

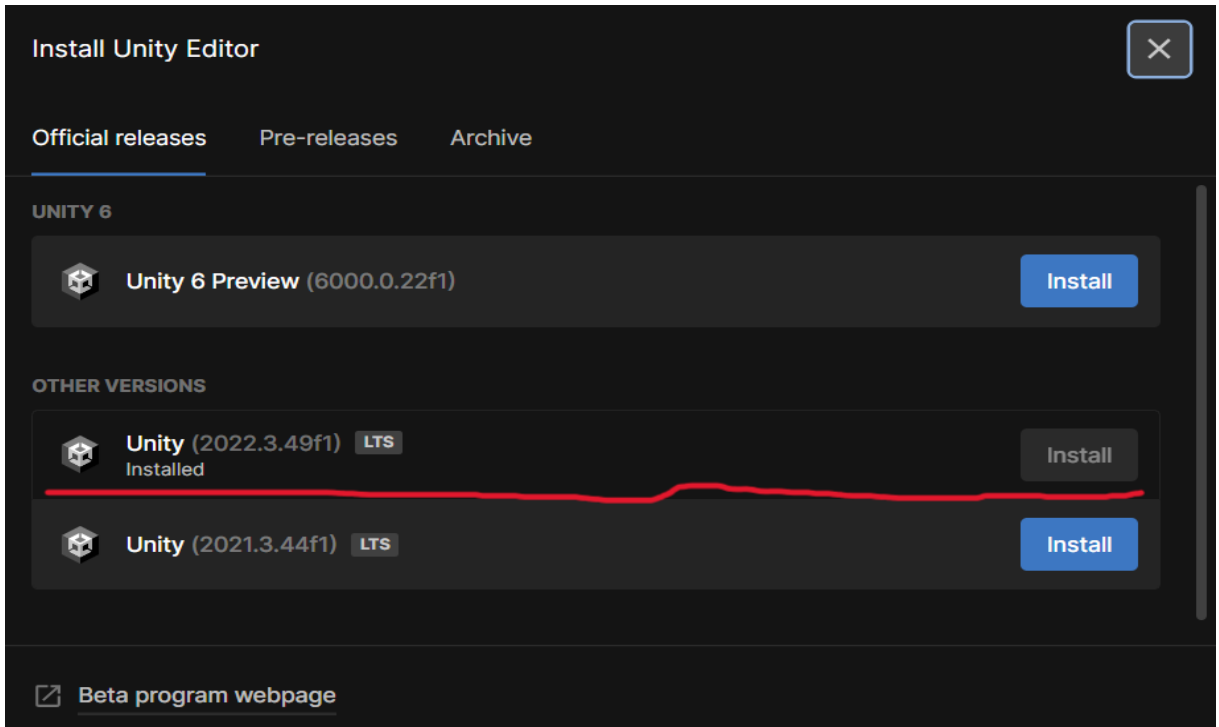
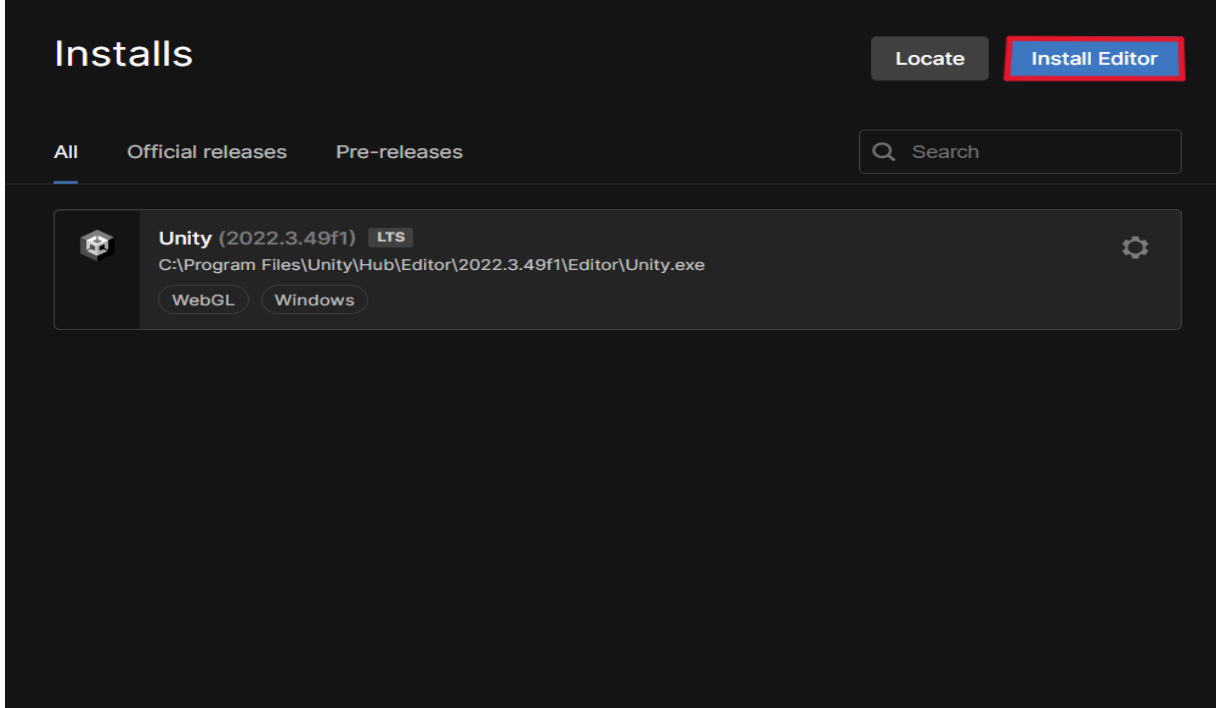
Yusuf Güney 22360859041

Oyun Programlama

Hafta 1 Rapor

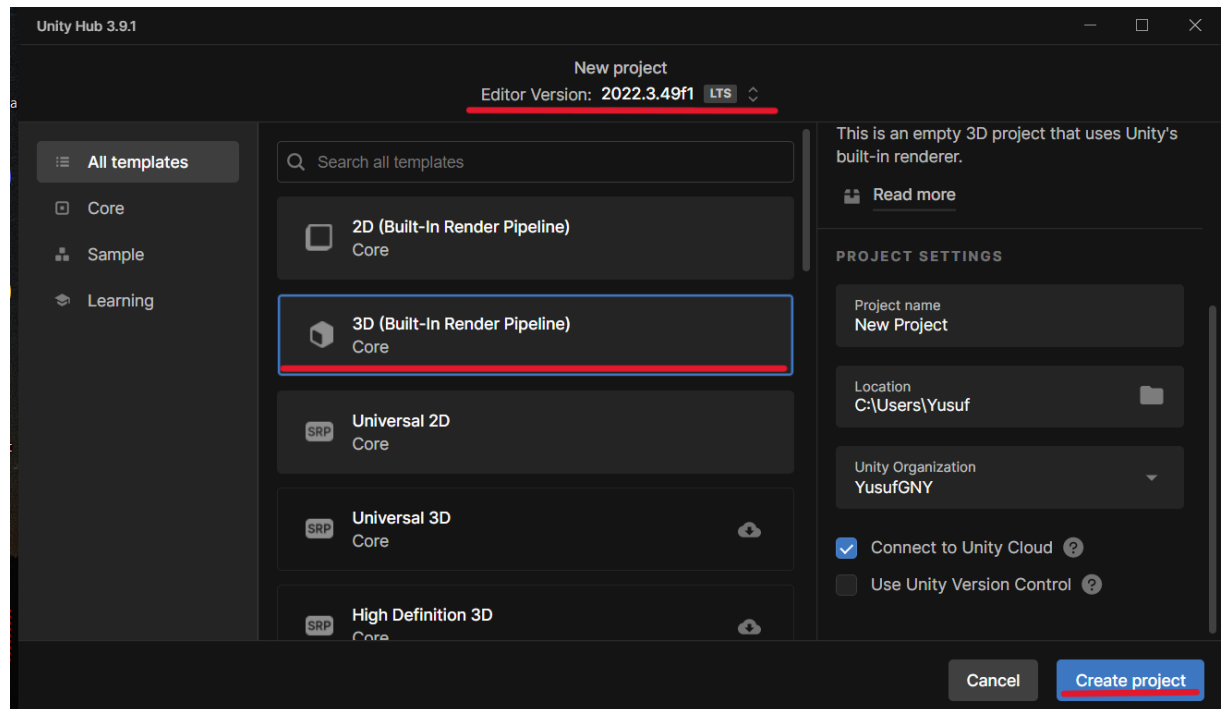
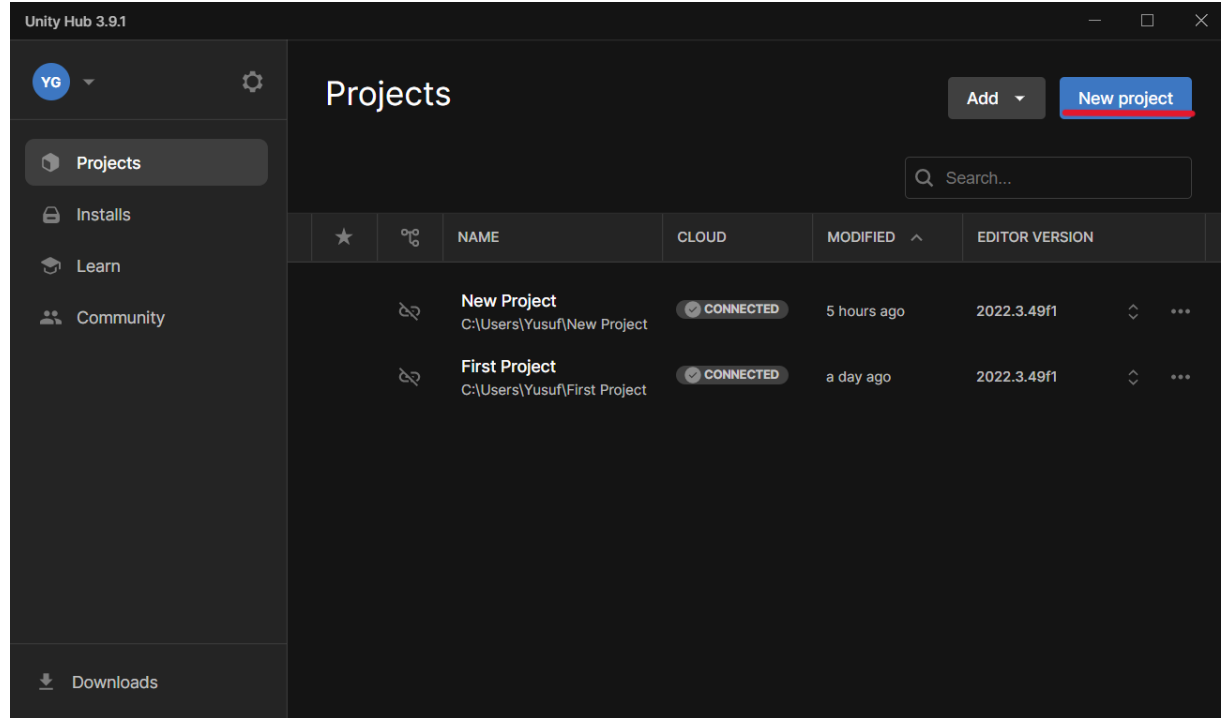
1-Unity'de yeni proje oluşturma

Unity hesabına giriş yaptıktan sonra Installs sekmesinden Installs Editor'e tıklayarak Unity(2022.3.49f1) i kuralım.



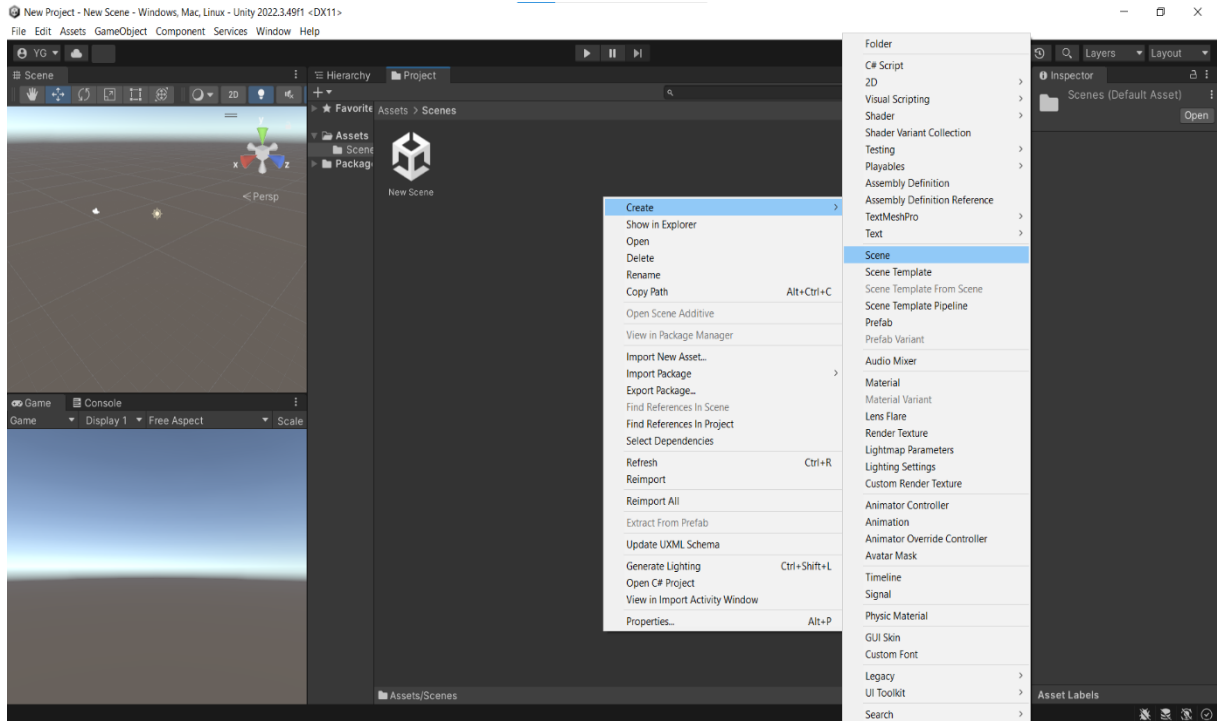
Daha sonra Project sekmesinden New Project'e tıklayarak yeni proje oluşturabiliriz.

Projelerden 3D(Built-In Render Pipeline) ı seçiyoruz ver Create Project'te tıklayarak projemizi başlatıyoruz burada dikkat etmemiz gereken Editor Versionu 2022.3.49f1 girmeliyiz.

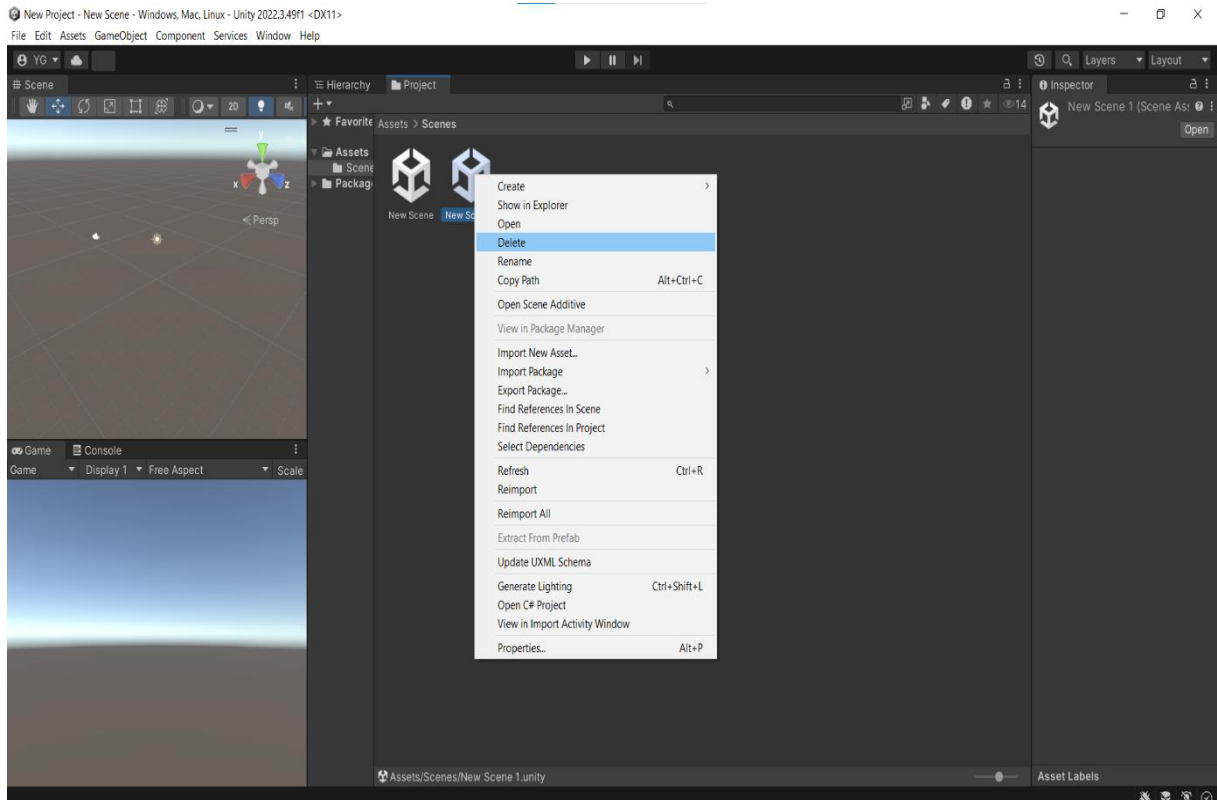


2- Sahne ekleme ve silme

Proje sayfasında sahne eklemek için Project bölümünde Assets dosyasının içindeki Scene dosyasına sağ tıklayarak Create bölümünden scene'ye tıklayarak yeni bir sahne ekleme işlemini yerine getirebiliriz.

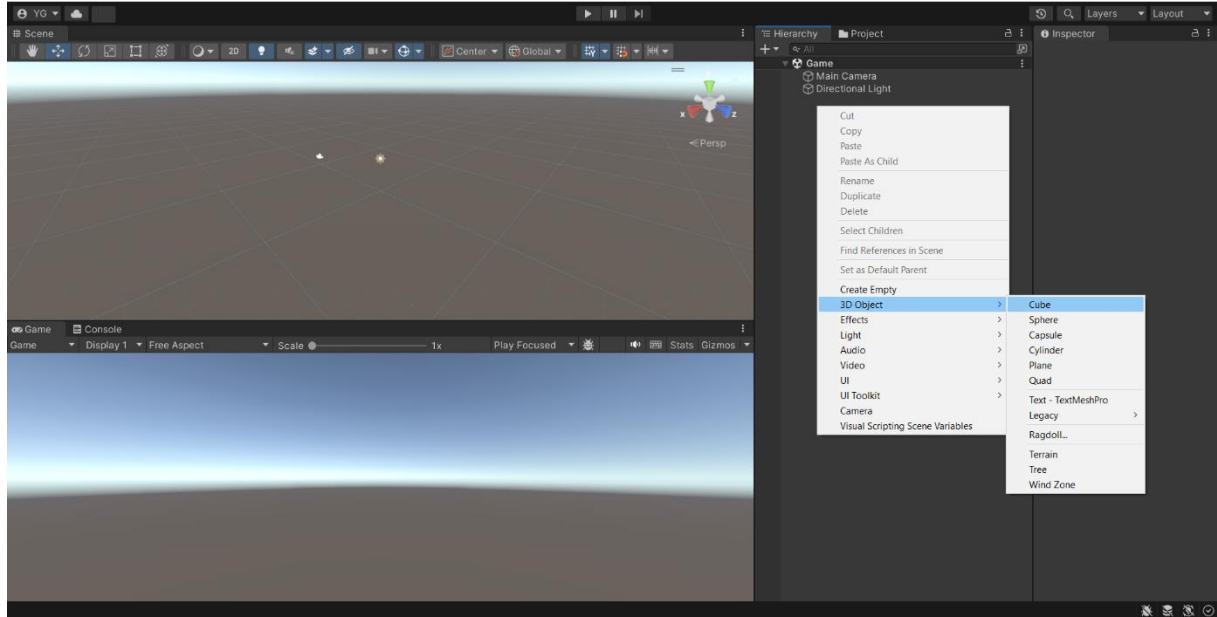


Sahne silme işleminde ise oluşturulan veya var olan sahnenin üzerine gelip sağ tıklayıp Delete seçeneğine tıklayarak sahneyi silebiliriz.



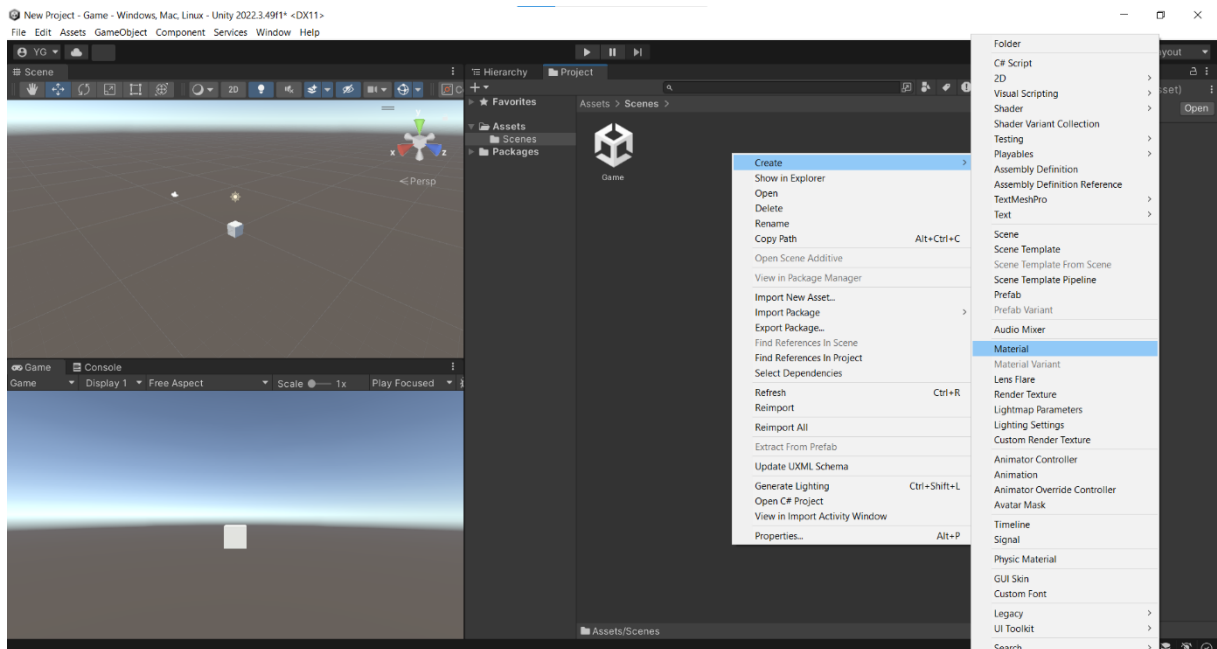
3- Sahneye nesne(küp) ekleme

Proje sayfasında Hierarchy sekmesinde sağ tıklayarak 3D Object bölümünden Cube seçeneğine tıklayarak yeni bir nesne oluşturabiliriz. Bu işlem sahneye bir küp ekleyecektir aynı nesneden daha fazla eklemek için sahnede cismin üzerine tıklayıp CTRL + D tuşları kullanılabilir.

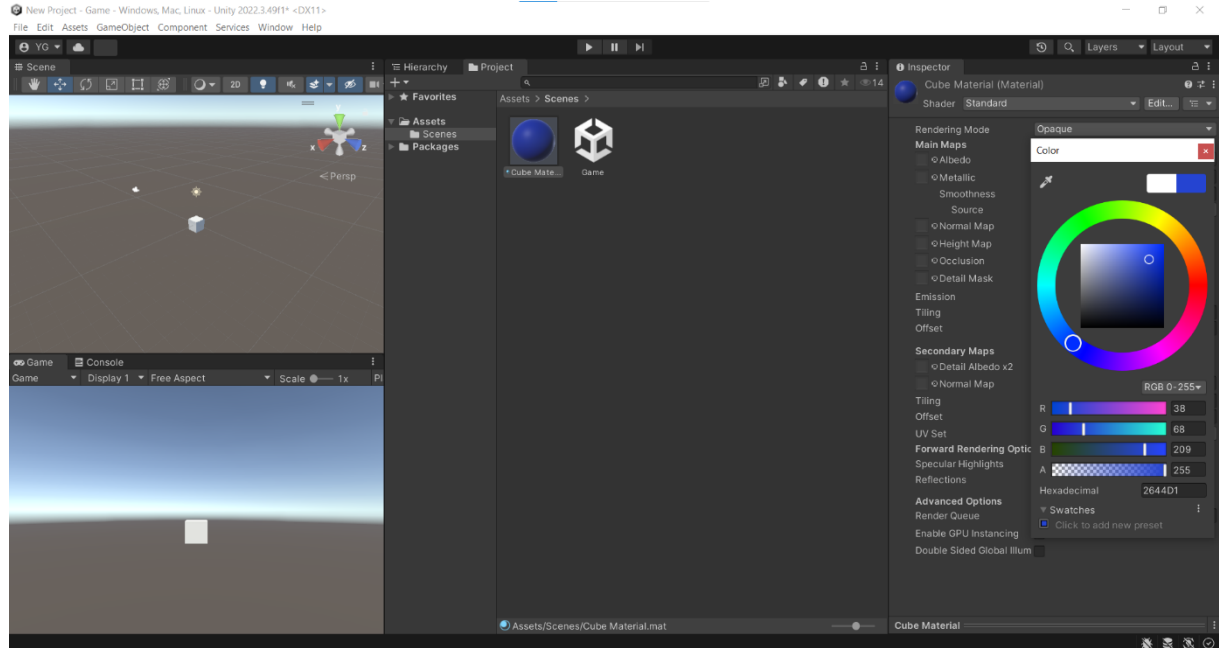


4 – Sahnedeki nesneye materyal ekleme

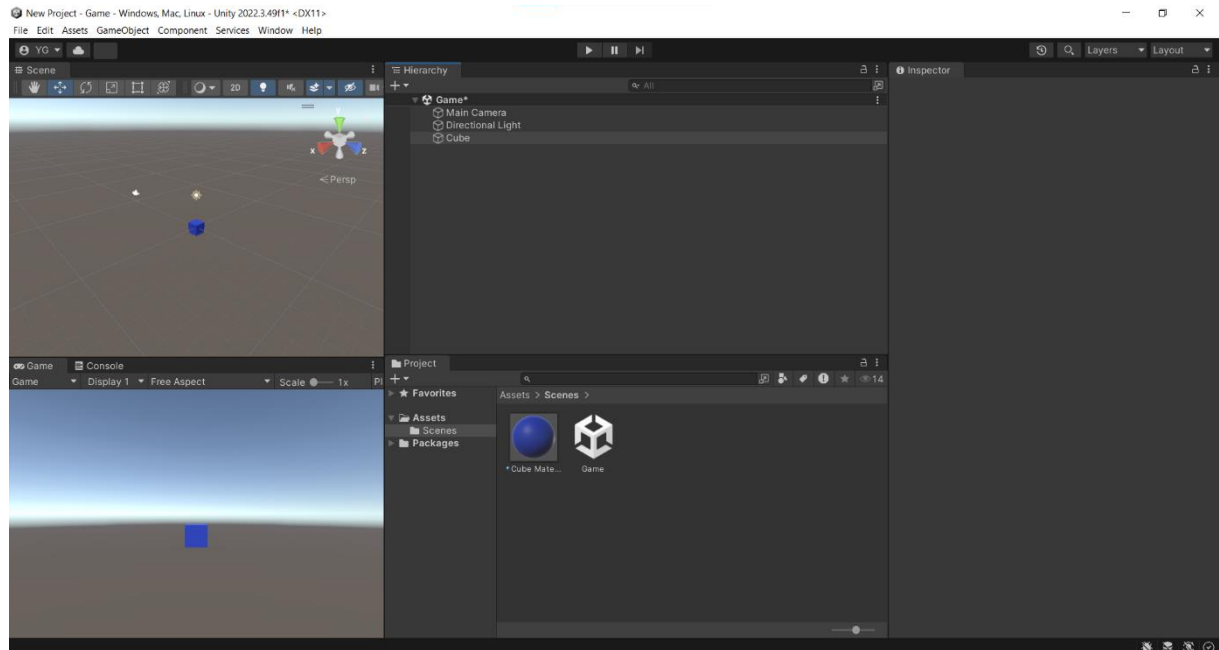
Sahnedeki nesneye materyal eklemek için önce bir materyal oluşturmalıyız Scenes klasörünün içine sağ tıklayarak Create bölümünden Material’i seçerek yeni bir materyal oluşturabiliriz.



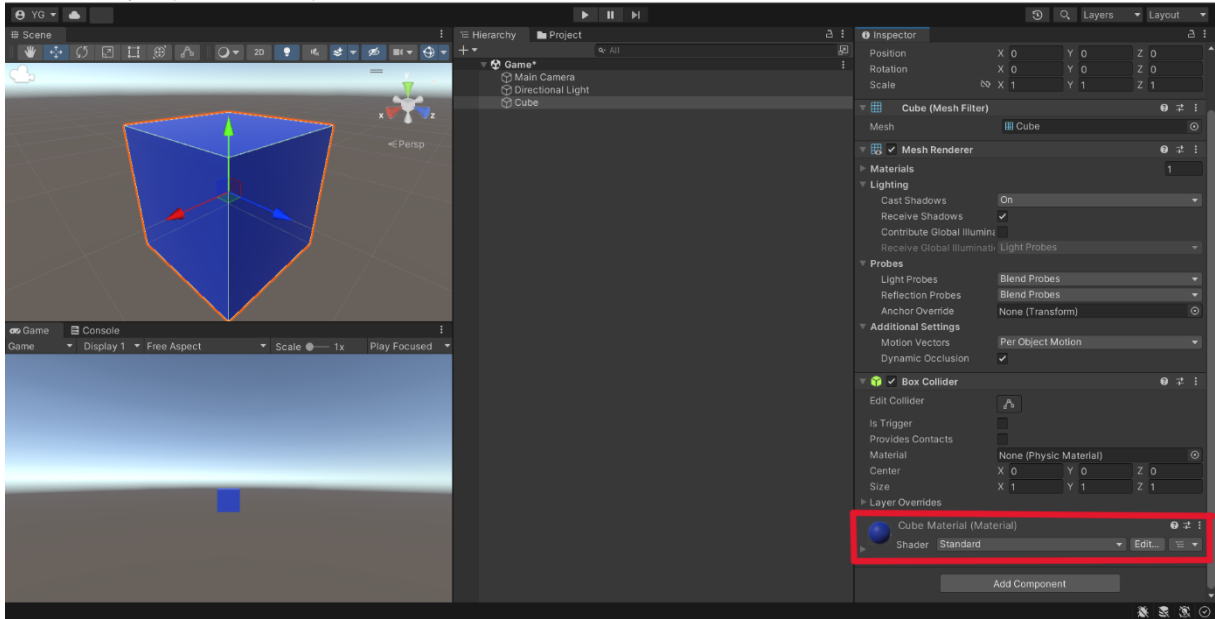
Daha sonra bu materyale renk ekleyebilmek için Material'e tıklayarak Inspector sekmesinden materyale verilecek rengi ayarlayabiliriz.



Nesneye materyali eklemek için Scenes altındaki Cube material dosyamızı Hierarchy altındaki Cube nesnesine sürükleyerek nesneye materyal ekleme işlemini tamamlayabiliriz.

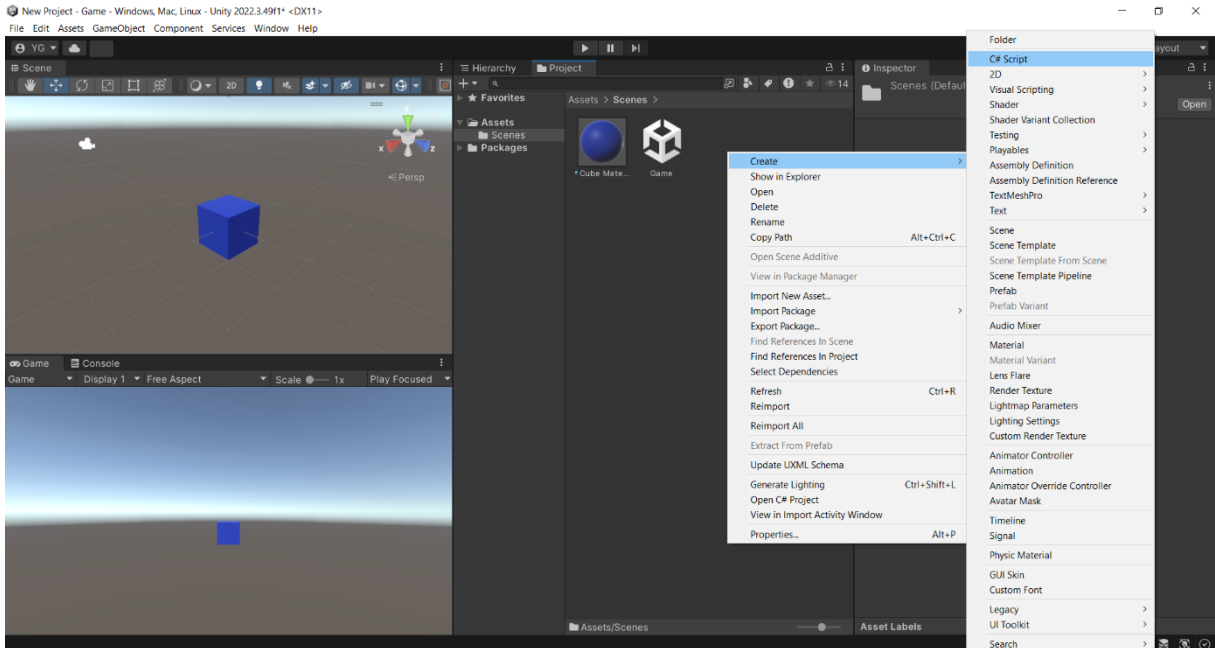


Materyalin nesneye eklendiğini anlayabilmek için Hierarchy bölümünden Cube nesnesine tıklayarak Inspector sekmesinde materyalin nesneye eklendiğini görebiliriz.



5 – Sahnedeki nesneye script ekleme

Öncelikle script oluşturmak için Scenes dosyasının içine sağ tıklayarak Create bölümünden C# Script'e tıklayarak yeni bir script dosyası oluşturalım

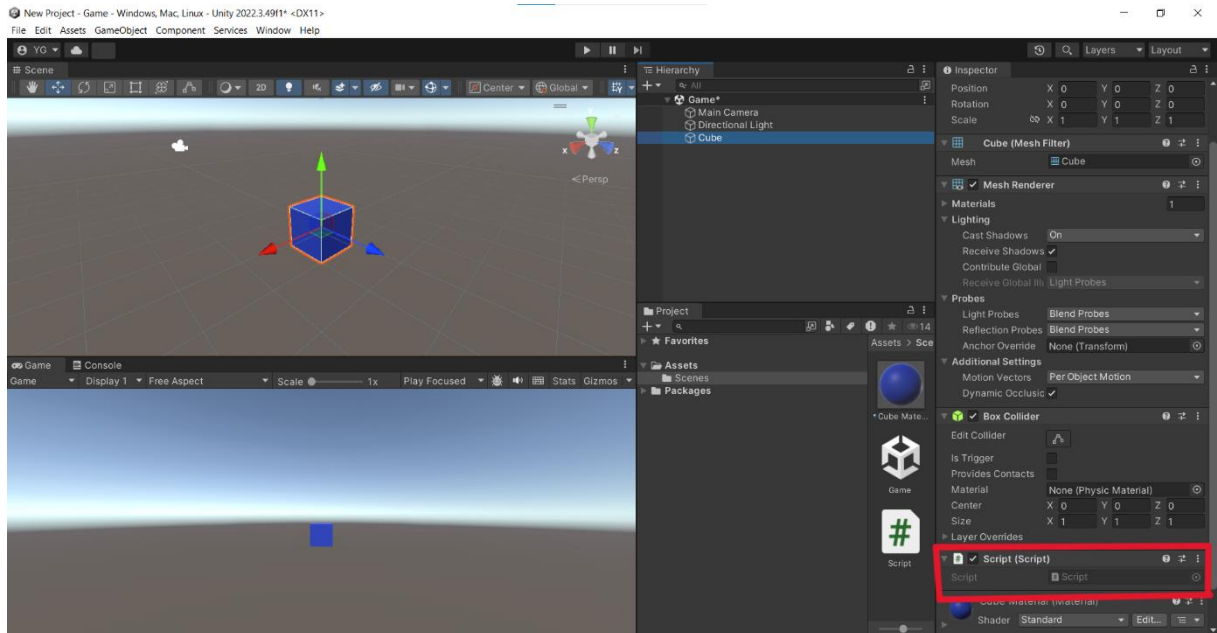


Script dosyasını Visual Studio Code editöründe açmak için ayarlar kısmından varsayılan editörün Visual Studio Code olduğunu kontrol ediniz.

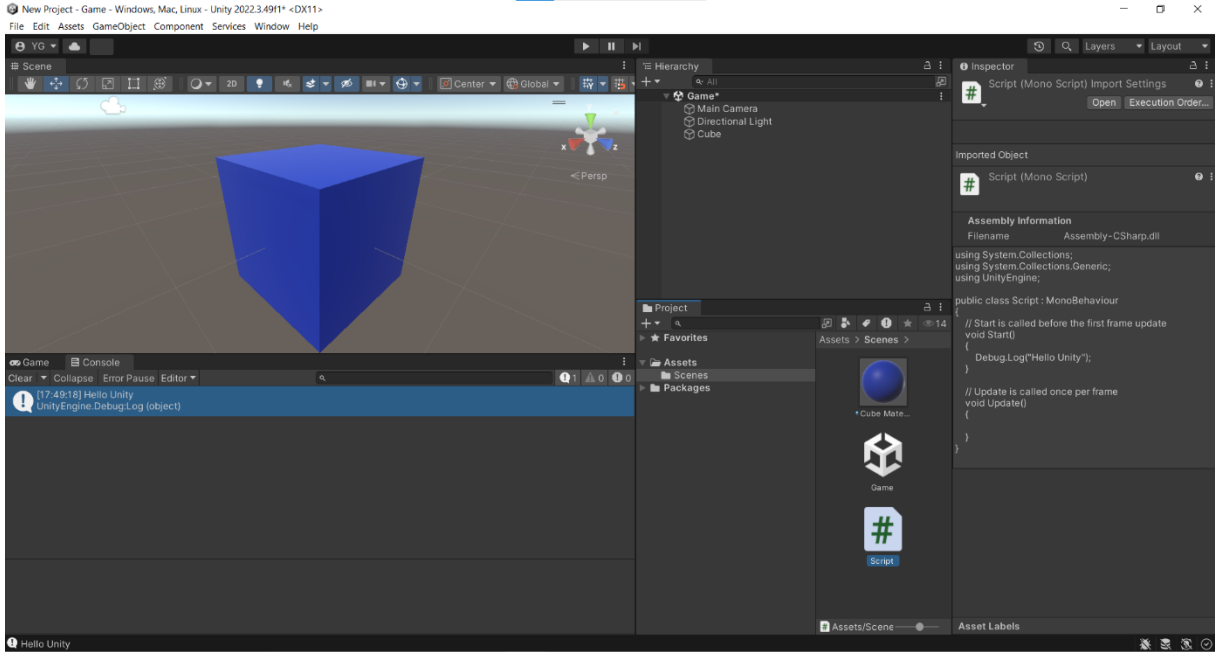
Script dosyasının ii bařlangıta ařağıdaki řekildeki gibi gzkr burada start fonksiyonu oyun bařladıėında bir kez update fonksiyonu ise her frame'de bir kez alıřır

```
Script.cs
Assets > Scenes > Script.cs > Script > Start
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 0 references
6 public class Script : MonoBehaviour
7 {
8     // Start is called before the first frame update
9     0 references
10    void Start()
11    {
12        Debug.Log("Hello Unity");
13    }
14
15    // Update is called once per frame
16    0 references
17    void Update()
18    {
19    }
20 }
```

Nesneye scripti eklemek iin script dosyasını Hierarchy sekmesindeki cube nesnesine taşıyabilirsiniz. Eklendiėini grmek iin de Inspector sekmesinde řekildeki gibi grebilirsiniz.



Son olarak script dosyasının çalıştığını anlayabilmek için start fonksiyonunu içerisine konsola Hello Unity yazdıran kodu çalıştıralım eğer çıktı konsolda çıkmazsa script dosyasında bir hata var demektir.



6- Script ile nesnenin konumunu bir kez değiştirme (Start Fonksiyonu)

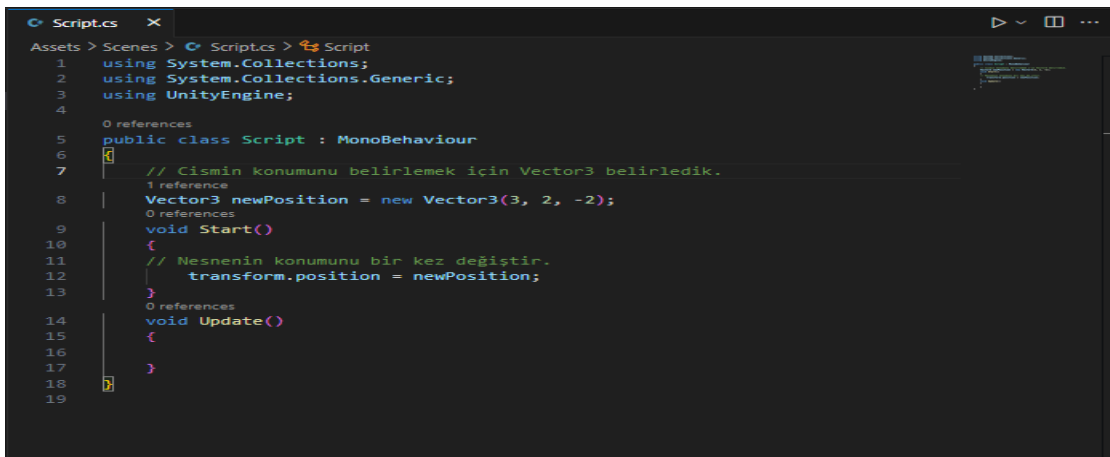
Bu işlemi isteyen C# kodu aşağıda görülmektedir

newPosition: Nesnenin taşınmasını istediğimiz yerin konumunu belirten bir Vector3 değişkenidir

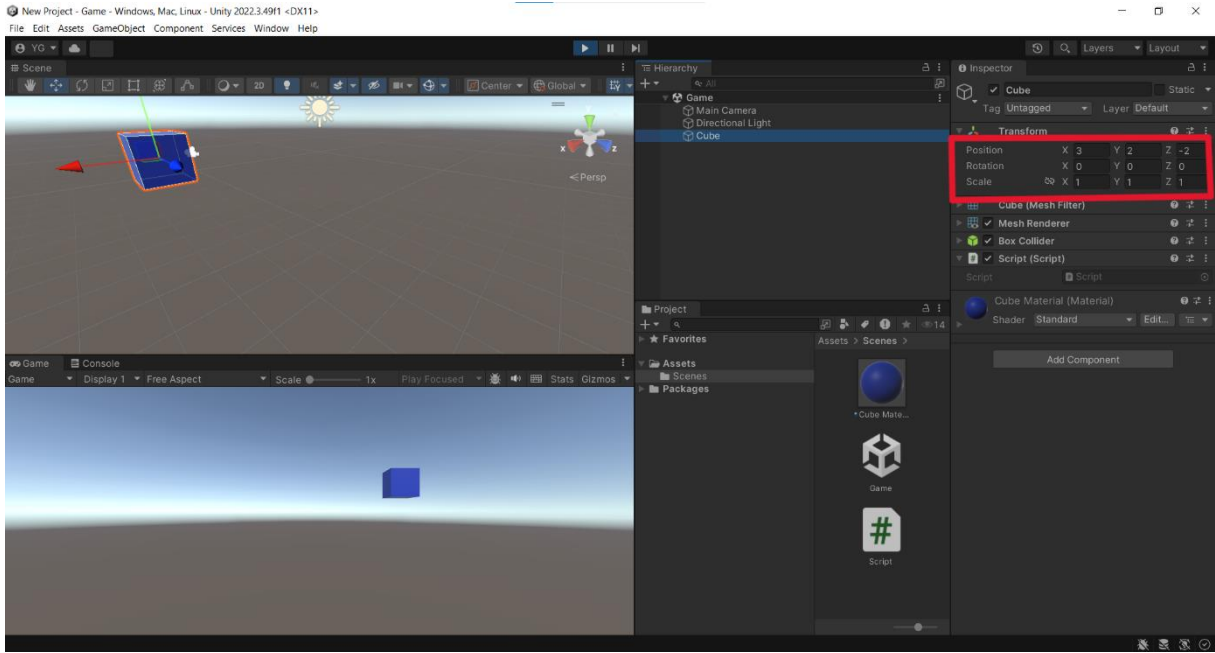
transform.position: Nesnenin şu anki pozisyonunu temsil eder bunu newPosition ile değiştirerek nesnenin konumunu ayarlayabiliriz

Start fonksiyonu sadece oyunu başladığında bir kez çağırılabilir bu yüzden nesnenin konumu bir defaya mahsus değiştirilebilir.

Oyun başladığında verilen koordinatlara göre nesnenin konumu değiştirilecektir.



Cismin konumunun deęişimini proje üzerinden görünümü



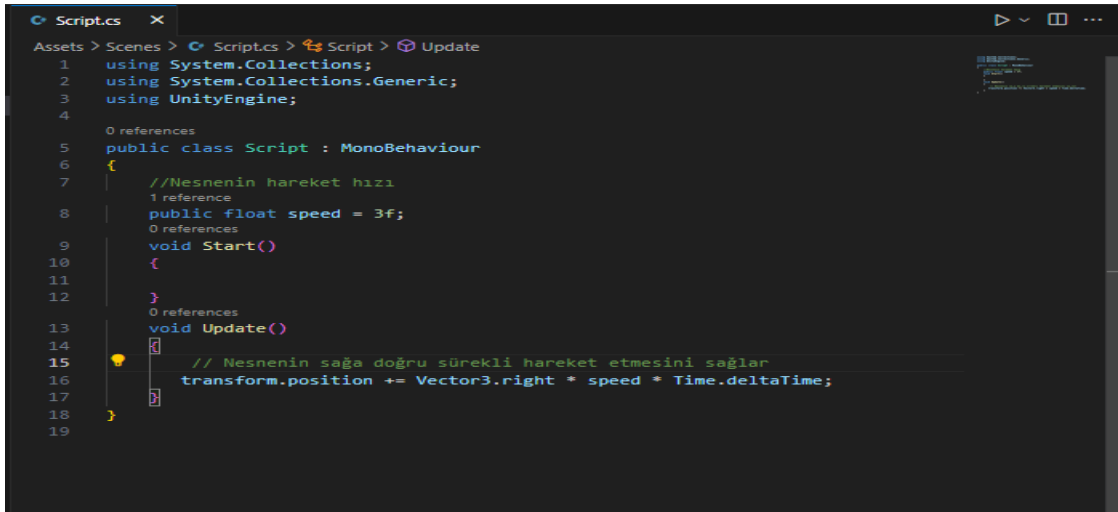
7-Script ile nesnenin konumunu sürekli deęiştirme (Update fonksiyonu)

Cismin konumunu sürekli deęiştiren kod aşağıda gözükmemektedir.

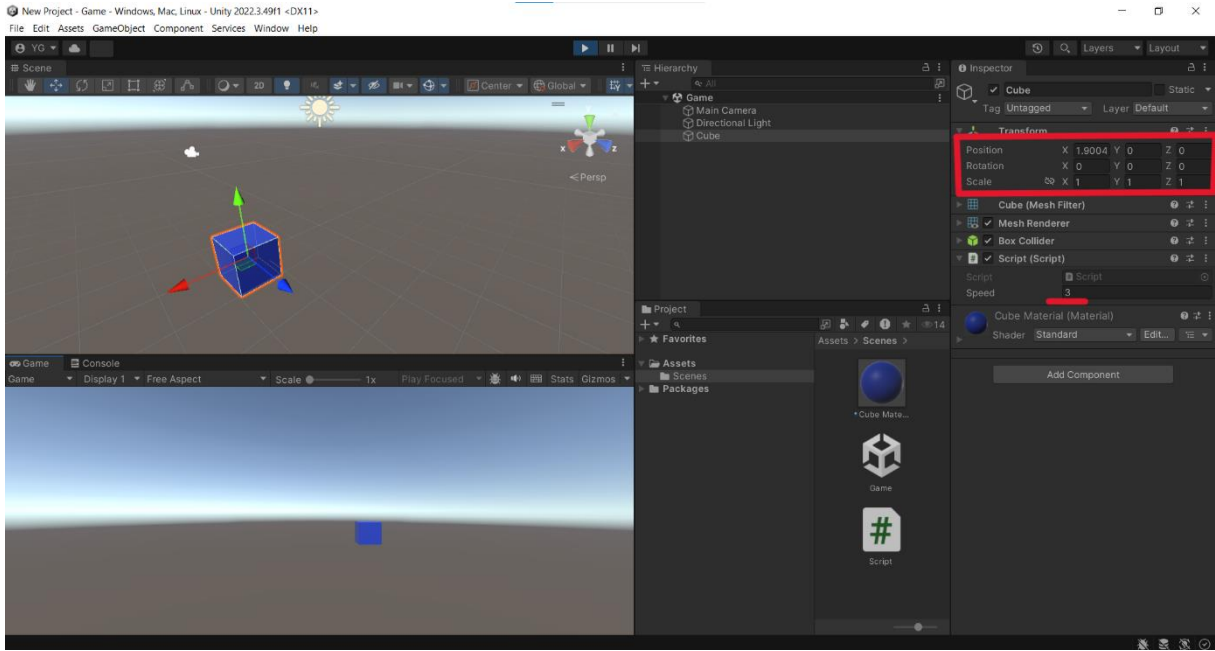
speed: Nesnenin hareket hızını belirler.

Vector3.right: Cismin x ekseninde sağa doğru hareketini ifade eder. Eğer farklı bir yöne hareket etmek isterseniz Vector3.left, Vector3.up, Vector3.down gibi vektörleri kullanabilirsiniz.

Time.deltaTime: Bu, frame rate'ine baęlı olarak hareketi sabit hızda yapar. Yani her frame'de geçen süre ile çarpılarak, nesnenin hızının frame hızına baęlı olmaması saęlanır.



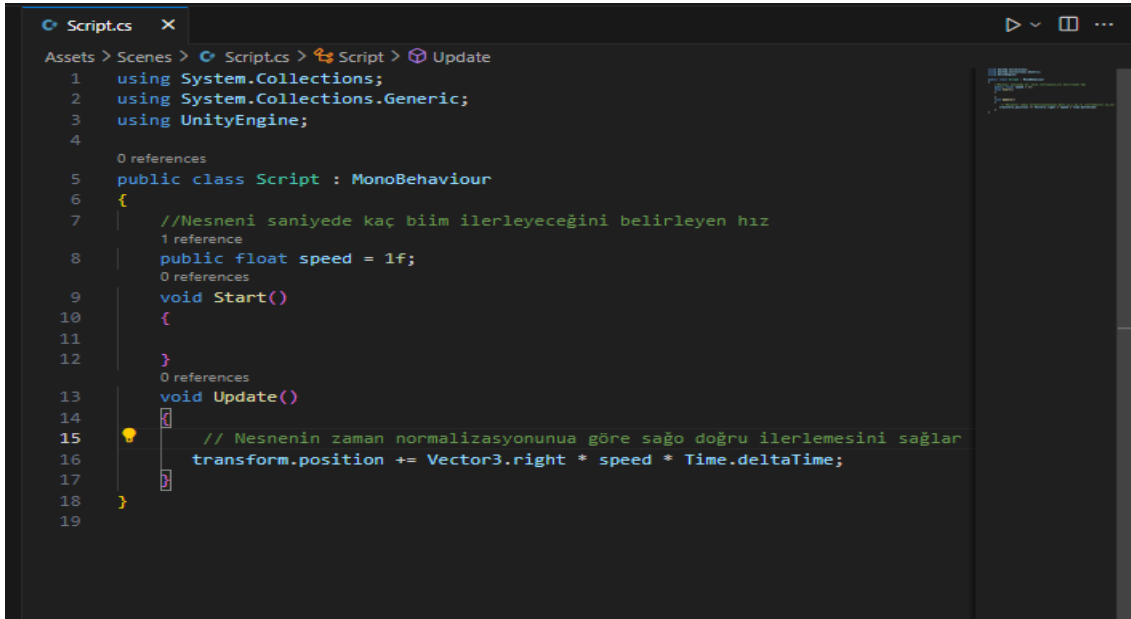
Cismin konumunu sürekli deęişiminin proje üzerinde görünümü



8 – Zamanın normalizasyonu (nesnenin saniyede bir birim ilerlemesi)

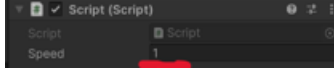
Speed = 1f nesnenin saniyede 1 birim ilerleyeceğini ifade eder

Time.deltaTime: Bu deęer son frame'den bu yana geen zamanı saniye cinsinden verir. Bu sayede, frame rate ne olursa olsun nesne sabit bir hızda ilerler.

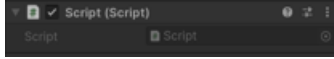


9 – Speed değışkeni tanımlama (public private farkı)

Public olarak tanımlanan bir değışken başka sınıflar ve Unity editörü tarafından erişilebilir ve değıştirilebilir.

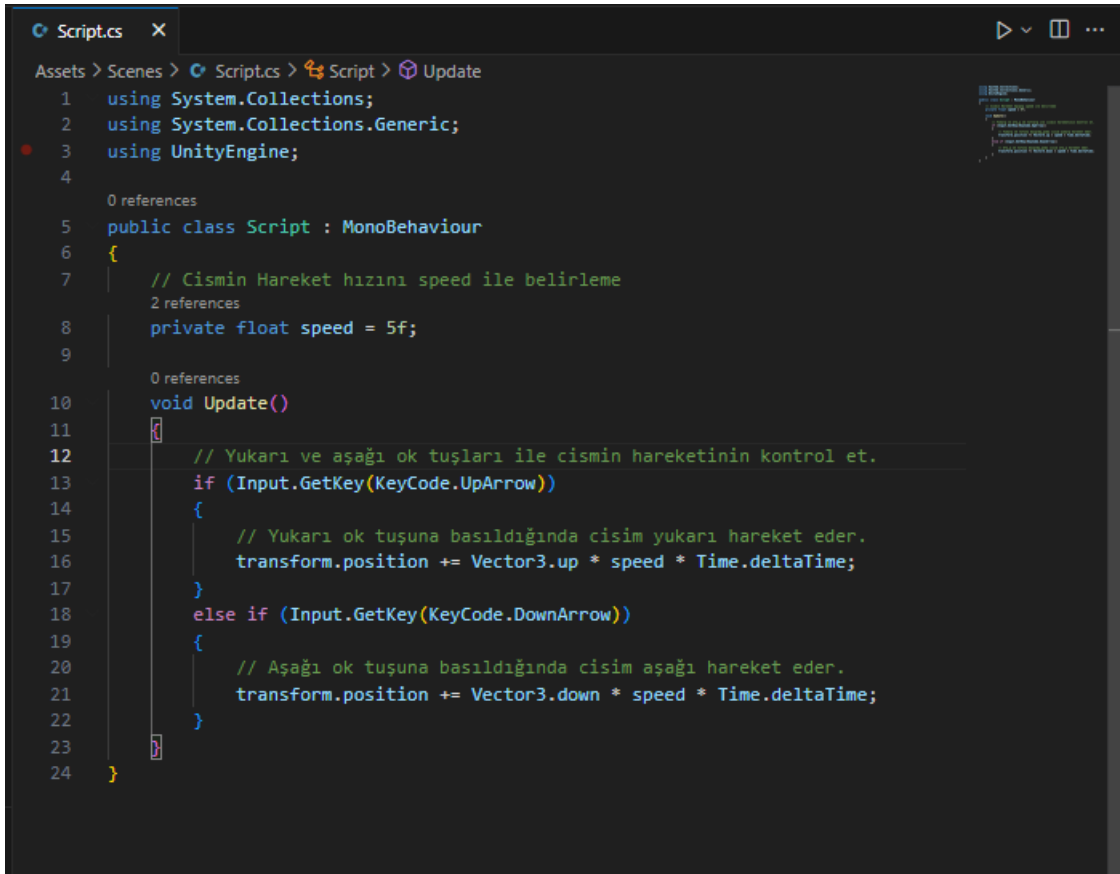


private olarak tanımlanan bir değışken sadece tanımlandığı sınıf içinde kullanılabilir, başka sınıflardan ve Unity editöründen erişilemez.



10-Klavyeden yön tuşları ile cismin hareketinin kontrolü (dikey eksen)

Unity'de klavyeden yön tuşları ile nesnenin dikey ekseninde (yukarı-aşağı) hareket ettirilmesini sağlayabilirsiniz. Bunun için Input.GetKey fonksiyonu kullanılabilir. Aşağıda yön tuşlarıyla (UpArrow ve DownArrow) dikey ekseninde hareketi kontrol eden bir örnek veriyorum.



Bonus: Yatay eksenin kontrolüne dahil edilmesi

Unity'de klavyeden yön tuşları ile nesnenin dikey ve yatay ekseninde (yukarı-aşağı-sağ-sol) hareket ettirilmesini sağlayabilirsiniz. Bunun için Input.GetKey fonksiyonu kullanılabilir. Aşağıda yön tuşlarıyla (UpArrow, DownArrow, RightArrow ve LeftArrow) dikey ekseninde hareketi kontrol eden bir örnek veriyorum.

```
Assets > Scenes > Scripts
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Script : MonoBehaviour
6  {
7      // Cismin Hareket hızını speed ile belirleme
8      private float speed = 5f;
9
10     void Update()
11     {
12         // Yukarı ve aşağı ok tuşları ile cismin hareketinin kontrol et.
13         if (Input.GetKey(KeyCode.UpArrow))
14         {
15             // Yukarı ok tuşuna basıldığında cisim yukarı hareket eder.
16             transform.position += Vector3.up * speed * Time.deltaTime;
17         }
18         else if (Input.GetKey(KeyCode.DownArrow))
19         {
20             // Aşağı ok tuşuna basıldığında cisim aşağı hareket eder.
21             transform.position += Vector3.down * speed * Time.deltaTime;
22         }
23         else if (Input.GetKey(KeyCode.RightArrow))
24         {
25             // Sağ ok tuşuna basıldığında cisim sağa hareket eder.
26             transform.position += Vector3.right * speed * Time.deltaTime;
27         }
28         else if (Input.GetKey(KeyCode.LeftArrow))
29         {
30             // Sol ok tuşuna basıldığında cisim sola hareket eder.
31             transform.position += Vector3.left * speed * Time.deltaTime;
32         }
33     }
34 }
```

Github Linki: [https://github.com/Yusuf-](https://github.com/Yusuf-Guney/Oyun_Programlama_Haftalik_Ders_Raporlari/blob/main/Unity_Hafta1_Rapor.pdf)

[Guney/Oyun_Programlama_Haftalik_Ders_Raporlari/blob/main/Unity_Hafta1_Rapor.pdf](https://github.com/Yusuf-Guney/Oyun_Programlama_Haftalik_Ders_Raporlari/blob/main/Unity_Hafta1_Rapor.pdf)