

# HAFTA 9 – RAPOR

DERS : OYUN PROGRAMLAMA

AD SOYAD : LÜTFÜ BEDEL

ÖĞRENCİ NO : 21360859030

GITHUB : [https://github.com/lutfubedel/BTU\\_Oyun\\_Programlama\\_Kodlari/tree/main/Hafta\\_9](https://github.com/lutfubedel/BTU_Oyun_Programlama_Kodlari/tree/main/Hafta_9)

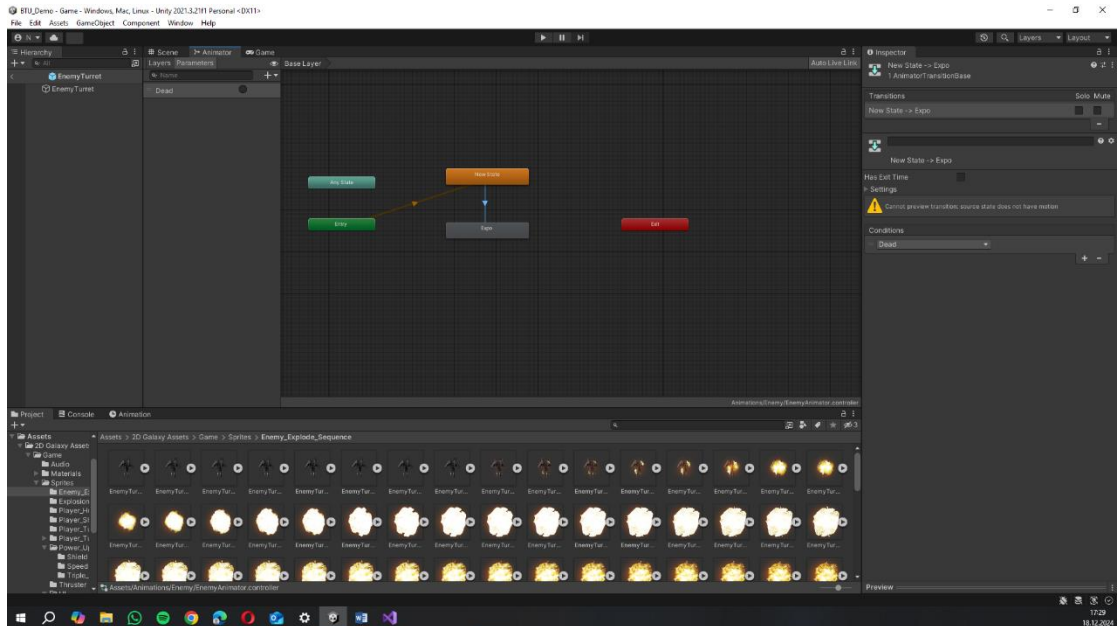
## BÖLÜM 1 : Düşman Gemilerin Yok Edilmesi

Düşman gemiler yok edildiği zaman bir patlama animasyonu oynaması gerekiyor. Bunun için 2D Galaxy Assets klasörü içerisinde bulunan “Enemy\_Explode\_Sequence” klasörü içerisinde bulunan spriteler kullanılarak “Expo” adında bir animasyon oluşturulur ve bu animasyon EnemyAnimator’ün içerisine eklenir.

Oyun başlar başlamaz animasyonun oynamaması için içi boş bir new State oluşturulur ve animasyona sağ tık yapılarak “Set as Layer Default State” ile default state olarak belirlenir. Objenin vurulup vurulmadığını kontrol etmek için animator içerisinde trigger türünde “Dead” isminde bir parametre oluşturulur. Dead aktif olduğunda expo animasyonu oynıyacak şekilde animasyonlar birbirine bağlanır.

Bu animasyon bir kez oynayacağı için “Loop Time” kapatılır ve “Dead” parametresi aktif olduktan sonra belli bir süre beklemek istemediğimiz için “Has Exit Time” kapatılır.

Bu animasyon enemy objesi bir laser objesi ile çarpıştığı zaman çalışacaktır. Bu sebepten “EnemyMovement” scripti içerisindeki “OnTriggerEnter2D” fonksiyonu güncellenir. Eğer çarpışan objenin tagı “Laser” ise önce oyuncunun scoreu artırılır ardından enemyAnimator içerisinde bulunan “Dead” parametresi aktif edilir ve expo animasyonu oynamaya başlar. Düşman bu durumda oyuncuya zarar vermemelidir bu yüzden önce objenin “moveSpeed” değerini sıfırlar bu şekilde patlayan obje olduğu yerde kalır. Animasyon oynarken oluşabilecek çarpışmaları önlemek için objenin colliderı kapatılır ve enemy objesi 2 saniye sonra oyundan silinir.



```

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);

        anim.SetTrigger("Dead");
        moveSpeed = 0;
        Destroy(other.gameObject);
        gameObject.GetComponent<BoxCollider2D>().enabled = false;
        Destroy(this.gameObject, 2f);
    }
}

```

## BÖLÜM 2 : Astreoidin Oyuna Eklenmesi

Başlangıçta astreoid sürekli olarak z ekseninde oyuncuya doğru gelmeli ve hareket ederken sürekli olarak kendi etrafında dönmeli. “Astreoid” scripti içerisinde Update fonksiyonunda transform.Rotate() fonksiyonu ile objenin sürekli olarak z ekseninde dönmesi sağlanır. Benzer şekilde objenin transform.position değerine sürekli olarak objeyi z ekseninde hareket ettirecek bir Vector3 eklenerek objenin z ekseninde hareket etmesi sağlanır.

```

Unity İletisi | 0 başvuru
private void Update()
{
    transform.Rotate(new Vector3(0, 0, 1 * rotateSpeed * Time.deltaTime));
    transform.position += new Vector3(0, -1 * moveSpeed * Time.deltaTime, 0);
}

```

Astreoidin patlama animasyonu enemy patlama animasyonu ile aynı şekilde hazırlanır yani “Laser” tagında herhangi bir obje ile çarpışır ise animator içerisindeki “Dead” triggerı tetiklenir ve patlama animasyonu oynamaya başlar. Animasyon oynarken player’ı zarar vermemesi için önce hareket hızı sıfırlanır ardından colliderı kapatılır. 2 saniye sonrada obje yok edilir.

```

@ 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);

        anim.SetTrigger("Dead");
        moveSpeed = 0;
        Destroy(other.gameObject);
        gameObject.GetComponent<BoxCollider2D>().enabled = false;
        Destroy(this.gameObject, 2f);

        GameObject.FindWithTag("SpawnManager").GetComponent<SpawnManager>().isGameStarted = true;
    }
}

```

### BÖLÜM 3 : Spawn Routinlerin Başlatılması

Astreoid objesi yok edilmeden enemy ve bonus objelerinin oluşmasını istemeyiz bu yüzden "SpawnManager" scripti içerisinde bazı değişiklikler yapılmalıdır. Öncelikle astreoid objesi yok edildikten sonra "SpawnManager" scripti içerisinde tanımlanmış olan "isGameStarted" değişkenini true yapar. SpawnManager scripti içerisinde tüm coroutinlerin çağrılarını Start fonksiyonundan "Spawner" isimli bir fonksiyona taşınır. Eğer "isWorking" değişkeni false ise yani coroutinler daha önce çalışmamışlar ise ve "isGameStarted" değişkeni true ise Spawner fonksiyonu çağırılır ve enemy, bonus coroutinleri çalışmaya başlar.

```

public bool isGameStarted;
public bool isWorking;

@ Unity İletisi | 0 başvuru
private void Update()
{
    if (!isWorking)
    {
        Spawner();
    }
}

1 başvuru
public void Spawner()
{
    if (isGameStarted)
    {
        enemySpawnerWorking = true;
        tripleShotSpawnerWorking = true;

        StartCoroutine(EnemySpawner());
        StartCoroutine(TripleShotSpawner());
        StartCoroutine(SpeedBonusSpawner());

        isWorking = true;
    }
}

```

## BÖLÜM 4 : Thruster ve Hasar Animasyonlarının Eklenmesi

Player uzay gemisinin daha güzel görünmesi için hemen arkasına animasyonlu bir thruster eklenir. Bu thruster assetler içerisindeki spriteler kullanılarak oluşturulmuş bir animasyona sahiptir. Bu animasyonun da sürekli oynamasını istediğimiz için Loop Time işaretlenir.

Player hasar aldıkça hasar aldığını göstermek için sağ ve sol motor olmak üzere iki tane yanma animasyonu eklenir. Bu animasyonlar assets klasörü içerisindeki spriteler kullanılarak oluşturulmuş bir animasyona sahiptir.

Eğer oyuncunun can değeri 2 ye düşerse sol motordaki yanma animasyonu görünür hale gelir. Eğer tekrar hasar alırsa bu sefer sağ motordaki yanma animasyonu görünür hale gelicektir.

```
if (life == 2)
    leftEngine.SetActive(true);

if (life == 1)
    rightEngine.SetActive(true);
```

