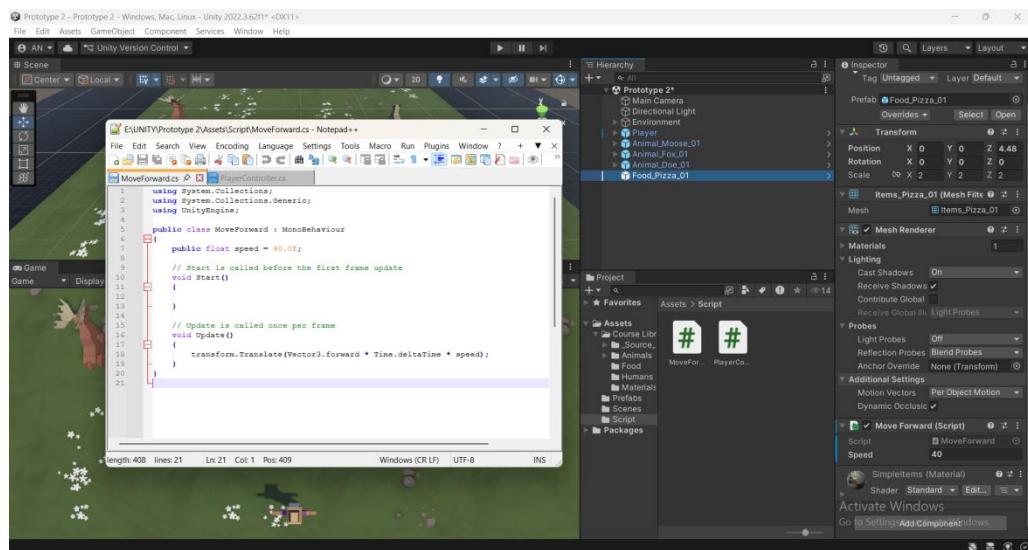
2.1.5 Clean Up Code and Variables

Melakukan penyempurnaan kode dengan menambahkan batas gerak pemain di sisi kiri dan kanan layar menggunakan variabel xRange.



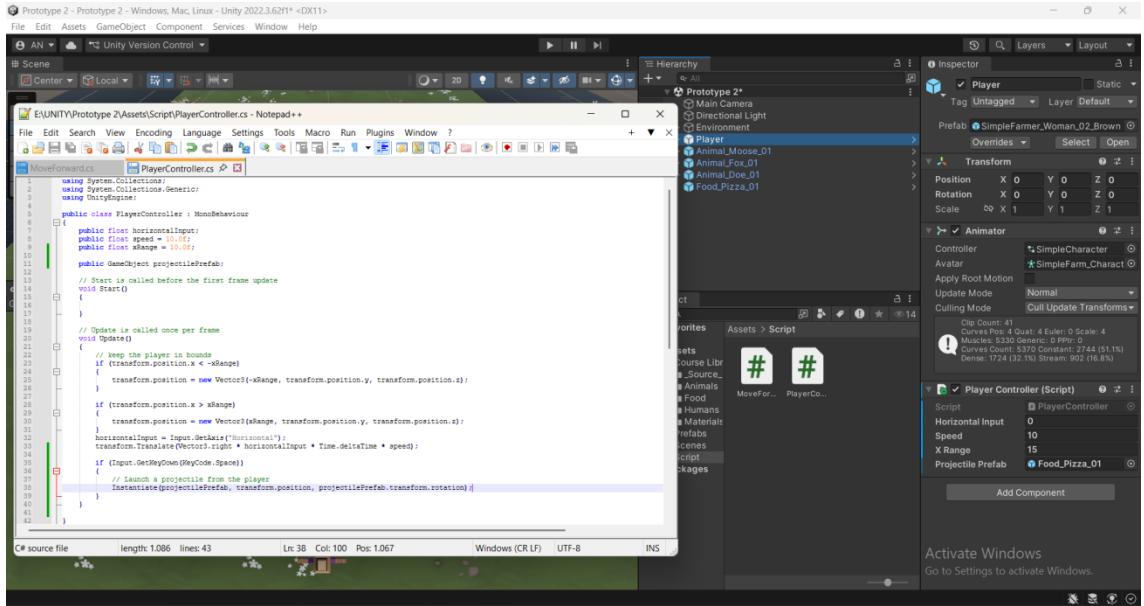
2.1.6 Creating and Controlling Projectiles in Unity

Pembuatan script baru bernama MoveForward yang berfungsi untuk mengatur pergerakan objek makanan. Dalam skrip tersebut, ditambahkan variabel public float speed untuk mengatur kecepatan gerak.



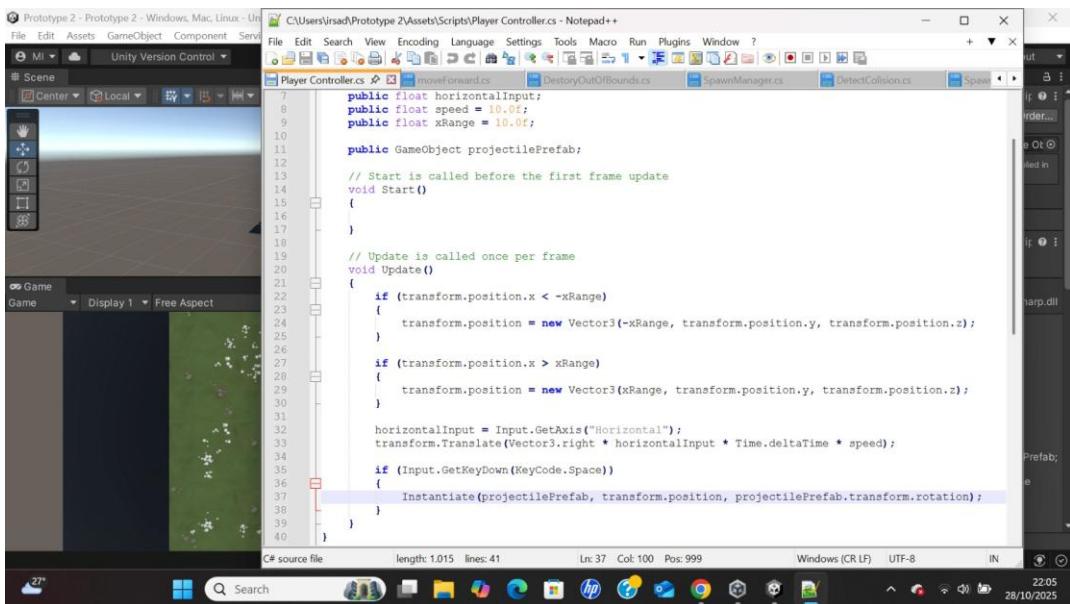
Folder baru bernama Prefabs dibuat untuk menyimpan objek makanan sebagai Original Prefab. Pada skrip PlayerController.cs, ditambahkan variabel public GameObject projectilePrefab dan dihubungkan ke Projectile Prefab di panel

Inspector. Pemain dapat menembakkan projektil dengan menekan spasi, menggunakan metode Instantiate untuk memunculkan projektil di posisi pemain dengan rotasi prefab.

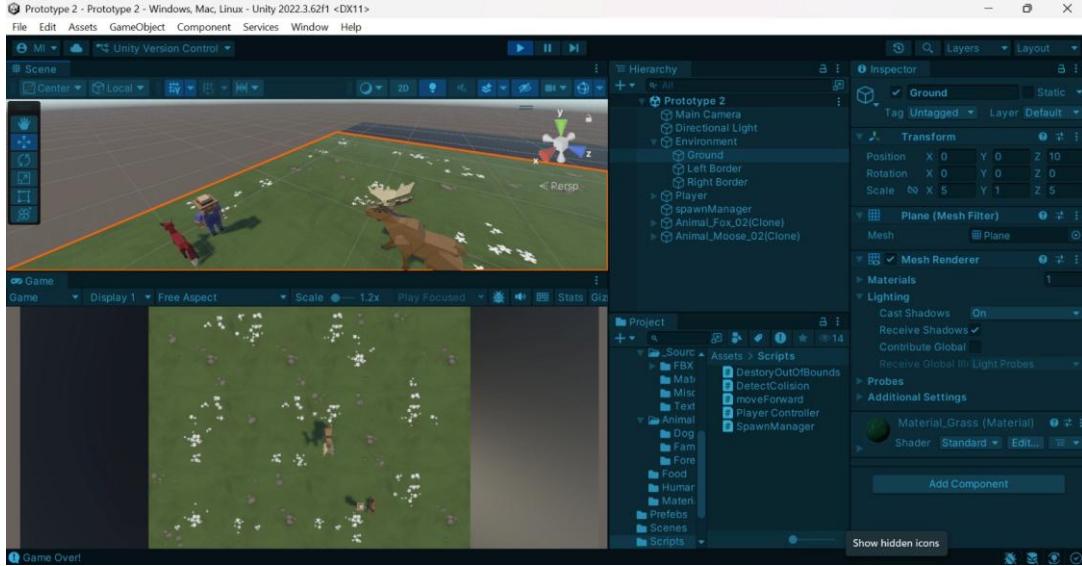


2.1.7 Create a Spawn Manager

Menampilkan suasana saat proses pengujian game di Unity dengan menggunakan bahasa pemrograman C#. Editor Unity di sisi kiri dan Notepad++ di kanan yang sedang menampilkan script bernama PlayerController.cs. Script ini berfungsi untuk mengatur pergerakan karakter pemain (player) dan aksi saat menembakkan projektil. Terdapat pengaturan agar pemain tidak melewati batas area permianan dengan variabel xRange.

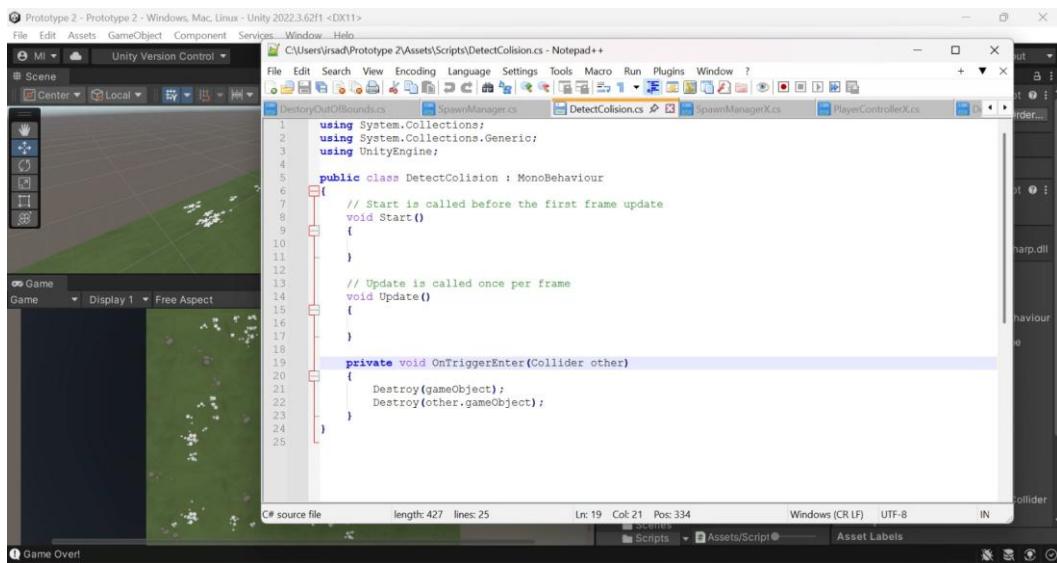


Memperlihatkan tampilan Unity ketika proyek dijalankan. Di dalamnya terlihat karakter pemain beserta hewan-hewan (seperti rusa dan rubah) yang menjadi bagian dari gameplay. Panel Hierarchy menunjukkan susunan objek dalam scene, seperti Player, Ground, Left Border, dan Right Border. Sementara panel Project memperlihatkan berbagai folder dan script yang digunakan, termasuk script yang sedang diuji.



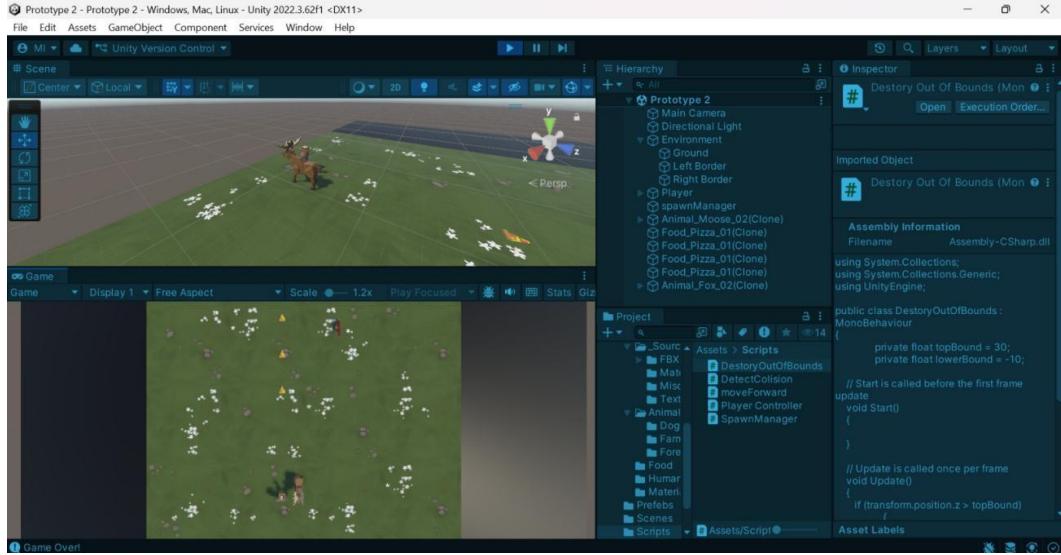
2.1.8 Spawn an animal if S is pressed

Memperlihatkan tampilan Notepad++ yang sedang menampilkan script berjudul DestroyOutOfBounds.cs. Untuk menghapus objek yang keluar dari batas area permainan,dibagian atas maupun bawah layar.



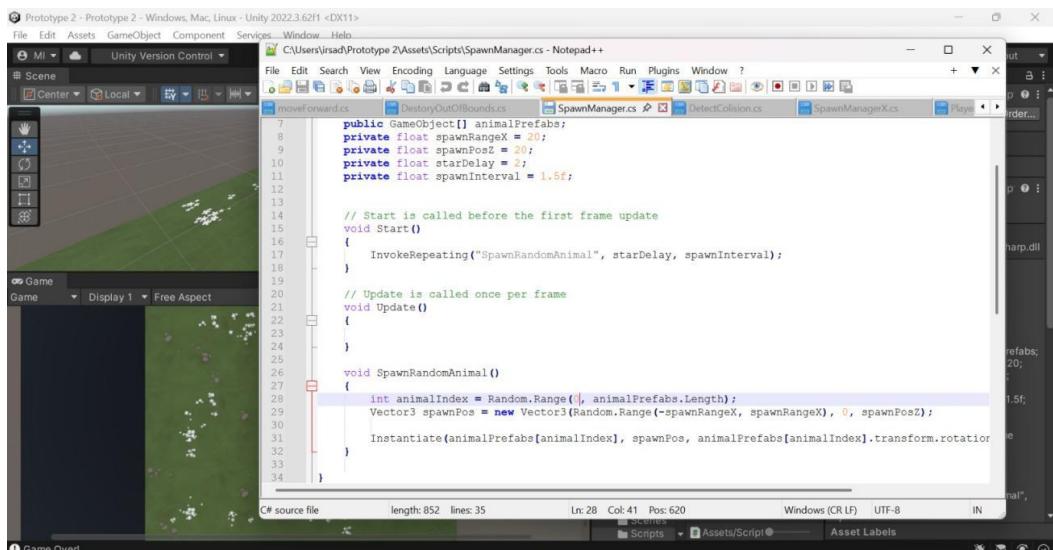
Memperlihatkan hasil penerapan script tersebut di dalam Unity Editor. Di sini tampak berbagai objek dalam scene seperti hewan dan makanan yang muncul di

area permainan. Pada panel Inspector, script DestroyOutOf Bounds sudah terpasang dan aktif, memastikan setiap objek yang keluar dari area permainan otomatis dihapus.



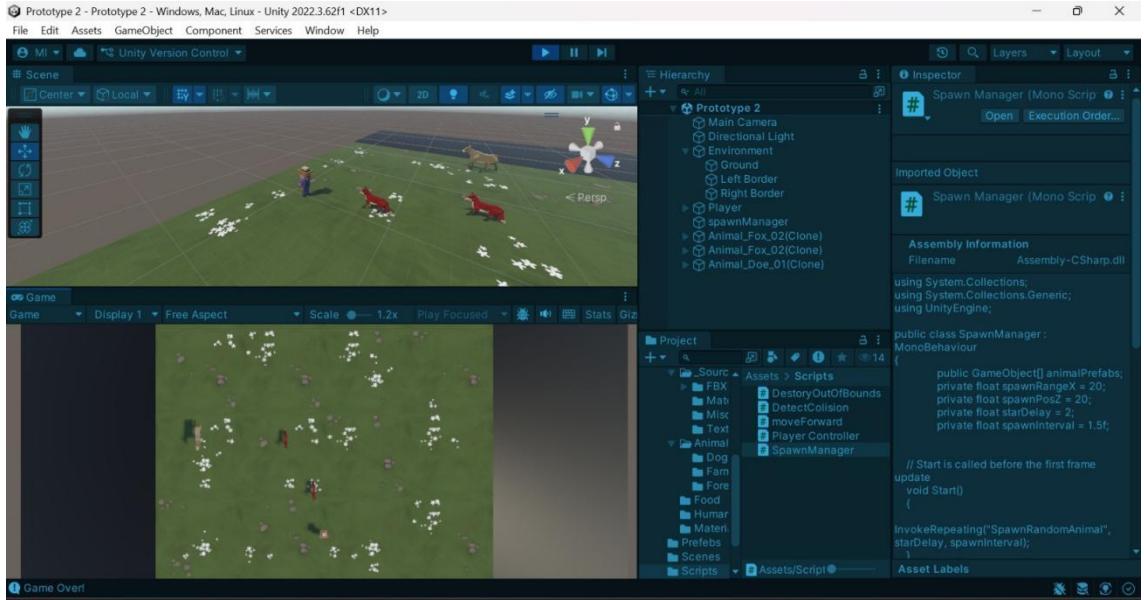
2.1.9 Spawn random animals from array

Memperlihatkan penulisan script SpawnManager.cs di Notepad++, berfungsi untuk mengatur kemunculan objek secara acak di dalam permainan. Script ini menggunakan metode InvokeRepeating() agar objek hewan muncul secara berkala pada posisi acak di area permainan. Nilai seperti spawnRangeX, spawnPosZ, dan spawnInterval mengontrol lokasi serta waktu kemunculan tiap objek, sehingga permainan terasa dinamis.



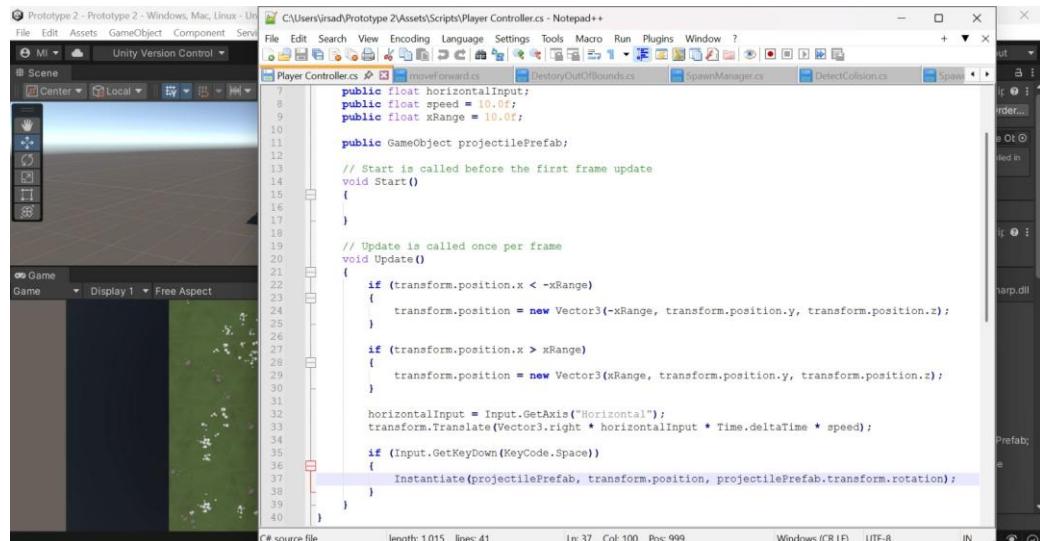
Menunjukkan hasil penerapan script tersebut, hewan seperti rusa dan rubah muncul

secara acak di medan permainan, menandakan sistem spawn bekerja dengan baik. Panel Hierarchy menampilkan objek-objek hasil duplikasi (clone) dari prefab hewan, sementara panel Inspector menunjukkan bahwa script SpawnManager aktif dan berjalan sesuai pengaturan.



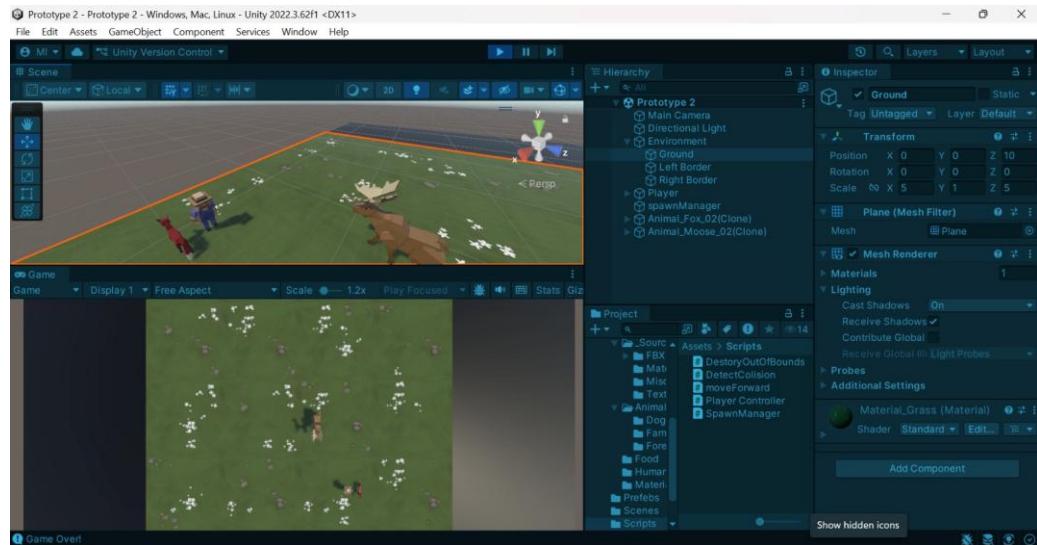
2.1.10 Randomize the spawn location

Memperlihatkan script DetectCollision.cs yang sedang dibuka di Notepad++. Script ini berfungsi mendeteksi tumbukan antara dua objek dalam permainan. Ketika dua objek saling bertabrakan, fungsi OnTriggerEnter() akan dijalankan untuk menghapus keduanya dari area permainan.

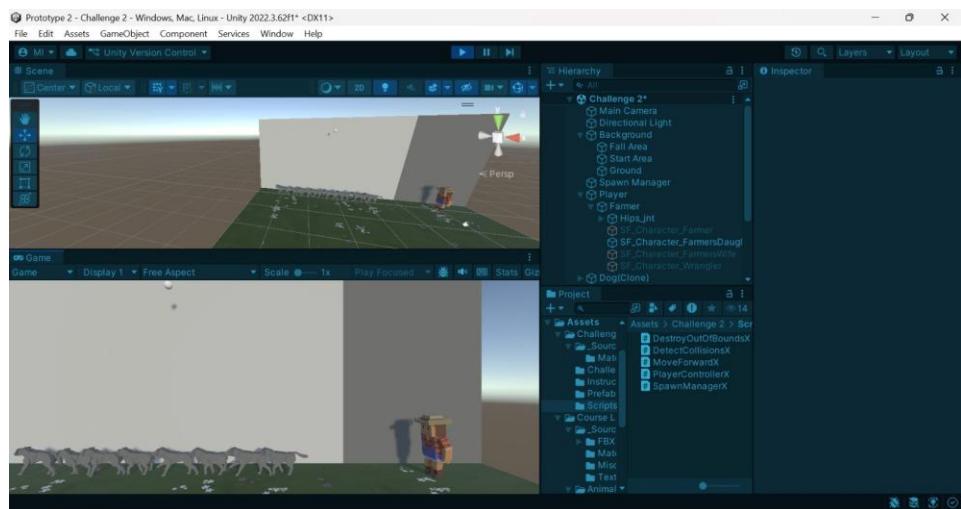


Tampilan Unity Editor saat script tersebut diterapkan. Terlihat beberapa hewan

dan objek yang aktif, seperti rubah dan rusa, yang bergerak dan berpotensi saling bertumbukan. Pada panel Inspector, script DetectCollision sudah diaktifkan sebagai komponen dari objek terkait, memastikan sistem dapat mengenali dan menghapus objek saat terjadi kontak.



2.1.11 CHALLENGE 2 - PLAY FETCH



Menunjukkan proyek Unity bertajuk Prototype 2 Challenge 2. Di jendela Scene dan Game, terlihat karakter petani (Farmer) serta beberapa anjing (Dog(Clone)) yang muncul di area permainan. Di panel Hierarchy, berbagai elemen penting proyek terdaftar, seperti kamera utama, cahaya, area permainan, hingga pengatur pemunculan objek (Spawn Manager). Ini memperlihatkan proses pengujian permainan, di mana objek-objek telah aktif dan disimulasikan di lingkungan 3D Unity.

The screenshot shows the Unity Editor interface with the PlayerController.cs script open in the code editor. The script contains logic for player movement and projectile spawning. A specific line of code is highlighted in blue:

```

    if (Input.GetKeyDown(KeyCode.Space))
    {
        Instantiate(projectilePrefab, transform.position, projectilePrefab.transform.rotation);
    }

```

Memperlihatkan kode C# bernama SpawnManagerX.cs yang dibuka di Notepad++. Bagian skrip tersebut berfungsi mengatur waktu kemunculan objek secara acak. Baris kode spawnInterval = Random.Range(3f, 5f); menentukan jarak waktu acak antara tiga hingga lima detik untuk memunculkan objek baru. Melalui fungsi InvokeRepeating, sistem menjalankan perintah SpawnRandomBall() secara berulang berdasarkan interval tersebut. Skrip ini mengatur ritme munculnya objek di permainan agar tidak monoton.

The screenshot shows the Unity Editor interface with the DestroyOutOfBounds.cs script open in the code editor. The script contains logic for destroying objects that move outside the boundaries of the game area. A specific line of code is highlighted in blue:

```

    Debug.Log("Game Over");
    Destroy(gameObject);
}

```

Memperlihatkan skrip lain berjudul DestroyOutOfBoundsX.cs. Fungsinya adalah menghapus objek yang keluar dari batas arena permainan agar tidak membebani sistem. Pada kode terlihat dua kondisi: satu untuk menghapus objek ketika posisi X melewati batas kiri, dan satu lagi ketika posisi Y lebih rendah dari batas bawah. Mekanisme ini memastikan bahwa objek yang sudah tidak terlihat di layar akan dihapus secara otomatis.

BAB III

PENUTUP

3.1 Kesimpulan

Proyek Prototype 2 pada Unity Junior Programmer Pathway memberikan pemahaman dasar tentang pembuatan mekanik permainan menggunakan Unity dan C#. Melalui script seperti PlayerController, SpawnManagerX, dan DestroyOutOfBoundsX, peserta belajar mengatur pergerakan karakter, pemunculan objek acak, serta pembatasan area permainan. Kegiatan ini melatih kemampuan berpikir logis, pemecahan masalah, dan penerapan kode secara efektif dalam membangun gameplay yang seimbang dan responsif.