

HAFTA 8 – RAPOR

DERS : OYUN PROGRAMLAMA

AD SOYAD : LÜTFÜ BEDEL

ÖĞRENCİ NO : 21360859030

GITHUB : https://github.com/lutfubedel/BTU_Oyun_Programlama_Kodlari/tree/main/Hafta_8

BÖLÜM 1 : Puan Sistemi

Öncelikle oyuncunun score değerini tutabilmek için “UIManager” scripti içerisinde int türünde “score” değişkeni tanımlanır. Ardından çarpışmaları kontrol ettiğimiz “EnemyMovement” scriptinde “OntriggerEnter2D” fonksiyonu içerisinde eğer enemy objesinin çarpıştığı obje bir laser objesi ise objeler yok olmadan önce ekranda “UIManager” tagına sahip obje bulunup içerisinde tanımlanmış olduğumuz score değişkeni arttırılır.

```
© Unity İletisi | 0 başvuru
private void OnTriggerEnter2D(Collider2D other)
{
    if(other.CompareTag("Player"))
    {
        other.GetComponent<PlayerMovement>().life -= 1;
        Destroy(this.gameObject);
    }
    if(other.CompareTag("Laser"))
    {
        UIManager uiManager = GameObject.FindWithTag("UIManager").GetComponent<UIManager>();
        uiManager.score += Random.Range(50, 100);

        Destroy(other.gameObject);
        Destroy(this.gameObject);
    }
}
```

BÖLÜM 2 : Puanın ve Canın Ekranda Görünmesi

UIManager içerisinde “TMP_Text” türünde scoreText isminde bir TextMeshPro objesi oluşturulur ve scoreText objesinin text değerine score verilir.

Canımızı göstermek için öncelikle canvas içerisine sağ üst köşeye bir Image eklenir ve ismi lifelImage olarak verilir. LifelImage için kullanılacak olan 4 farklı sprite tutabilmek için Sprite türünde bir dizi oluşturulur. Bir for döngüsü yardımıyla sprite dizisi içerisinde oyuncunun o andaki canına denk gelen sprite lifelImage in sprite’na verilir.

```

private void Update()
{
    scoreText.text = "Score : " + score.ToString();

    for (int i = 0; i < lifeSprites.Length; i++)
    {
        if (i == player.life)
        {
            lifeImage.sprite = lifeSprites[i];
        }
    }
}

```

BÖLÜM 3 : GameOver Ekranı Oluşturma

Öncelikle canvas içerisinde bir gameOver sahnesi oluşturulur ve oyunun başlangıcında setActive değeri false olarak verilir. Eğer oyuncunun canı yani life değeri 0 olursa o zaman gameOver objesi tekrar aktif edilir. Ekrandaki Game Over yazısını flicker etkisi oluşturmak için IEnumerator kullanılır. Her 0.5 saniyede bir Game Over yazısı tutan değişkenin text değeri sıfırlanır ve 0.5 saniye sonra tekrar "Game Over" değerini alır.

Oyuncunun oyun bittikten sonra tekrar başlatabilmesi için oyun bittiyse ve oyuncu R tuşuna basarsa kontrolleri yapıldıktan sonra "SceneManager" kullanılarak oyun sahnesinin tekrar yüklenmesi sağlanır.

```

if(player.isDead)
{
    gameOverPanel.SetActive(true);
    StartCoroutine(GameOverFlickerRoutine());
}

if (player.isDead && Input.GetKeyDown(KeyCode.R))
{
    SceneManager.LoadScene(1);
}

1 başvuru
IEnumerator GameOverFlickerRoutine()
{
    while (true)
    {
        gameOverPanel.transform.GetChild(0).GetComponent<TextMeshProUGUI>().text = "GAME OVER";
        yield return new WaitForSeconds(1f);
        gameOverPanel.transform.GetChild(0).GetComponent<TextMeshProUGUI>().text = " ";
        yield return new WaitForSeconds(1f);
    }
}

```

BÖLÜM 4 : Ana Menü Oluşturma ve Oyunun Başlatılması

Öncelikle “Menu” isminde yeni bir sahne oluşturulur. Sahne içerisinde canvas üzerinde “Play” isminde bir buton eklenir. Bu sahnedeki tüm menu işlemlerini kontrol etmek için “MenuManager” isminde bir script oluşturulur. Bu script içerisinde “StartGame” isminde public olarak bir fonksiyon oluşturulup SceneManager ile bir sonraki sahnenin tekrar yüklenmesi sağlanır. Oluşturulan bu fonksiyon sahneye eklediğimiz “Play” butonuna basıldığı zaman çağırılacak şekilde Button componentine eklenir.

```
0 başvuru
public void StartGame()
{
    SceneManager.LoadScene(1);
}
```

