

Oyun Programlama Ödevi – Hafta 1

Ad: Eren

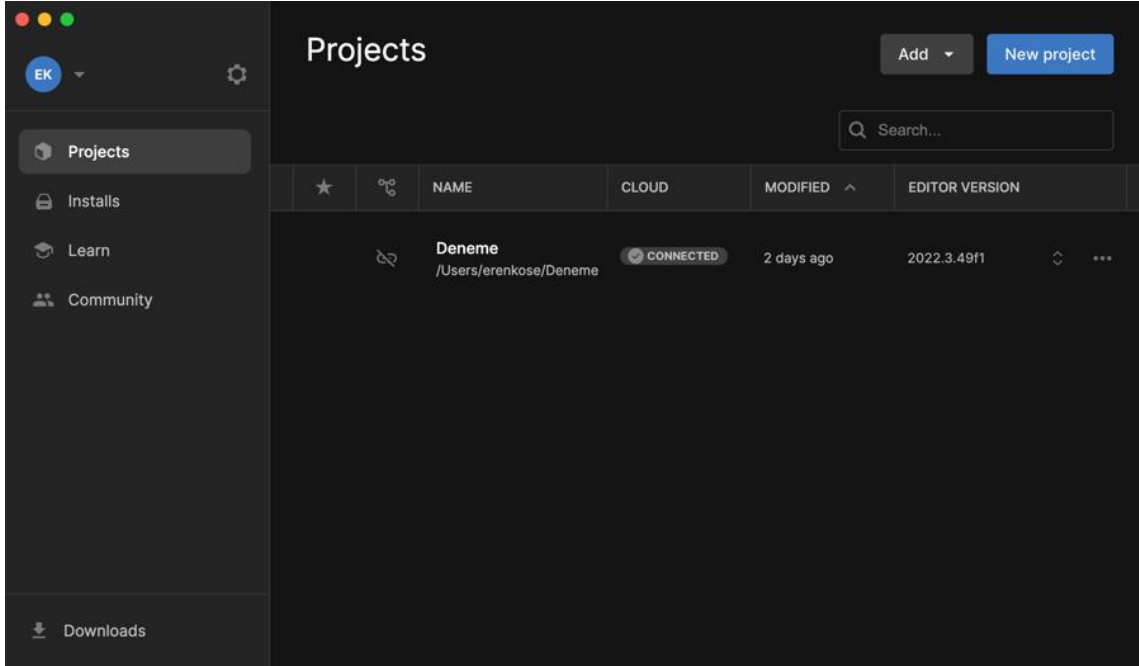
Soyad: Köse

Numara: 22360859075

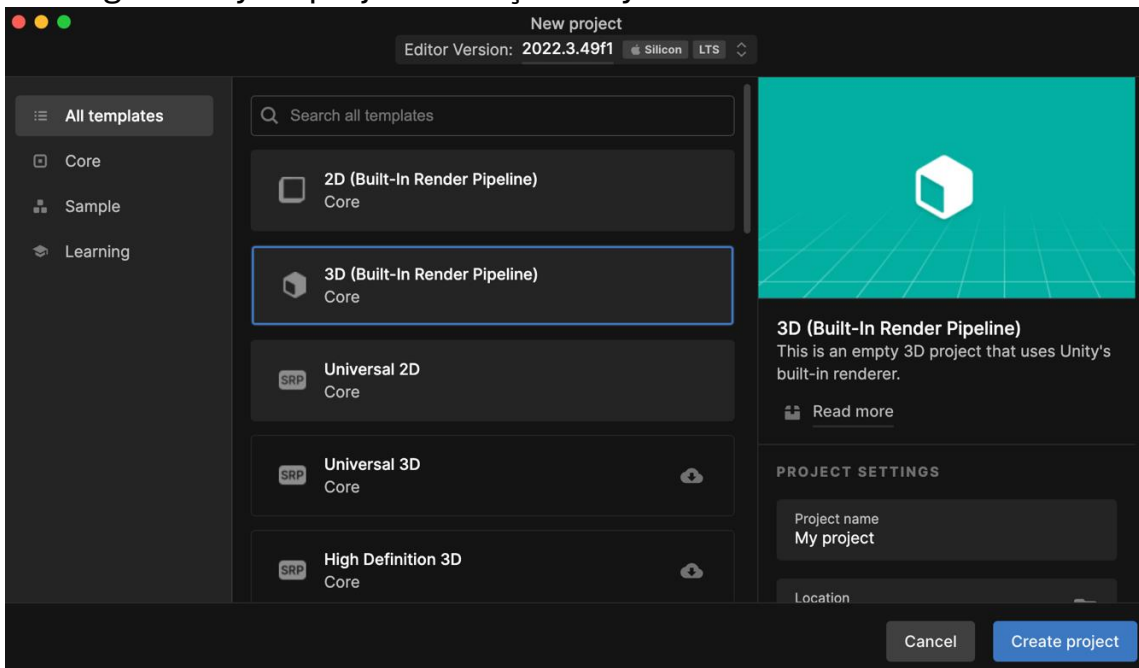
GitHub Kod Linki: https://github.com/erennkose/btu-oyun-programlama/blob/main/Hafta1/Cube_sc.cs

• Yeni Proje Oluşturma

Unity Hub uygulamasını açıp giriş yapıyoruz. Gelen ekranda soldaki seçeneklerden **Projects** kısmına tıklıyoruz. Projects kısmı açılınca sağ üstte **New Project** yazan butona tıklıyoruz.

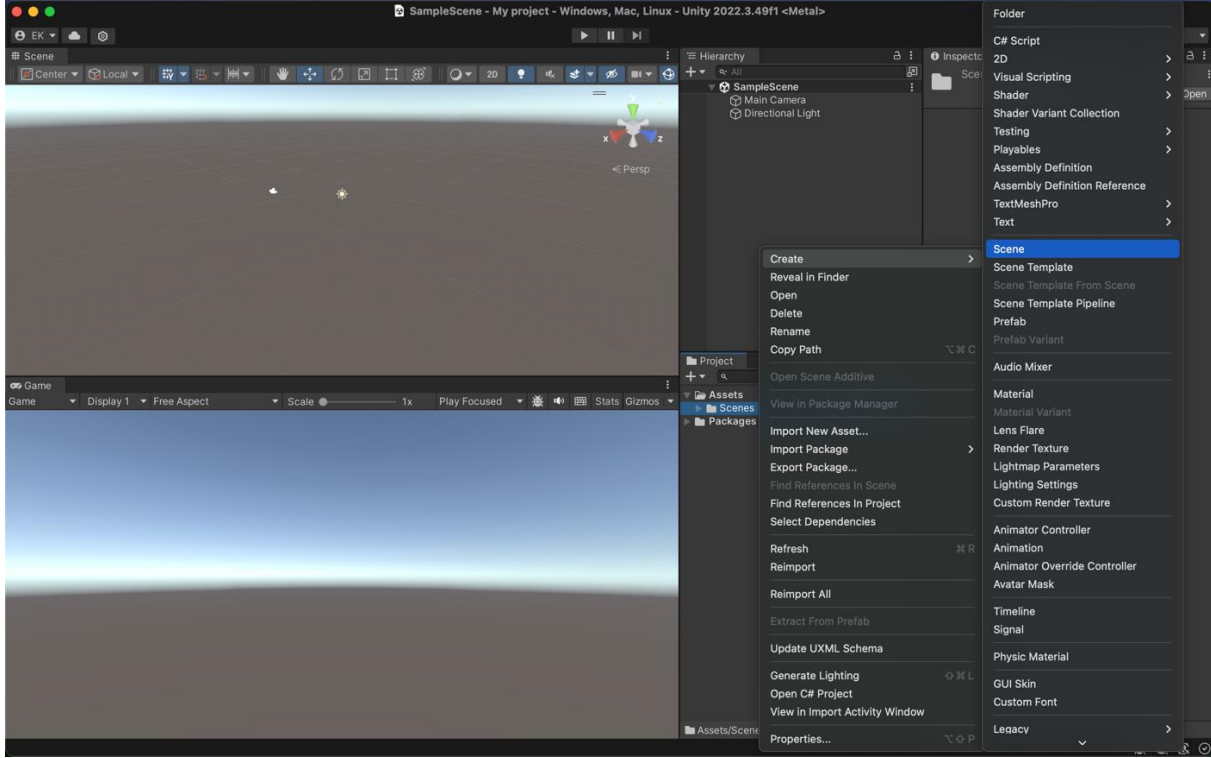


Tıkladıktan sonra gelen ekranda soldan **All templates** seçeneğine tıklıyoruz ve gelen seçeneklerden **3D (Built-In Render Pipeline)** seçeneğini (Yapılmak istenen projeye göre diğer seçenekler de seçilebilir) seçiyoruz ve **Create Project** butonuna tıkladığımızda yeni projemiz oluşturuluyor.

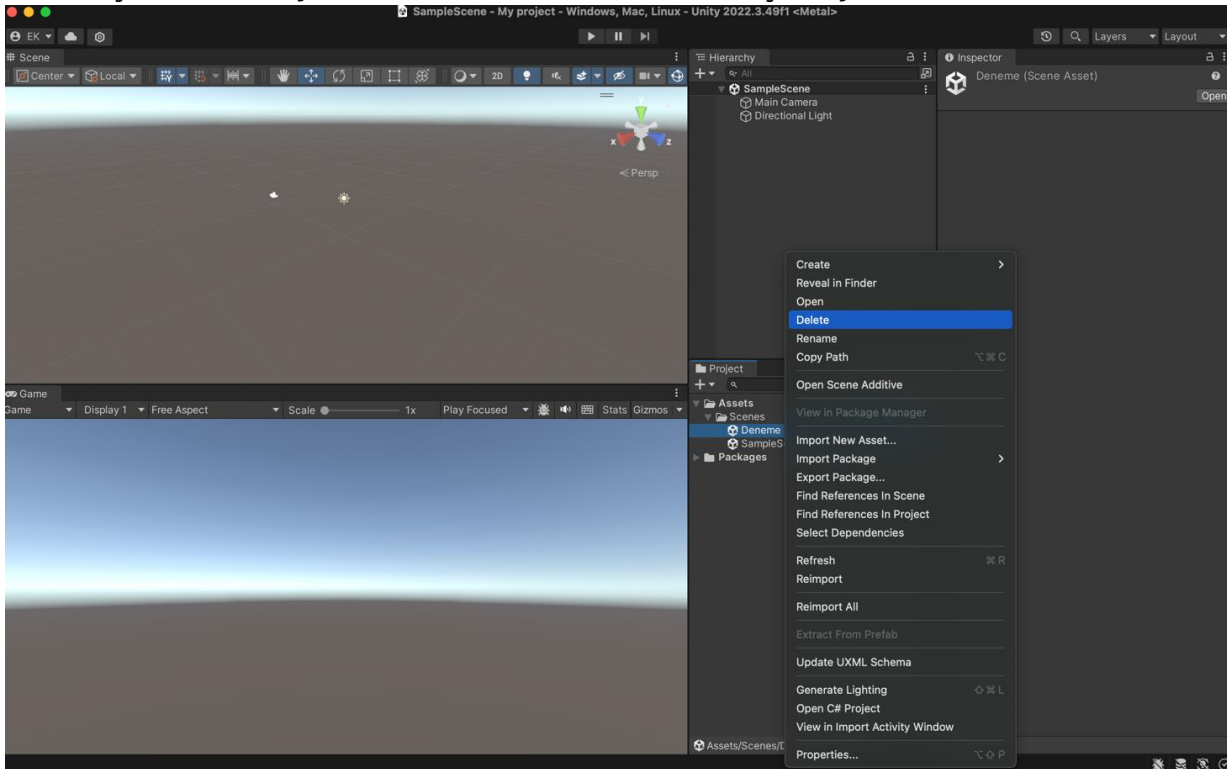


• Sahne Ekleme-Silme, Sahneye Nesne Ekleme

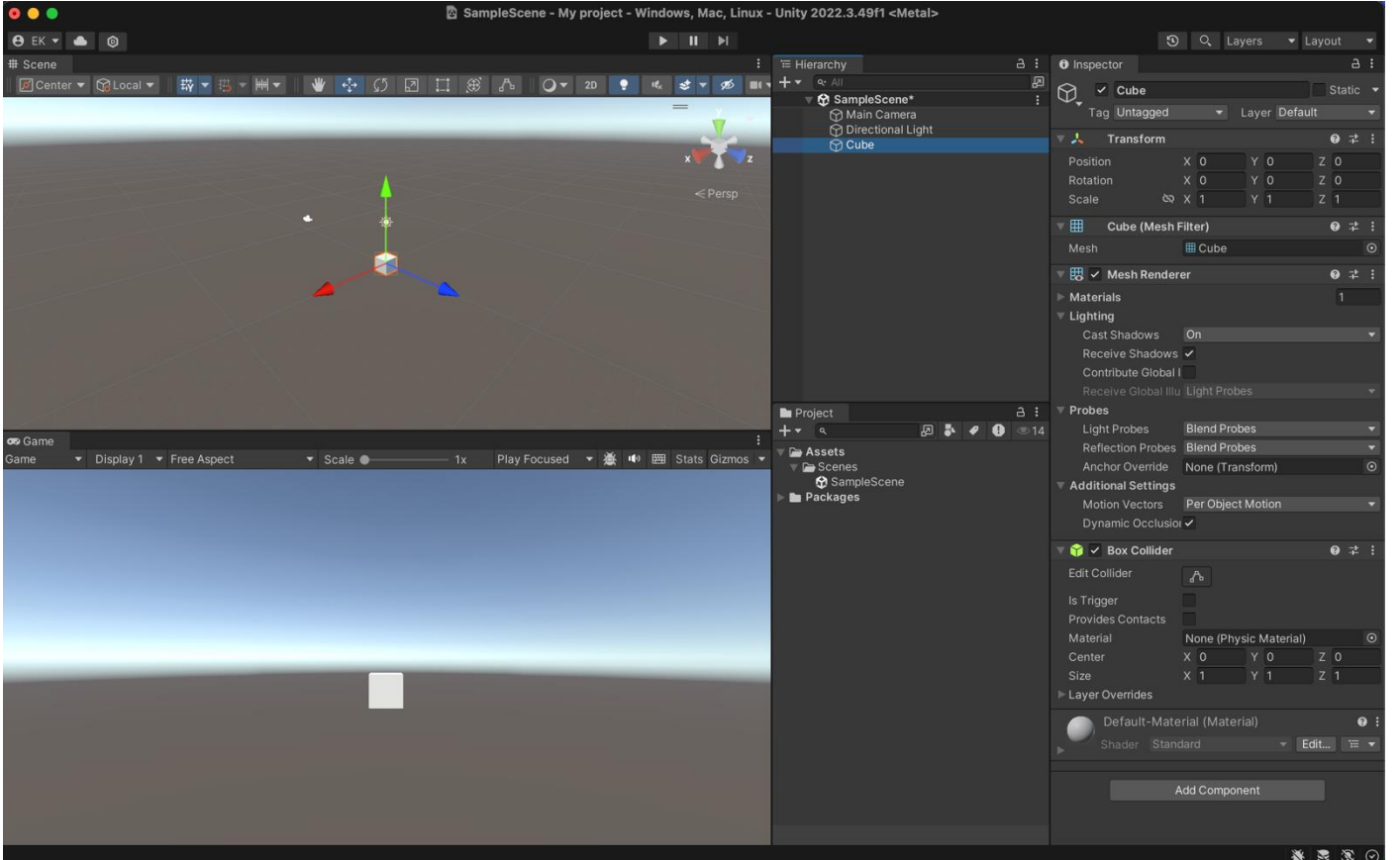
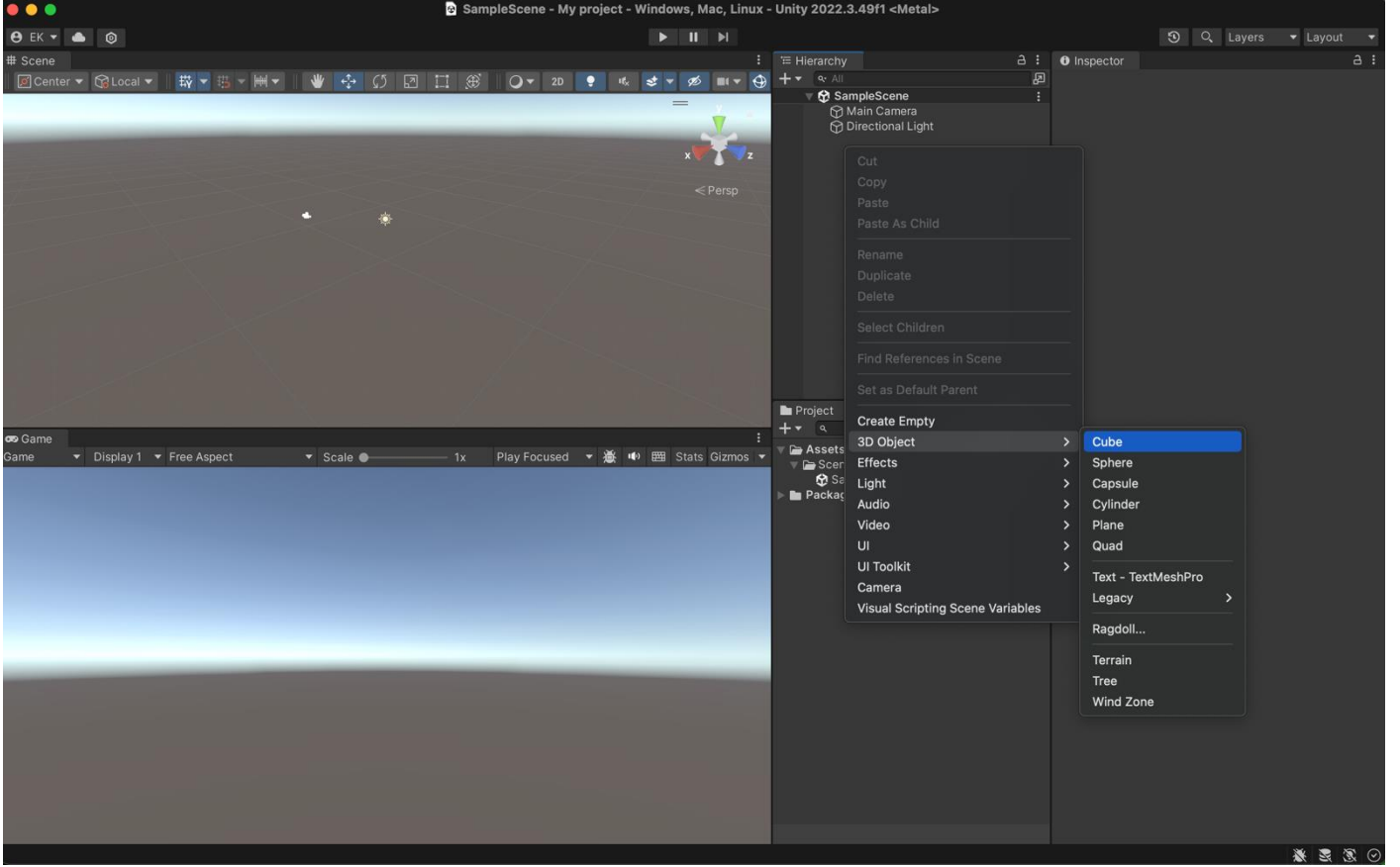
Projeyi oluşturma işlemini bitirdikten sonra önümüze proje içi işlemlerin yapıldığı ekranımız gelecektir. Bu ekranda sahne ekleme işlemi için ekranda gözükten **Project** klasöründe bulunan **Assets** klasörü altındaki **Scenes** klasörüne sağ tık atıyoruz. Ardından gelen seçenekler arasından **Create > Scene** seçeneklerini seçiyoruz. Oluşan yeni sahnemizi isimlendiriyoruz ve bu adımlarla sahne ekleme işlemi tamamlıyoruz.



Herhangi bir sahneyi silmek için ise aynı şekilde **Scenes** klasörüne ulaşıyoruz fakat sağ tık atmak yerine klasörü açıyoruz. Klasörün altında silmek istediğimiz sahneyi bulup sahneye sağ tık attıktan sonra **Delete** seçeneğine tıklıyoruz. Çıkan Pop-Up'ta silme işlemini onayladıktan sonra sahne silinmiş oluyor.

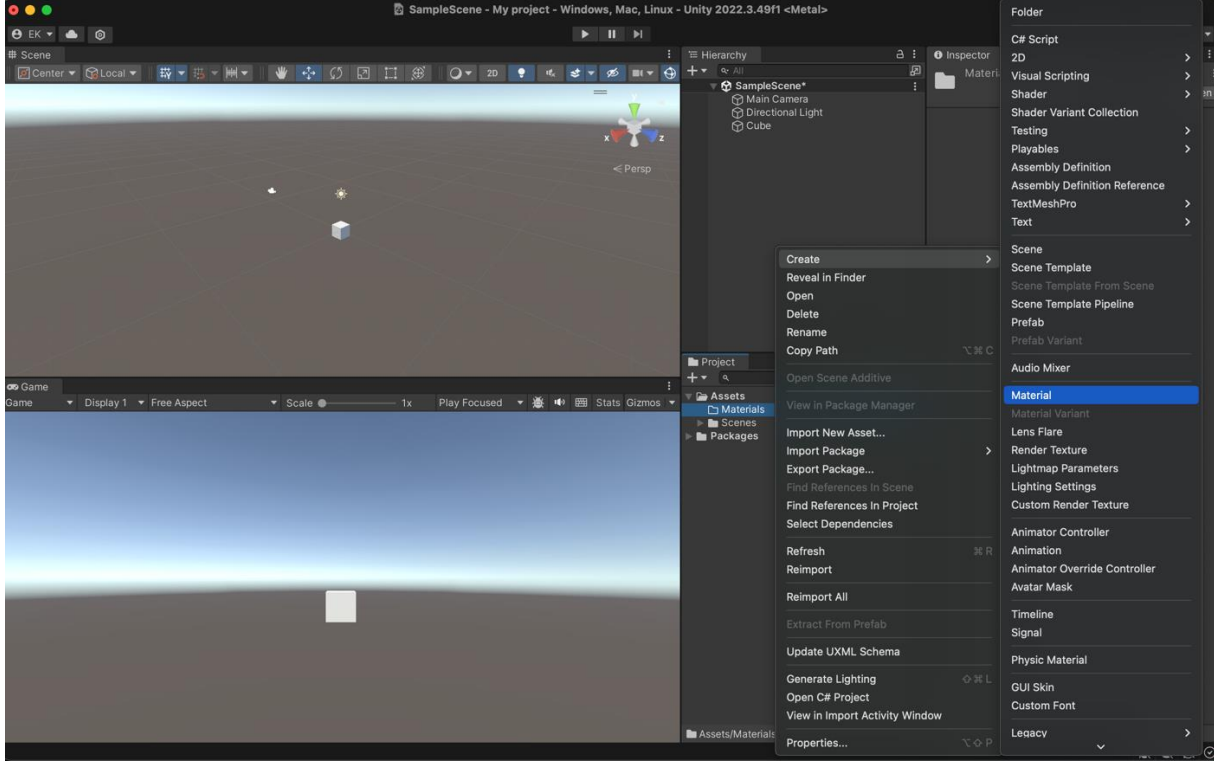


Sahneye nesne eklemek için ise **Project > Assets > Scenes** klasöründen nesne eklemek istediğimiz sahneye çift tıklıyoruz. Ekrandaki **Hierarchy** kısmında sahnemize erişiyoruz. Ardından sahnemizin altındaki Main Camera ve Directional Lightın altında bulunan boşluğa sağ tık atıyoruz. Gelen seçeneklerden **3D Objects > Cube** şeklinde Cube seçeneğimizi seçiyoruz ve küp nesnemiz oluşturulmuş oluyor.

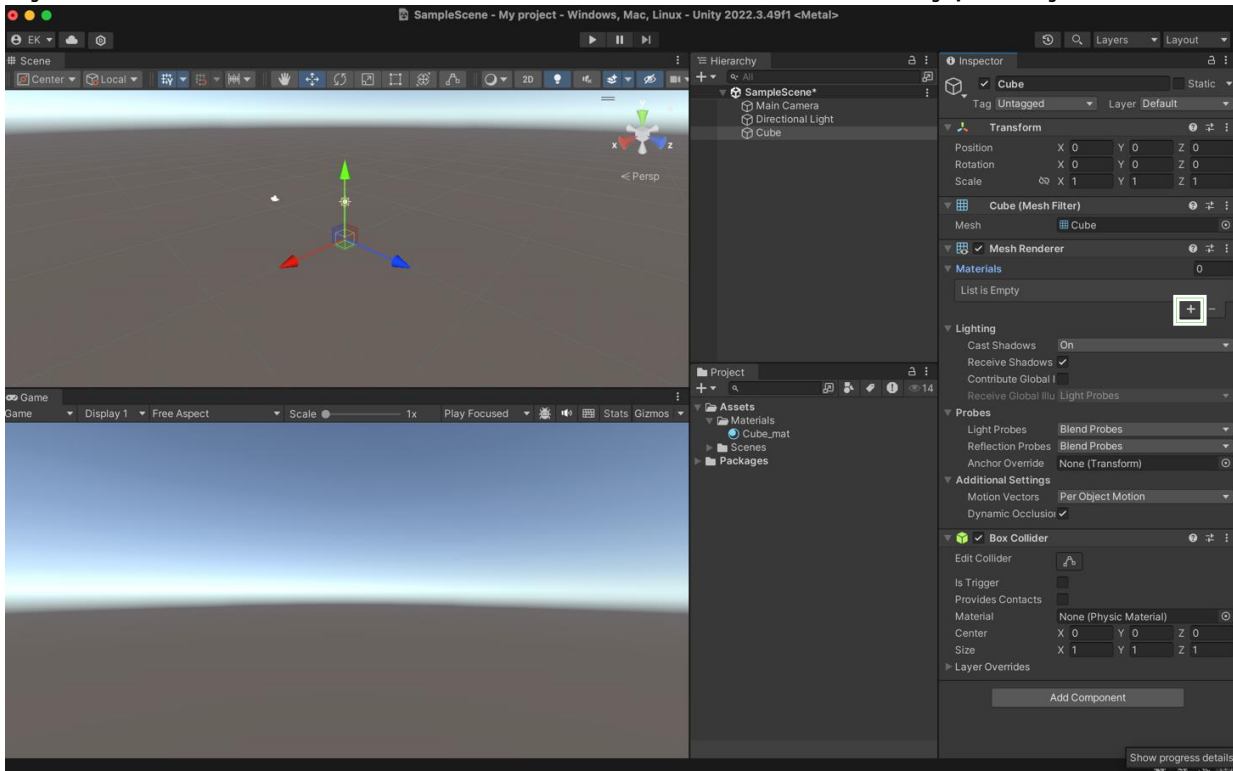


• Sahnedeki Nesneye Materyal Ekleme

Nesnemize materyal eklemek için öncelikle materyal oluşturmamız gerekiyor. Bunun için ise önce **Project** klasörü altındaki **Assets** klasörünün altına **Materials** adında bir klasör (Düzen sağlama amaçlı bir klasör) açıyoruz. **Materials** adlı klasörümüzü oluşturduktan sonra klasörümüze sağ tık atıyoruz. Çıkan seçeneklerden **Create > Material** şeklinde **Material** seçeneğini seçiyoruz. Ardından gerekli isimlendirmeyi de yaptığımızda materyal oluşmuş oluyor.

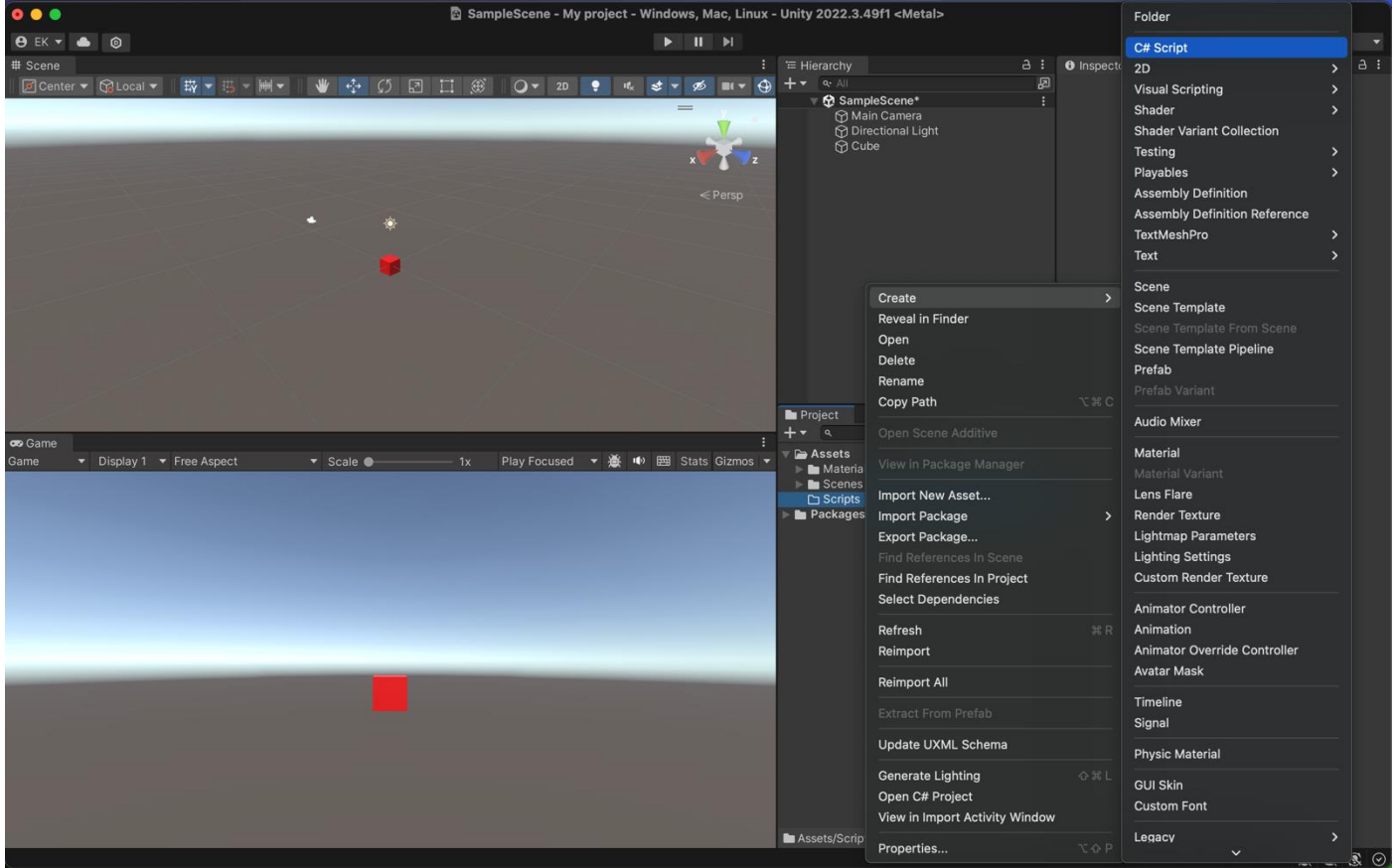


Oluşturduğumuz materyali nesnemize eklemek için ise oluşturulan materyali **sürükle-bırak** şeklinde nesnemizin üstüne bırakmamız yeterli olacaktır. Başka bir yöntem ise **Hierarchy** kısmından nesnemize tıkladıktan sonra **Inspector** bölümünde açılan **Mesh Renderer** kısmı altındaki **Materials** kısmını açıp **+** tuşuna basmaktır.

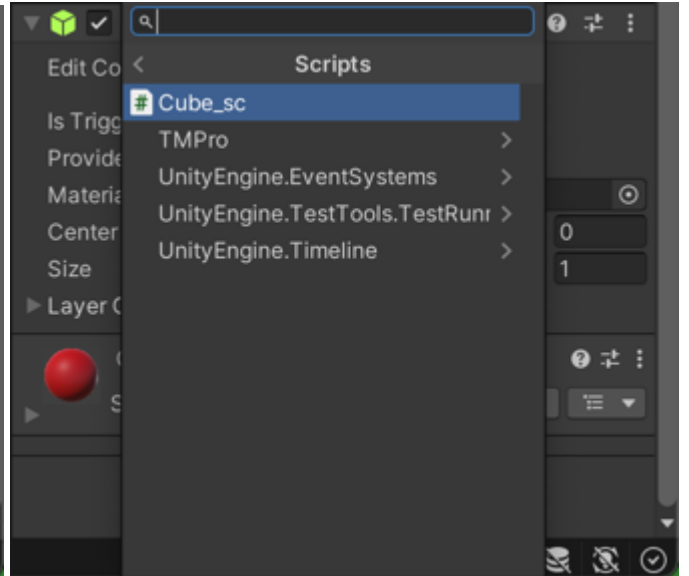
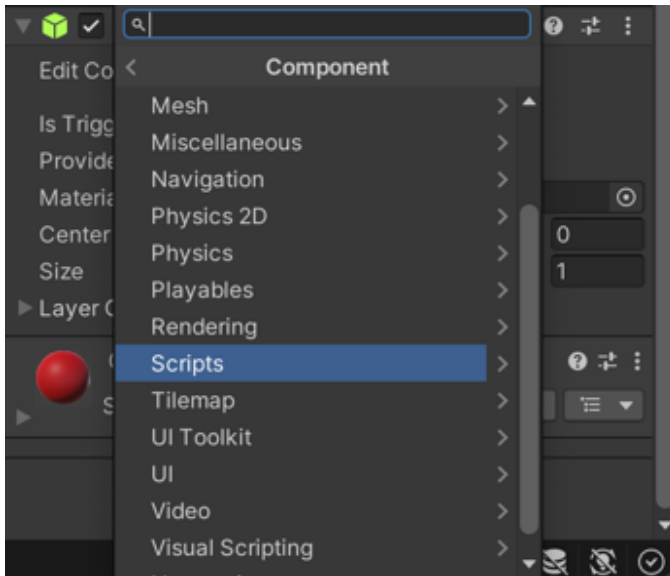
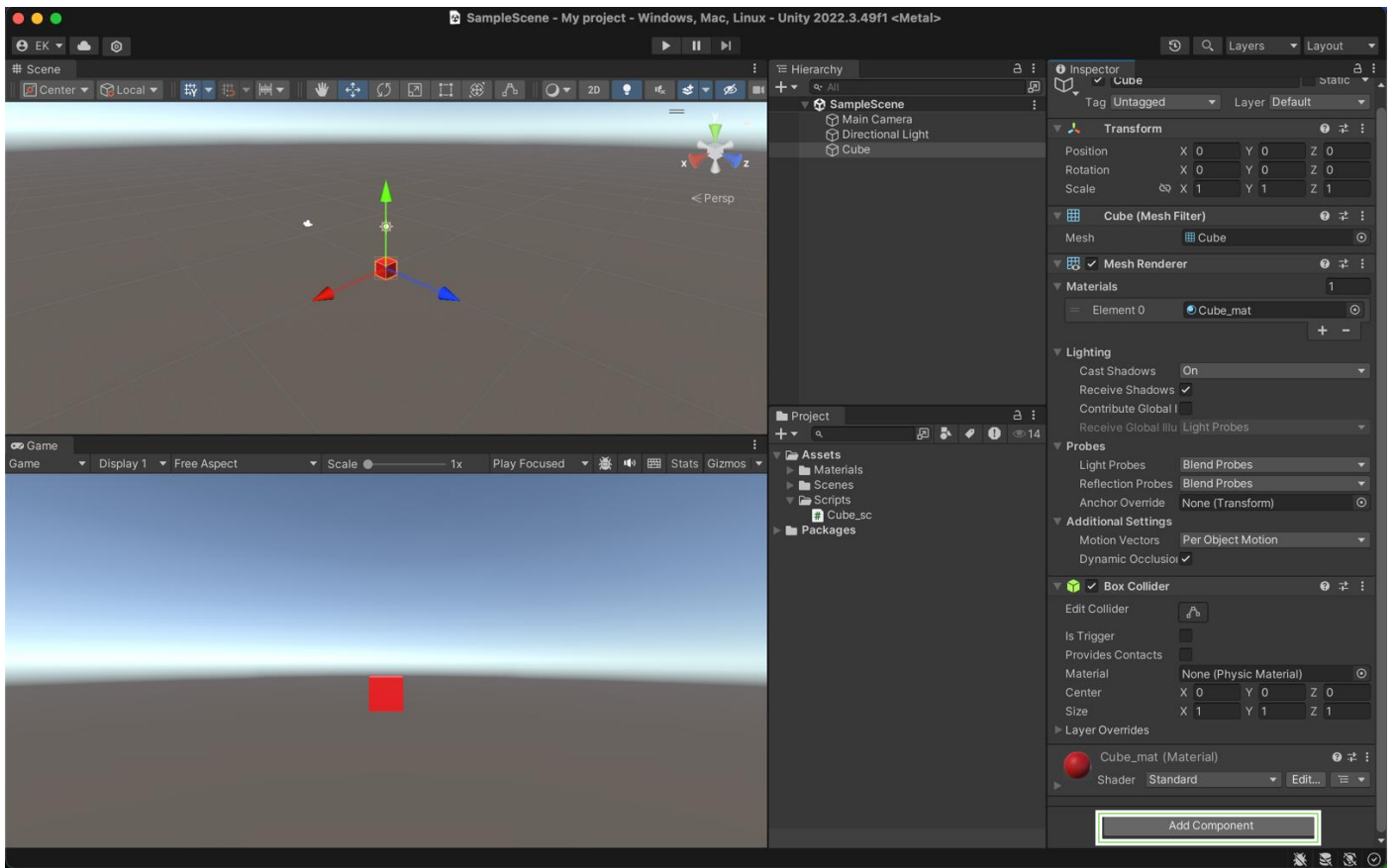


- **Sahnedeki Nesneye Script Ekleme**

Nesnemize script eklemek için öncelikle script oluşturmamız gerekiyor. Bunun için ise önce **Project** klasörü altındaki **Assets** klasörünün altına **Scripts** adında bir klasör (Düzen sağlama amaçlı bir klasör) açıyoruz. **Scripts** adlı klasörümüzü oluşturduktan sonra klasörümüze sağ tık atıyoruz. Çıkan seçeneklerden **Create > C# Script** şeklinde **C# Script** seçeneğini seçiyoruz. Ardından gerekli isimlendirmeyi de yaptığımızda script oluşmuş oluyor.



Oluşturduğumuz scripti nesnemize eklemek için ise oluşturulan scripti **sürükle-bırak** şeklinde nesnemizin üstüne bırakmamız yeterli olacaktır. Başka bir yöntem ise **Hierarchy** kısmından nesnemize tıkladıktan sonra **Inspector** bölümünün en altında bulunan **Add Component** butonunu kullanmaktır. Bu butona tıkladıktan sonra açılan Pop-Up sekmesinden **Scripts** seçeneğini seçiyoruz. Bu seçeneği seçtikten sonra projemizde bulunan scriptler arasından eklemek istediğimiz scripti seçiyoruz. Böylece istenen scripti nesnemize eklemiş oluyoruz.



- **Script İle Nesnenin Konumunu Bir Kez Değiştirme**

Bu işlemi scriptimizin içinde bulunan **Start** fonksiyonunu kullanarak yapabiliriz. Bunun sebebi ise Start fonksiyonunda bulunan işlemlerin oyunun ilk ekran karesinden önce gerçekleştirilmesidir. Bu yüzden **oyun başlatıldığında yalnızca bir kez gerçekleşmesini istediğimiz işlemleri Start fonksiyonuna yazabiliriz**. transform.position kullanılarak nesneye bir konum atanabilir. Aşağıda bulunan Start fonksiyonuyla oyun başladığında yalnızca bir kez nesnemizi y ekseninde 2 birim aşağı götürebiliriz.

```
void Start()
{
    transform.position = new Vector3(0,-2,0);
}
```

- **Script İle Nesnenin Konumunu Sürekli Değiştirme**

Bu işlemi scriptimizin içinde bulunan **Update** fonksiyonunu kullanarak yapabiliriz. Bunun sebebi ise update fonksiyonunda bulunan işlemlerin oyunun her bir ekran karesinde gerçekleştirilmesidir. Bu yüzden **oyun başlatıldığında her ekran karesinde gerçekleşmesini istediğimiz işlemleri Update fonksiyonuna yazabiliriz**. Aşağıda bulunan Update fonksiyonuyla oyun başladığında sürekli olarak nesnemizi y ekseninde 1 birim aşağı götürebiliriz.

```
void Update()
{
    transform.Translate(new Vector3(0,-1,0));
}
```

- **Zamanın Normalizasyonu**

Script ile nesnenin konumunu sürekli değiştirirken bunun bir saniyede y ekseninde 1 adım aşağı olmadığını fark etmişsinizdir. Ne kadar süreyle 1 adım aşağı gideceği kullanılan bilgisayara, bilgisayarın saniye başına ekran karesine göre değişebilmektedir. Her bilgisayar için de saniyede y ekseninde 1 adım aşağı gitmesini istiyorsak zamanı normalize etmemiz gerekir. Bu ise **Time.deltaTime** kullanılarak yapılır. Bu kod parçası sayesinde bir saniyedeki ekran karesine bağlı olan işlemin saniye başına bir kere yapılmasını sağlayabiliriz.

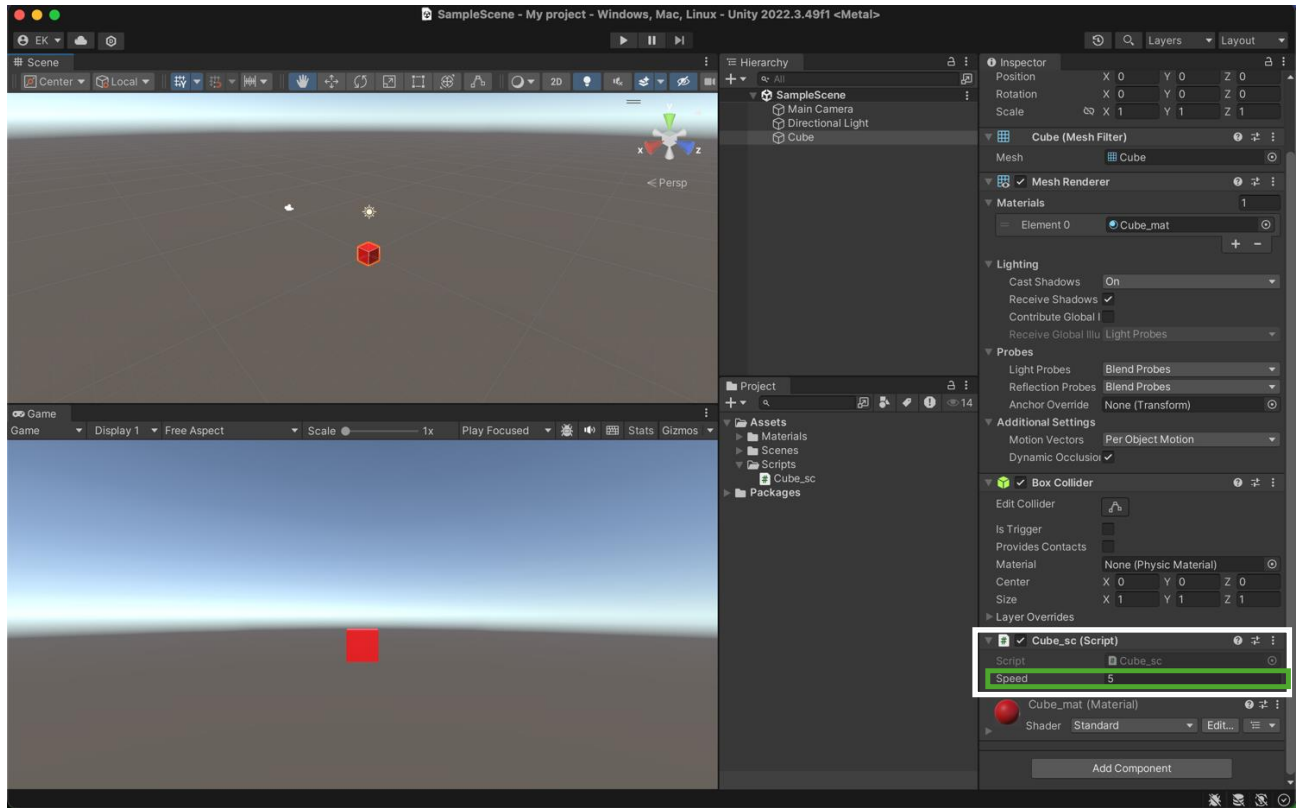
```
void Update()
{
    transform.Translate(new Vector3(0,-1,0) * Time.deltaTime);
}
```

- **Speed Deęiřkeni Tanımlama**

Speed deęiřkenimiz nesnenin hızını dñzenlemek, takip etmek iin kullanılmaktadır. Nesnemizin classı iinde public veya private řekilde oluřturulabilir. **Public** řekilde oluřturmamız durumunda **script zerinden** bu deęiřkene ulařılıp dñzenlenebildięi gibi **Inspector kısmından da** eriřilip dñzenlenebilir. Fakat bu deęiřkeni **private** řekilde oluřturacak olursak bu deęiřken **yalnızca script zerinden** deęiřtirilip grntlenebilir.

```
0 bařvuru
public float speed = 5;

2 bařvuru
private float privateSpeed = 4;
```



- **Klavyeden Yön Tuşları İle Nesnenin Hareketinin Kontrolü**

Nesnemizin hareketinin kontrolünü sağlamak, nesnemizin scriptinin içerisinde bulunan Update fonksiyonuna yazdığımız kodlarla yapılır. Her bir ekran karesinde istenen tuşların klavye inputu kontrol edilir. Input varsa input alınan tuşa göre nesnemizin hareketi sağlanabilir. Hareketin daha düzenli ve pürüzsüz olması amacıyla saniyede bir hareket ve hız durumu da hesaba katılarak bu işlem yapılır. Hız değişkeniyle (speed) hız durumu kontrol edilmese hız sabit olur ve dinamik olarak nesnenin kontrol edilmesi imkanı kaybolur. Saniyede bir hareket kontrolü (Time.deltaTime) olmazsa da hareket miktarı saniye başına ekran karesi miktarına göre değişiklik gösterir, optimizasyon bozulur. Yatay hareket için alınan inputlar horizontal adlı değişkene, dikey hareket için alınan inputlar vertical adlı değişkene kaydedilir. Sağ ve sol yön tuşları ile yatay hareket, yukarı ve aşağı yön tuşları ile dikey hareket sağlanır. transform.Translate fonksiyonu kullanılarak da inputlara bağlı değişikliklerin gerçekleşmesi sağlanır.

```
void Update()
{
    float horizontal = Input.GetAxis("Horizontal");
    float vertical = Input.GetAxis("Vertical");
    transform.Translate(new Vector3(1,0,0) * Time.deltaTime * speed * horizontal);
    transform.Translate(new Vector3(0,1,0) * Time.deltaTime * speed * vertical);
}
```