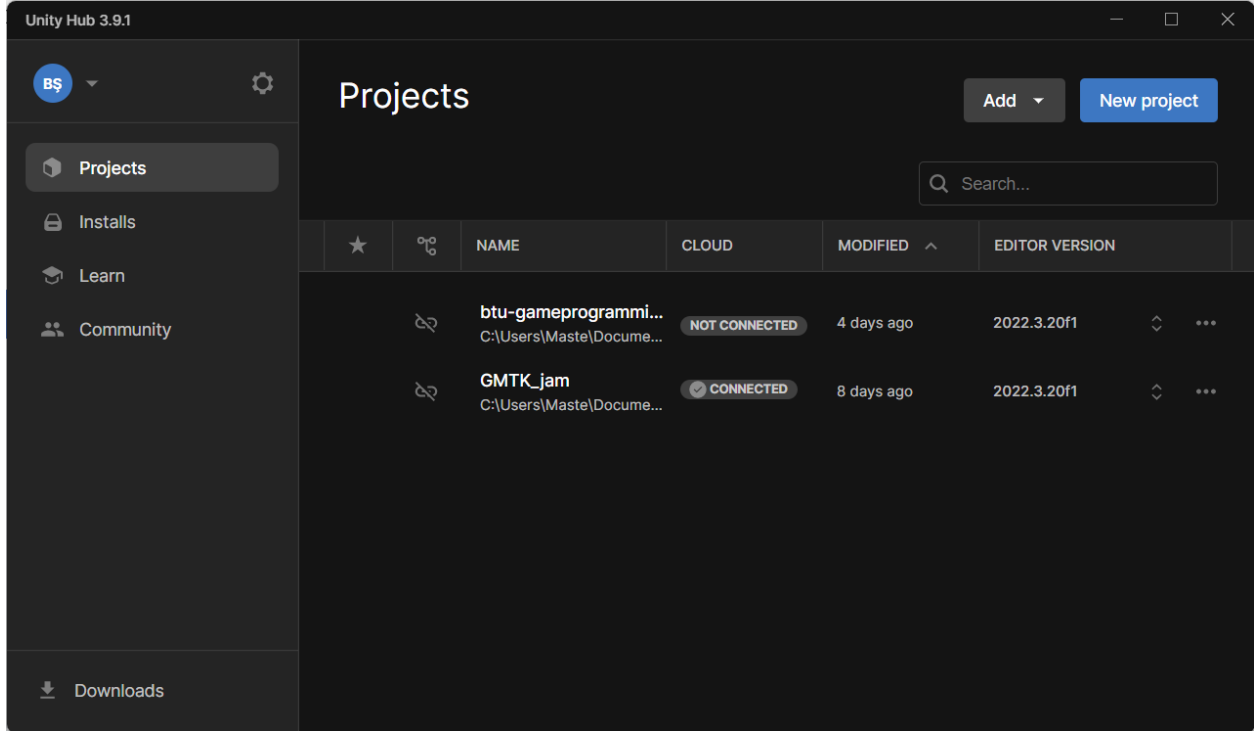


# 1.Hafta Ders Raporu – Batuhan Şengül

## Unity

### Unity Hub

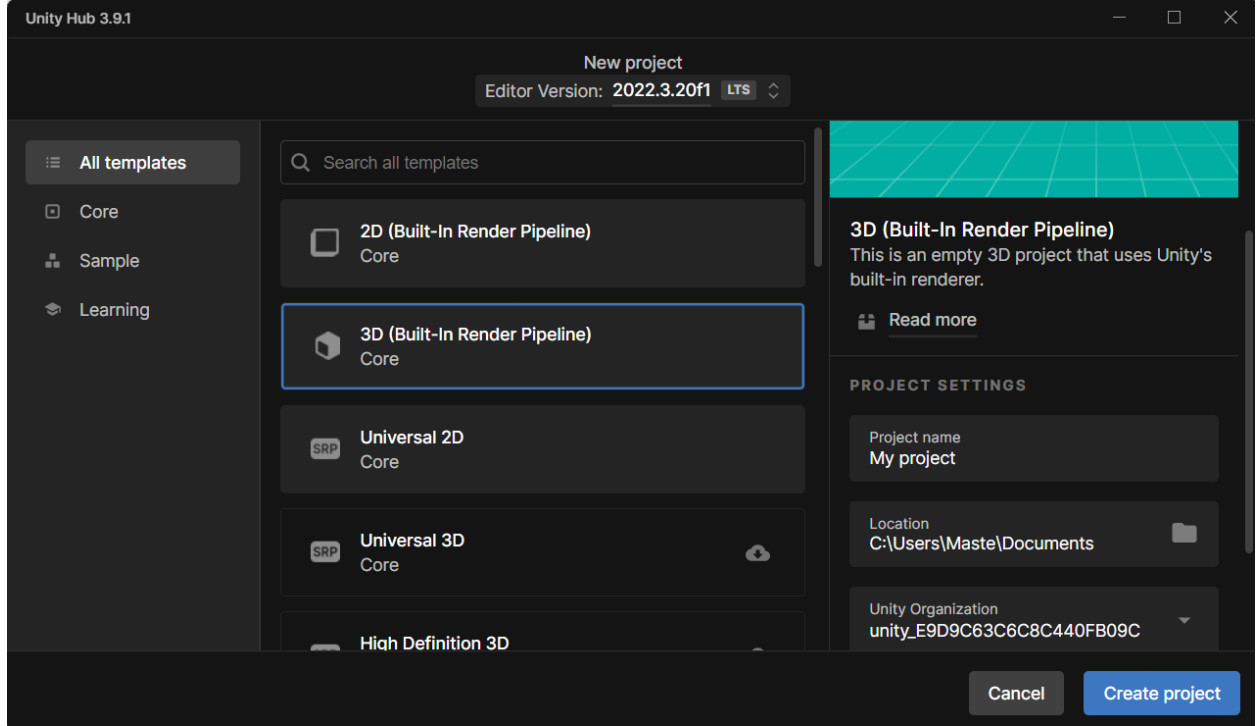


### 1. Sol Sekmeler

- **Projects:** Mevcut projelerin listelendiği sekme. Bu sekmede daha önce oluşturduğunuz projelere erişebilir, projeleri düzenleyebilir veya başlatabilirsiniz.
- **Installs:** Bilgisayarınızda yüklü olan Unity Editor sürümlerini gösterir. Yeni Unity sürümleri buradan eklenebilir.
- **Learn:** Unity'nin öğretici (tutorial) içeriklerine ve öğrenme kaynaklarına erişim sağlar.
- **Community:** Unity topluluğuna erişim sağlayan bir sekmedir. Forumlar, yardım ve rehber içeriklere ulaşmak için kullanılır.
- **Downloads:** İndirilen veya indirilmekte olan Unity sürümleri ve diğer araçların yönetildiği bölümdür.

## 2. Yeni Proje Oluřturma

Project sekmesin Unity Hub açıldıđı zaman ilk karřımıza çıkan penceredir. Sađ üstten “New Project” tuřuna basılarak yeni proje oluřturulma ekranı açılır.



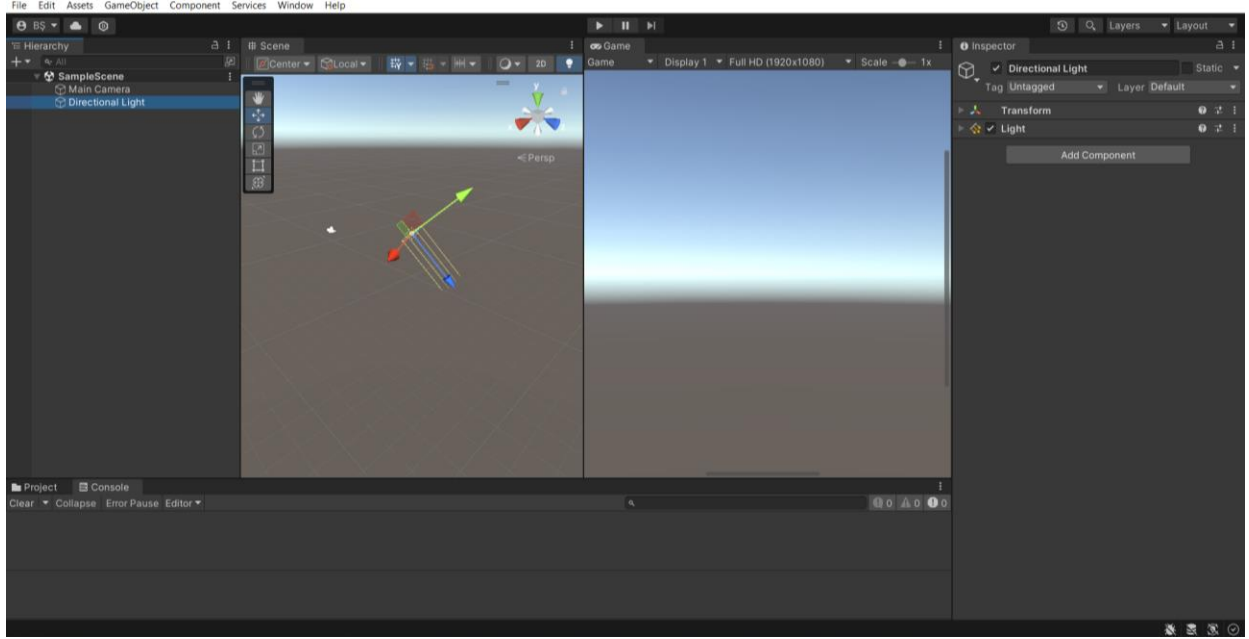
**Template Seçimi:** Yapılacak oyun için 2D, 3D, URP (Universal Render Pipeline) gibi şablonlar arasından seçim yapılır

**Proje İsmi:** Projeye bir isim verilir.

**Proje Dizinini Seçme:** Projeyi bilgisayarınızda nereye kaydedeceğinizi belirlemeniz gerekir.

**Create Project:** Tüm ayarları yaptıktan sonra "Create Project" butonuna basarak yeni projeyi oluřturulur.

## Unity Arayüzü – Ana Bileşenler



- **Hierarchy (Sol Üst)**

Bu panel, oyun sahnesinde (SampleScene) bulunan oyun nesnelerinin listelendiği yerdir. Unityde yeni bir proje oluşturulduğu zaman sahne içerisinde Main Camera ve Directional Light oyun nesneleri varsayılan olarak oluşturulur.

- **Scene (Ortada Sol)**

Sahne oyun nesnelerinin bulunduğu uzaydır, Oyun tarzına göre 2 boyut ya da 3 boyutlu olabilir. Burada oyun nesnelerinin konumu, büyüklüğü, rotasyonu gibi özellikler değiştirilebilir. Sahne araçları(Tools):

- **Görüş Aracı(View Tool):** Fare kullanarak sahne üzerinde kamerayı oynatmaya olanak tanır. Fare ile sol tıklama sahne içinde kamerayı kaydırır, sağ tıklama kamerayı döndürür.
- **Hareket Aracı(Move Tool):** Sahne içinde seçilen objenin yerel ya da küresel eksenlerde hareket ettirilmesine olanak tanır.
- **Döndürme Aracı(Rotate Tool):** Sahne içinde seçilen objenin merkezine ya da dayanağına(pivot) göre eksenlerde döndürülmesine olanak tanır.
- **Ebat Aracı(Scale Tool):** Sahne içinde seçilen objenin ebatlarının eksenlere göre değiştirilmesine(büyütme ya da küçültmesine) olanak tanır.

- **Dönüştürme Aracı(Transform Tool):** Yukarıda açıklanan özelliklerin yapılmasına olanak tanır.

- **Game (Ortada Sağ)**

Bu sekme, sahneyi oyun oynarken nasıl görüneceğini gösterir. Oyunun görüntüsü kamera oyun nesnesi tarafından oluşturulur.

- **Inspector (Sağ Üst)**

Hierarchy de seçili olan oyun nesnesinin bileşenlerinin(component) ince ayarlarının yapılabildiği alandır. Her oyun nesnesinin kendisine özel ve varsayılan bileşenleri vardır. Örnekteki seçili oyun nesnesindeki;

- **Transform Bileşeni:** Her oyun nesnesinde bulunur ve ilgili oyun nesnesinin uzaydaki konum, rotasyon ve boyut bilgilerini içerip isteğe bağlı düzenlenmesine olanak tanır.
- **Light Bileşeni:** Opsiyonel bir bileşendir, ışık yayma özellikleri gösterecek oyun nesnelerinde bulunur. Işığın yayılım tipi, rengi vb. ayarların isteğe bağlı düzenlenmesine olanak tanır.

- **Project (Sol Alt)**

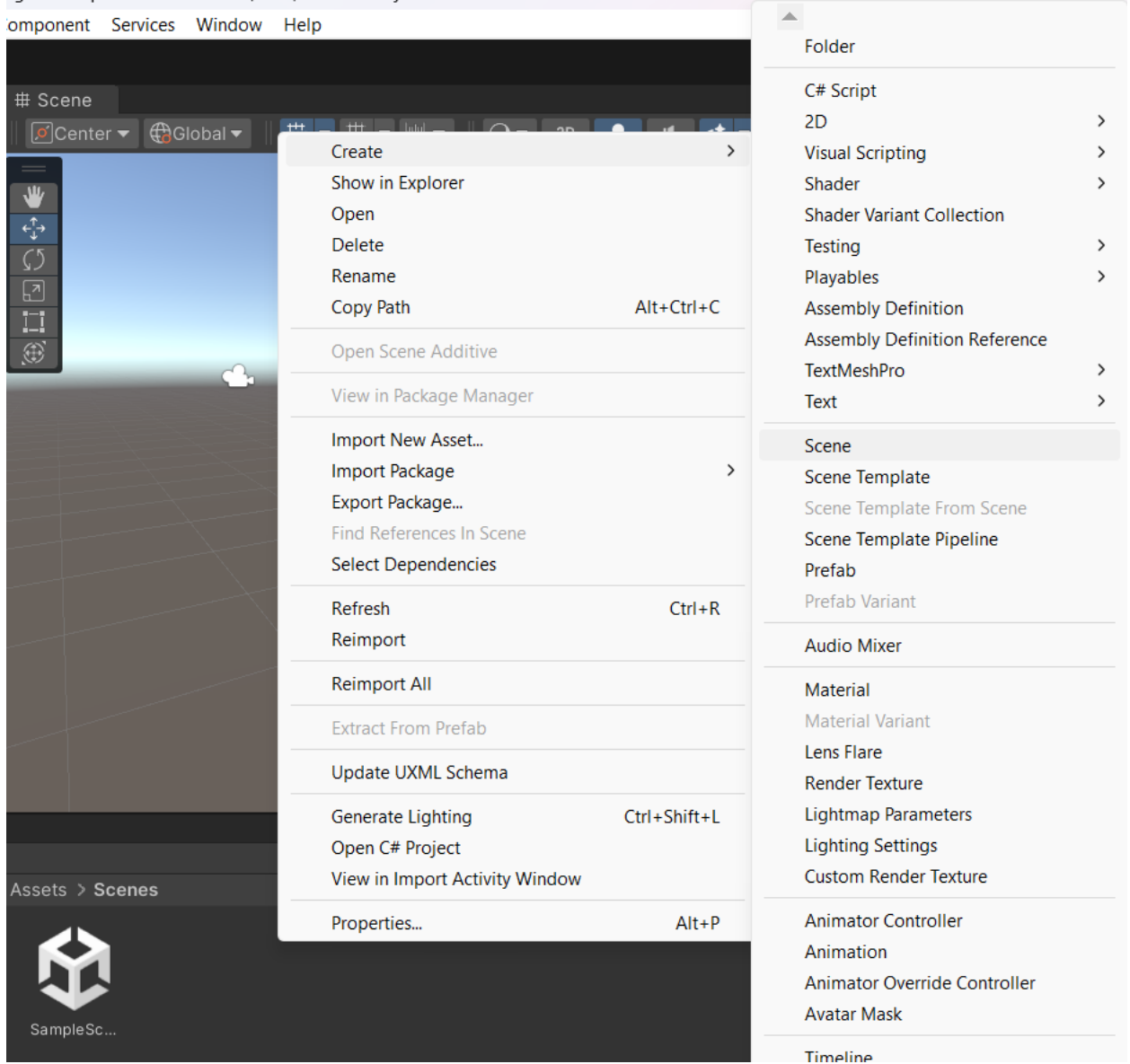
Bu sekme, proje içindeki tüm varlıkları (assets) barındırır. Oyuna sonradan oluşturulacak varlıklar da burada hiyerarşik bir yapıda bulunmaktadır.

- **Console (Sol Alt, Project'in Yanında)**

Hata mesajlarını veya debug loglarını buradan gözükmektedir. Genelde oyun geliştirirken scriptlerin dönütleri de burada gösterilebilir.

# Scene(Sahne)

## Sahne Oluřturma



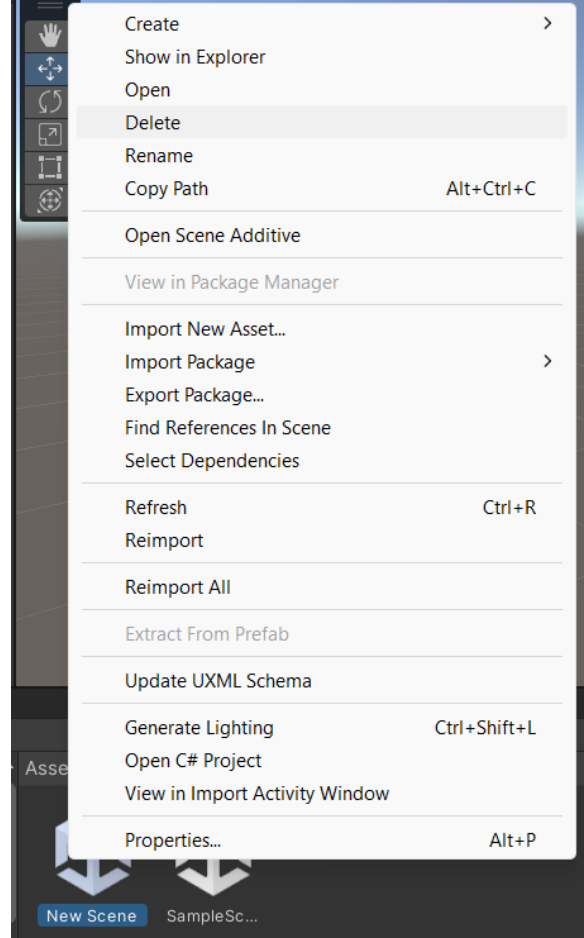
Sol alt kısımda bulunan **Project** panelinde, **Scenes** klasörüne sağ tıklayıp **Create > Scene** seçeneği ile de yeni sahneler oluşturulur.

## Sahne Silme

**Project** penceresinde, silmek istediğin sahneyi bulunur (genelde bu sahneler **Assets** klasörünün altında **Scenes** adında bir klasörde yer alır).

Silmek istediğin sahneye sağ tıklayıp **Delete** seçeneği seçilir.

Bir sahneyi silmek geri alınamaz bir işlem olduğundan dikkatli olunmalıdır. Eğer bir sahne üzerinde önemli değişiklikler yapıldıysa silmeden önce yedeğinin alınması mantıklı olacaktır.

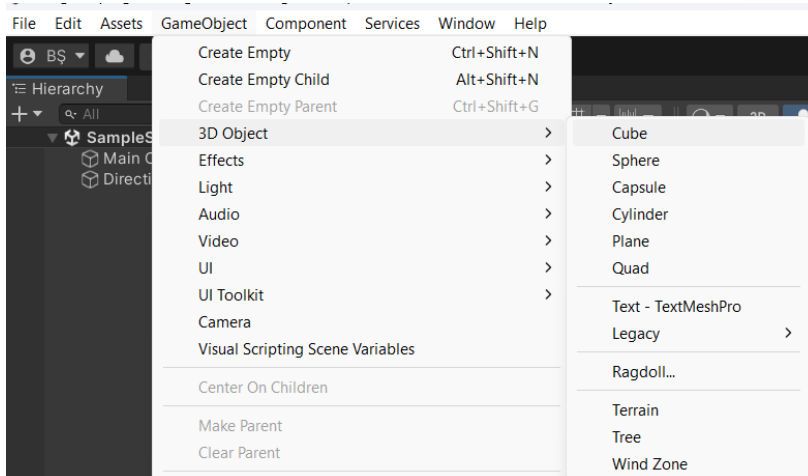


## Sahneye Nesne (Küp) Ekleme

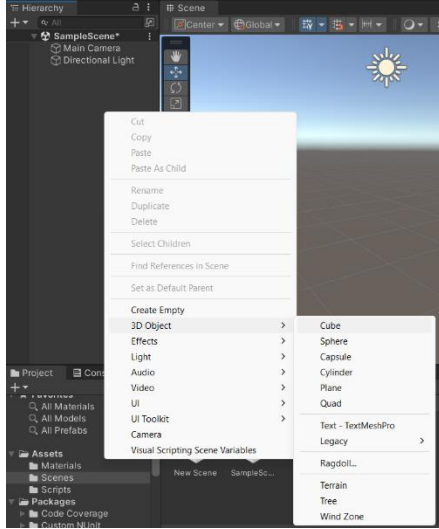
- İlk Yöntem

Sahne penceresine sağ tıklayın veya üst menüdeki **GameObject** menüsüne git.

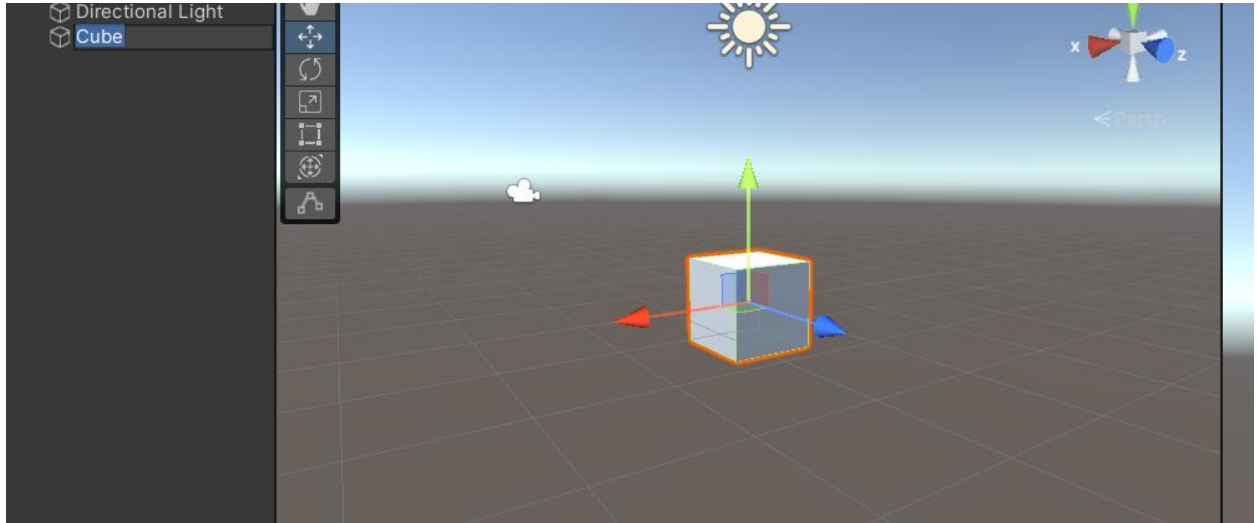
**3D Object > Cube** seçeneğini seç. Bu işlem sahneye bir küp nesnesi ekleyecektir.



- İkinci Yöntem



**Hierarchy** sekmesi içinde sağ tıklanarak **3D Object > Cube** seçilerek yeni bir küp oyun objesi oluşturulur.



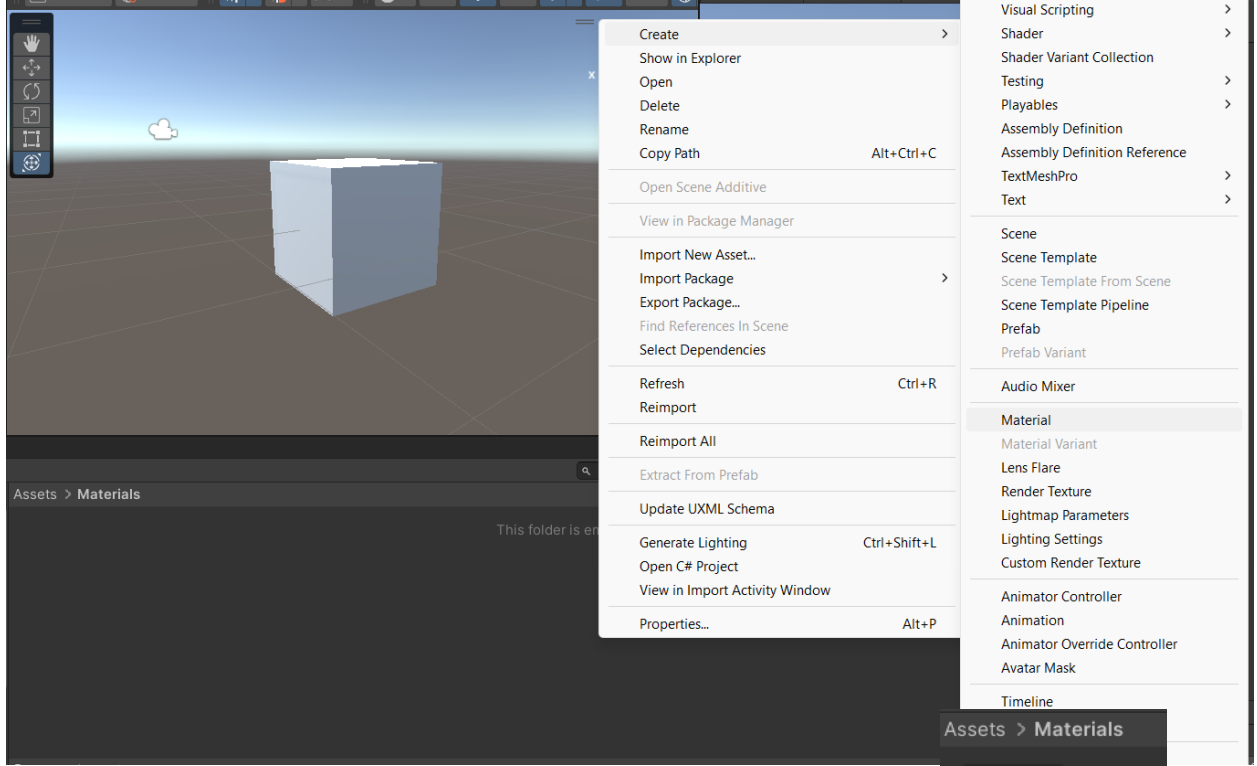
Eklenen küp, sahnede belirli bir pozisyonda görünecektir. **Scene** penceresinden küpü seçip taşıyabilir, döndürebilir ya da boyutları değiştirilebilir.

Sağ taraftaki **Inspector** penceresinde, küpün **Transform** özelliklerinden konum, boyut ve rotasyon gibi ayarları düzenlenebilir.

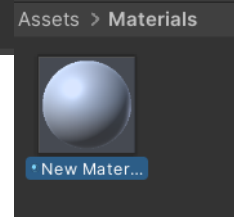
# Materyal(Material)

## Yeni Materyal Oluřturma

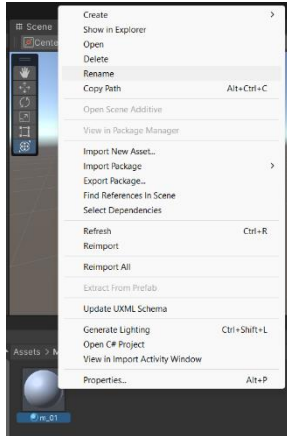
Proje sekmesi içinde farenin sađ tuřuna tıklanarak sırasıyla Oluřtur(Create) -> Materyal(Material) seřilerek yeni materyal oluřturulur.



Oluřturulan materyal yandaki gibi g z k r.

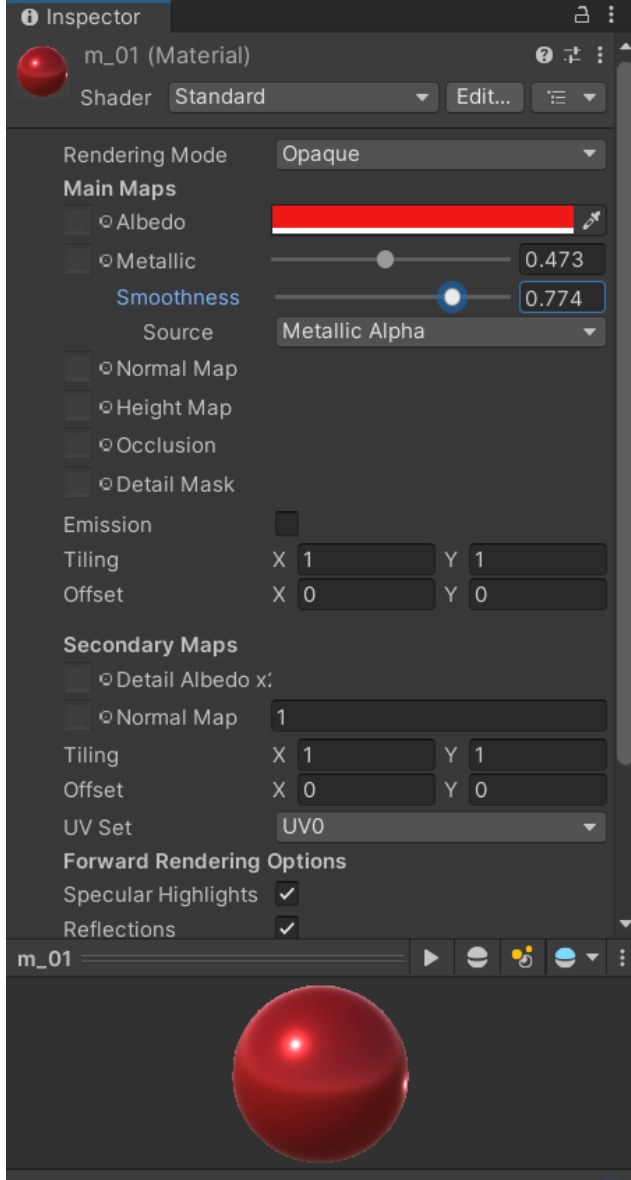


Oluřturulan materyale sađ tıklanarak ismi deđiřtirilebilir(Rename).  
Bu  rnekte materyalin adı “m\_01” olarak deđiřtirilmiřtir.



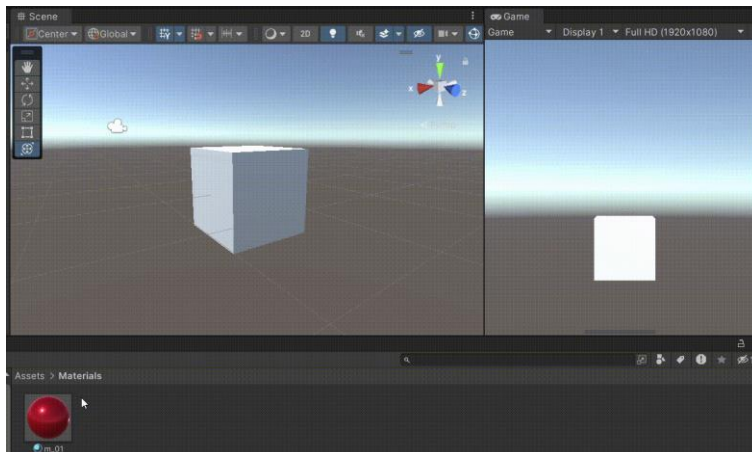


## Materyal Özellikleri



Inspector üzerinden oluşturulan materyale tıklandığında özellikleri bu şekilde gözükmetedir. Standard Shaderdaki bazı ayarlar:

- Rendering Mode: Opak, Transparan gibi materyal özelliği seçilebilir.
- Albedo: Renk ayarlarının yapıldığı alandır.
- Metallic: Metalik parlaklık ayarının yapıldığı alandır.
- Smoothness: Pürüzsüzlük/Matlık ayarının yapıldığı alandır.
- Normal Map: Materyalin detay haritası başka bir deyişle ışığın derinlik haritasıdır.
- Height Map: Materyalin yükseklik haritasıdır.
- Occlusion: Nesnenin saydamlık kullanıp kullanmayacağını ve eğer öyleyse hangi tür karıştırma modunu kullanacağını seçiminin yapıldığı alandır.
- Detail Mask: Detay dokusunun uygulanması için modelinizin belirli alanlarını maskeleyenize olanak tanıdığı alandır.
- Emmison: Materyal yüzeyinin ışık yayma özelliklerinin kontrolünün yapıldığı alandır.



## Oyun Nesnesinin Materyalini Değiştirmek

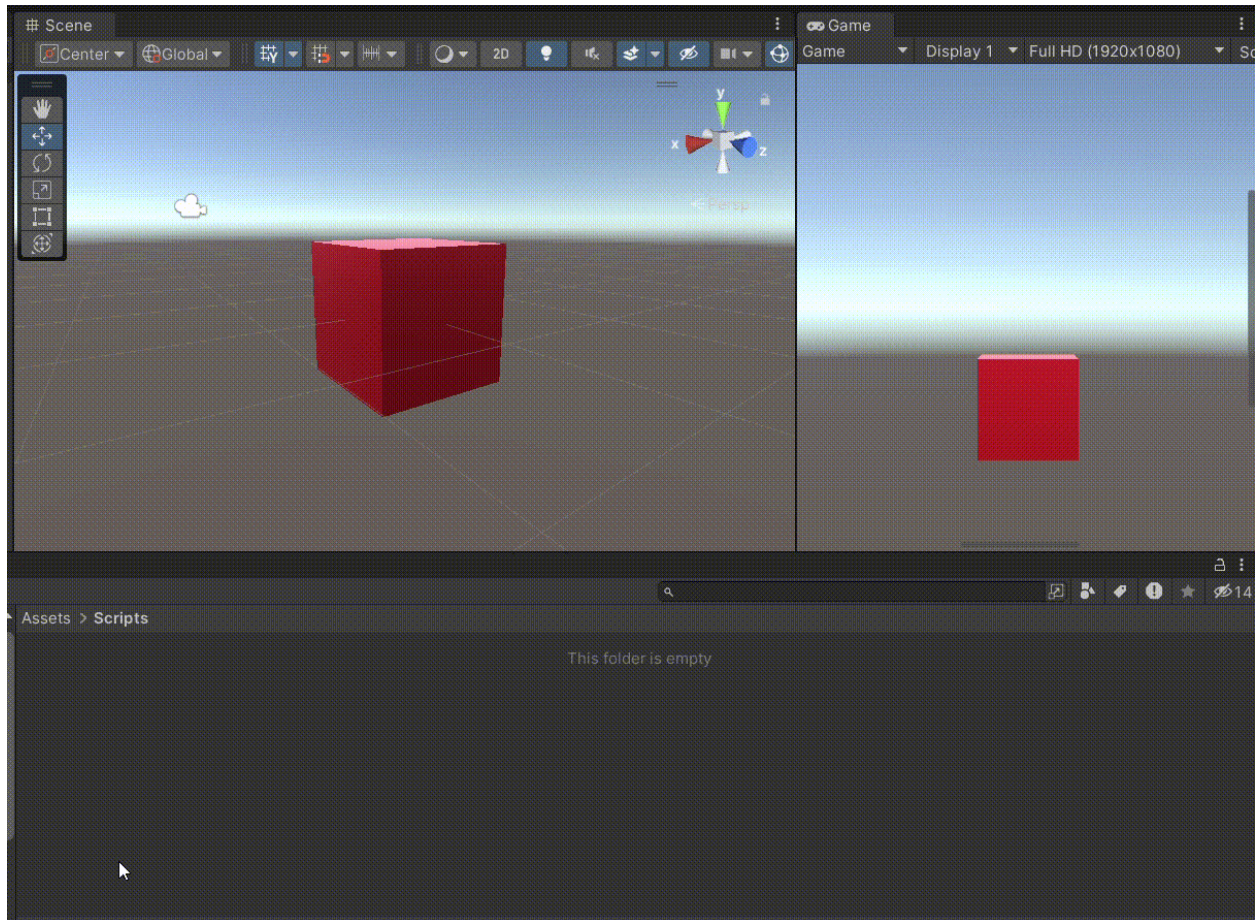
Oluşturduğumuz materyali sahnedeki herhangi bir objeye sürükleyerek ilgili objenin materyalini yandaki gibi değiştirebiliriz. (GIF DOSYASI PDF ÜZERİNDE GÖZÜKMÜYOR)

# Scriptler

Unity’de scriptler, oyun nesnelerinin (GameObject) davranışlarını kontrol etmek için kullanılan yazılım bileşenleridir. Genellikle **C#** dilinde yazılırlar ve Unity’de bir nesneye atanarak o nesnenin ne yapacağını belirlerler. Bir script, oyunun mantığını (logic), fiziksel etkileşimleri, kullanıcı girdilerini ve oyun dünyasındaki olayları yönetir.

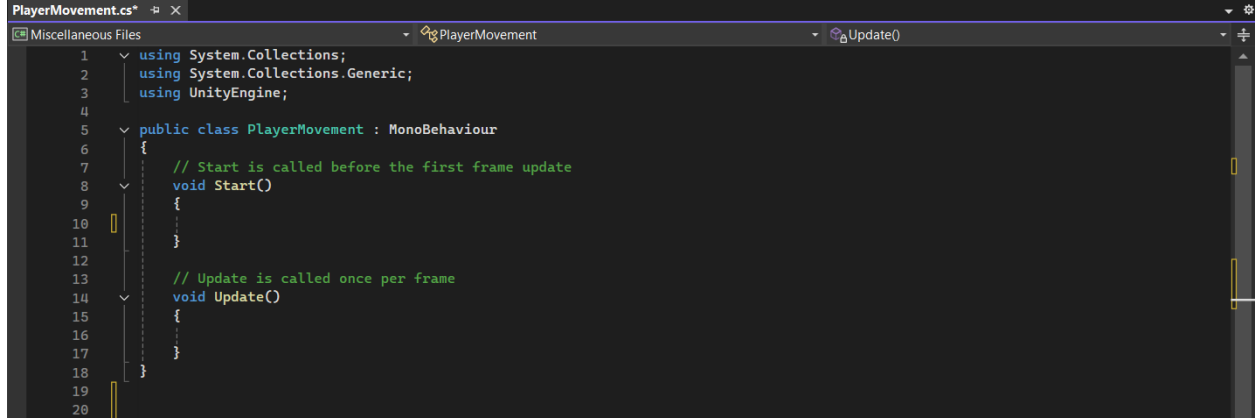
## Script Oluşturma

Project panelinin içerisinde sırasıyla fare sağ tıklanır -> Create -> C# Script seçilerek yeni bir script oluşturulup ismi düzenlenir. Aşağıdaki örnekte bir script oluşturulup adı “PlayerMovement” yapılmıştır.(GIF DOSYASI PDF ÜZERİNDE GÖZÜKMÜYOR)



## Varsayılan Script İçeriği

Oluşturulan bir script içerisinde bazı fonksiyonlar varsayılan olarak oluşur. Bunları açıklamak gerekirse:



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerMovement : 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
20
```

- **Start() Fonksiyonu**

**Start** fonksiyonu, bir oyun nesnesi ilk kez etkinleştirildiğinde **bir kez** çalışır. Oyun başladığında ya da nesne sahnede aktif hale geldiğinde çalışır.

Genellikle oyun nesnelerinin başlangıçtaki ayarlarını yapmak için kullanılır. Örnek olarak ilgili oyun nesnesinin başlangıç konumunu(spawnpoint) ayarlamak vb.

- **Update() Fonksiyonu**

**Update** fonksiyonu, her karede (frame) bir kez çalışır. Yani, oyunun her bir kare güncellemesinde otomatik olarak çağrılır. Bu da saniyede birçok kez (FPS'ye bağlı olarak) çalışacağı anlamına gelir. Bu hususa dikkat edilmediği takdirde farklı kullanıcı sistemlerinde dengesizliklere sebep olabilir.

Genellikle nesnelerin sürekli ve dinamik değişimlerini takip etmek için kullanılır. Örneğin, oyuncu girdilerini kontrol etmek (klavye, fare veya gamepad), karakter hareketlerini düzenlemek, fiziksel hesaplamaları sürekli güncellemek gibi görevler için kullanılır.

## Script ile Oyun Nesnesini Hareket Ettirmek

```
public class PlayerMovement : MonoBehaviour
{
    [SerializeField]
    private float speed=3f;

    private float horizontalVal, verticalVal;
    ✎ Unity Message | 0 references
    void Start()
    {
        Debug.Log("Game Started");
    }

    ✎ Unity Message | 0 references
    void Update()
    {
        horizontalVal = Input.GetAxis("Horizontal");
        verticalVal = Input.GetAxis("Vertical");

        //Oyun nesnesinin Sag veya Sola hareketi
        transform.Translate(horizontalVal* Vector3.right * speed * Time.deltaTime);

        //Oyun nesnesinin Yukari veya Asagi hareketi
        transform.Translate(verticalVal * Vector3.up * speed * Time.deltaTime);

        //Ekran ciktilari
        if (horizontalVal != 0) Debug.Log("Horizontal value: " +horizontalVal);

        if (verticalVal != 0) Debug.Log("Vertical value: " + verticalVal);
    }
}
```

Bu kod, Unity'de bir oyun nesnesinin hareketini kontrol eden bir script örneğidir. Kod içerisinde bulunan bileşenler:

- **Değişkenler**

[SerializeField]

Bu etiket, Unity Editor'da özel bir değişkenin görünmesini sağlar, böylece Unity arayüzünde bu değişkeni ayarlayabilirsiniz. Normalde private değişkenler Unity arayüzünde görünmez, ancak bu etiketle görünür hale gelir.

**private float speed;** Bu değişken, nesnenin hareket hızını belirler ve Unity Editor üzerinden ayarlanabilir.

**horizontalVal** ve **verticalVal**: Bu değişkenler, oyuncunun girdiği yatay ve dikey hareketleri saklar. Bunlar daha sonra Update fonksiyonunda oyuncunun hangi yöne hareket ettiğini belirlemek için kullanılır.

- **Debug.Log(...)**

Bu fonksiyon konsola içerikleri yazdırmakta kullanılır.

*horizontalVal = Input.GetAxis("Horizontal");*

Bu satır, yatay (sol-sağ) girdi değerini alır. Bu değer genellikle -1 (sol) ile 1 (sağ) arasında değişir.

*verticalVal = Input.GetAxis("Vertical");*

Bu satır, dikey (yukarı-aşağı) eksenindeki girdiyi alır. Bu değer de -1 (aşağı) ile 1 (yukarı) arasında değişir.

- **Nesnenin Hareketi**

*transform.Translate(horizontalVal \* Vector3.right \* speed \* Time.deltaTime);*

Bu satır, nesnenin sağa veya sola hareket etmesini sağlar. horizontalVal, oyuncunun yatay ekseninde hareket etme isteğini gösterir. Vector3.right sabiti, X ekseninde sağ yönü temsil eder. Hız(speed), hareketi ayarlamak için çarpılır. Time.deltaTime, bu işlemin saniye başına olacak şekilde düzenlenmesi için kullanılır, yani hareket FPS'den bağımsız hale getirilir.

*transform.Translate(verticalVal \* Vector3.up \* speed \* Time.deltaTime);*

Bu satır, nesnenin yukarı veya aşağı hareket etmesini sağlar. verticalVal ile dikey girdiyi alır, Vector3.up ise yukarı yönü temsil eder. Hız(speed), hareketi ayarlamak için çarpılır. Time.deltaTime, bu işlemin saniye başına olacak şekilde düzenlenmesi için kullanılır, yani hareket FPS'den bağımsız hale getirilir.

- **Ekran Çıktıları**

*if (horizontalVal != 0) Debug.Log("Horizontal value: " + horizontalVal);*

Eğer yatay eksenindeki girdi sıfır değilse (oyuncu hareket ediyorsa), o anki yatay değeri konsola yazdırır.

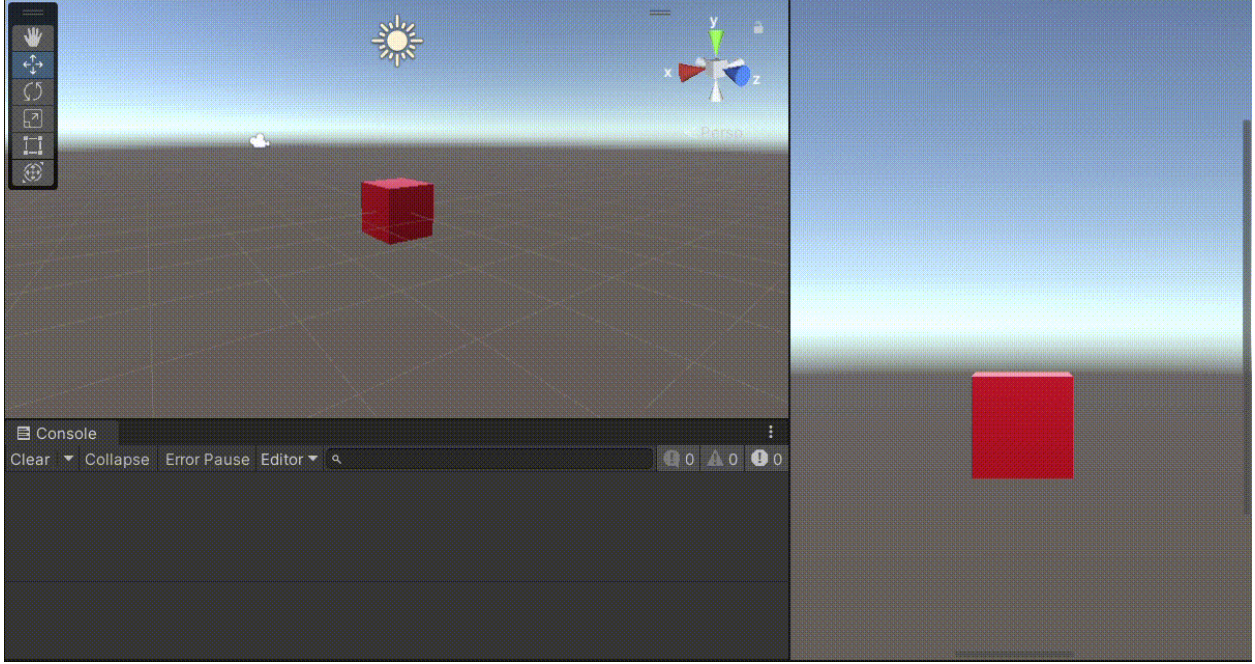
*if (verticalVal != 0) Debug.Log("Vertical value: " + verticalVal);*

Aynı şekilde, eğer dikey eksenindeki girdi sıfır değilse (oyuncu yukarı veya aşağı hareket ediyorsa), o anki dikey değeri konsola yazdırır.



## Örnek Scriptin Gösterimi

Özetle bu script, klavye girdilerine göre bir oyun nesnesinin X ve Y eksenlerinde hareket etmesini sağlar. Hız(speed) değişkeni ile nesnenin hareket hızı kontrol edilir bu değişken Inspector üzerinden ayarlanabilir. Debug.Log komutları, oyuncunun hareket ettiği yönleri konsola yazdırır. Uygulama görüntüleri aşağıda gösterilmiştir.



## Kaynakça

<https://docs.unity3d.com/Manual/>

<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UIBasicLayout.html>

<https://chatgpt.com/>

## Proje Kodu ve Github Repo

Kod: <https://github.com/bathuchan/btu-gameprogramming-BatuhanSengul/blob/main/Reports/1.Hafta/PlayerMovement.cs>

Proje Repo: <https://github.com/bathuchan/btu-gameprogramming-BatuhanSengul>

## Hazırlayan

Batuhan Şengül – 20360859008- bathu.sengul@gmail.com