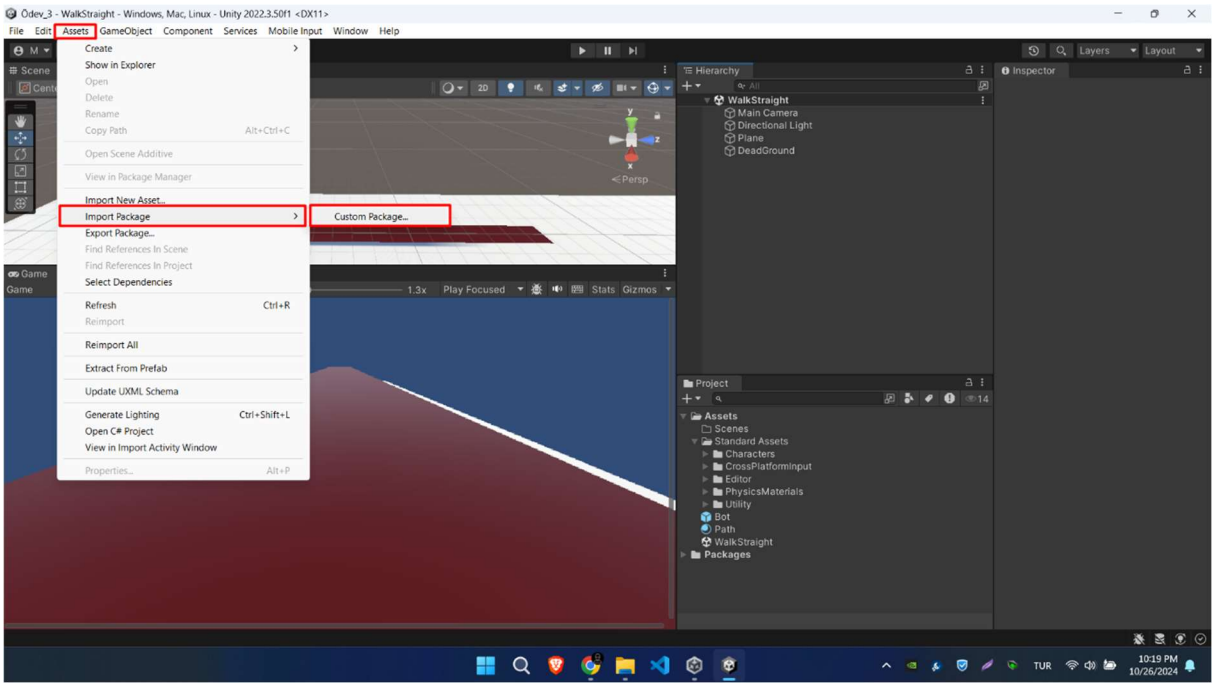
Bilgisayar Oyunlarında Yapay Zekâ - Ödev 3

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];
}
```

C: > Users > Mehmet > Ödev_3 > Assets > Scripts > DNA_sc.cs > DNA_sc > Combine

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
    3 references
5   public class DNA_sc : MonoBehaviour
6   {
        9 references
7       List<int> genes = new List<int>();
8
        5 references
9       int dnaLength = 0;
10
        3 references
11      int maxValues = 0;
12
        0 references
13      public DNA_sc(int l, int v)
14      {
15          dnaLength = l;
16          maxValues = v;
17          SetRandom();
18      }
19
        1 reference
20      public void SetRandom()
21      {
22          genes.Clear();
23          for (int i = 0; i < dnaLength; i++)
24          {
25              genes.Add(Random.Range(0, maxValues));
26          }
27      }
28
        0 references
29      public void SetInt(int pos, int value)
30      {
31          genes[pos] = value;
32      }
33
```

```
33
        0 references
34      public void Combine(DNA_sc d1, DNA_sc d2)
35      {
36          for (int i = 0; i < dnaLength; i++)
37          {
38              if (i < dnaLength / 2.0)
39              {
40                  int c = d1.genes[i];
41                  genes[i] = c;
42              }
43              else
44              {
45                  int c = d2.genes[i];
46                  genes[i] = c;
47              }
48          }
49      }
50
        0 references
51      public void Mutate()
52      {
53          genes[Random.Range(0, dnaLength)] = Random.Range(0, maxValues);
54      }
55
        0 references
56      public int GetGene(int pos)
57      {
58          return genes[pos];
59      }
60
```

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()
    {

    }
}
```

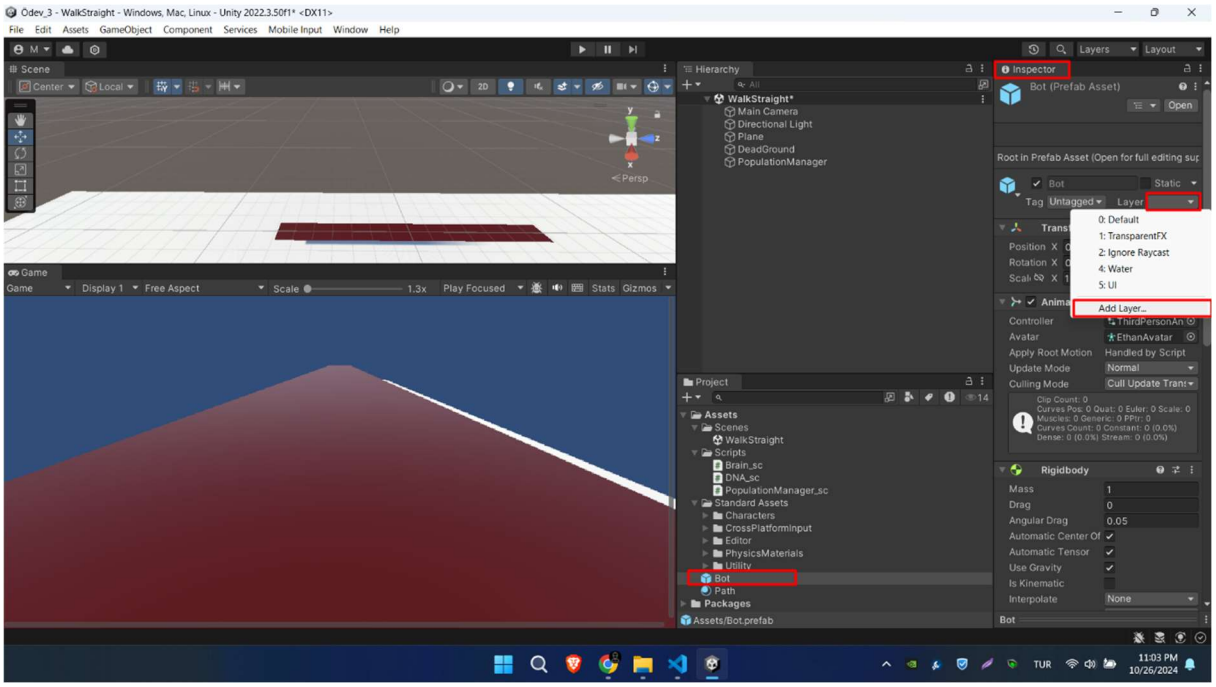
Assets > Scripts > Brain_sc.cs > Brain_sc > dna_sc

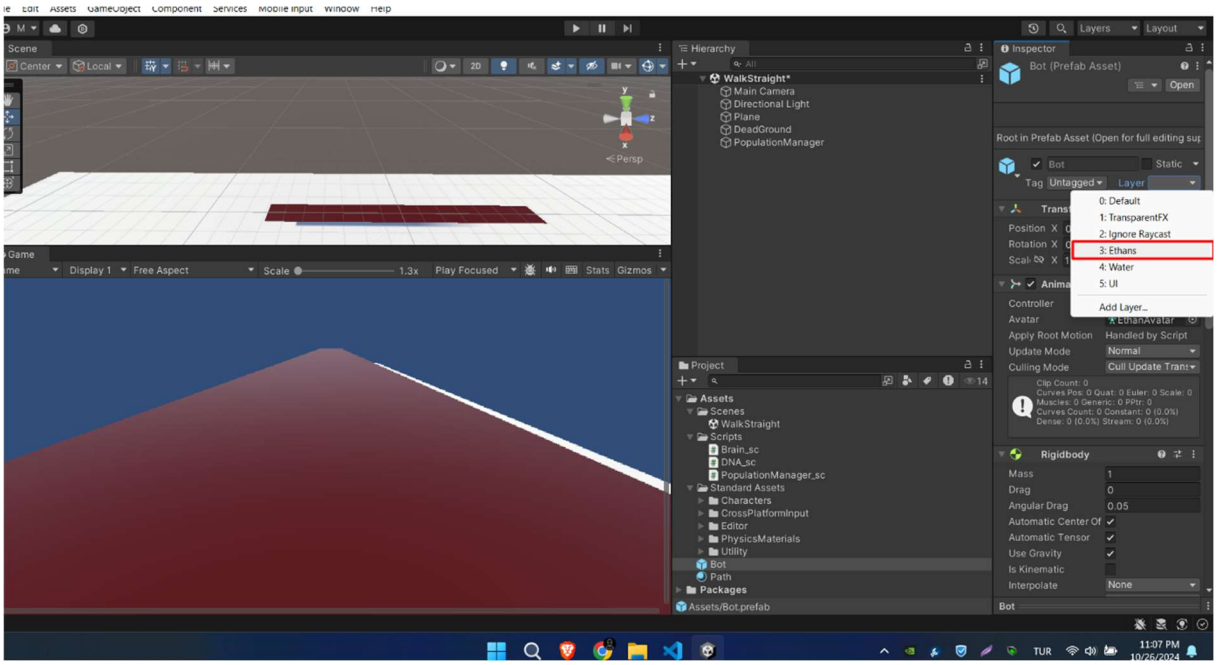
```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityStandardAssets.Characters.ThirdPerson;
5
6   // This script is used to control the AI of the enemy
7   [RequireComponent(typeof(ThirdPersonCharacter))]
    0 references
8   public class Brain_sc : MonoBehaviour
9   {
        1 reference
10      public int DNALength = 1;
        2 references
11      public float timeAlive;
        7 references
12      public DNA_sc dna_sc;
13
        2 references
14      private ThirdPersonCharacter m_Character;
        2 references
15      private Vector3 m_Move;
        0 references
16      private bool m_Jump;
        3 references
17      bool alive = true;
18
        0 references
19      void OnCollisionEnter(Collision obj)
20      {
21          if (obj.gameObject.tag == "dead")
22          {
23              alive = false;
24          }
25      }
26
```

```
0 references
public void Init()
{
    // Initialize DNA
    // 0 forward
    // 1 back
    // 2 left
    // 3 right
    // 4 jump
    // 5 crouch
    dna_sc = new DNA_sc(DNALength, 6);
    m_Character = GetComponent<ThirdPersonCharacter>();
    timeAlive = 0;
    alive = true;
}

0 references
private void FixedUpdate()
{
    // read DNA
    float h = 0;
    float v = 0;
    bool crouch = false;
    bool jump = false;

    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;
    m_Character.Move(m_Move, crouch, jump);
    if (alive)
    {
        timeAlive += Time.deltaTime;
    }
}
```

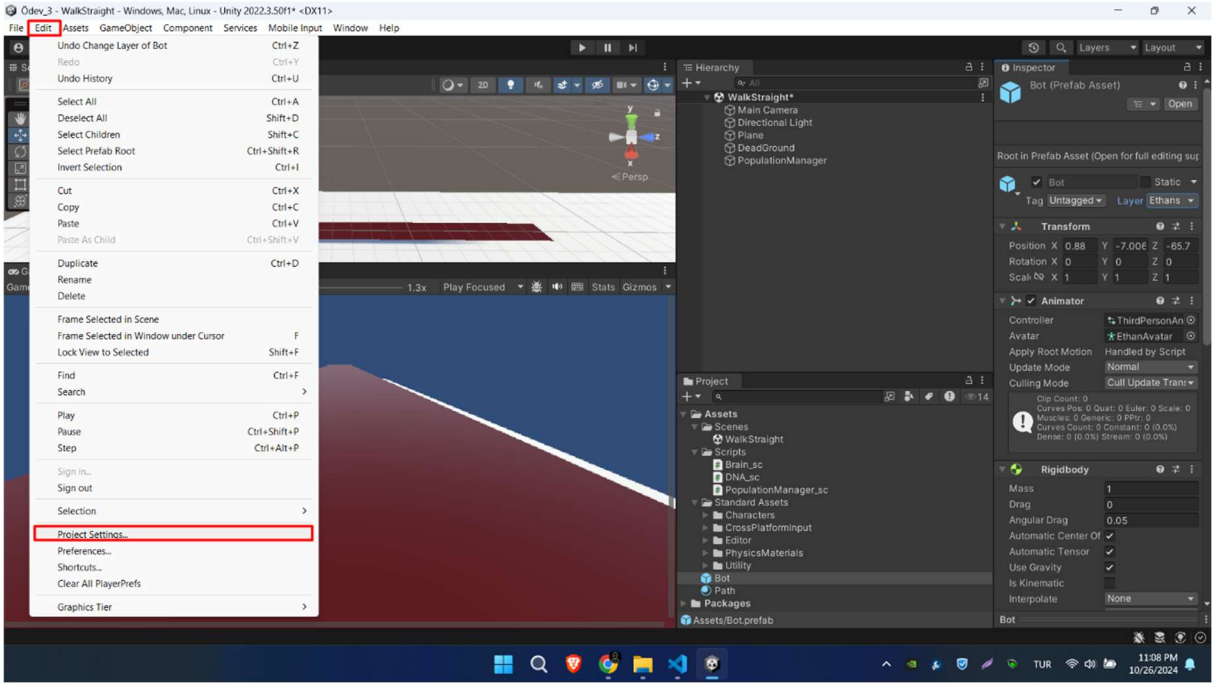
3 – Karakterlerin birbirine çarpma durumunu ayarlama

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

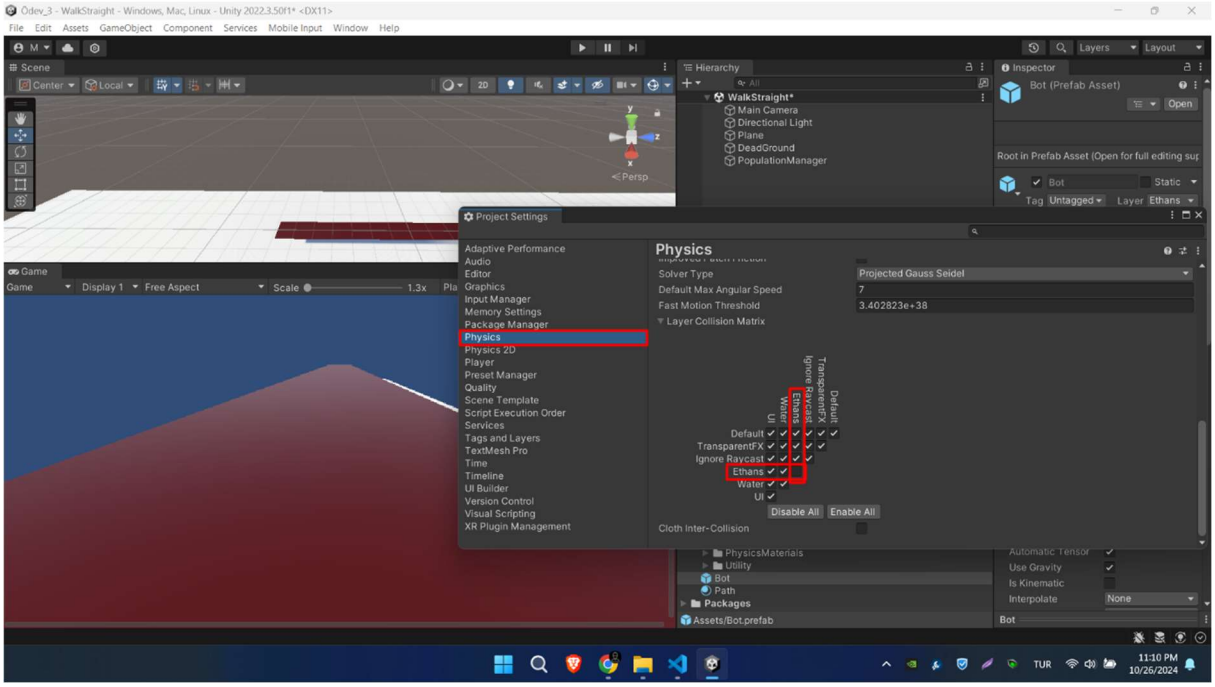



Edit Assets GameObject Component Services Mobile Input Window Help
Scene
Center Local 2D
Persp
Game
Display 1 Free Aspect Scale 1.3x Play Focused Stats Gizmos
Hierarchy
WalkStraight*
Main Camera
Directional Light
Plane
DeadGround
PopulationManager
Project
Assets
Scenes
WalkStraight
Scripts
Brain_sc
DNA_sc
PopulationManager_sc
Standard Assets
Characters
CrossPlatformInput
Editor
PhysicsMaterials
Utility
Bot
Path
Packages
Assets/Bot.prefab
Inspector
Bot (Prefab Asset)
Open
Root in Prefab Asset (Open for full editing sup
Bot
Static
Tag Untagged
Layer
0: Default
1: TransparentFX
2: Ignore Raycast
3: Ethans
4: Water
5: UI
Add Layer...
Controller
Avatar
EthanAvatar
Apply Root Motion
Handled by Script
Update Mode
Normal
Culling Mode
Cull Update Trans
Rigidbody
Mass
Drag
Angular Drag
Automatic Center Of
Automatic Tensor
Use Gravity
Is Kinematic
Interpolate
None
Bot
TUR
11:07 PM
10/26/2024

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
        }
    }
}
```

```
0 references
public class PopulationManager_sc : MonoBehaviour
{

    2 references
    public GameObject botPrefab;
    1 reference
    public int populationSize = 50;
    6 references
    List<GameObject> population = new List<GameObject>();
    4 references
    public static float elapsed = 0;
    1 reference
    public float trialTime = 5;
    2 references
    int generation = 1;

    6 references
    GUIStyle guiStyle = new GUIStyle();

    0 references
    void OnGUI()
    {
        guiStyle.fontSize = 25;
        guiStyle.normal.textColor = Color.white;
        GUI.BeginGroup(new Rect(10, 10, 250, 150));
        GUI.Box(new Rect(0, 0, 140, 140), "Stats", guiStyle);
        GUI.Label(new Rect(10, 25, 200, 30), "Gen: " + generation, guiStyle);
        GUI.Label(new Rect(10, 50, 200, 30), string.Format("Time: {0:0.00}", elapsed), guiStyle);
        GUI.Label(new Rect(10, 75, 200, 30), "Population: " + population.Count, guiStyle);
        GUI.EndGroup();
    }

    2 references
    GameObject Breed(GameObject parent1, GameObject parent2)
    {
        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);
        Brain_sc b = offspring.GetComponent<Brain_sc>();
        if (Random.Range(0, 100) == 1) // mutation
        {
            b.Init();
            b.dna_sc.Mutate();
        }
        else
        {
            b.Init();
            b.dna_sc.Combine(parent1.GetComponent<Brain_sc>().dna_sc, parent2.GetComponent<Brain_sc>().dna_sc);
        }
        return offspring;
    }

```

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

```
1 reference
void BreedNewPopulation()
{
    List<GameObject> sortedList = population.OrderBy(o => (o.GetComponent<Brain_sc>().timeAlive)).ToList();

    population.Clear();

    // breed upper half of sorted list
    for (int i = (int)(sortedList.Count / 2.0f) - 1; i < sortedList.Count - 1; i++)
    {
        population.Add(Breed(sortedList[i], sortedList[i + 1]));
        population.Add(Breed(sortedList[i + 1], sortedList[i]));
    }

    // destroy all parents and previous population
    for (int i = 0; i < sortedList.Count; i++)
    {
        Destroy(sortedList[i]);
    }
    generation++;
}
```

```
    // Start is called before the first frame update
    0 references
    void Start()
    {
        for (int i = 0; i < populationSize; i++)
        {
            Vector3 startingPos = new Vector3(this.transform.position.x + Random.Range(-2, 2),
                this.transform.position
                this.transform.position (field) GameObject PopulationManager_sc.botPrefab
            GameObject b = Instantiate(botPrefab, startingPos, this.transform.rotation);
            b.GetComponent<Brain_sc>().Init();
            population.Add(b);
        }
    }

    // Update is called once per frame
    0 references
    void Update()
    {
        elapsed += Time.deltaTime;
        if (elapsed >= trialTime)
        {
            BreedNewPopulation();
            elapsed = 0;
        }
    }
```

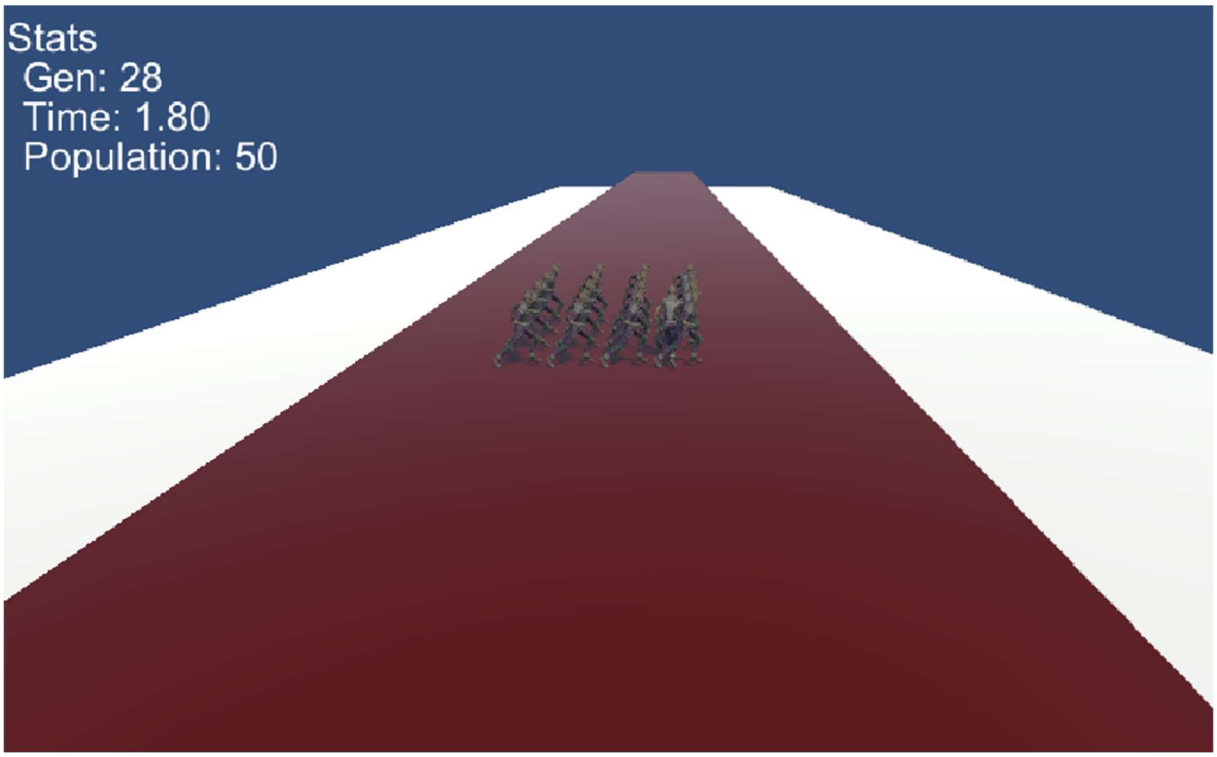
```
    2 references
    private ThirdPersonCharacter m_Character;
    2 references
    private Vector3 m_Move;
    0 references
    private bool m_Jump;
    3 references
    bool alive = true;

    0 references
    public float distanceTravelled;
    1 reference
    Vector3 startPosition;

    0 references
    void OnCollisionEnter(Collision obj)
    {
        if (obj.gameObject.tag == "dead")
        {
            alive = false;
        }
    }
```

```
0 references
public void Init()
{
    // Initialize DNA
    // 0 forward
    // 1 back
    // 2 left
    // 3 right
    // 4 jump
    // 5 crouch
    dna_sc = new DNA_sc(DNALength, 6);
    m_Character = GetComponent<ThirdPersonCharacter>();
    timeAlive = 0;
    alive = true;
    startPosition = this.transform.position;
}
```

```
0 references
46  private void FixedUpdate()
47  {
48      // read DNA
49      float h = 0;
50      float v = 0;
51      bool crouch = false;
52      bool jump = false;
53
54      if (dna_sc.GetGene(0) == 0) v = 1;
55      else if (dna_sc.GetGene(0) == 1) v = -1;
56      else if (dna_sc.GetGene(0) == 2) h = -1;
57      else if (dna_sc.GetGene(0) == 3) h = 1;
58      else if (dna_sc.GetGene(0) == 4) jump = true;
59      else if (dna_sc.GetGene(0) == 5) crouch = true;
60
61      m_Move = v * Vector3.forward + h * Vector3.right;
62      m_Character.Move(m_Move, crouch, jump);
63      if (alive)
64      {
65          timeAlive += Time.deltaTime;
66          distanceTravelled = Vector3.Distance(this.transform.position,
67          startPosition);
68      }
69  }
```

```
1 reference
void BreedNewPopulation()
{
    List<GameObject> sortedList = population.OrderBy(o => (o.GetComponent<Brain_sc>().distanceTravelled)).ToList();

    population.Clear();

    // breed upper half of sorted list
    for (int i = (int)(sortedList.Count / 2.0f) - 1; i < sortedList.Count - 1; i++)
    {
        population.Add(Breed(sortedList[i], sortedList[i + 1]));
        population.Add(Breed(sortedList[i + 1], sortedList[i]));
    }

    // destroy all parents and previous population
    for (int i = 0; i < sortedList.Count; i++)
    {
        Destroy(sortedList[i]);
    }
    generation++;
}
```

6 – Oyunu başlatıyoruz.



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