

Unity ile Script Oluřturma

Selin Esra z

Bilgisayar Mühendisliđi

20360859089

4. Sınıf

23.05.2024

İçindekiler

1. Unity Nedir?
2. Script Nedir?
3. Unity ve C#
4. MonoBehaviour
5. Temel Metotlar
6. Collision ve Trigger
7. Input Kullanımı
8. Uygulama
9. Kaynakça

Unity Nedir?

Unity, oyun geliřtiricilerin 2D ve 3D oyunlar, sim¼lasyonlar, görsel efektler ve diğeri interaktif deneyimler oluřturmasını sađlayan pop¼ler bir oyun motorudur. Ücretsiz olarak indirilebilen Unity, zengin bir araç seti sunar ve farklı platformlarda (bilgisayarlar, mobil cihazlar, konsollar vb.) çalıřabilen oyunlar geliřtirmenize olanak tanır.

3/17

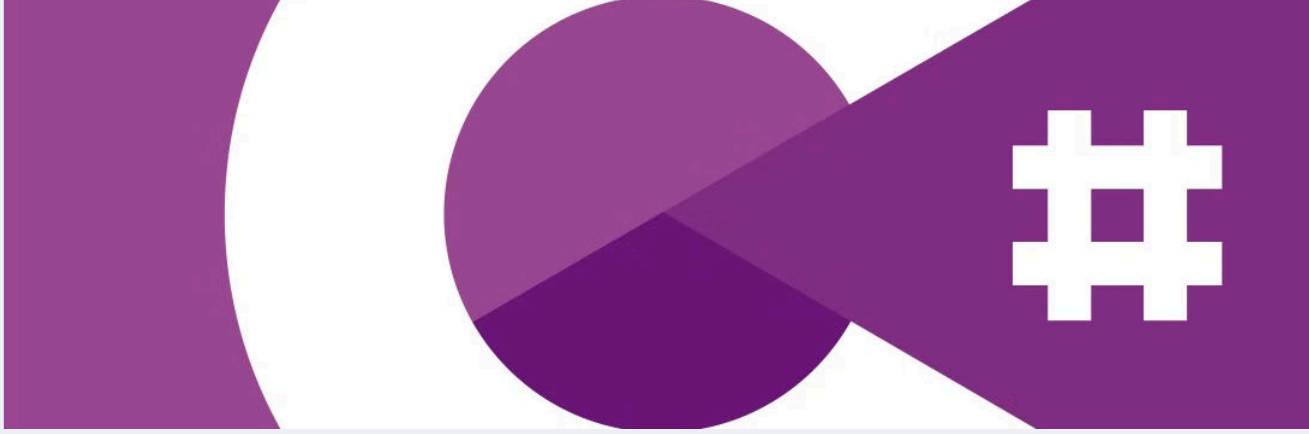


Script Nedir?

Unity'de script, oyun nesnelere davranış kazandırmak için kullanılan kod parçalarıdır.

Scriptler, oyun dünyasında dinamik ve interaktif unsurlar oluşturmak için kullanılır. Örneğin, bir karakterin hareket etmesi, bir nesnenin fiziksel tepkisi veya bir kullanıcı etkileşimi scriptler ile kontrol edilir.

4/17



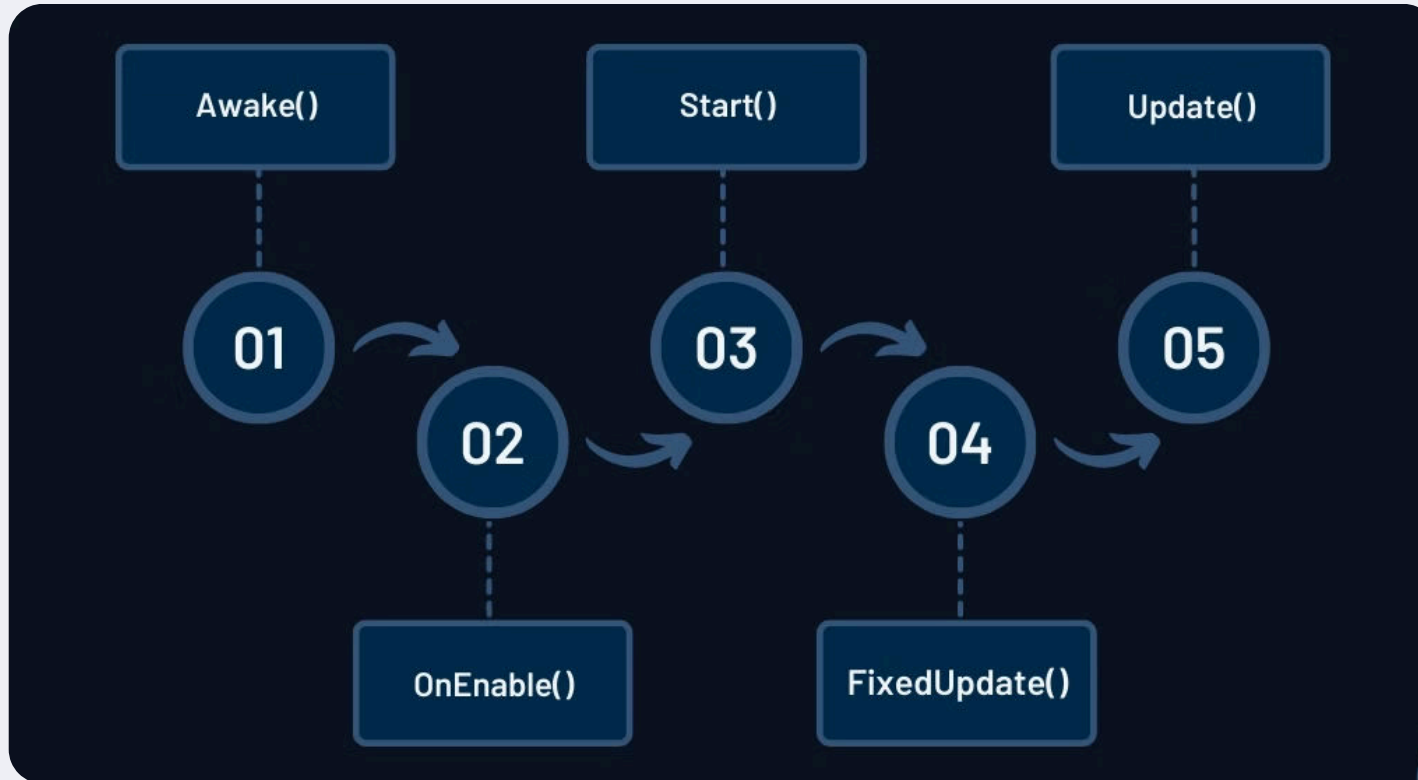
Unity ve C#

Unity, oyun ve uygulama geliştirme platformu olarak C# programlama dilini kullanır. C# dili, Unity'nin güçlü ve esnek bir şekilde kullanılabilmesini sağlar, çünkü modern ve nesne yönelimli bir dildir. Unity'deki scriptlerin çoğu, oyun nesnelerine dinamik davranışlar kazandırmak için C# ile yazılır. Bu ilişki sayesinde, geliştiriciler kompleks oyun mekaniği ve etkileşimleri kolayca oluşturabilirler.

5/17

MonoBehaviour

Unity'deki scriptlerin çoğu, MonoBehaviour sınıfından türetilir ve bu sınıfın özelliklerini ve metodlarını kullanır.



Temel Metotlar

Awake, Start, Update

Awake()

- Oyun çalışmaya başladığında (oyun uyandığı anda) bütün fonksiyonlardan önce çalışan fonksiyondur.
- Awake metodu, nesne oluşturulduğunda çağrılır ve nesnenin oyun dünyasında yerini aldığı anlamına gelir.

Start()

- Awake'den sonra ve ilk frame'den önce çağrılır.
- Genellikle, oyun başında gerekli başlangıç ayarlarını yapmak ve nesne üzerindeki bileşenleri etkinleştirmek için kullanılır.

Update()

- Her frame'de bir kez çağrılır.
- Çalışma hızı oyunun çalıştığı cihaza göre değişir. Bunun nedeni Update() fonksiyonunun belirli bir çalışma hızına sahip olmamasıdır.
- Kullanıcı girişi, fizik etkileşimleri ve oyun mantığı burada güncellenir.

```
Assembly-CSharp  MazeCell  Start()
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class MazeCell : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11      }
12
13      // Update is called once per frame
14      void Update()
15      {
16
17      }
18  }
19
```


Temel Metotlar

FixedUpdate ve LateUpdate

FixedUpdate()

- Awake ve Start fonksiyonları çalıştıktan sonra sürekli olarak çalışan ve sabit bir çalışma hızına sahip olan fonksiyondur
- Genel olarak fizikle alakalı işlemlerde kullanılır.Örnek vermek gerekirse iki objenin çarpışması gibi işlemlerde kullanılır.

LateUpdate()

- Update metodundan sonra her frame'de bir kez çağrılır.
- Çalışma hızı çalıştığı cihazın performansına göre değişir.
- Genel olarak kamera kontrolleri gibi işlemlerde tercih edilir.Örneğin obje hareket ettiğinde kamera da onu takip edicektir.

Collision ve Trigger

Collision

- **Collision (Çarpışma):** İki fiziksel nesne birbirine temas ettiğinde gerçekleşen etkileşimdir. Örneğin, iki araba çarpıştığında veya oyuncu karakterinin duvara çarptığında bir collision (çarpışma) meydana gelir. Bu tür etkileşimler, genellikle fiziksel çarpışmaları ve çarpışma sonuçlarını kontrol etmek için kullanılır.

Trigger

- **Trigger (Tetikleyici):** Belirli bir bölgede veya alanın içinde bulunan bir nesnenin başka bir nesneyle etkileşime girmesini sağlayan bir bölge veya alan. Trigger, çarpışma olmadan gerçekleşir ve bir nesne tetikleyici bölgeye girdiğinde veya çıktığında belirli bir eylemi gerçekleştirmek için kullanılabilir. Örneğin, oyuncu karakteri bir kapıya yaklaştığında kapının açılması gibi.

Input Kullanımı

Kullanıcıdan input alma ve buna göre aksiyon gerçekleştirme. Örneğin, tuşa basıldığında zıplama.

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        GetComponent<Rigidbody>().AddForce(Vector3.up * 5, ForceMode.Impulse);
    }
}
```

11/17

Kaynakça

Learn

1)Unity Ana Fonksiyonlar(Unity-101)

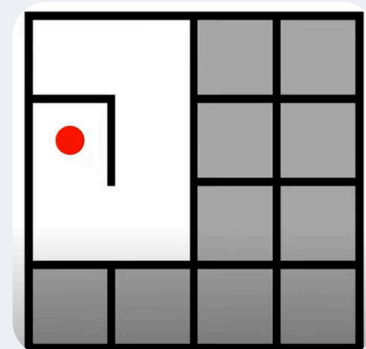
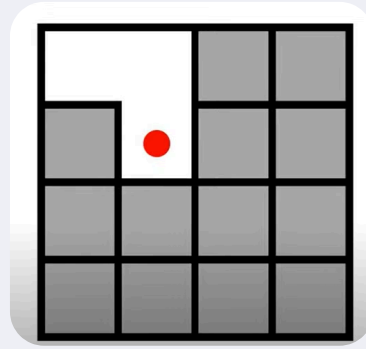
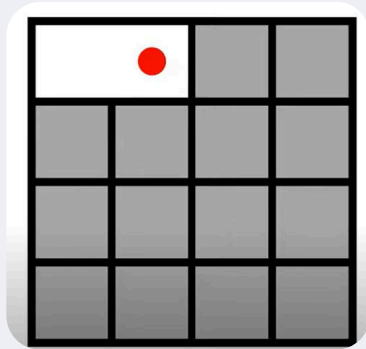
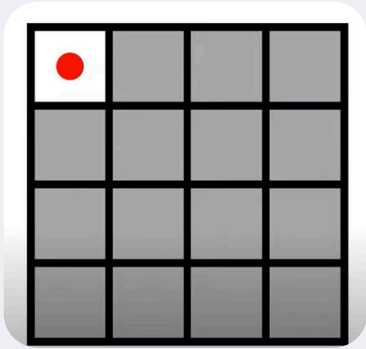
2)Unity Temas Fonksiyonları(Unity-101)

12/17

Uygulama

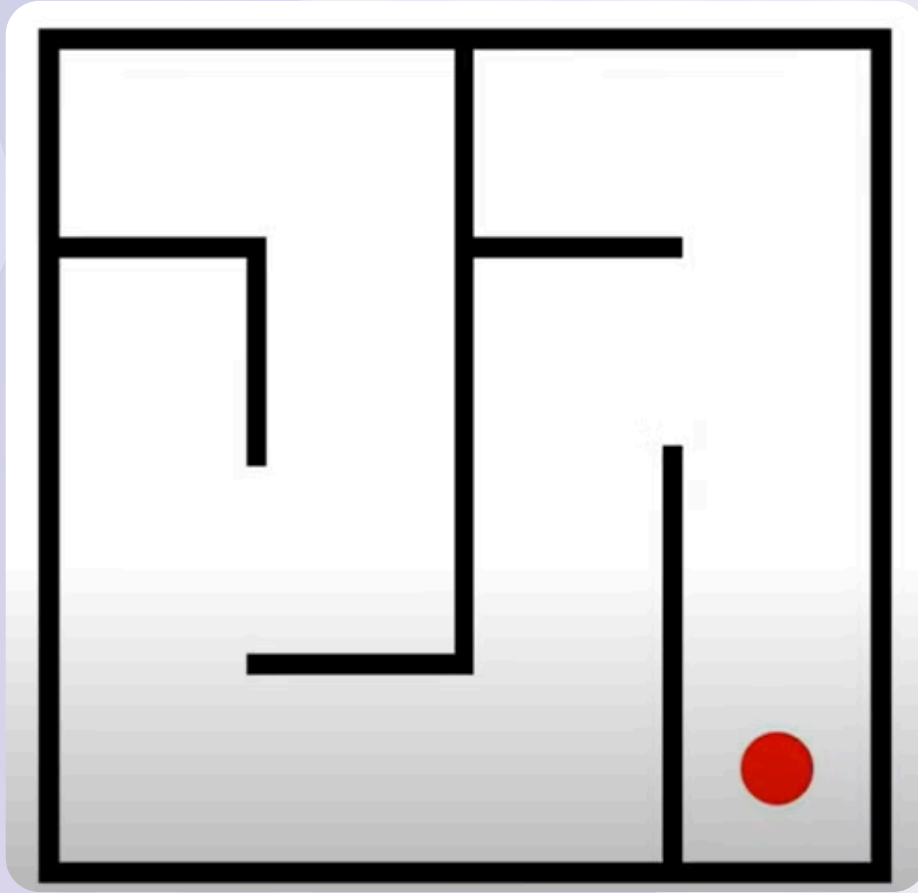
13/17

Uygulama



14/17

Uygulama



Soru-Cevap

16/17

Teşekkürler!

17/17