

Oyun Programlama Ödevi – Hafta 2

Ad: Eren

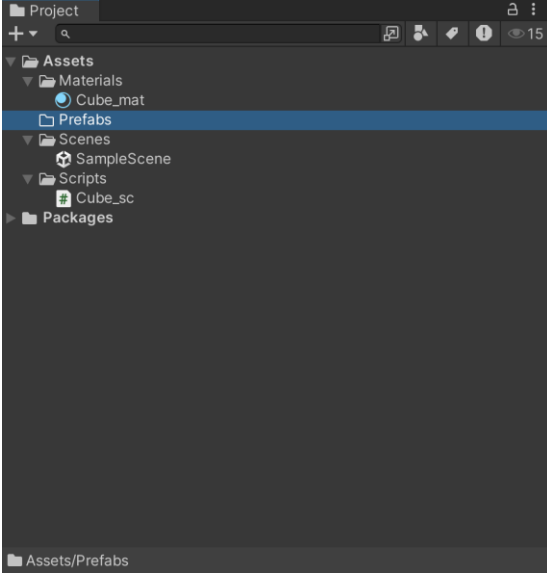
Soyad: Köse

Numara: 22360859075

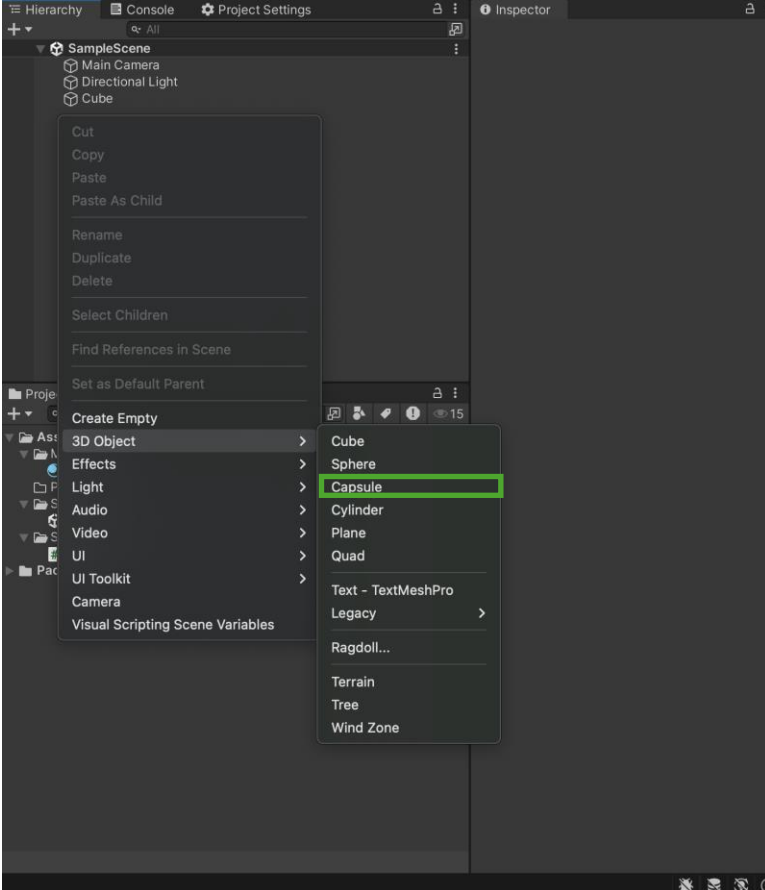
GitHub Kod Linki: <https://github.com/erennkose/btu-oyun-programlama/tree/main/Hafta2>

• **Laser Prefab’i Oluşturma**

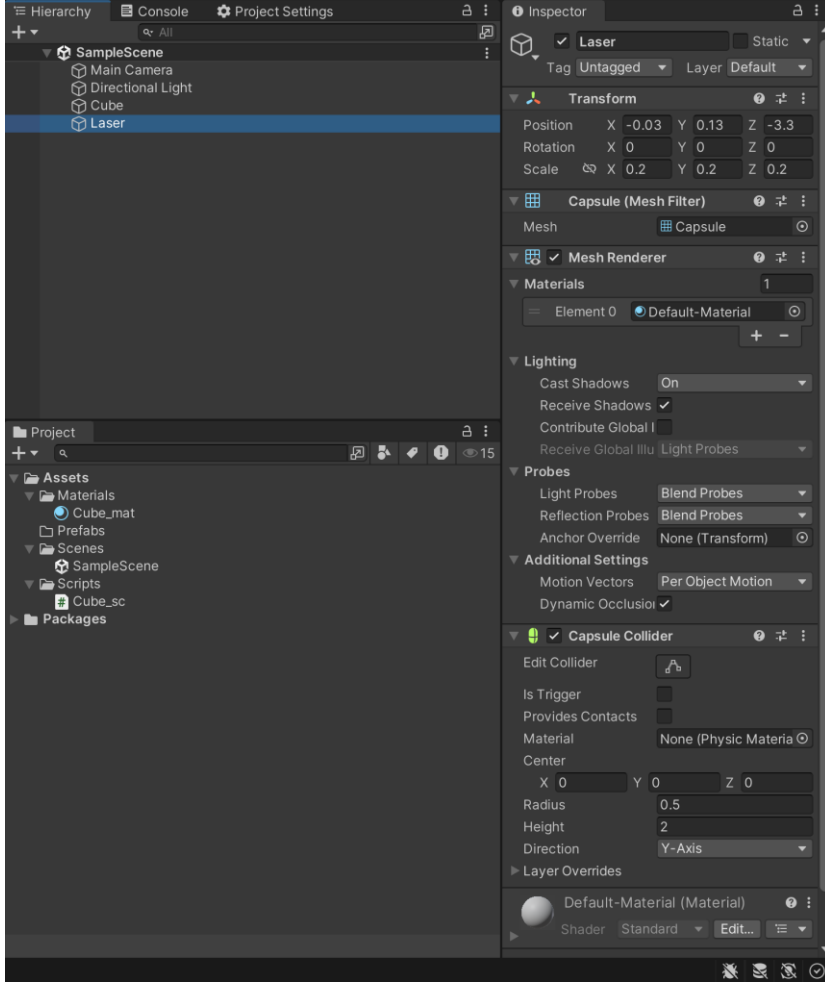
Laser Prefab’i oluşturmak için öncelikle projemizin düzeni için **Assets** klasörüne sağ tıklayıp **Create > Folder** seçeneklerini izleyerek **Assets** klasörü altında **Prefabs** adında bir klasör oluşturuyoruz.



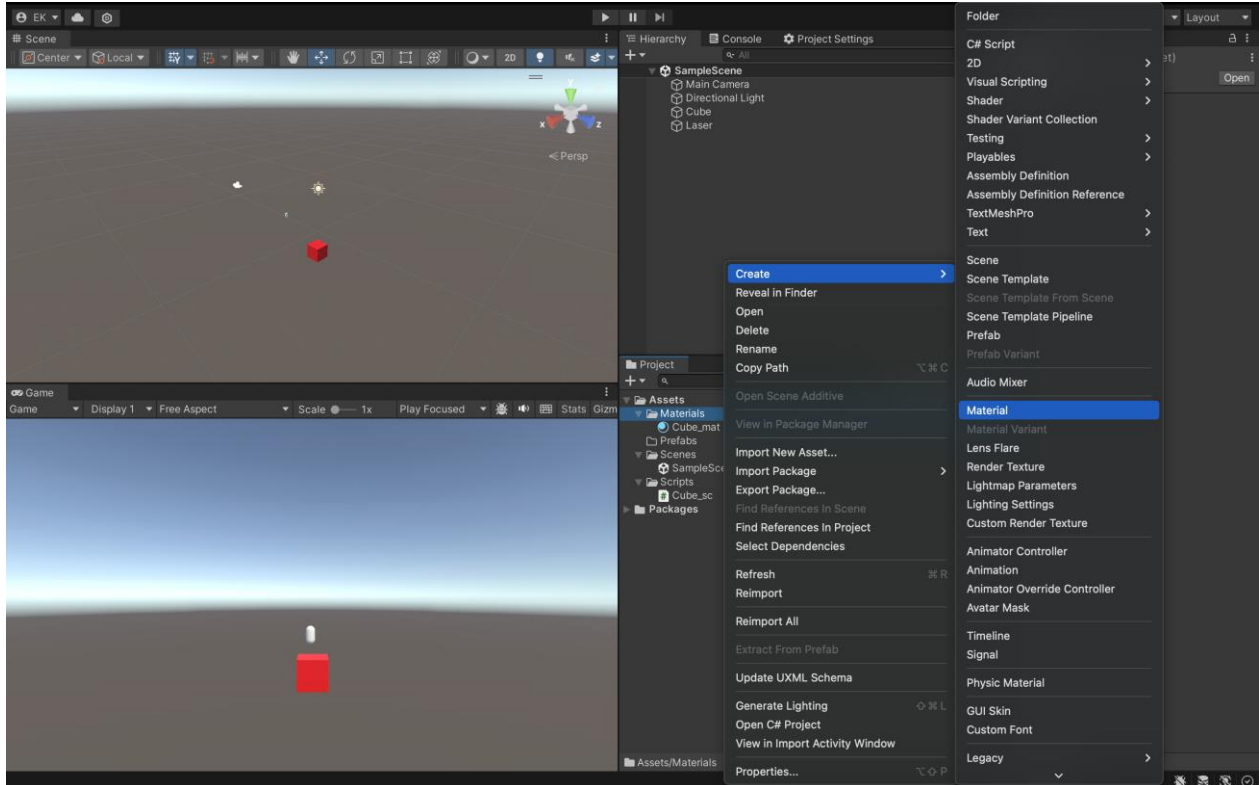
Klasörümüzü oluşturduktan sonra **Hierarchy** bölümünde açık olan sahnemizin altındaki boşluğa sağ tık atıyoruz ve **3D Object > Capsule** seçeneklerini izleyerek laserimiz için bir **Capsule** objesi oluşturmuş oluyoruz.



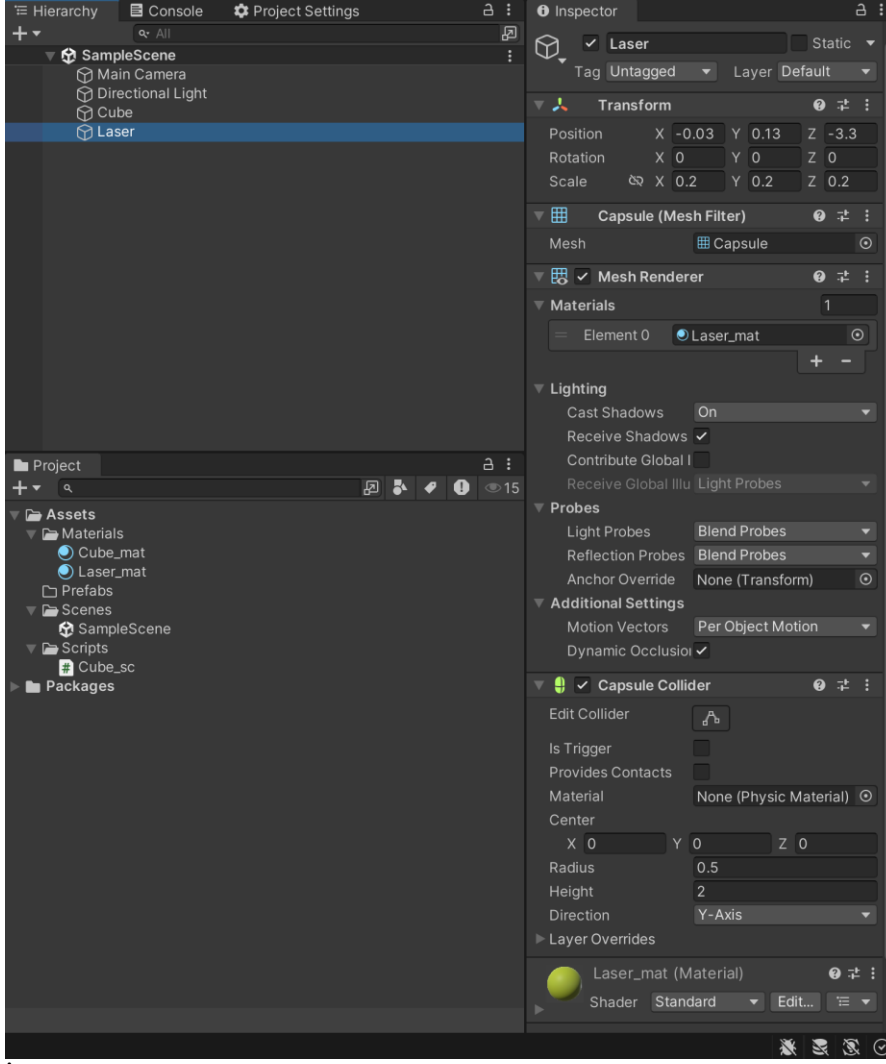
Bu adımların ardından oluşturduğumuz Laser objesinde, Prefab halinde istediğimiz özellikleri ayarlıyoruz. **Hierarchy** kısmından **Laser nesnemize** tıkladığımızda **Inspector** bölümünde açılan özelliklerden **Transform** bölümüne geliyoruz. Burada **Scale** yazan kısımdan oluşturulacak Laserlerimizin boyutunu x,y,z koordinatlarına göre ayarlıyoruz.



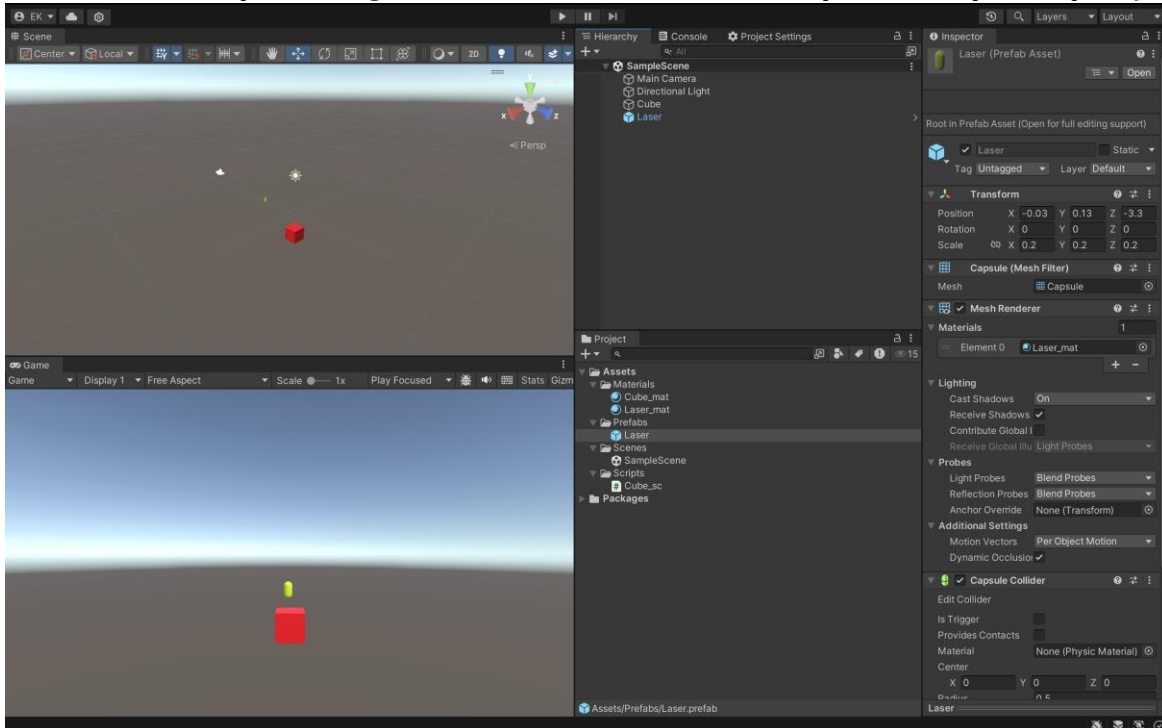
Laserimizin boyutunu ayarladıktan sonra ise Laser Prefab'imizde kullanılacak olan materyalimizi **Assets** klasörü altındaki **Materials** klasörüne sağ tık atıp **Create > Material** seçeneklerini izleyerek oluşturuyoruz.



Ardından oluşturduğumuz materyale isim veriyoruz ve bu materyali sürükleyip Laser objemizin üzerine bırakıyoruz. Böylece nesnemize materyal eklemiş oluyoruz.



İstediğimiz özellikleri ayarladıktan sonra ise **sürükle-bırak** yöntemiyle Laser objemizi sürükleyerek önceden oluşturmuş olduğumuz **Prefabs** klasörümüzün içine bırakıyoruz. Bu şekilde Laser Prefab'imizi, Laser nesnemiz için eklediğimiz tüm özellikleri barındırır bir şekilde oluşturmuş oluyoruz.



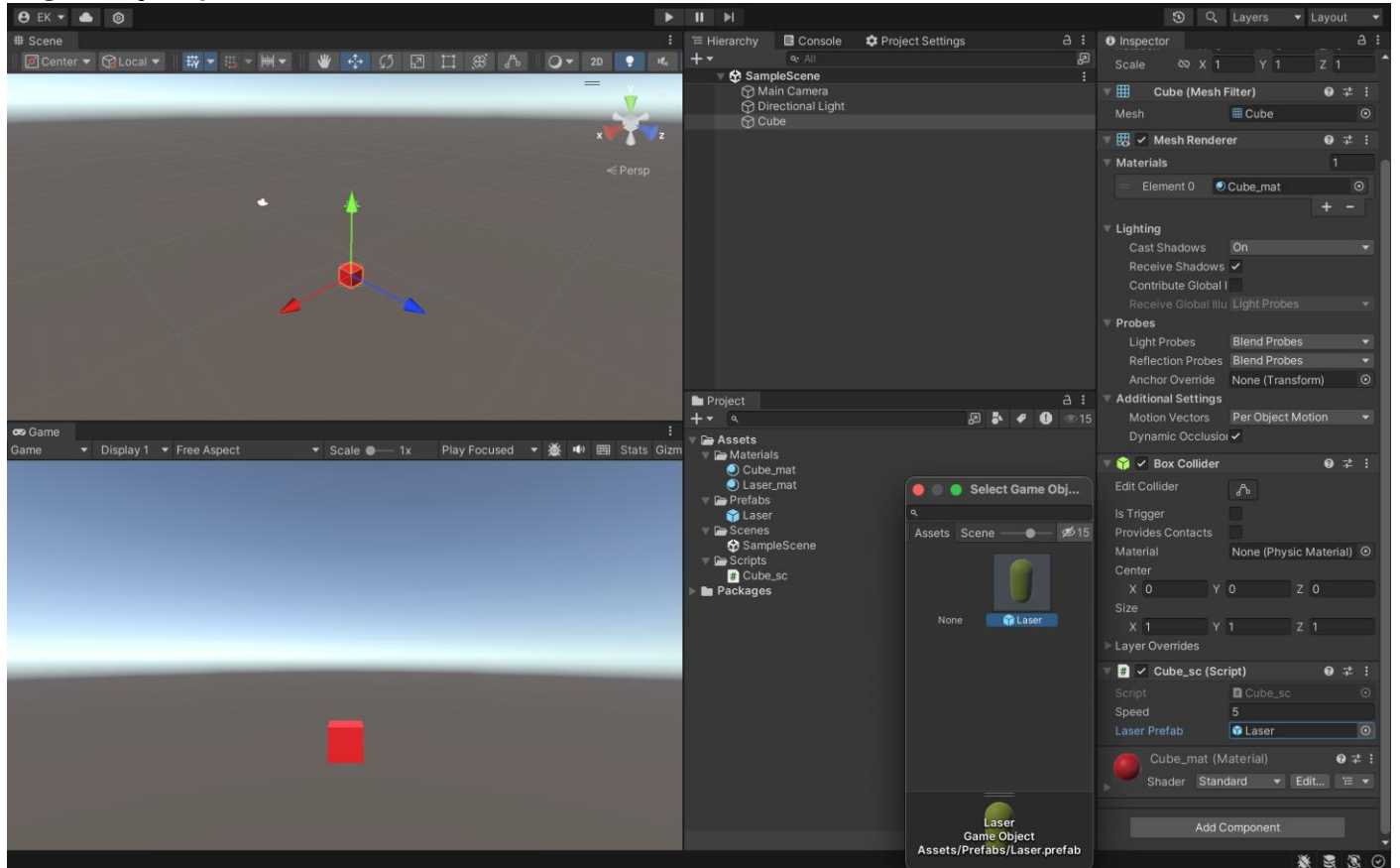
Prefab oluşturduktan sonra **Hierarchy** kısmından Laser nesnemizi artık silebiliriz.

• Space Tuşuna Basıldığında Laser Prefab'den Laser Oluşturma

Space tuşuna basıldığında Laser oluşturma işlemini yapmak için önceden oluşturmuş olduğumuz küpümüzün scriptine gitmemiz gerekiyor. **Scripts** kısmından **küpümüze atanan scripte** tıklıyoruz ve kod editörümüzde açıyoruz. Scriptimiz açılınca boşluk tuşunu her frame'de bir kontrol etmemiz gerekeceğinden **Update** fonksiyonumuzu buluyoruz ve bu fonksiyon içine kodumuzu yazıyoruz. Yazacağımız kod ile hem space tuşuna basılma durumunu kontrol edeceğiz ve laser oluşturacağız, hem de oluşturulan laserlerimizin her saniye bir birim yukarı hareket etmesini sağlayacağız. Öncelikle Laser Prefab'imize kod içerisinden erişebilmek için yapmamız gereken önemli bir işlem var. Giriş yaptığımız scriptteki classın içerisinde bir obje tanımlıyoruz.

```
public GameObject laserPrefab;
```

Scriptimizi kaydedip Unity uygulamamıza geri dönüyoruz. **Hierarchy** kısmından küpümüze tıkladığımızda **Inspector** kısmında görünen scriptimizin altındaki özelliklerde **Laser Prefab** obje değişkenini görebiliriz. Bu değişkenin sağ kısmındaki noktaya tıklıyoruz ve açılan küçük Pop-Uptan **Assets** kısmına tıkladığımızda görünen prefablerimizden bu objeye atanmasını istediğimiz prefabi seçiyoruz. Bu sayede artık kodumuzda kullandığımız laserPrefab nesnesiyle Laser prefabimiz bağlanmış oluyor.



Artık prefabimize kod üzerinden erişebildiğimize göre Space butonu kontrolü ve her saniyede bir birim yukarı gitme işlemlerine geçebiliriz. Space butonunu kontrol etmek için Update fonksiyonu içerisinde `Input.GetKeyDown(KeyCode.Space)` kod parçasını kullanacağız.

```
if (Input.GetKeyDown(KeyCode.Space)){  
    FireFunction();  
}
```

Yukarıdaki if durumu içerisinde kullandığımız FireFunction fonksiyonunu ise kendimiz oluşturacağız. Bu fonksiyonda **seri atışlar arasında bekleme süresi ekleyeceğiz** ve Instantiate fonksiyonunu kullanarak prefabimizden nesne oluşturacağız.

Seri atışlar arasında bekleme süresi eklemek için öncelikle birkaç değişken tanımlamamız gerekiyor. Bunları sınıfımızın içinde, fonksiyonların dışında tanımlıyoruz.

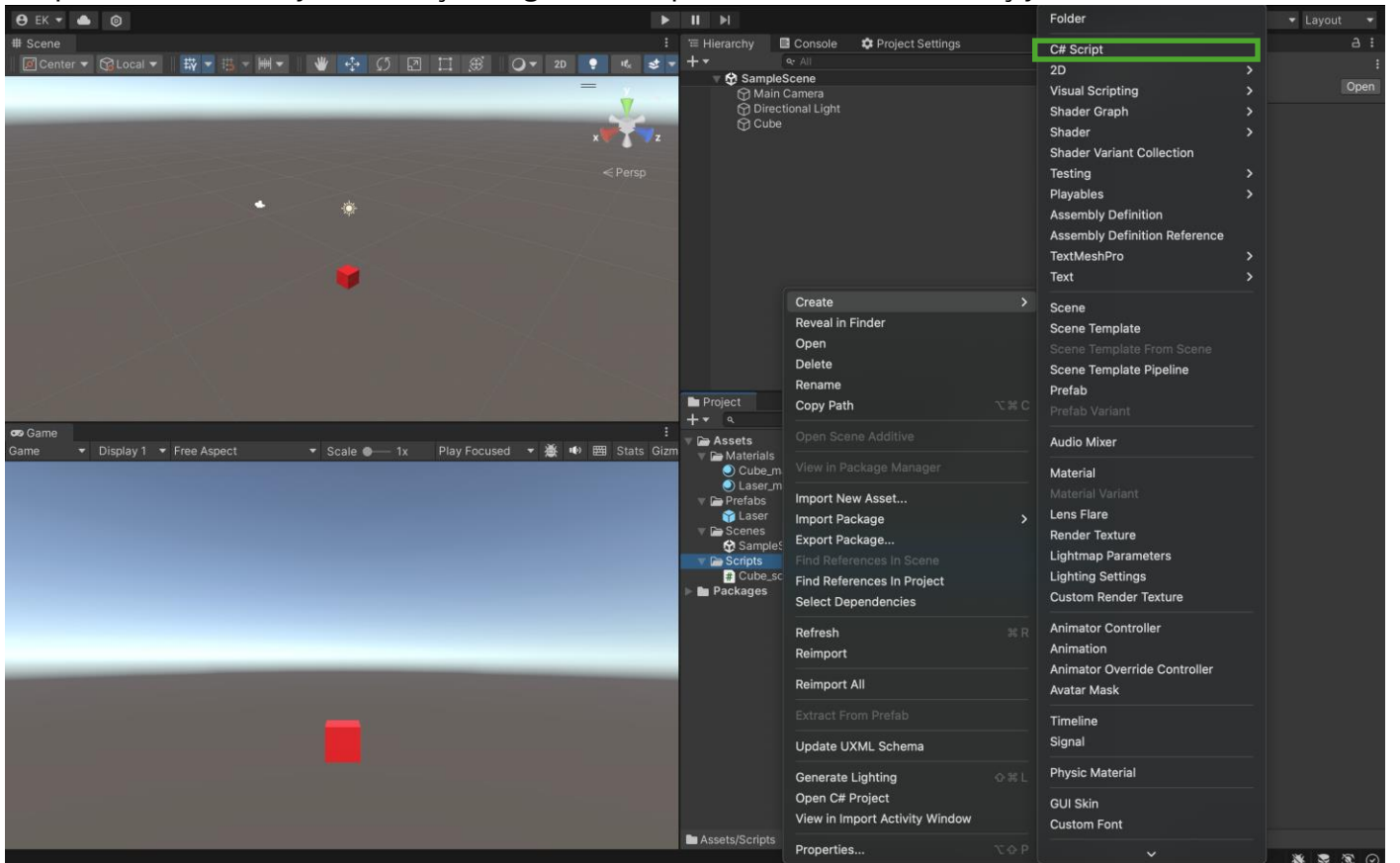
```
float fireRate = 0.5f;  
2 başvuru  
float nextFire = 0f;
```

Bu değişkenlerden **fireRate** değişkeni, her iki atış arasında beklenmesi gereken süreyi belirtmekte. **nextFire** değişkeni ise bir sonraki atışımızın vaktini belirtecek ve her framede güncellenecek. İlk atışımız 0 anından sonra herhangi bir anda olabileceğinden dolayı bu değişkene base değer olarak 0 atıyoruz. Şimdi bu değişkenleri kullanarak FireFunction fonksiyonumuza bakabiliriz.

```
void FireFunction(){  
    if (Time.time > nextFire){  
        Instantiate(laserPrefab, transform.position + new Vector3(0, 0.6f, 0), Quaternion.identity);  
        nextFire = Time.time + fireRate;  
    }  
}
```

Bu kodda Quaternion.identity ile laserimizin bakacağı yönü belirtiyoruz. Vector3 içine girdiğimiz parametrelerle gelen sonucu transform.positiona ekleyerek de laserimizin küpümüzden biraz daha yukarıda oluşturulmasını sağlıyoruz. Instantiate fonksiyonuna ilk parametre olarak ise oluşturulmasını istediğimiz nesnenin prefabini giriyoruz ve bu kod parçası sayesinde prefabimizden nesne oluşturulmasını sağlıyoruz. if ile şu anki zamanı gelecek atışımızın zamanına kıyaslıyoruz ve bu kıyasa göre prefabden nesne oluşturulmasına izin veriyoruz. nextFire değişkenini ise her atış yaptığımızda güncelliyoruz. **Bu sayede seri atışlar arasına bekleme süresi koymuş oluyoruz.**

Artık prefabimizi kullanarak nesnemizi oluşturabildiğimize göre, bu **nesnemizi her saniye bir birim yukarı hareket ettirme**, oluşturduğumuz **nesnenin ekrandan kaybolduğunda yok edilmesini sağlama** gibi durumları barındıracak olan prefab scriptimizi oluşturabiliriz. Bunun için öncelikle prefabimize ekleyeceğimiz scriptimizi, **Assets > Scripts** klasörümüz altında oluşturacağız. **Script** klasörümüze sağ tık atıp **Create > C# Script** seçeneklerini izliyoruz. Ardından oluşturulan scriptimizi isimlendiriyoruz. Oluşturduğumuz scripti kod editörümüzden açıyoruz.



Scriptimizi kod editörümüzde açtığımıza göre artık kodumuzu yazabiliriz. Atışlarımızın hareketi için öncelikle bir fonksiyon oluşturuyoruz. Ardından bu fonksiyonumuzu her framede bir çalıştırmak istediğimiz için **Update** fonksiyonu içerisinde kullanacağız.

```
void Update()
{
    CalculateMovement();
}
```

CalculateMovement fonksiyonumuzda laser nesnemizi prefabimizden oluşturduğumuzda her saniyede bir birim yukarı hareket etmesini istiyoruz. Bunu sağlamak için öncelikle Vector3 direction = new Vector3(0, 1, 0); kod parçasını kullanarak bir birim yukarıyı belirten bir vektör nesnesi oluşturuyoruz. Ardından fonksiyonlarımızın dışında, classımızın içinde; oluşturulan nesnemizin hızını belirtmek amacıyla bir speed değişkeni tanımlıyoruz.

```
public int speed = 20;
```

Sonrasında transform.Translate fonksiyonunu kullanarak nesnemizin hareket etmesini sağlıyoruz. Saniyede bir birim hareketini ve istediğimiz hız ve yönde hareketini sağlamak için transform.Translate fonksiyonuna parametremizi (direction * Time.deltaTime * speed) şeklinde giriyoruz. Bu şekilde **nesnemizin oluşturulduktan sonra her saniye bir birim yukarı hareket etmesini** sağlamış oluyoruz. Nesnemiz ekrandan kaybolunca silinmesini sağlamak için ise kullandığımız aspecte (oyunun ekran boyutuna) göre silineceği sınırı belirliyoruz. Sınırı belirledikten sonra if koşulu ile oluşturulan nesnemizin y pozisyonunun belirlediğimiz sınırı geçip geçmediğini kontrol ediyoruz. **Bu sınırı geçmesi durumunda** ise Destroy(this.gameObject); kod parçasını kullanarak **nesnemizin projeden yok edilmesini sağlıyoruz**. Böylece teorik olarak sonsuz sayıda nesne üretmekten ve bilgisayarı yormaktan kaçınmış oluyoruz.

```
void CalculateMovement(){
    Vector3 direction = new Vector3(0, 1, 0);
    transform.Translate(direction * Time.deltaTime * speed);
    if (transform.position.y >= 8){
        Destroy(this.gameObject);
    }
}
```

Scripti tamamladıktan sonra ise bu **scriptimizi prefabimize entegre etmemiz gerekiyor**. Bu entegrasyonu yapmamamız halinde projemiz istenilen şekilde çalışmayacaktır. Bunun için ise sürükle-bırak yöntemini kullanacağız. **Assets > Scripts** klasörümüz altında prefabimiz için oluşturduğumuz scripti tutuyoruz ve sürükleyerek **Assets > Prefabs** klasörümüz altındaki **Laser** için oluşturduğumuz klasörün üzerinde bırakıyoruz. Projemizi kaydediyoruz ve scriptimizin prefabimize entegrasyonu tamamlanmış oluyor.

