

# OYUN PROGRAMLAMA DERS RAPORU 1

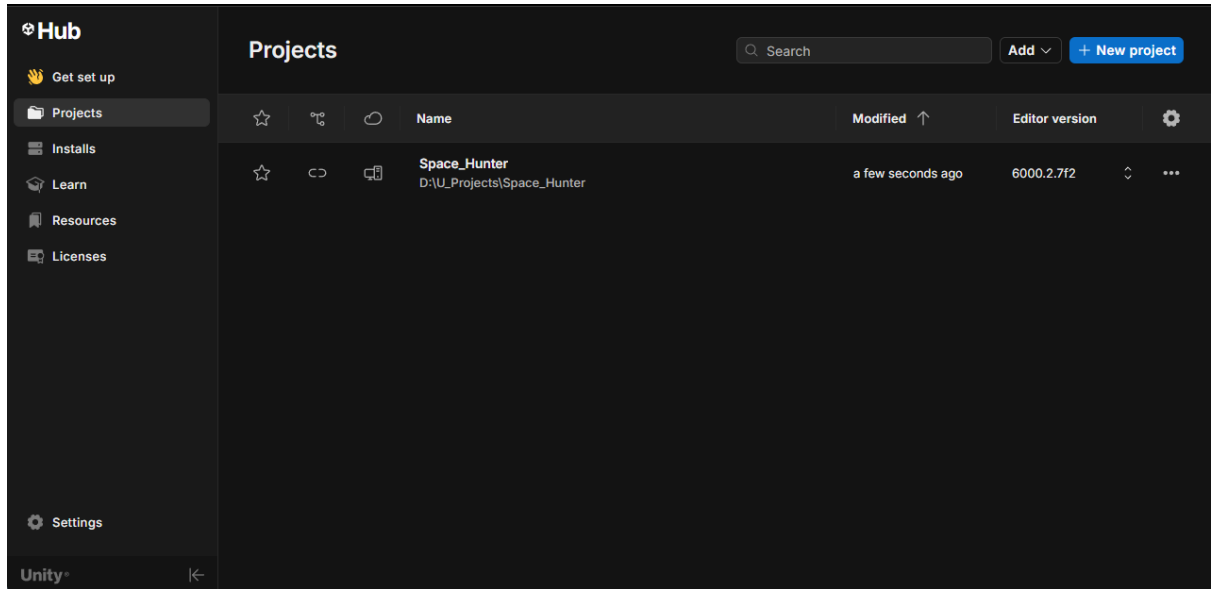
AD-SOYAD: Kubilay İnanç

Öğrenci No: 22360859047

İlk ders raporu olarak Unity uygulamasında aşağıdaki temel konuları inceledik.

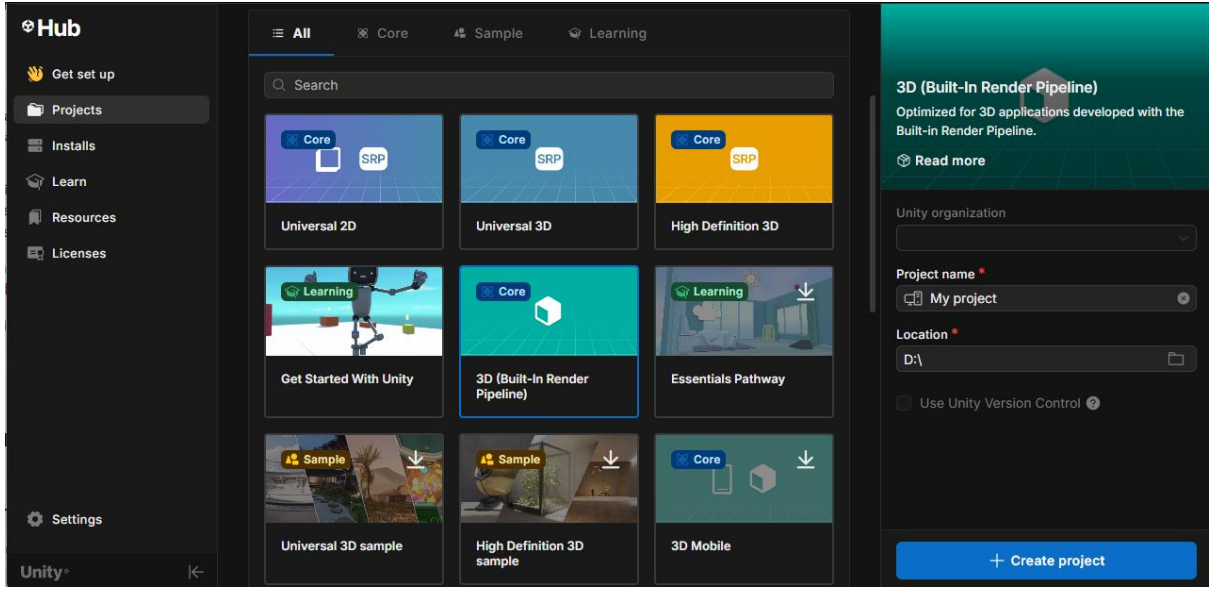
## 1.Yeni Proje Oluşturma:

Şekil1’de de görüldüğü üzere yeni bir proje oluştururken “Projects” kısmından sağ üstteki “New Project” adlı mavi renkteki butonuna tıklıyoruz.



Şekil1

Ardından Şekil2’deki gibi bir ekran gelecek ve burada bizim için hazır olan “pipeline”lardan kendi projemize uygun olanı seçiyoruz. Ben 3D(Built-in Render) seçtim. Burada da proje ismini verip, bulutta veya cihazda depolanabileceğini seçebiliyorsunuz.

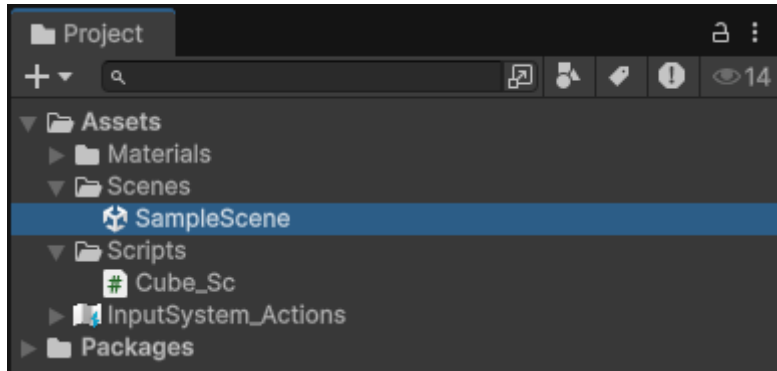


Şekil2

## 2.Sahne Ekleme-Silme:

Bunun için açılan ekranda “Şekil3”teki gibi “Project” bölmesine gelmeniz gerekiyor. Burada “Scenes” klasörünün altında mevcut sahneler bulunur. Bunu isterseniz sağ tıklayarak “delete”i seçerek silebilirsiniz.

Yeni bir sahne eklemek içinse “Şekil3”te de görülen yine Project bölmesindeki “+” sekmesine tıklayarak yeni sahnenizi oluşturabilirsiniz.



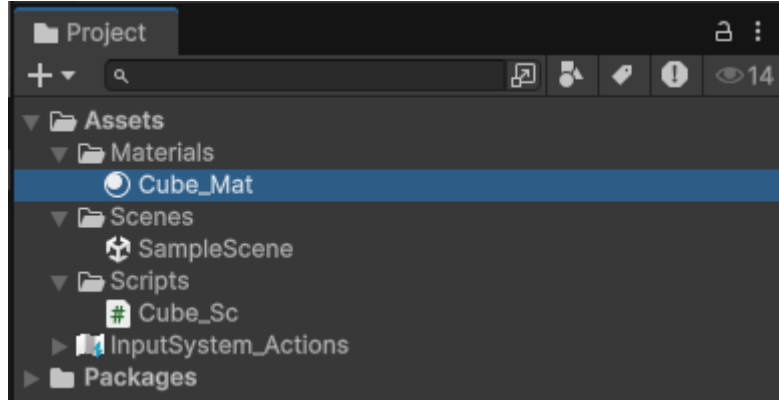
Şekil3

## 3.Sahneye Nesne Ekleme:

Bunun için “Şekil4”teki “Hierarchy” bölmesinden sol üstteki “+” sekmesinden “3D Objects” seçeniğinden oluşturmak istediğiniz nesneyi seçmeniz gerekiyor.

## 4.Sahnedeki Nesneye Materyal Ekleme:

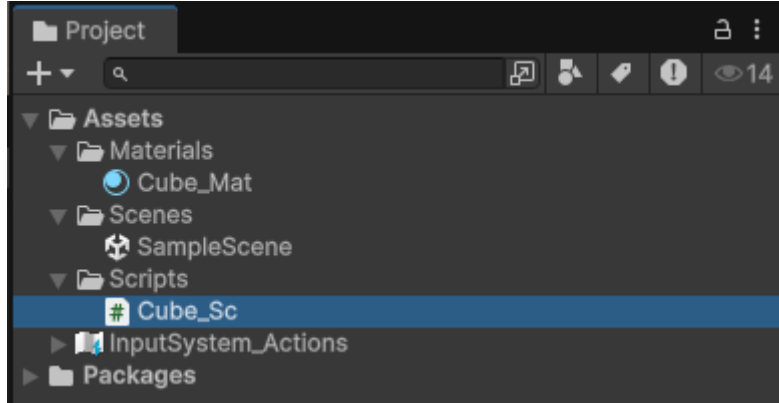
Bunun için “Şekil4”te görüldüğü üzere “Project” bölmesinden “+” kısmından “Material”ı seçmemiz gerekiyor. Bunu sürükleyip “Hierarchy”deki nesnemizin üzerine bırakırsak bu materyali o nesneye bağlamış oluruz.



Şekil4

## 5.Sahnedeki Nesneye Script Ekleme:

Bunun için “Şekil5”te de görüldüğü üzere Project bölmesinden “+” sekmesine tıklayınca “MonoBehaviour Script” seçeneğine tıklayınca bir script oluşturuyoruz. Bunu da sürükleyip nesnemize bırakırsak birbirine bağlamış oluruz.



Şekil5

## 6.Script ile Nesnenin konumunu Bir Kez Değiştirme:

Bir kez değişiklik yapmak istediğimiz için start() fonksiyonu içerisine alttaki kodu yazmamız gerekiyor.

```
transform.position += new Vector3(0f, 0f, 0f); //başlangıç konumuna getirir.
```

## 7.Script ile Nesnenin konumunu Sürekli Değiştirme:

Sürekli değişiklik yapmak istediğimiz için update() fonksiyonu içerisine alttaki kodu yazmamız gerekiyor.

```
transform.position += Vector3.right; //sağa doğru hareket ettirir.
```

## 8.Zamanın Normalizasyonu:

Bu farklı FPS değerlerine sahip sistemlerde aynı çalışmasını sağlar. Bu yüzden `time.deltaTime` ile çarpmamız gerekiyor. Yani,

```
transform.position += Vector3.right * Time.deltaTime;
```

## 9.Speed değişkenini tanımlama:

Bunun için fonksiyonların üzerinde bir değişken tanımlamalıyız.

Public olursa Unity Inspector penceresinde görünür ve başka scriptler erişebilir.

Private olursa sadece bu script erişir ve Inspector'da görünmez.

```
public float speed = 5f;
```

## 10.Klavye Tuşları ile Kontrol Etme(Dikey):

Bunun için Unity üzerinden klavye tuşlarınıza hangi ad ile erişebildiğine bakmanız gerekiyor. Örneğin Default olarak dikey için "Vertical" kullanılır. Bunu da `GetAxis()` fonksiyonu ile alabiliriz.

```
float verticalInput = Input.GetAxis("Vertical");  
transform.position += Vector3.up * verticalInput * speed * Time.deltaTime;
```

## 11.Klavye Tuşları ile Kontrol Etme(Dikey + Yatay):

```
// Dikey (↑ ↓ veya W/S)  
float verticalInput = Input.GetAxis("Vertical");  
  
// Yatay (← → veya A/D)  
float horizontalInput = Input.GetAxis("Horizontal");  
  
// Toplam hareket vektörü (x = yatay, y = dikey)  
Vector3 movement = new Vector3(horizontalInput, verticalInput, 0f);  
  
// Normalize et (çapraz hareketlerde hız artmasın)  
if (movement.magnitude > 1f)  
    movement.Normalize();  
  
// Nesneyi hareket ettir (FPS bağımsız)  
transform.position += movement * speed * Time.deltaTime;
```

Input.GetAxis ile yön tuşlarından bilgi alınıyor.

Vector3(x,y,z) ise bunu birleştiriyor.

Ardından hız artmaması için Normalize ediyoruz.

En nihayetinde de nesneyi hareket ettiriyoruz.

## 12.Sonuç:

Unity uygulamasına kısa bir giriş yapıldı.