

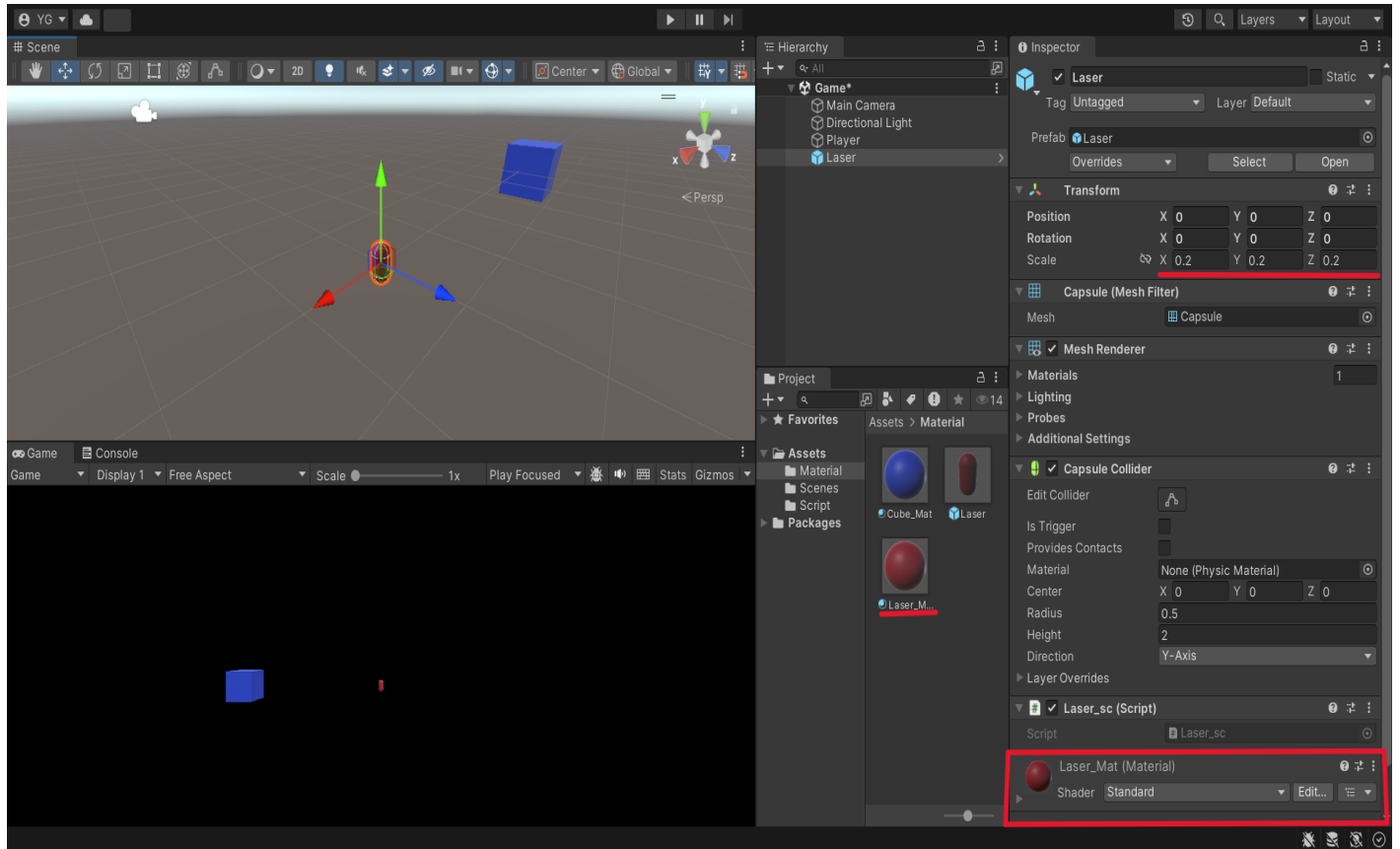
Yusuf Güney 22360859041

Oyun Programlama

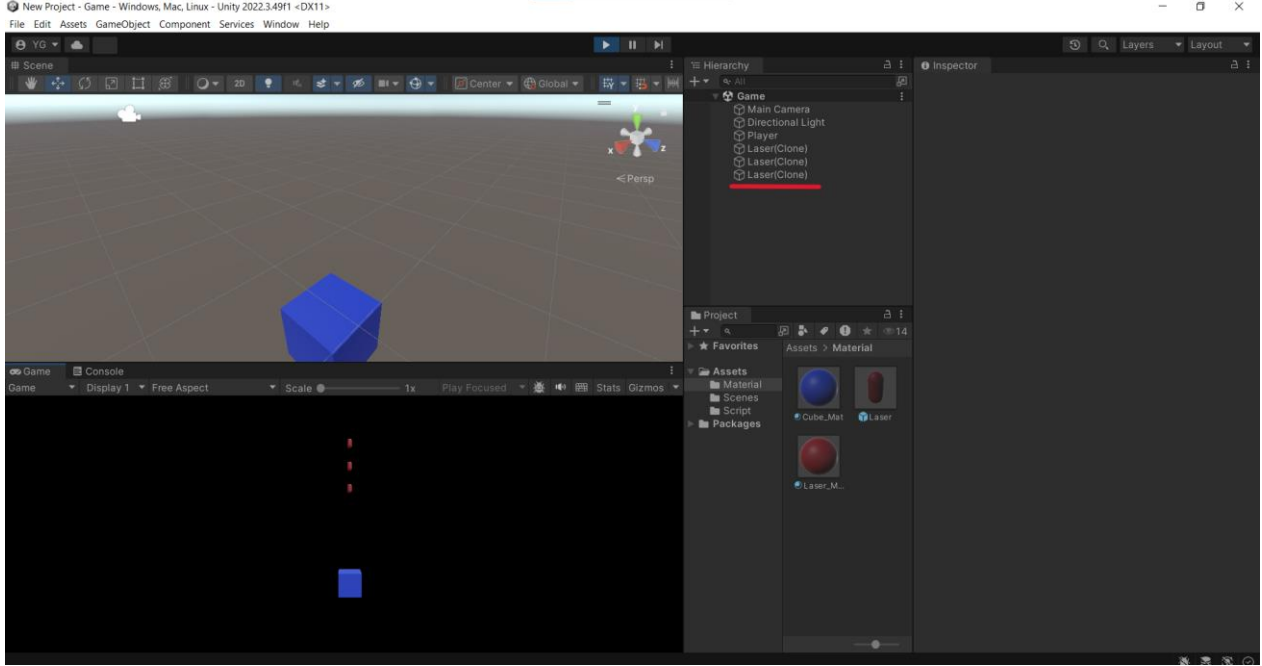
Hafta 2 Rapor

1-Laser Prefab'i oluşturma (capsul'e ekle, ölçeklendir, materyal ekle)

Laser Prefab'i oluşturmak için öncelikle Game kısmından bir kapsül oluşturuyoruz daha sonra bu kapsülü Project kısmına ekliyoruz inspector kısmından kapsülün laser gibi gözükmesi için scale'ini değiştiriyoruz. Son olarak Laser_Mat materyalini laser'e ekleyerek Laser Prefab'i oluşturuyoruz.



2-Space tuşuna basıldığında Laser prefab'den laser oluştur (her saniye bir birim yukarı hareket etsin, ekrandan kaybolduğunda oyun nesnesi yok edilsin)



Burada lazerin küpten çıkıp hareket ettiğini ve sahne dışına çıktığında yok olduğunu görüyoruz bunun için lazerin küpten çıktığını gösteren **shoting()** fonksiyonu ve lazerin hareketini kontrol eden **CalculateLaserMovement()** fonksiyonlarına bakalım.

```
void shooting(){
    if(Input.GetKeyDown(KeyCode.Space) /*&& Time.time > nextFire*/){
        Instantiate(laserPrefab, transform.position + new Vector3(0,0.8f,0),Quaternion.identity);
        //nextFire = Time.time + fireRate;
    }
}
```

Shoting() fonksiyonu lazerin fırlatılma(ateş edilme) işlemini gerçekleştirir.

if(Input.GetKeyDown(KeyCode.Space)) Bu kısım, kullanıcının klavyede Space tuşuna basıp basmadığını kontrol eder. Eğer Space tuşuna basılırsa, if bloğundaki işlemler gerçekleştirilir.

Instantiate() fonksiyonu önceden tanımlanmış bir objeyi sahneye yerleştirir.

İlk parametre olan laserPrefab bizim lazer objemizdir.

İkinci parametre olan **transform.position + new Vector3(0, 0, 8f, 0)** lazerin hangi konumda oluşturulacağını belirtir. Burada lazerin küpten 0.8 birim yukarıda oluşturulması gerektiği söylenmek istenmiştir.

Üçüncü parametre olan **Quaternion.identity** lazerin varsayılan rötasyonla oluşturulacağını belirtir.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  0 references
6  public class Laser_sc : MonoBehaviour
7  {
8      1 reference
9      float speed = 8.0f;
10
11      // Update is called once per frame
12      0 references
13      void Update()
14      {
15          CalculateLazerMovement();
16      }
17
18      1 reference
19      void CalculateLazerMovement(){
20          transform.Translate(Vector3.up * Time.deltaTime * speed);
21          if(transform.position.y > 7.0f) {
22              Destroy(this.gameObject);
23          }
24      }
25  }
```

CalculateLazerMovement() fonksiyonu bir lazerin yukarı doğru hareket etmesini ve belirli bir mesafeden sonra yok olmasını sağlar.

Speed = 8.0f lazerin atış hızını belirtir.

transform.Translate(Vector3.up * Time.deltaTime * speed) bu fonksiyon lazerin konumunu günceller.

Vector3.up yukarı yönde hareketi temsil eder.

Time.deltaTime, her bir kare arasındaki zamanı alır, bu sayede hareket kareye bağımlı olmaz ve daha akıcı hale gelir.

Kısacası bu fonksiyon lazeri yukarı doğru saniyede 8 birim hareket etmesini sağlar.

if(transform.position.y > 7.0f) { Destroy(this.gameObject); } eğer lazerin pozisyonu y ekseninde 7.0 den büyükse if bloğunun içine girer ve Destroy komutu çalışır ver lazer objesi sahneden yok edilir. Buradaki 7.0 değeri sahnenin dışını ifade eder.

3- Seri atışlar arasında bekleme süresi ekleyin

Seri atışlar için bekleme süresi eklemek için **fireRate** ve **nextFire** değişkenlerin tanımlıyoruz

FireRate: İki atış arasındaki bekleme süresi

NextFire: Bir sonraki atışın yapılacağı zaman

Bu kodu önceden yazdığımız **shoting()** fonksiyonunun içine ekliyoruz.

```
1 reference
float fireRate = 0.5f;

2 references
float nextFire = 0f;
```

```
void shoting(){
    if(Input.GetKeyDown(KeyCode.Space) && Time.time > nextFire){
        Instantiate(laserPrefab, transform.position + new Vector3(0,0.8f,0),Quaternion.identity);
        nextFire = Time.time + fireRate;
    }
}
```

Burada eklediğimiz kodlar atışlar arasında bir bekleme süresi koyarak lazerin sürekli ateşlenmesini engeller. fireRate süresi geçmeden yeni bir atış yapılmasına izin verilmez, bu sayede "seri atış" engellenmiş olur.

4-Atışla ilgili kod için bir fonksiyon oluşturun

Lazerin atışı için tanımlanan **shoting()** fonksiyonu

```
shoting();
}
1 reference
void shoting(){
    if(Input.GetKeyDown(KeyCode.Space) && Time.time > nextFire){
        Instantiate(laserPrefab, transform.position + new Vector3(0,0.8f,0),Quaternion.identity);
        nextFire = Time.time + fireRate;
    }
}
```

Küpün hareketi için tanımlanan **CalculateMovement()** fonksiyonu

Burada küpün ekran dışına çıktığında diğer taraftan gelmesini sağlamak için if blokları güncellenmiştir ve cismin uzay gemisi gibi gözükmesi için belirli bir konumdan sonra hareketi kısıtlanmıştır.

```
1 reference
void CalculateMovement(){

    float horizontal = Input.GetAxis("Horizontal");
    float vertical = Input.GetAxis("Vertical");
    Vector3 direction = new Vector3(horizontal, vertical,0);
    transform.Translate(direction*Time.deltaTime*speed);

    if(transform.position.y >= 0){// y nin maximum değeri
        transform.position = new Vector3(transform.position.x,0,0);
    }
    else if(transform.position.y <=-3.9f){// y nin minimum değeri
        transform.position = new Vector3(transform.position.x,-3.9f,0);
    }
    if(transform.position.x >=15.2f){// x in maximum değeri
        transform.position = new Vector3(-15.2f,transform.position.y,0);
    }
    else if(transform.position.x <=-15.2f){// x nin minimum değeri
        transform.position = new Vector3(15.2f,transform.position.y,0);
    }
}
```

Lazerin hareketi için tanımlanan **CalculateLazerMovement()** fonksiyonu

```
1 reference
void CalculateLazerMovement(){
    transform.Translate(Vector3.up * Time.deltaTime * speed);
    if(transform.position.y > 7.0f) {
        Destroy(this.gameObject);
    }
}
```

Kodların tamamına Github sayfamdan da erişebilirsiniz

Github Linki:

