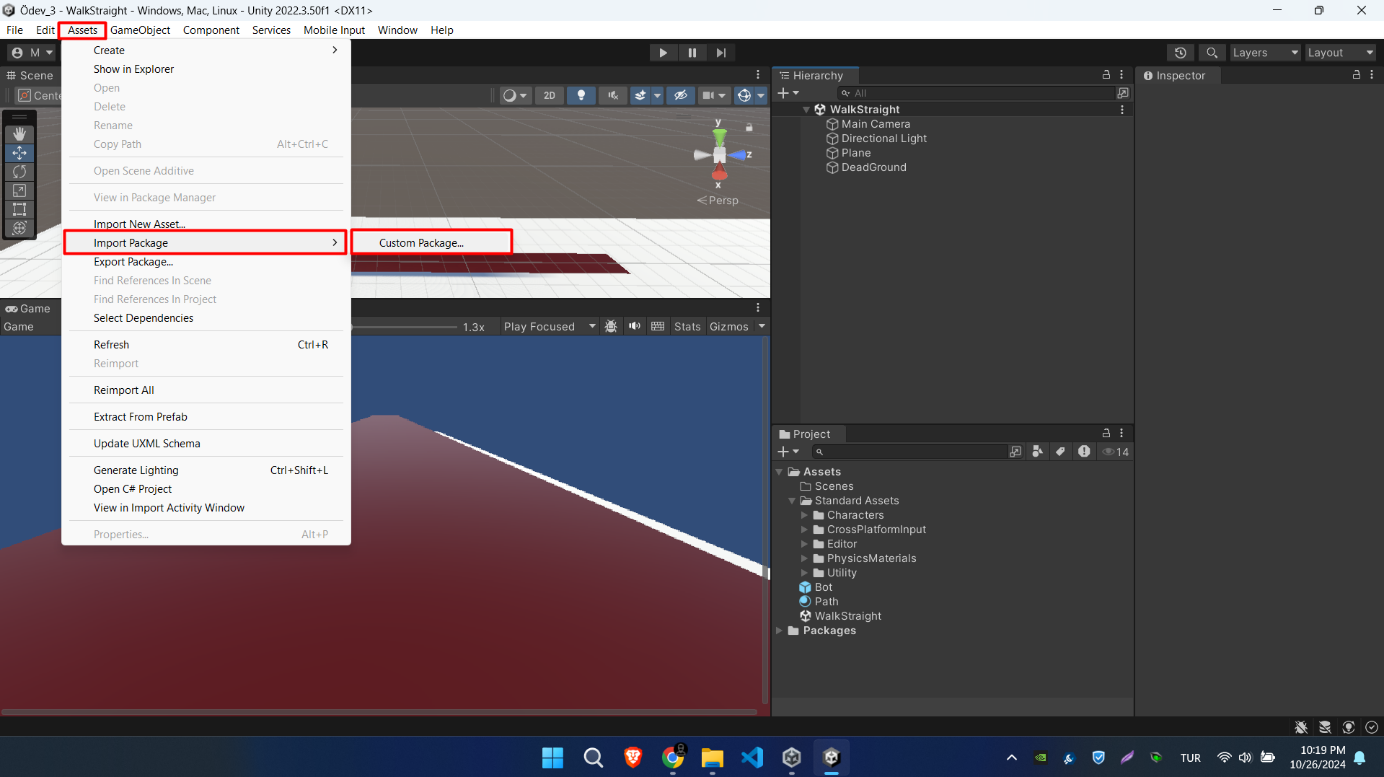
Bilgisayar Oyunlarında Yapay Zekâ - Ödev 3

1 – Asset Import etme

1.1 - Üst menüdeki Assets->Import Package->Custom Package kısmından “EthanWalker2022.unitypackage” adlı dosyayı seçiyoruz.



2 – Script dosyalarını oluşturma

2.1 – Scripts adında bir klasör oluşturuyoruz. Klasör içine DNA\_sc scriptini oluşturup gerekli kodları ekliyoruz:

List<int> genes = new List<int>(); // Gen listesi için

int dnaLength = 0; // DNA uzunluğu

int maxValues = 0; // Maksimum değerler

// Yapıcı metot, DNA uzunluğu ve maksimum değerleri alır

public DNA\_sc(int l, int v)

{

    dnaLength = l;

    maxValues = v;

    SetRandom(); // Rastgele genler oluştur

}

// Rastgele genler oluşturur

public void SetRandom()

{

    genes.Clear(); // Gen listesini temizle

    for (int i = 0; i < dnaLength; i++)

    {

        genes.Add(Random.Range(0, maxValues)); // Rastgele gen ekle

    }

}

// Belirli bir pozisyondaki geni ayarlar

public void SetInt(int pos, int value)

{

    genes[pos] = value;

}

// İki DNA'yı birleştirir

public void Combine(DNA\_sc d1, DNA\_sc d2)

{

    for (int i = 0; i < dnaLength; i++)

    {

        if (i < dnaLength / 2.0)

        {

            int c = d1.genes[i];

            genes[i] = c;

        }

        else

        {

            int c = d2.genes[i];

            genes[i] = c;

        }

    }

}

// Rastgele bir geni mutasyona uğratır

public void Mutate()

{

    genes[Random.Range(0, dnaLength)] = Random.Range(0, maxValues);

}

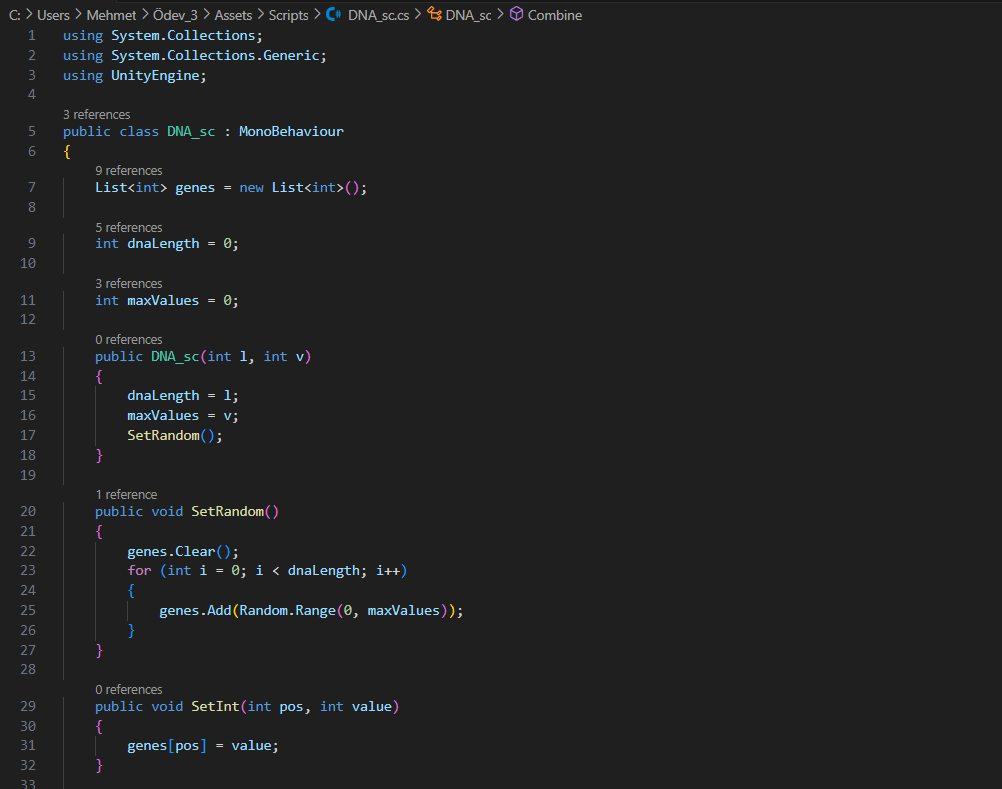
// Belirli bir pozisyondaki geni döndürür

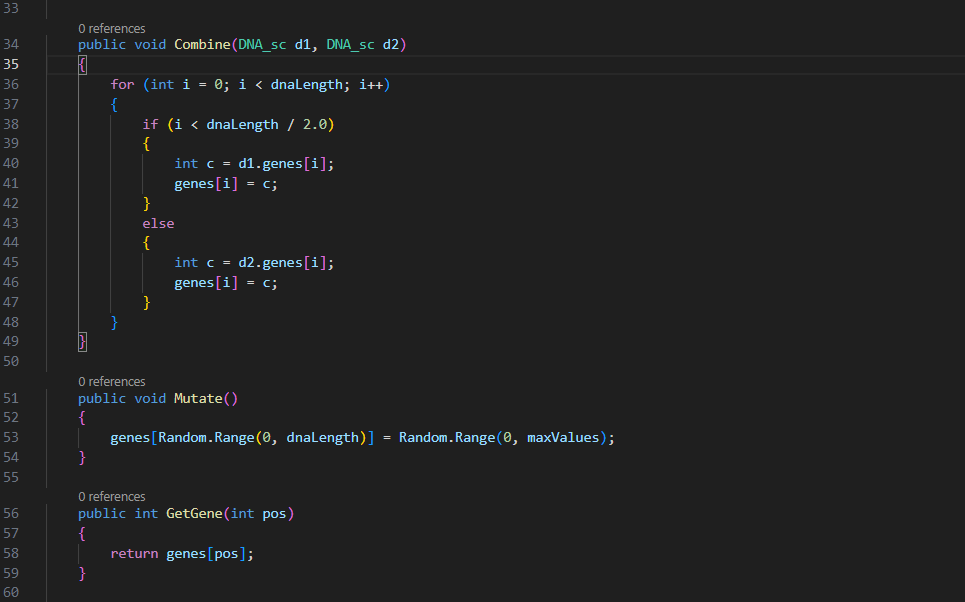
public int GetGene(int pos)

{

    return genes[pos];

}





2.1 – Brain\_sc adında bir script oluşturuyoruz. Oluşturduğumuz scripti Bot prefab ile ilişkilendiriyoruz.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityStandardAssets.Characters.ThirdPerson;

[RequireComponent(typeof(ThirdPersonCharacter))]

public class Brain\_sc : MonoBehaviour

{

    public int DNALength = 1; // DNA uzunluğu

    public float timeAlive; // Canlı kalma süresi

    public DNA\_sc dna\_sc; // DNA nesnesi

    private ThirdPersonCharacter m\_Character; // Üçüncü şahıs karakteri

    private Vector3 m\_Move; // Hareket vektörü

    private bool m\_Jump; // Zıplama durumu

    bool alive = true; // Canlılık durumu

    void OnCollisionEnter(Collision obj)

    {

        // Çarpışma olduğunda "dead" etiketi varsa canlılık durumu false olur

        if (obj.gameObject.tag == "dead")

        {

            alive = false;

        }

    }

    public void Init()

    {

        // DNA'yı başlat

        // 0 ileri

        // 1 geri

        // 2 sol

        // 3 sağ

        // 4 zıpla

        // 5 çömel

        dna\_sc = new DNA\_sc(DNALength, 6);

        m\_Character = GetComponent<ThirdPersonCharacter>();

        timeAlive = 0;

        alive = true;

    }

    private void FixedUpdate()

    {

        // DNA'yı oku

        float h = 0;

        float v = 0;

        bool crouch = false;

        bool jump = false;

        // DNA'ya göre hareket ve zıplama belirle

        if (dna\_sc.GetGene(0) == 0) v = 1;

        else if (dna\_sc.GetGene(0) == 1) v = -1;

        else if (dna\_sc.GetGene(0) == 2) h = -1;

        else if (dna\_sc.GetGene(0) == 3) h = 1;

        else if (dna\_sc.GetGene(0) == 4) jump = true;

        else if (dna\_sc.GetGene(0) == 5) crouch = true;

        m\_Move = v \* Vector3.forward + h \* Vector3.right; // Hareket vektörünü belirle

        m\_Character.Move(m\_Move, crouch, jump); // Karakteri hareket ettir

        if (alive)

        {

            timeAlive += Time.deltaTime; // Canlı kalma süresini artır

        }

    }

    // Start fonksiyonu, başlatıldığında bir kez çağrılır

    void Start()

    {

    }

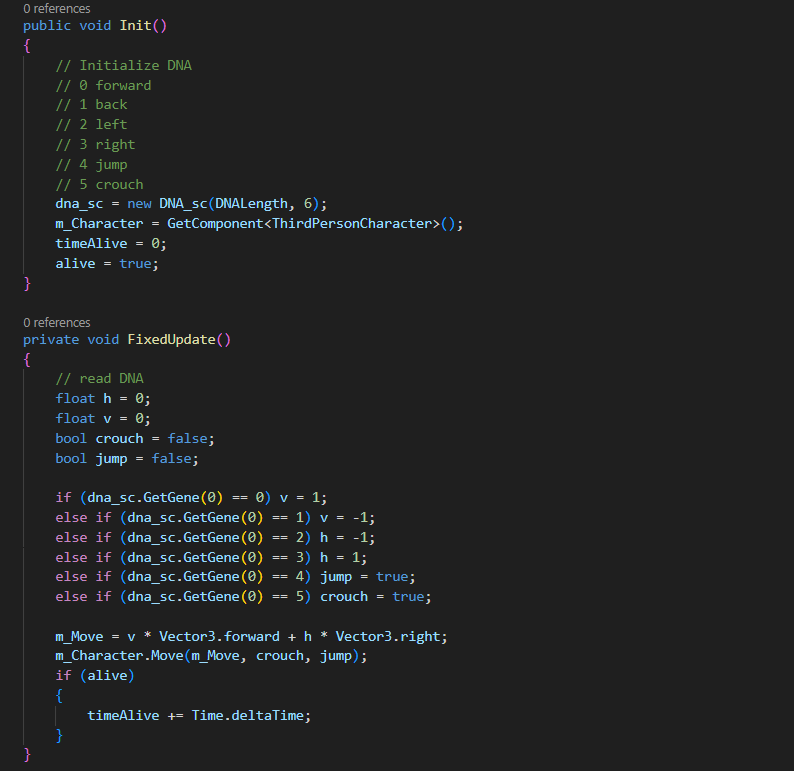
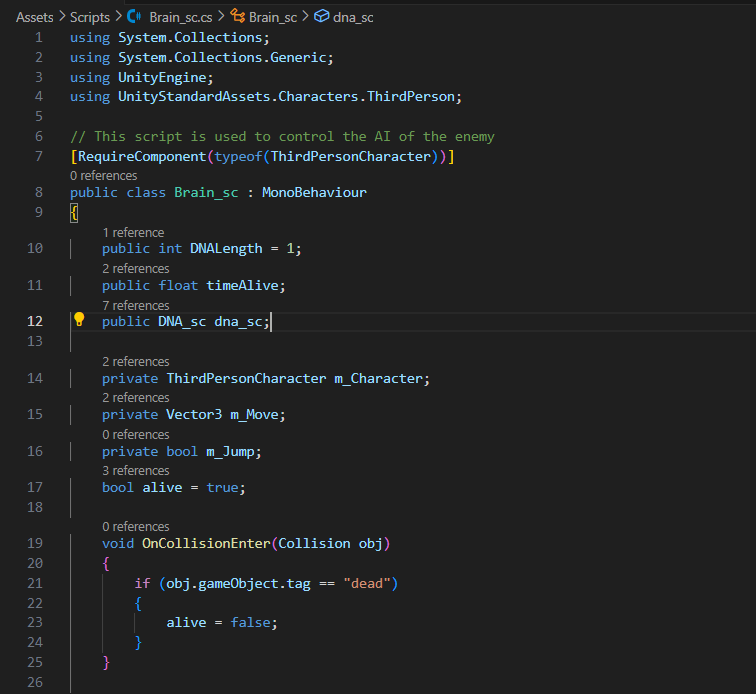
    // Update fonksiyonu, her karede bir kez çağrılır

    void Update()

    {

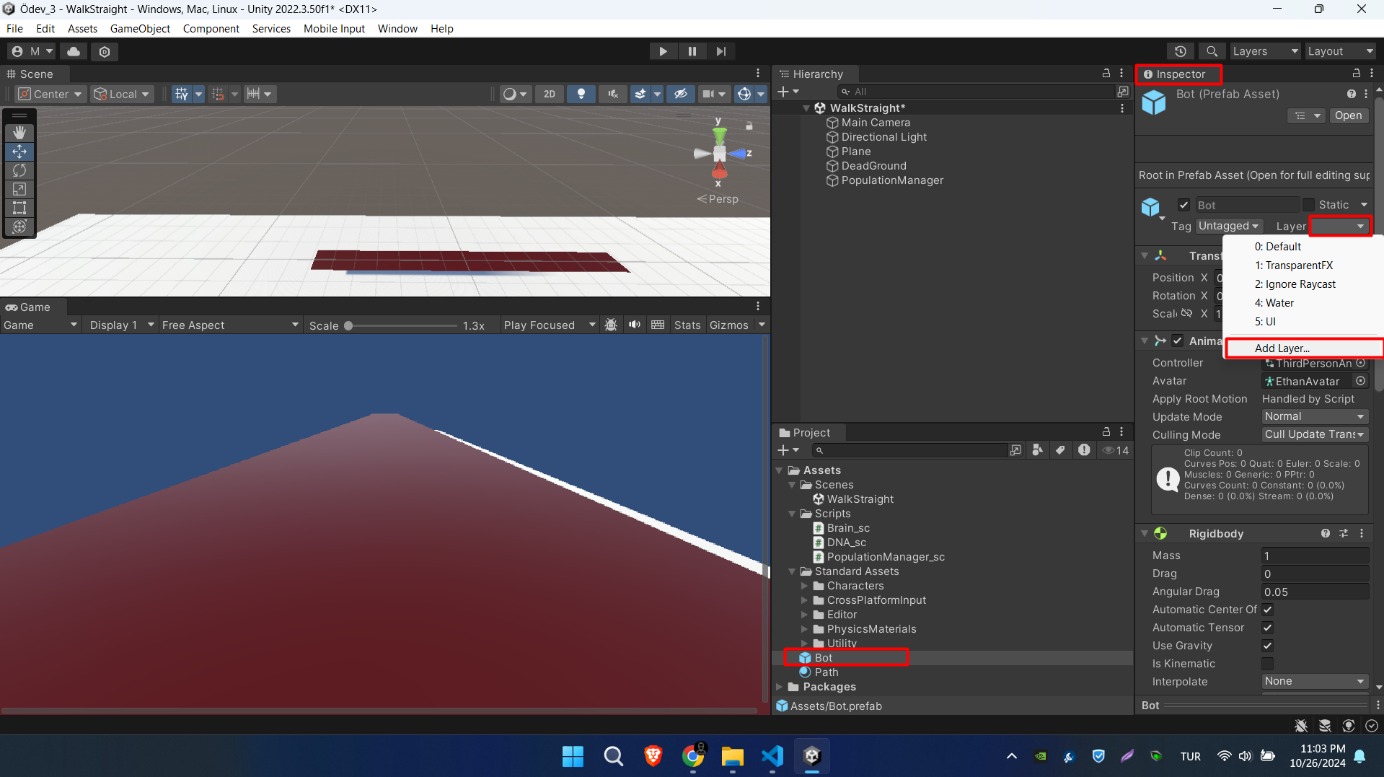
    }

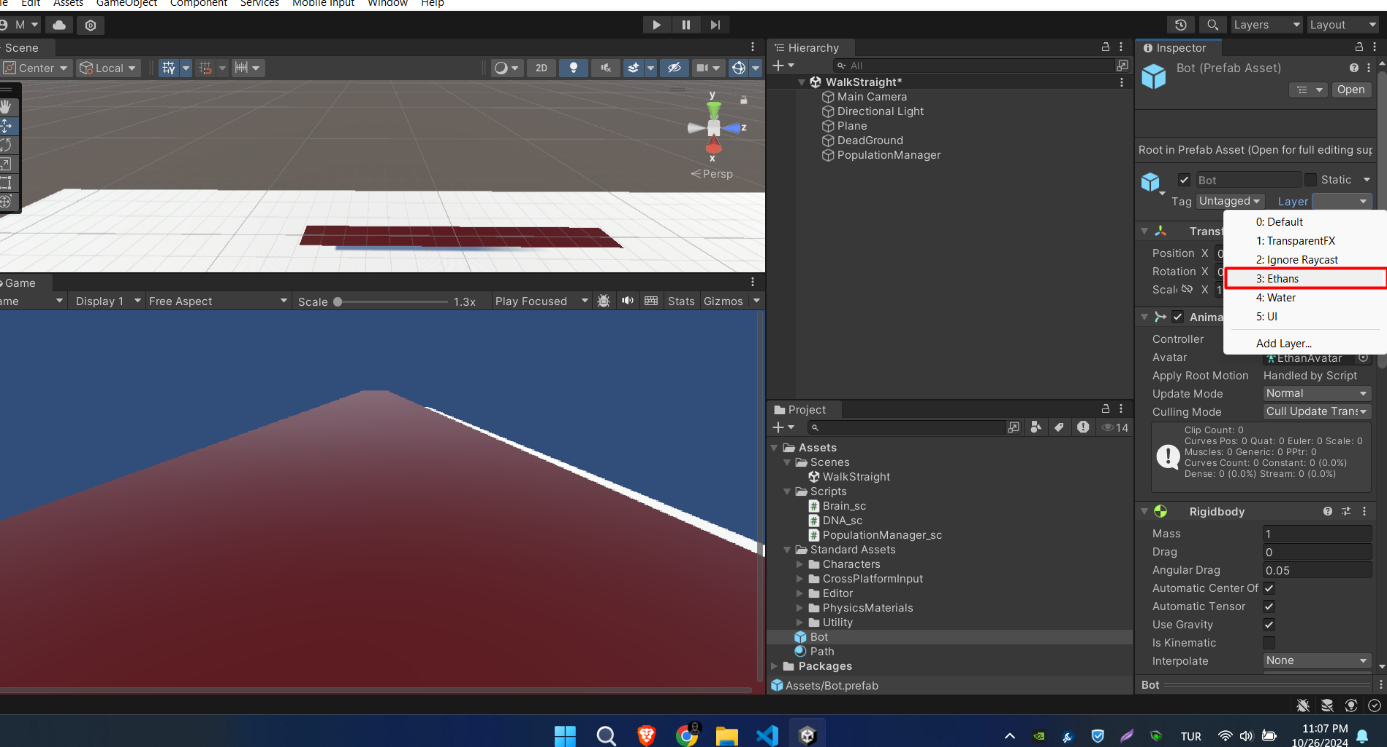
}



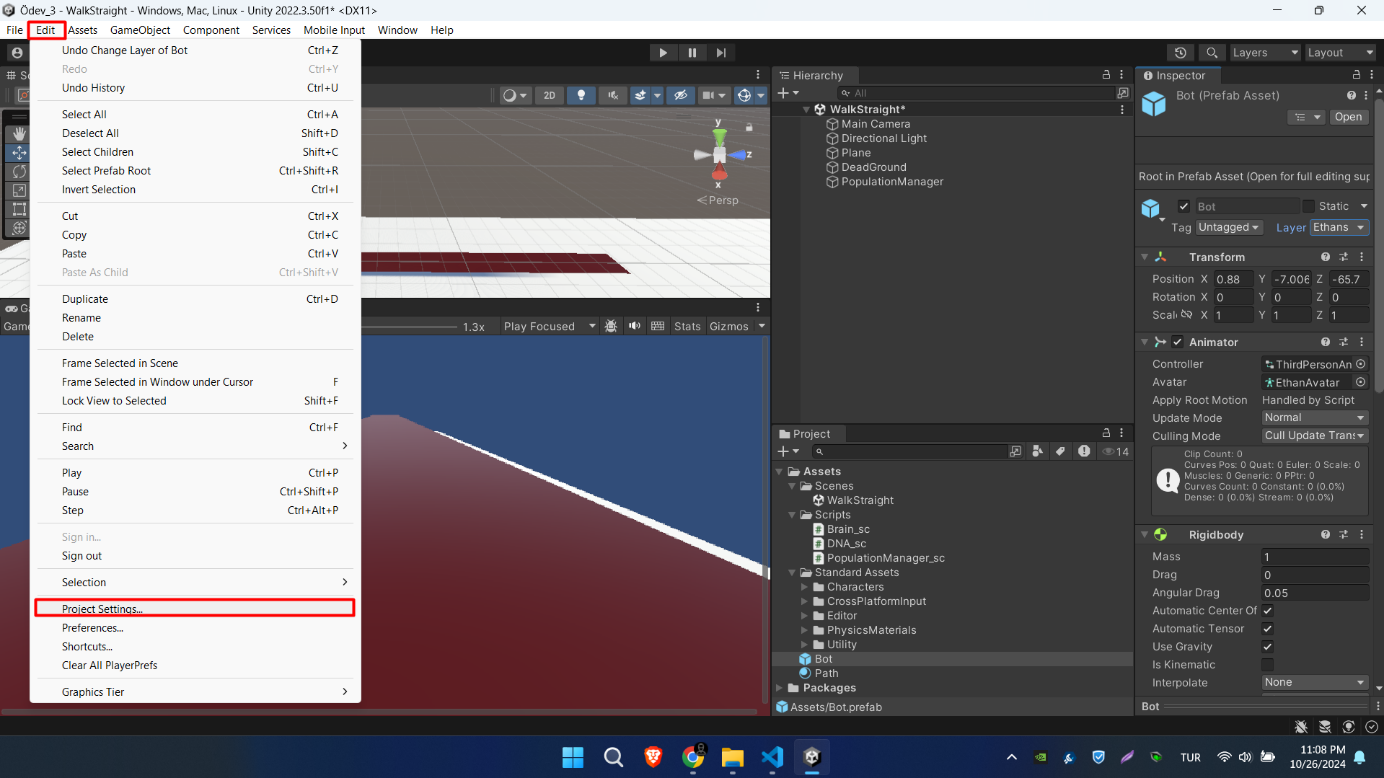
3 – Karakterlerin birbirine çarpma durumunu ayarlama

3.1 – Bot prefab’ı üzerine tıklıyoruz. Inspector alanında Layer kısmında Add Layer seçeneğini seçiyoruz. Herhangi boş bir alana Ethans yazıyoruz.

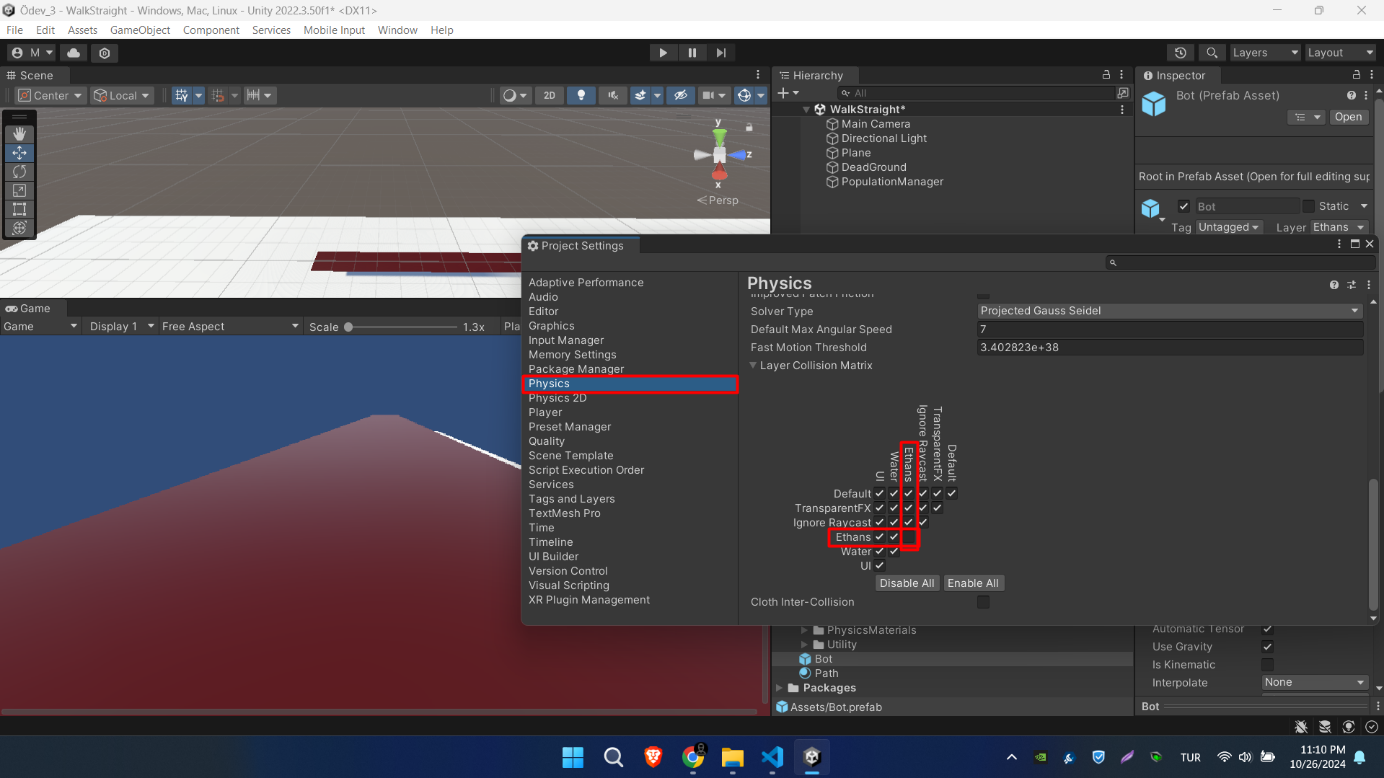




3.2 – Layer ayarları için Edit->Project Settings kısmına giriyoruz.



Açılan menüden Physics kısmından Ethans kesişimi olan tiki kaldırıyoruz.



4 –Boş bir nesne oluşturup adını PopulationManager koyuyoruz. PopulationManager\_sc adında bir script oluşturup scripti PopulationManager nesnesine bağlıyoruz. Ardından script içine aşağıdaki kodu ekliyoruz.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.Linq;

public class PopulationManager\_sc : MonoBehaviour

{

    public GameObject botPrefab; // Bot prefab referansı

    public int populationSize = 50; // Popülasyon boyutu

    List<GameObject> population = new List<GameObject>(); // Popülasyon listesi

    public static float elapsed = 0; // Geçen süre

    public float trialTime = 5; // Deneme süresi

    int generation = 1; // Nesil sayısı

    GUIStyle guiStyle = new GUIStyle(); // GUI stil objesi

    void OnGUI()

    {

        guiStyle.fontSize = 25; // Yazı tipi boyutu

        guiStyle.normal.textColor = Color.white; // Yazı rengi

        GUI.BeginGroup(new Rect(10, 10, 250, 150)); // GUI grubu başlat

        GUI.Box(new Rect(0, 0, 140, 140), "Stats", guiStyle); // İstatistik kutusu

        GUI.Label(new Rect(10, 25, 200, 30), "Gen: " + generation, guiStyle); // Nesil etiketi

        GUI.Label(new Rect(10, 50, 200, 30), string.Format("Time: {0:0.00}", elapsed), guiStyle); // Zaman etiketi

        GUI.Label(new Rect(10, 75, 200, 30), "Population: " + population.Count, guiStyle); // Popülasyon etiketi

        GUI.EndGroup(); // GUI grubu bitir

    }

    GameObject Breed(GameObject parent1, GameObject parent2)

    {

        // Yeni botun başlangıç pozisyonu

        Vector3 startingPos = new Vector3(this.transform.position.x + Random.Range(-2, 2),

            this.transform.position.y,

            this.transform.position.z + Random.Range(-2, 2));

        GameObject offspring = Instantiate(botPrefab, startingPos, this.transform.rotation); // Yeni bot oluştur

        Brain\_sc b = offspring.GetComponent<Brain\_sc>(); // Beyin bileşenini al

        if (Random.Range(0, 100) == 1) // Mutasyon kontrolü

        {

            b.Init(); // Beyni başlat

            b.dna\_sc.Mutate(); // Mutasyon uygula

        }

        else

        {

            b.Init(); // Beyni başlat

            b.dna\_sc.Combine(parent1.GetComponent<Brain\_sc>().dna\_sc, parent2.GetComponent<Brain\_sc>().dna\_sc); // DNA birleştir

        }

        return offspring; // Yeni botu döndür

    }

    void BreedNewPopulation()

    {

        // Popülasyonu mesafeye göre sırala

        List<GameObject> sortedList = population.OrderBy(o => (o.GetComponent<Brain\_sc>().timeAlive)).ToList();

        population.Clear(); // Popülasyonu temizle

        // Sıralı listenin üst yarısını çiftleştir

        for (int i = (int)(sortedList.Count / 2.0f) - 1; i < sortedList.Count - 1; i++)

        {

            population.Add(Breed(sortedList[i], sortedList[i + 1])); // Yeni bot ekle

            population.Add(Breed(sortedList[i + 1], sortedList[i])); // Yeni bot ekle

        }

        // Tüm ebeveynleri ve önceki popülasyonu yok et

        for (int i = 0; i < sortedList.Count; i++)

        {

            Destroy(sortedList[i]); // Botu yok et

        }

        generation++; // Nesil sayısını artır

    }

    // Başlangıçta çağrılır

    void Start()

    {

        for (int i = 0; i < populationSize; i++)

        {

            // Yeni botun başlangıç pozisyonu

            Vector3 startingPos = new Vector3(this.transform.position.x + Random.Range(-2, 2),

                this.transform.position.y,

                this.transform.position.z + Random.Range(-2, 2));

            GameObject b = Instantiate(botPrefab, startingPos, this.transform.rotation); // Yeni bot oluştur

            b.GetComponent<Brain\_sc>().Init(); // Beyni başlat

            population.Add(b); // Popülasyona ekle

        }

    }

    // Her karede çağrılır

    void Update()

    {

        elapsed += Time.deltaTime; // Geçen süreyi artır

        if (elapsed >= trialTime)

        {

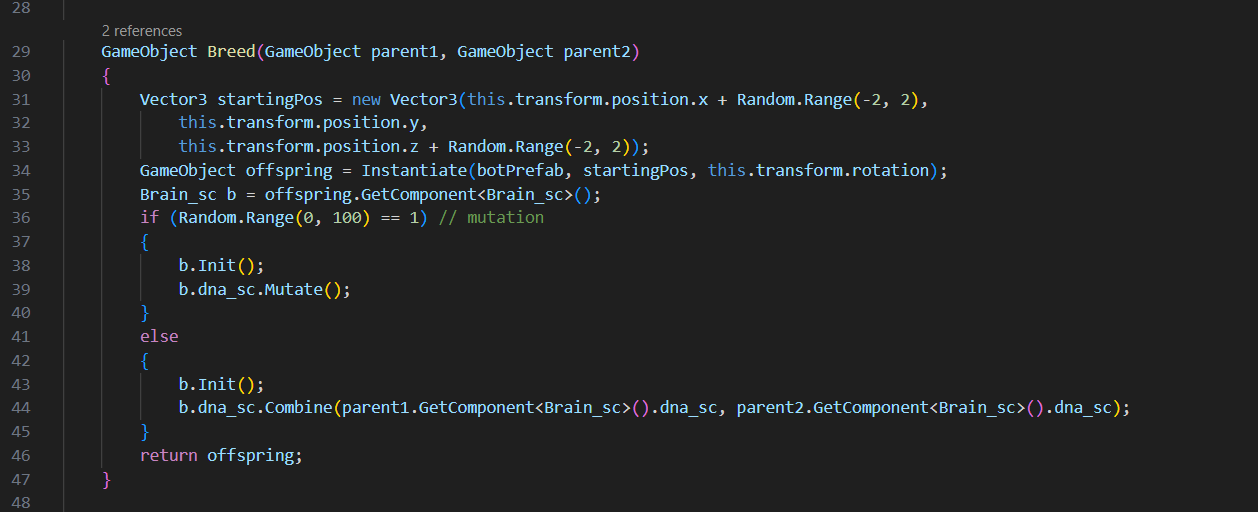
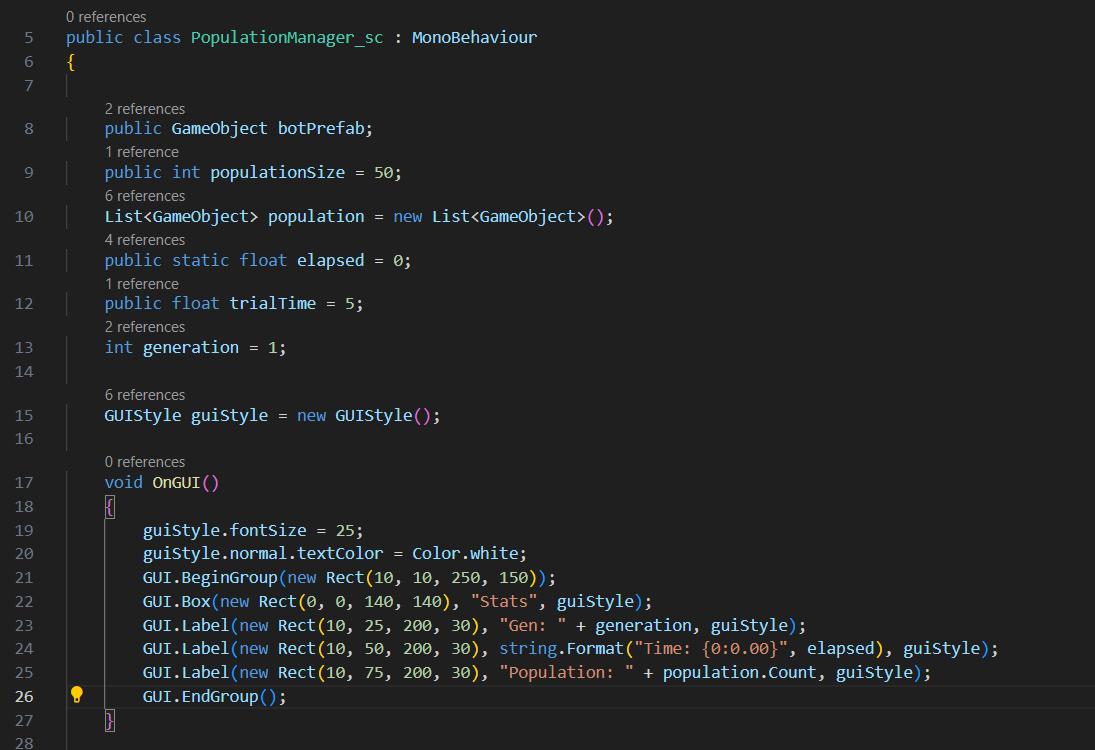
            BreedNewPopulation(); // Yeni popülasyon oluştur

            elapsed = 0; // Geçen süreyi sıfırla

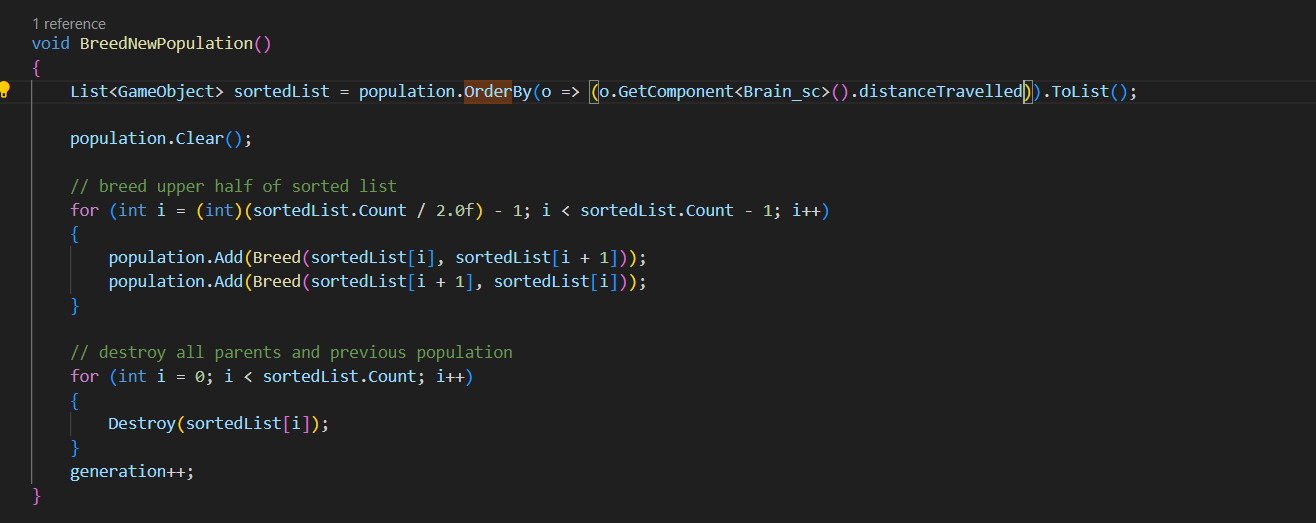
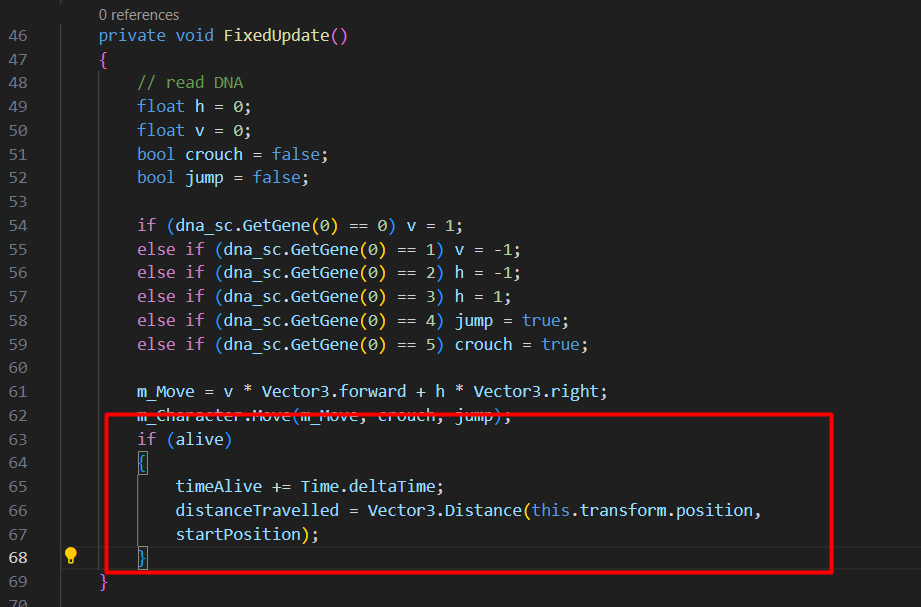
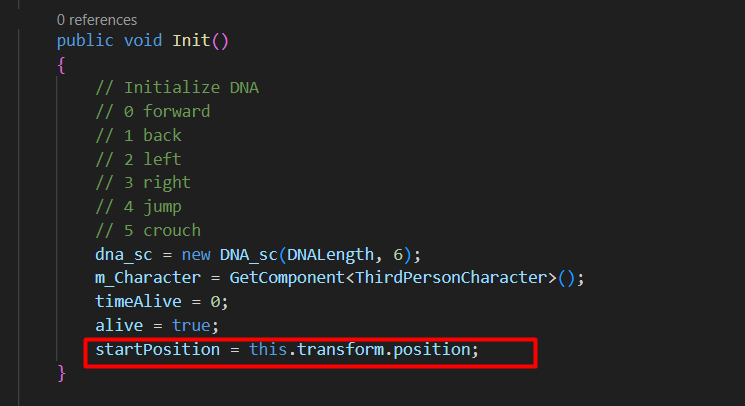
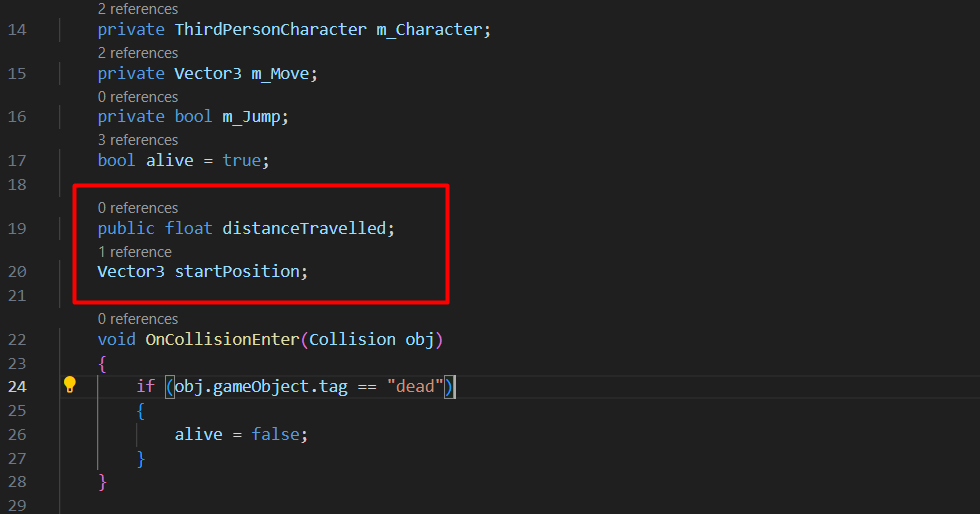
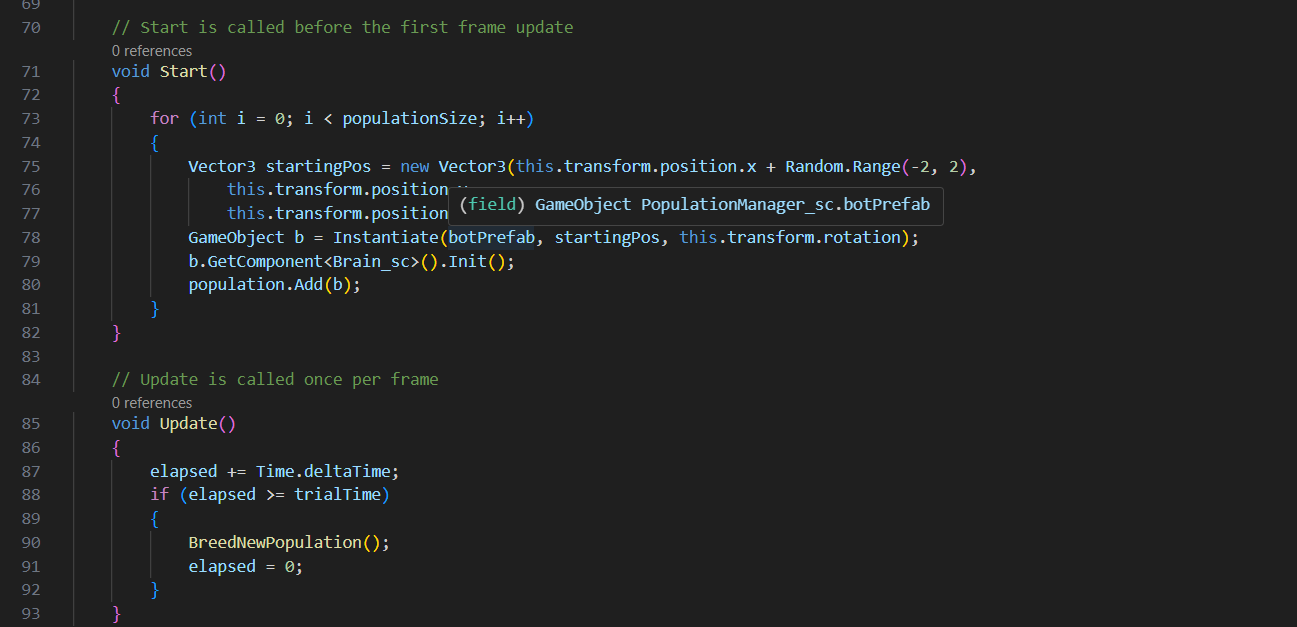
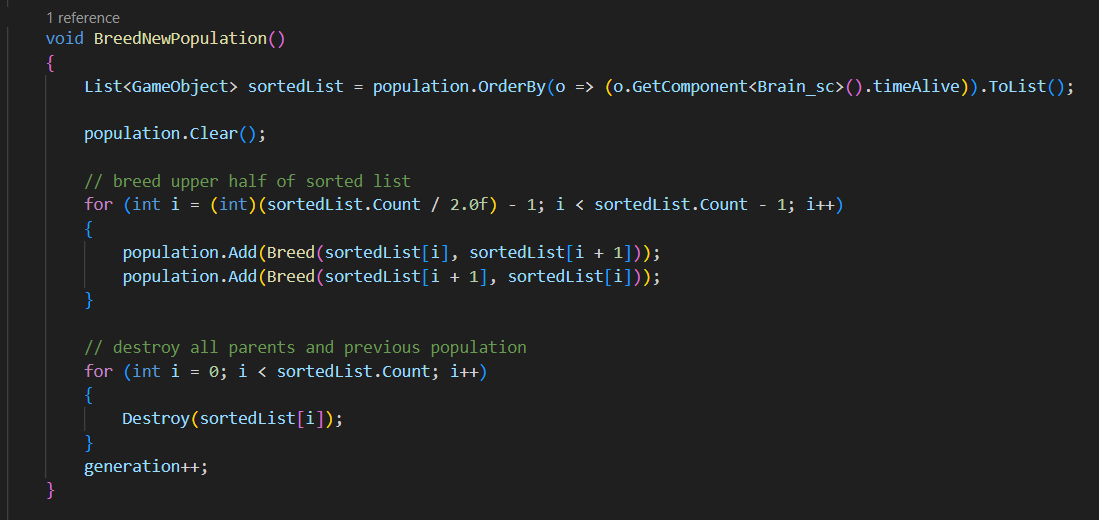
        }

    }

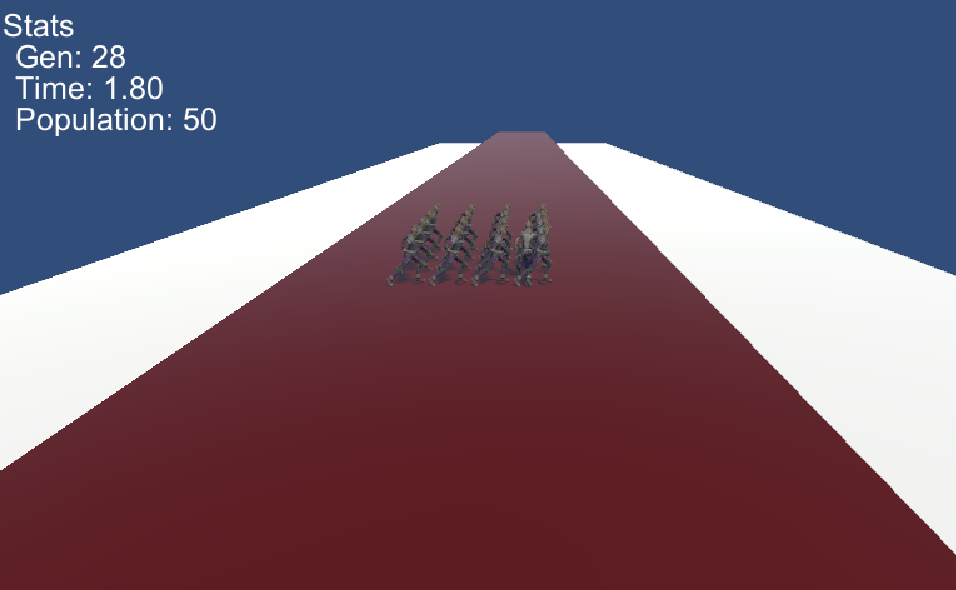
}



5 – Gidilen uzaklık ve başlangıç pozisyonu ile ilgili değişiklikler için aşağıdaki kısımları değiştiriyoruz.



6 – Oyunu başlatıyoruz.



Github Linki: <https://github.com/mehmetgencdal/BilgisayarOyunlarindaYapayZeka/tree/main/%C3%96dev_3>