HAFTA 3 – RAPOR

DERS : OYUN PROGRAMLAMA

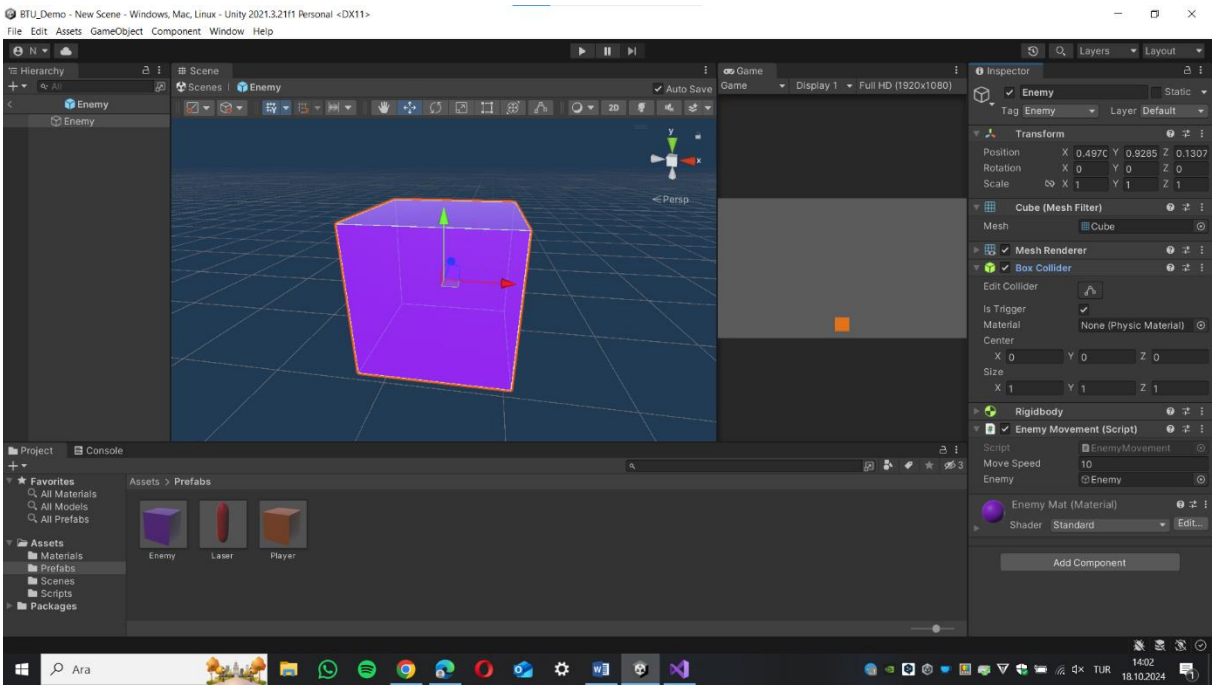
AD SOYAD : LÜTFÜ BEDEL

ÖGRENCİ NO : 21360859030

GITHUB : https://github.com/lutfubedel/BTU_Oyun_Programlama_Kodlari/tree/main/Hafta_3

BÖLÜM 1 : ENEMY PREFAB OLUŞTURMA

Öncelikle enemy objesi olarak ekrana yeni bir küp objesi ekledikten sonra içerisine enemyMat ismindeki materyal eklenir ve “Enemy” tagı eklenir. Ardından çarpışma ve diğer fiziksel işlemleri gerçekleştirmek için içerisine “Rigidbody” ve “Box Collider” componentleri eklenir. Oyunumuzda yer çekimine ihtiyaç duymadığımız için öncelikle Rigidbody componenti içerisindeki Use Gravity seçeneği kapatılır ve çarpışmaların gerçekleştirilebilmesi için Box Collider içerisindeki isTrigger seçeneği tiklenir. Oluşturulan enemyi objesi prefab olarak kullanılabilmesi için Prefabs klasörüne bırakılır.



BÖLÜM 2 : ENEMY HAREKETİNİ SAĞLAYAN KODUN YAZILMASI

Enemy objemizin 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 eksininde sürekli hareketi sağlanır.

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

Enemy objesinin sahnenin dışına çıkması durumunda yeni bir obje oluşmasını ve eski objenin yok edilmesi gerekir. Bunun için objenin pozisyonun y değeri -4.5f den küçük ise obje sahne dışına çıkmış demektir. Bu durumda ekranın üst kısmıdan x ekseninde rastgele olacak şekilde yeni bir enemy objesi oluşturur ardından eski enemy objemiz kendisini yok eder.

```
public float moveSpeed;
public GameObject enemy;

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

    if(transform.position.y < -4.5f)
    {
        Instantiate(enemy, new Vector3(Random.Range(-8.5f, 8.5f), 6.5f, 0), Quaternion.identity);
        Destroy(this.gameObject);
    }
}
```

BÖLÜM 3 : OnTriggerEnter FONKSİYONU VE ÇARPIŞMALARIN TESPİTİ

Çarpışmaların algılanması ve çarpışma anında ne olucağını karar verebilmemiz için enemynin içerisindeki scriptte onTriggerEnter fonksiyonu tanımlanır. Öncelikle çarpan objenin tagına göre yapılacak işlemler değişkenlik göstermeli bu sebepten if yapısı ile çarpan objenin tagı kontrol edilmeli. Eğer çarpan objenin tagı “Laser” a eşit ise önce yeni bir enemy objesi oluşturulur ardından çarpışmaya giren laser objesi ve enemy objesi yok edilir.

BÖLÜM 4 : OYUNCU HASAR ALMA İŞLEMLERİ

Eğer çarpıştığı objenin tagı “Player” ise önce player içerisindeki PlayerMovement scriptinde bulunan life değişkeni bir azaltılır. Ardından yeni bir enemy objesi oluşturulduktan sonra çarpışan enemy objesi kendisini yok eder.

```
Unity İletisi | 0 başvuru
private void OnTriggerEnter(Collider other)
{
    if(other.CompareTag("Player"))
    {
        other.GetComponent<PlayerMovement>().life -= 1;
        Instantiate(enemy, new Vector3(Random.Range(-8.5f,8.5f),6.5f,0), Quaternion.identity);
        Destroy(this.gameObject);
    }
    if(other.CompareTag("Laser"))
    {
        Instantiate(enemy, new Vector3(Random.Range(-8.5f, 8.5f), 6.5f, 0), Quaternion.identity);
        Destroy(other.gameObject);
        Destroy(this.gameObject);
    }
}
```