

HAFTA 6 – RAPOR

DERS : OYUN PROGRAMLAMA

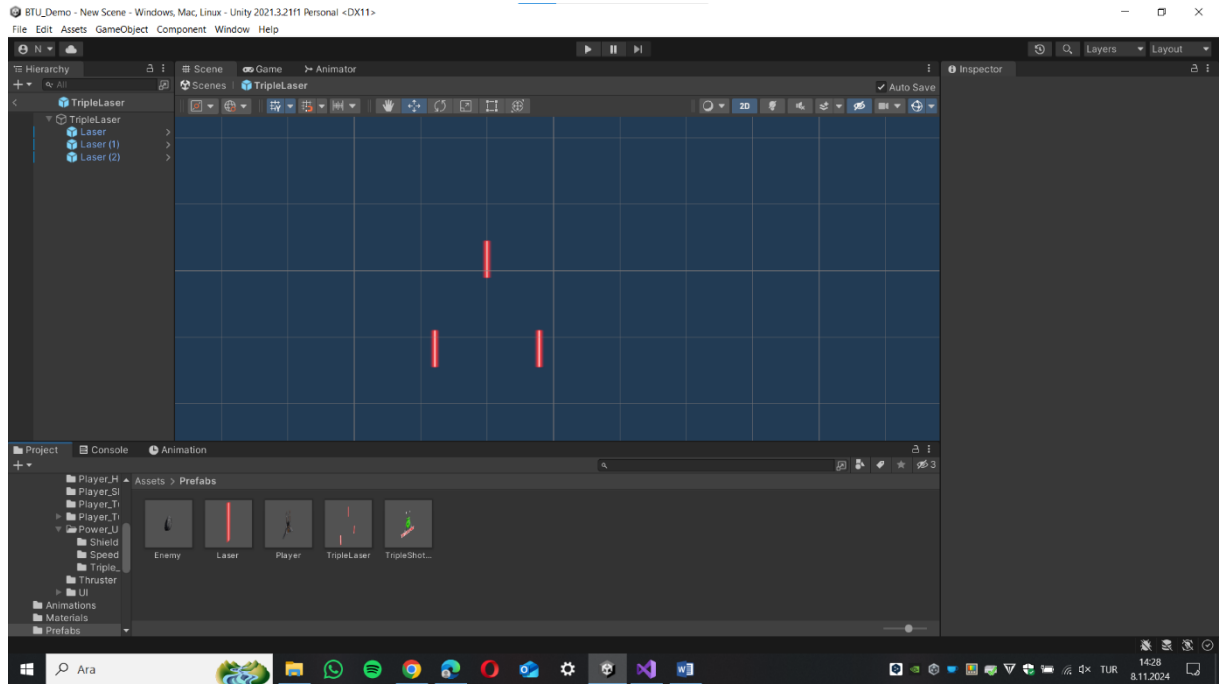
AD SOYAD : LÜTFÜ BEDEL

ÖĞRENCİ NO : 21360859030

GITHUB : https://github.com/lutfubedel/BTU_Oyun_Programlama_Kodlari/tree/main/Hafta_6

BÖLÜM 1 : Üçlü Laserin Oluşturulması

Daha önceden oluşturulmuş olan laser prefablarını kullanarak playerın üzerindeki atış noktalarına birer laser objesi bırakılır bu şekilde laserların konumları belirlenmiş olur. Laser objelerin konumları belirlendikten sonra boş bir gameobject oluşturulur ve “TripleLaser” olarak adlandırılır. Oluşturulan 3 laser objesi child obje olarak TripleLaser objesine eklenir. Ardından oluşturulan TripleLaser objesi Prefabs klasörüne eklenir.



BÖLÜM 2 : Üçlü Bonusun Oluşturulması

Assets klasörü içerisinde bulunan üçlü bonus spriteni kullanarak bir bonus objesi oluşturulur ve bu obje “TripleShotBonus” olarak adlandırılır. Üçlü atışın gerçekleştirilebilmesi için Playerın TripleShotBonus objesi ile temas etmesi gerekiyor. Çarpışmaların algılanabilmesi için öncelikle TripleShotBonu objesine “CircleCollider2D” , “Rigidbody2D” ve animasyonların kullanılabilmesi için “Animator” componenti eklenir ve çarpışmaları kontrol eden “TripleShotBonus” adında bir script eklenir.

BÖLÜM 3 : FireLaser Scriptinin Güncellenmesi

Öncelikle script içerisinde oluşturulmak istenen “tripleLaser” objesi ve oluşturulup oluşturulmayacağını kontrol eden bool türünde “isTripleLaserActive” adında bir değişken tanımlanır.

```
public GameObject tripleLaser;  
public bool isTripleLaserActive;
```

If else yapısı kullanılarak isTripleLaserActive true ise tripleLaser, false ise laser objesini oluşturacak şekilde düzenlenir.

Bir kez isTripleLaserActive true olduktan sonra sürekli olarak tripleLaser oluşturmasını önlemek için “ChangeTripleLaserActive” isminde bir fonksiyon oluşturulur ve içerisinde isTripleLaserActive false yapılır. Bu fonksiyon Invoke fonksiyonu ile eğer isTripleLaserActive true ise 10sn sonra isTripleLaserActive değişkeni false yapılır.

```
Unity Betiği (1 varlık başvurusu) | 1 başvuru  
public class Fire : MonoBehaviour  
{  
    public GameObject laser;  
    public GameObject tripleLaser;  
  
    public bool canFire;  
    public bool isTripleLaserActive;  
  
    Unity İletisi | 0 başvuru  
    private void Update()  
    {  
        1 başvuru  
        private void LaserFire()  
        {  
            if (Input.GetKeyDown(KeyCode.Space) && canFire)  
            {  
                if(!isTripleLaserActive)  
                {  
                    Instantiate(laser, transform.position, Quaternion.identity);  
                }  
                else  
                {  
                    Instantiate(tripleLaser, transform.position, Quaternion.identity);  
                }  
  
                canFire = false;  
                Invoke(nameof(ChangeCanFire), 0.5f);  
  
                if (isTripleLaserActive)  
                    Invoke(nameof(ChangeTripleLaserActive), 10f);  
            }  
        }  
  
        1 başvuru  
        private void ChangeCanFire()  
        {  
            1 başvuru  
            private void ChangeTripleLaserActive()  
            {  
                isTripleLaserActive = false;  
            }  
        }  
    }  
}
```

BÖLÜM 4 : Üçlü Bonusun Scripti

TripleBonus objesinin sürekli olarak y ekseninde aşağıya doğru hareket etmesini istiyoruz. Bunun için objenin içerisine yeni bir script eklenir. Bu scriptde Update fonksiyonu içerisinde transform.Translate() fonksiyonu kullanılarak objenin y ekseninde sürekli hareketi sağlanır.

```
transform.Translate(new Vector3(0, -1 * moveSpeed * Time.deltaTime, 0));
```

TripleBonus objesinin sahnenin dışına çıkması durumunda objenin yok edilmesi gerekir. Bunun için objenin pozisyonun y değeri -4.5f den küçük ise obje sahne dışına çıkmış demektir bu durumda objemiz kendisini yok eder.

Eğer tripleBonus objesi "Player" tagına sahip bir objeye çarpar ise player objesi içerisinde bulunan "Fire" scriptindeki "isTripleLaserActive" değişkeni true yapılır ardından obje kendini yok eder.

```
Unity Betiği (1 varlık başvurusu) | 0 başvuru
public class TripleShotBonus : MonoBehaviour
{
    public float moveSpeed;

    Unity İletisi | 0 başvuru
    private void Update()
    {
        transform.Translate(new Vector3(0, -1 * moveSpeed * Time.deltaTime, 0));

        if (transform.position.y < -4.5f)
        {
            Destroy(this.gameObject);
        }
    }

    Unity İletisi | 0 başvuru
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.gameObject.CompareTag("Player"))
        {
            collision.gameObject.GetComponent<Fire>().isTripleLaserActive = true;
            Destroy(this.gameObject);
        }
    }
}
```

BÖLÜM 5 : SpawnManager ile Üçlü Bonusun Oluşturulması

Öncelikle GameObject türünde oluşturulacak olan üçlü bonusu tutan “tripleShotBonus” ve enemy objesini oluşturan coroutine ni kontrol eden bool türünde “bonusSpawnerWorking” isminde bir değişken tanımlanır.

Ardında zaman ile ilgili işlemleri yapabilmek için “BonusSpawner” adında ve IEnumerator türünde bir coroutine oluşturulur. Üçlü Bonusun oluşturulma işlemi tüm oyun boyunca devam edeceği için öncelikle bir while döngüsü oluşturulur ve “bonusSpawnerWorking” değişkeninin değeri true olduğu sürece enemy objesinin oluşturulması sağlanır

Ardından Instantiate fonksiyonu ile enemy prefabı ekranın üst kısmında x konumu rastgele olacak şekilde ve parent objesi “parentObj” olacak şekilde oluşturulur.

```
Instantiate(tripleShotBonus, new Vector3(Random.Range(-8.5f, 8.5f), 6.5f, 0), Quaternion.identity);
```

Son olarak aynı anda birden fazla üçlü bonus objesi oluşturmalarını önlemek adına “WaitForSeconds” ile bir sonraki üçlü bonus objesini oluşturulana kadar 10 saniye beklemesi sağlanır.

```
yield return new WaitForSeconds(10);
```

```
1 başvuru
IEnumerator BonusSpawner()
{
    while (bonusSpawnerWorking)
    {
        Instantiate(tripleShotBonus, new Vector3(Random.Range(-8.5f, 8.5f), 6.5f, 0), Quaternion.identity);
        yield return new WaitForSeconds(10);
    }
}
```

BÖLÜM 6 : Üçlü Bonusun Detaylarının Eklenmesi

Bonusun daha güzel görünmesi için animasyon eklenir. Animasyonları kontrol eden bir Animator ve bir animasyon oluşturulur. Spriteler ile animasyon oluşturmak için tüm spriteler seçilir ve “animasyon” panelinde recording aktif iken tüm spriteler animasyon ekranının sağ tarafına atılır. Ardından oluşturulan bu animasyon tutup “Animator” panelinde daha önce oluşturduğumuz “TripleAnimator” içerisine atılır. Bu animasyonun sürekli oynamasını istediğimiz için animator ekranı içerisinde “TripleAnim” objesine çift tıklanır ve “Loop Time” özelliği aktif hale getirilir.

