

HAFTA 2 – RAPOR

DERS : OYUN PROGRAMLAMA

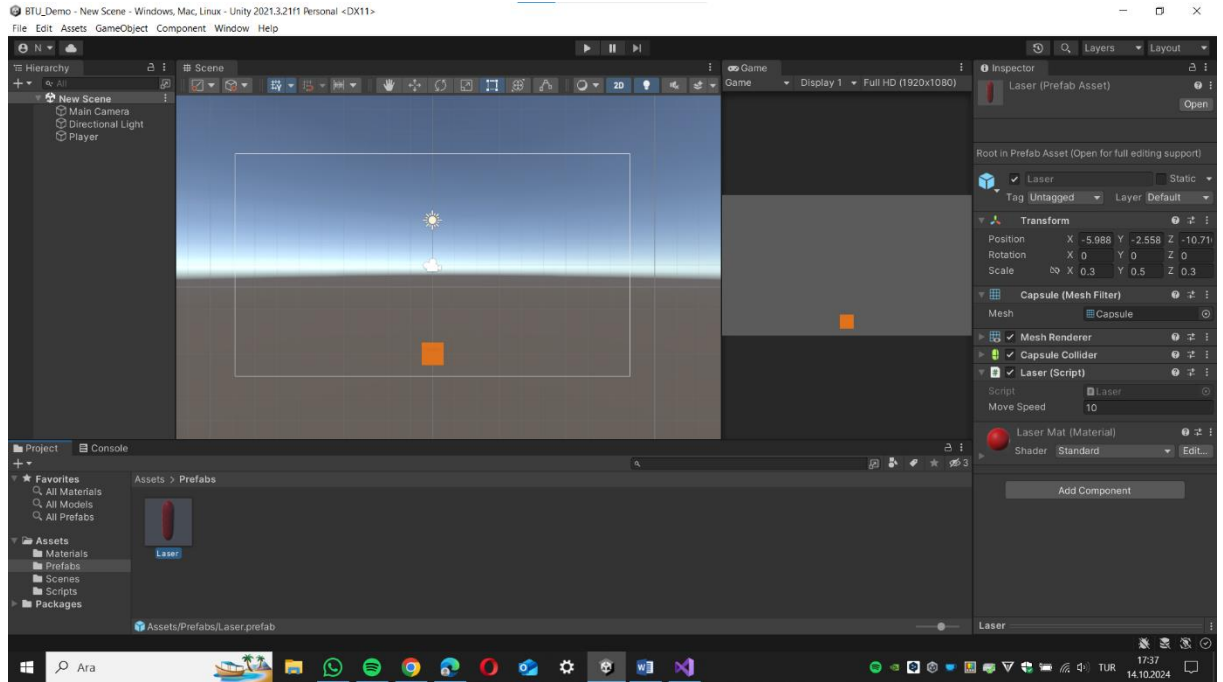
AD SOYAD : LÜTFÜ BEDEL

ÖĞRENCİ NO : 21360859030

GITHUB : https://github.com/lutfubedel/BTU_Oyun_Programlama_Kodlari/tree/main/Hafta_2

BÖLÜM 1 : LASER PREFABI OLUŞTURMA

Öncelikle sahnemize bir kapsül objesi ekliyoruz. Ardından bu kapsül objesinin Transform componentinin içerisindeki Scale değerlerini x : 0.3 y : 0.5 z : 0.3 olarak ayarlarız son olarak daha önce oluşturulan LaserMat adındaki materyal objeye ekleriz. Oluşturulan bu obje Laser ismi ile Prefabs isimli klasöre atılır.



BÖLÜM 2 : Laser Oluşturma Scripti

Öncelikle oluşturulmak istenen obje GameObject olarak “laser” ismi ile tanımlanır ardından playerın tekrar ateş edip edemeyeceğini kontrol etmek için bool türünde “canFire” isminde bir değişken tanımlanır.

“LaserFire” adında void döndüren bir fonksiyon oluşturulur ve tüm bu ateş etme kodu içerisine yazılır.

Öncelikle oyuncunun space tuşuna bir kez bastığını kontrol etmek için aşağıdaki fonksiyon kullanılır.

```
Input.GetKeyDown(KeyCode.Space)
```

Eğer Input fonksiyonu true değeri döndürürse ve aynı zamanda canFire değeri de true ise daha önce tanımlanmış olan laser objesi aşağıdaki Instantiate fonksiyonu ile sahnede oluşturulur.

```
Instantiate(laser, transform.position, Quaternion.identity);
```

Bu fonksiyon Update fonksiyonu içerisinde çağırıldığından ötürü herhangi bir önlem alınmaması durumunda bir kez space tuşuna basıldığı anda birden fazla laser objesi oluşturacaktır. Bunun önüne geçmek için ilk laser objesi oluşturulduğu anda canFire değişkeninin değeri false olarak değiştirilir.

Oluşan bu son durumda ise oyuncu bir kez space tuşuna bastığında bir laser objesi oluşturulur ardından canFire false değerini alır. Oyuncu tekrar space tuşuna basması durumunda ise canFire false olduğu için hiçbir şey olmaz. Bu sebepten ötürü bi süre sonra canFire değişkenini tekrar true olarak değiştirmemiz gerekiyor.

Bunun için öncelikle geriye bir şey döndürmeyen ChangeCanFire isminde bir fonksiyon oluşturulur ve içerisinde canFire değişkeninin değeri true olarak güncellenir.

Oyuncunun laserları aynı anda veya ard arda oluşturmaması için oluşturulan ChangeCanFire fonksiyonu Invoke fonksiyonu kullanılarak çağırılır. Bu şekilde canFire değişkeninin değeri false olduktan kaç saniye sonra tekrar true olacağını yani tekrar ne zaman oyuncunun laser objesi oluşturabileceği kontrol edilir.

Invoke fonksiyonu içerisine önce çağırılmak istenen fonksiyonun ismi ardından çağırılması için beklenecek süre parametre olarak girilir

```
Invoke(nameof(ChangeCanFire), 2f);
```

Unity Betiği (1 varlık başvurusu) | 0 başvuru

```
public class Fire : MonoBehaviour
```

```
{
```

```
    public GameObject laser;
```

```
    public bool canFire;
```

Unity İletisi | 0 başvuru

```
private void Update()
```

```
{
```

```
    LaserFire();
```

```
}
```

1 başvuru

```
private void LaserFire()
```

```
{
```

```
    if (Input.GetKeyDown(KeyCode.Space) && canFire)
```

```
    {
```

```
        Instantiate(laser, transform.position, Quaternion.identity);
```

```
        canFire = false;
```

```
        Invoke(nameof(ChangeCanFire), 2f);
```

```
    }
```

```
}
```

1 başvuru

```
private void ChangeCanFire()
```

```
{
```

```
    canFire = true;
```

```
}
```

```
}
```

BÖLÜM 3 : Laser Objesinin Hareketi ve Yok Edilmesi

Oluşturulan laser objesinin sürekli olarak yukarıya doğru hareket etmesi için laser objesinin Transform componenti içerisindeki position değeri değiştirilir.

```
transform.position += new Vector3(0, 1*moveSpeed*Time.deltaTime,0);
```

Ardından objenin sahne dışarısına çıktıktan sonra yok olması için Destroy fonksiyonu kullanılır ve objenin oluştuktan 3 saniye sonra sahne dışında olacağından ötürü Destroy fonksiyonuna 2. parametre olarak 3f değeri girilebilir.

```
Destroy(this.gameObject, 3f);
```

