



**BURSA TEKNİK ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİ (T) BÖLÜMÜ**  
**BİLGİSAYAR OYUNLARDA YAPAY ZEKA**

**Ödev-1 Raporu**

Github: <https://github.com/MoussaBane/BOYZ-NesneHareketi>

**MOUSSA BANE**

**24435004029**

**1. Yeni Proje Oluşturma:**

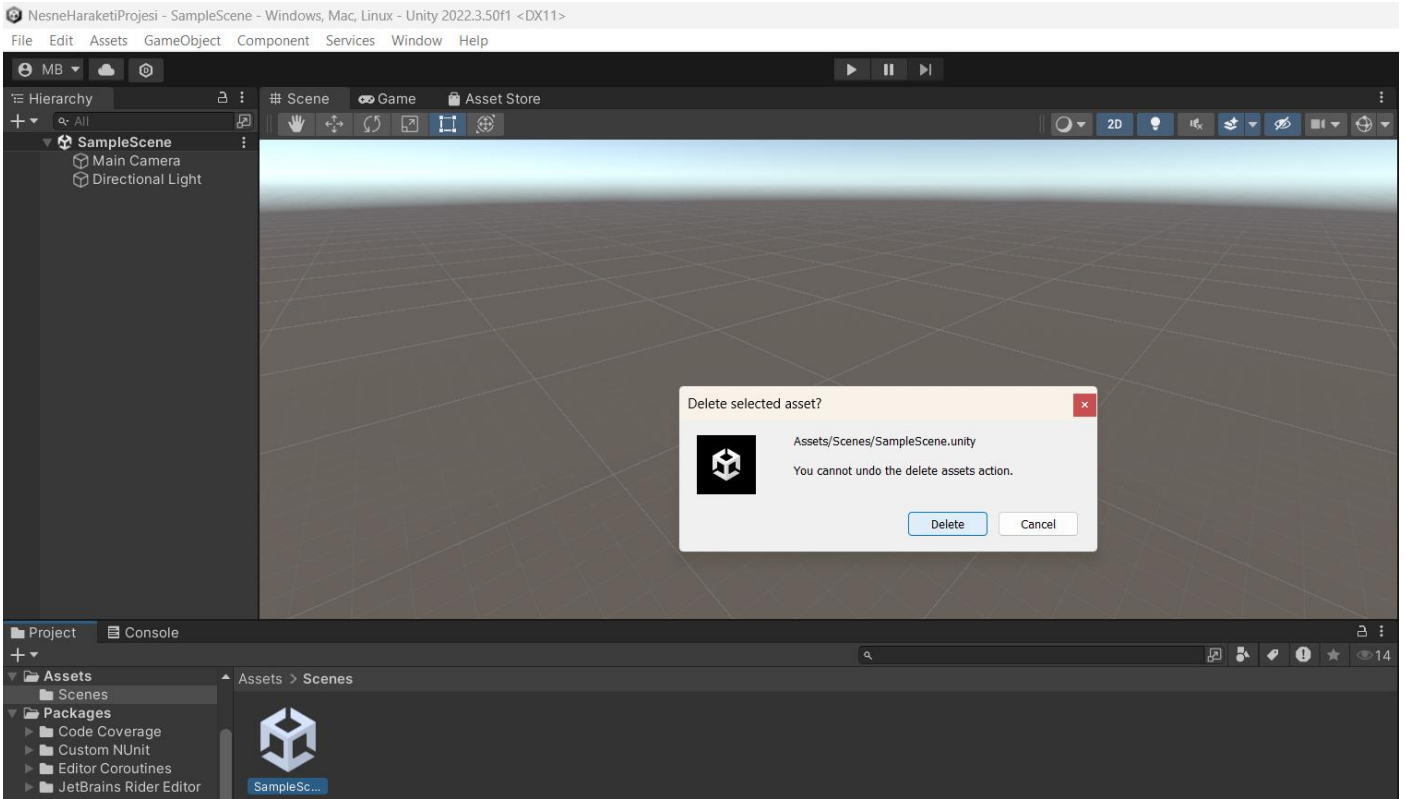
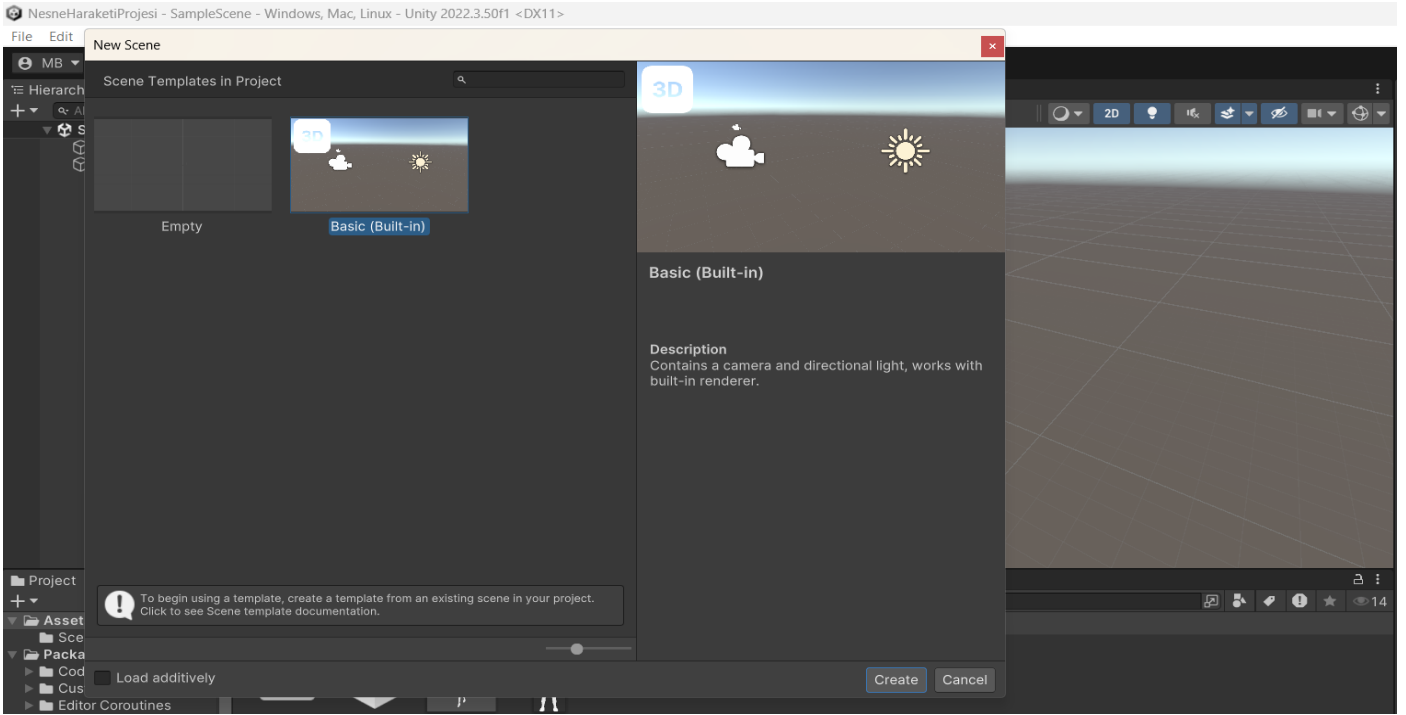
- Unity Hub'ı açın.
- **"New Project"** düğmesine tıklayın.
- Proje tipi olarak **"3D"** seçin.
- Projeyi adlandırın (örneğin, "Nesne Hareketi Projesi") ve **"Create"** düğmesine tıklayarak projeyi oluşturun.

Bu adımda, Unity'de çalışacağımız yeni bir proje oluşturuyoruz. 3D proje seçimi, sahnemizde küp gibi 3D nesnelerle çalışabilmemiz için gereklidir.

**2. Sahne Ekleme ve Silme:**

- Menüden **"File"** sekmesine tıklayın ve **"New Scene"** seçeneğini seçin. Yeni bir sahne eklenecektir.
- Eski bir sahneyi silmek için, **"Assets"** panelinde **"Scenes"** klasörünü açın. Silmek istediğiniz sahneye sağ tıklayıp **"Delete"** seçeneğini seçin.

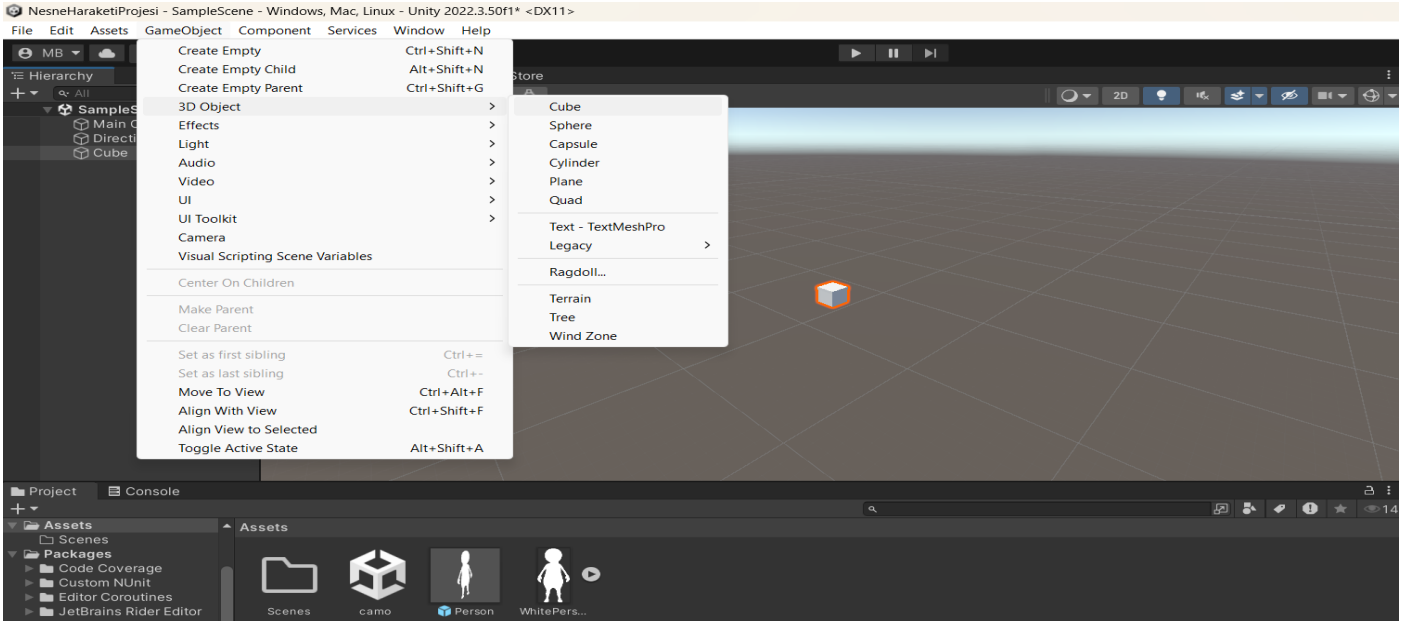
Her Unity projesinde bir veya birden fazla sahne (Scene) bulunur. Sahne, oyunda veya projede göreceğiniz her şeyin toplandığı yerdir. Bu adımda, yeni bir sahne ekleyip, gereksiz sahneleri silebilirsiniz.



### 3. Sahneye Nesne Ekleme (Küp):

- Menüden "**GameObject**" > "**3D Object**" > "**Cube**" seçeneğini seçin.
- Küp, sahnede ve "**Hierarchy**" panelinde belirecektir.

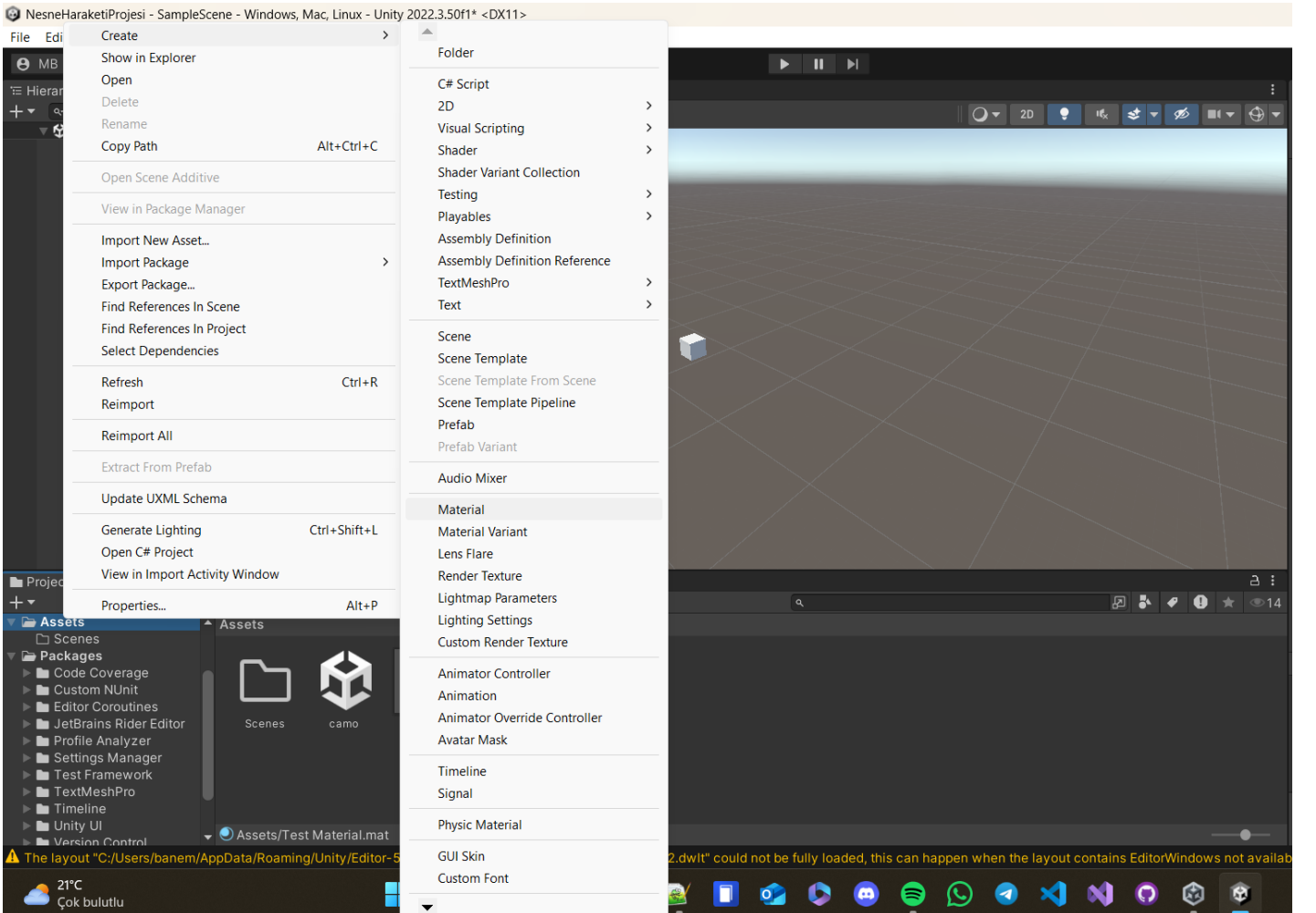
Küp, sahnede görsel olarak çalışacağımız bir nesne olacak. "Hierarchy" paneli, sahnedeki tüm nesnelerin listelendiği bir paneldir. Nesneleri bu panelden seçip düzenleyebilirsiniz.



#### 4. Nesneye Materyal Ekleme:

- **"Assets"** klasöründe sağ tıklayın ve **"Create > Material"** seçeneğini seçin.
- Materyali adlandırın (örneğin, "KüpMateryali").
- Küpün üzerine bu materyali sürükleyip bırakın.

Materyal, nesnelerin görsel özelliklerini (renk, doku, parlaklık vb.) tanımlar. Oluşturduğunuz materyali küp üzerine eklediğinizde, küpün dış görünüşü materyalin özelliklerine göre değişecektir.



## 5. Script Ekleme:

- "Assets" klasörüne sağ tıklayıp "Create > C# Script" seçeneğini seçin.
- Script'i adlandırın (örneğin, "KüpHareketi").
- Script'i küpe sürükleyin. Böylece script, küp nesnesine atanmış olur.

C# scriptleri, Unity'deki nesnelere davranış kazandırmak için kullanılır. Küp üzerine eklediğimiz script, küpün hareket etmesini veya belirli bir aksiyonu gerçekleştirmesini sağlar.

## 6. Nesnenin Konumunu Start Fonksiyonu ile Değiştirme:

- Script'i çift tıklayarak açın.
- Aşağıdaki kodu "Start" fonksiyonu içerisine yazın:

```
1 // Start is called before the first frame update
2 void Start()
3 {
4     transform.position = new Vector3(3, 4, 2); // Nesnenin x=3 y=4 z=2 yapar.
5 }
6
```

Start fonksiyonu, oyun başladığında bir defa çalışan bir fonksiyondur. Yukarıdaki kod ile küpün başlangıç konumunu değiştiriyoruz. Bu örnekte, küp sahnede başlangıçta x=3 y=4 et z=2 konumunda olacaktır.

## 7. Nesnenin Sürekli Hareket Etmesi (Update Fonksiyonu):

- Aynı script'te "Update" fonksiyonunu aşağıdaki gibi düzenleyin:

```
1 // Update is called once per frame
2 void Update()
3 {
4     transform.position += Vector3.forward * Time.deltaTime; // Nesneyi Z ekseninde hareket ettirir.
5 }
```

**Update** fonksiyonu, her karede (frame) tekrar tekrar çalışır. **Time.deltaTime**, her bir kare arasındaki zaman farkını verir. Bu sayede nesne, her saniyede sabit hızla ilerler. Bu kod, küpün Z eksenini boyunca sürekli olarak hareket etmesini sağlar.

## 8. Zamanın Normalizasyonu:

- **Time.deltaTime** kullanarak zaman farkını normalleştirdiğimizden, nesne her saniyede 1 birim ilerler. Bu, farklı cihazlardaki kare hızları arasındaki farkı telafi eder.

Bu adım, zamanın normalleştirilmesini sağlar. Yani, oyun farklı hızlarda kare (frame) oluştursa bile, nesnenin hareketi tutarlı olacaktır.

## 9. Speed Değişkeni Tanımlama (Public ve Private Farkı):

- Script'te bir speed değişkeni tanımlayın:

```
1 public float speed = 1.0f; // Public değişken
2 private float hiz = 1.0f; // Private değişken
3
```

**public** değişkenler, Unity Editor'da görünüp düzenlenebilirken, **private** değişkenler sadece script içinde kullanılır ve dışarıdan erişilemez. **speed** değişkeni nesnenin hareket hızını belirler.

## 10. Klavyeden Yön Tuşları ile Hareket Kontrolü (Dikey Eksen):

- Script'in "**Update**" fonksiyonunu aşağıdaki gibi güncelleyin:

```
1 // Update is called once per frame
2 void Update()
3 {
4     float moveVertical = Input.GetAxis("Vertical"); // Yön tuşları ile dikey eksen kontrolü
5     transform.position += Vector3.forward * moveVertical * speed * Time.deltaTime;
6 }
7
```

Bu adımda, yön tuşlarını kullanarak küpün dikey ekseninde (Z eksen) ileri veya geri hareket etmesini sağlıyoruz. **Input.GetAxis("Vertical")**, klavyedeki yön tuşlarını algılar.

### Bonus: Yatay Eksen Hareket Kontrolü

- Script'in "**Update**" fonksiyonuna yatay eksen kontrolü ekleyin:

```
1 // Update is called once per frame
2 void Update()
3 {
4     float moveVertical = Input.GetAxis("Vertical"); // Yön tuşları ile dikey eksen kontrolü
5     float moveHorizontal = Input.GetAxis("Horizontal"); // Yatay eksen kontrolü
6     Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);
7     transform.position += movement * speed * Time.deltaTime;
8 }
```

Bu adımda, hem yatay (X eksen) hem de dikey (Z eksen) hareketi kontrol edebiliyoruz. Böylece küp, yön tuşlarıyla hem sağa-sola hem de ileri-geri hareket ettirilebilir.