

Ad: Yusuf

Soy ad: Güney

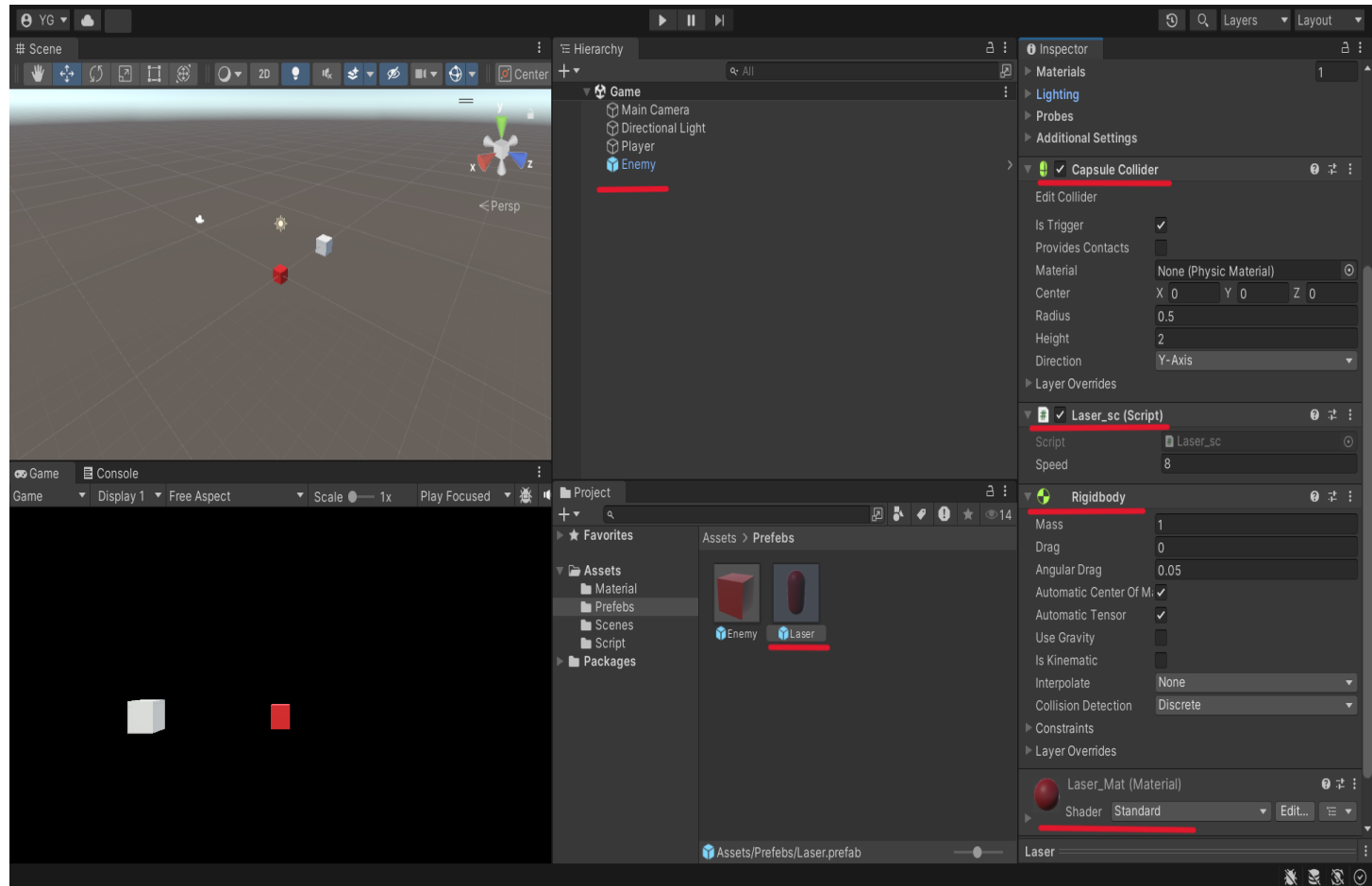
Numara: 22360859041

Ders: Oyun Programlama

Hafta 3 Rapor

1- Enemy prefab oluşturma

Öncelikle Enemy prefab'ı oluşturmak için Hierarchy sekmesinden bir GameObject oluşturdum ve buna "Enemy" ismini verdim. Daha sonra bu Enemy'e "Enemy_Mat" materyali ve "Enemy_sc" scriptini ekledim. Düşmanın fiziksel etkileşimlerini sağlamak için Collider ve Rigidbody bileşenlerini ekledim. (Collider da Is Trigger açık Rigidbody de ise Use Gravity seçeneklerini kapatıyoruz) Son olarak bu Enemy nesnesini Project sekmesinden Assets klasörünün alt dizini olan Prefabs klasörüne taşıdım. (Bu işlemi yaparken Enemy nesnesini override etmeyi de unutmamak gerekli)



2-Enemy hareketini sağlayan kodun yazılması

Enemy nesnesinin hareketi için **CalculateEnemyMovement()** fonksiyonunu aşağıdaki gibi yazabiliriz.

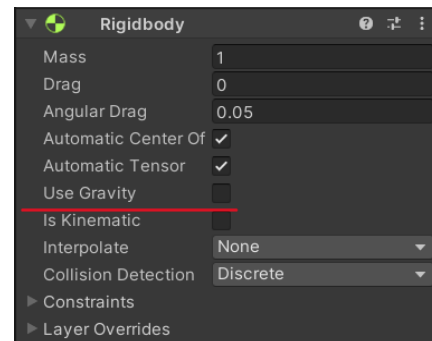
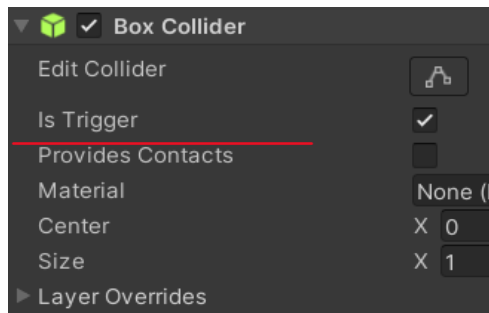
```
1 reference
void CalculateEnemyMovement(){
    transform.Translate(Vector3.down * speed * Time.deltaTime);
    if(transform.position.y < -5.4){
        float randomX = Random.Range(-9.4f, 9.4f);
        transform.position = new Vector3(randomX, 7.5f, 0);
    }
}
```

Tranform.Translate(Vector3.down * speed * Time.deltaTime) komutu nesnenin aşağı yönde hareketini sağlar. if koşulu cisim y ekseninde -5.4 konumunda yani cisim sahnenin dışına çıktığında tekrar oluşmasını sağlamak için kullanılır. Burada **Random.Range(-9.f, 9.4f)** komutu cismin x ekseninde rastgele bir konumda oluşmasını sağlar.

transform.position = new Vector3(randomX, 7.5f ,0) bu komut ise cismin x ekseninde söylenilen rastgele konumda y ekseninde sahnenin tepe noktasında z ekseninde ise 0 noktasında oluşmasını sağlar.

3-Rigidbody ve Collision component'lerinin eklenmesi

Rigidbody ve Collsion componentlerini eklemek için Enemy cismini seçip Inspector sekmesinde gözükmemektedir. Rigidbody eklemek için Add Compenent kısmından Rigidbody'yi seçerek ulaşabiliriz. Burada Rigidbody'i Enemy için seçiyoruz ve seçim yaptıktan sonra Use Gravity seçeneğini devre dışı bırakıyoruz. Bunun sebebi cismin hızını kendimiz ayarlamak istediğimiz içindir. Colliderı ise Player, Enemy ve Laser nesnelerinin hepsinde Is Trigger ekleyerek seçiyoruz.



4- OnTriggerEnter fonksiyonu ve çarpışmaların tespiti ile gerekli aksiyonların alınması

Enemy için **OnTriggerEnter()** fonksiyonu aşağıda görülmektedir.

```
0 references
void OnTriggerEnter(Collider other){
    if(other.tag == "Player"){
        //Canını bi azalt
        Destroy(this.gameObject);
    }
    else if(other.tag == "Laser"){
        Destroy(other.gameObject);
        Destroy(this.gameObject);
    }
}
```

OnTriggerEnter Metodu: Unity'de, bir "trigger collider" ile başka bir nesne çarpıştığında tetiklenen bir fonksiyondur. Bu fonksiyon, çarpışma gerçekleştiğinde neler yapılacağını tanımlar.

Collider other Parametresi: Çarpışan diğer nesneye erişim sağlar. "other", diğer nesneyi temsil eder.

If bloğunda eğer çarpışma nesnesi Player ise çarpışma esnasında enemy nesnesi yok edilir ve buraya daha sonra eklenecek olan player nesnesinin canı da adazlacaktır.

Else if bloğunun içinde eğer çarpışılan nesne Laser ise önce laser yok edilir daha sonra enemy nesnesi yok edilir.

5-Oyuncu hasar alma işlemlerinin yapılması

Oyuncunun hasar alma işlemini gerçekleştirmek için öncelikle Player_sc scriptinin içerisinde Playera bir lives değeri tanımlamalıyız.

```
int lives = 3;
```

Daha sonra Damage fonksiyonunda enemy nesnesiyle çarpışması esnasında canını bir azaltacak Damage fonksiyonunu aşağıdaki gibi yazabiliriz.

```
0 references  
public void Damage(){  
    lives--;  
  
    if(lives < 1){  
        Destroy(this.gameObject);  
    }  
}
```

Son olarak OnTriggerEnter fonksiyonunda daha önceden boş bıraktığımız Canını bir azalt satırına aşağıdaki kod satırını çalıştırarak Playerın hasar azalmasını gerçekleştirebiliriz.

```
Player_sc player_sc = other.transform.GetComponent<Player_sc>();  
player_sc.Damage();
```

Bu satır, çarpışan nesnedeki Player_sc adlı script bileşenine erişmeyi sağlar.

other.transform.GetComponent<Player_sc>() ifadesiyle, çarpışan nesnenin Player_sc bileşeni olup olmadığını kontrol eder ve varsa bu bileşeni player_sc değişkenine atar, böylece oyuncunun canını azaltmak gibi işlemler bu bileşen üzerinden yapılabilir.

Kodların tamamına Github sayfası üzerinden de ulaşabilirsiniz.

Github Linki: