



Ad: Hilal

Soyad: Kaya

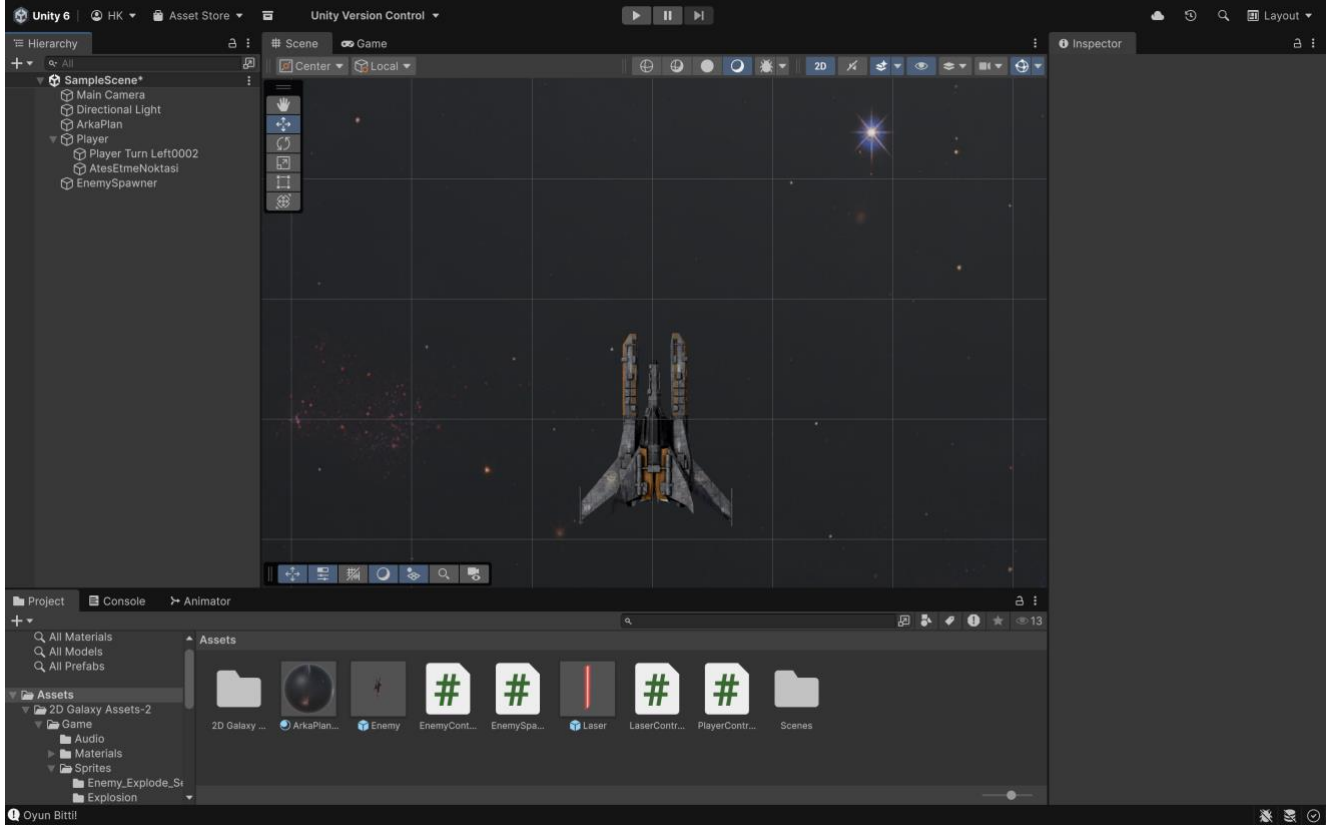
Numara: 22360859046

Ders: Oyun Programlama

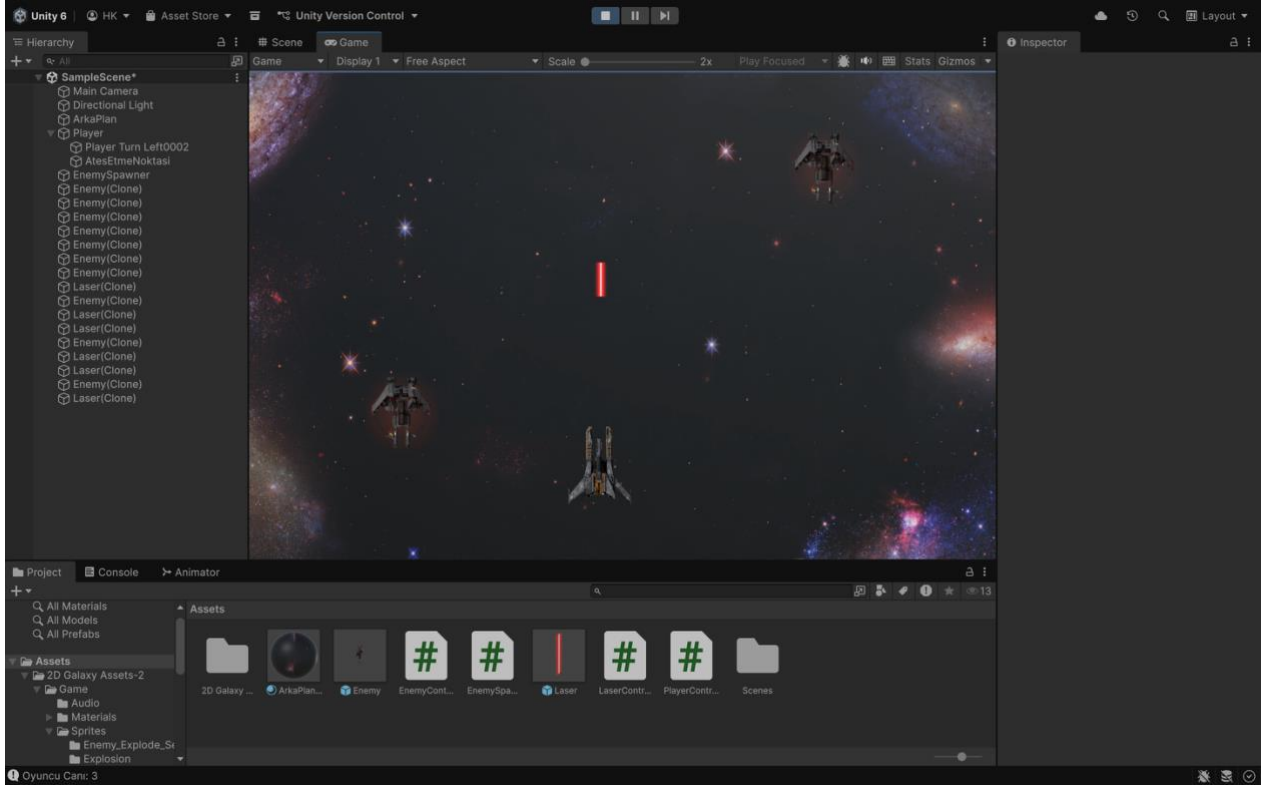
Ödev adı: Hafta 5

Geçen hafta bir uzay oyunu yapmıştık. Şimdi bu oyunu biraz daha geliştirip bazı ek özellikler ekleyeceğiz.

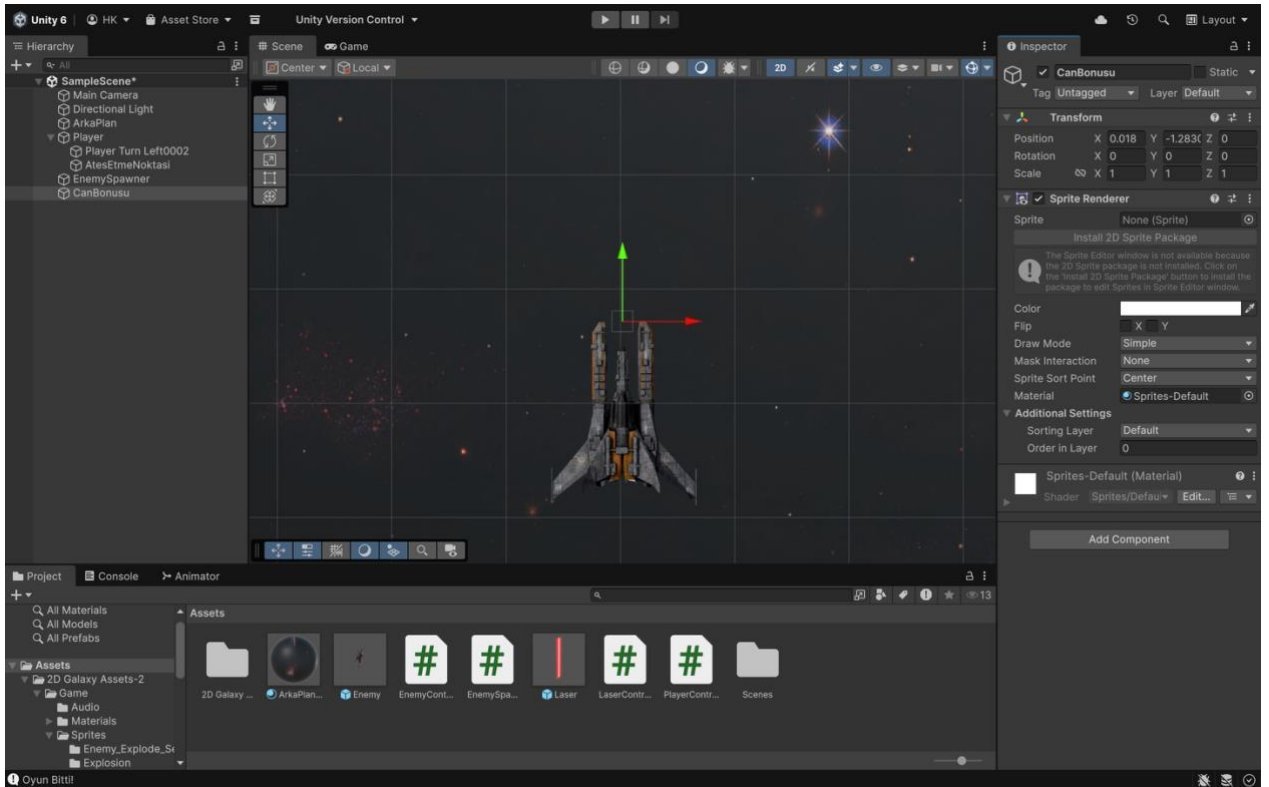
Player nesnemizin child'ı olarak boş bir nesne olan AtesEtmeNoktasi adında bir nesne oluşturuyoruz. Bunu laser prefabimizin çıkmasını istediğimiz konuma yerleştiriyoruz.



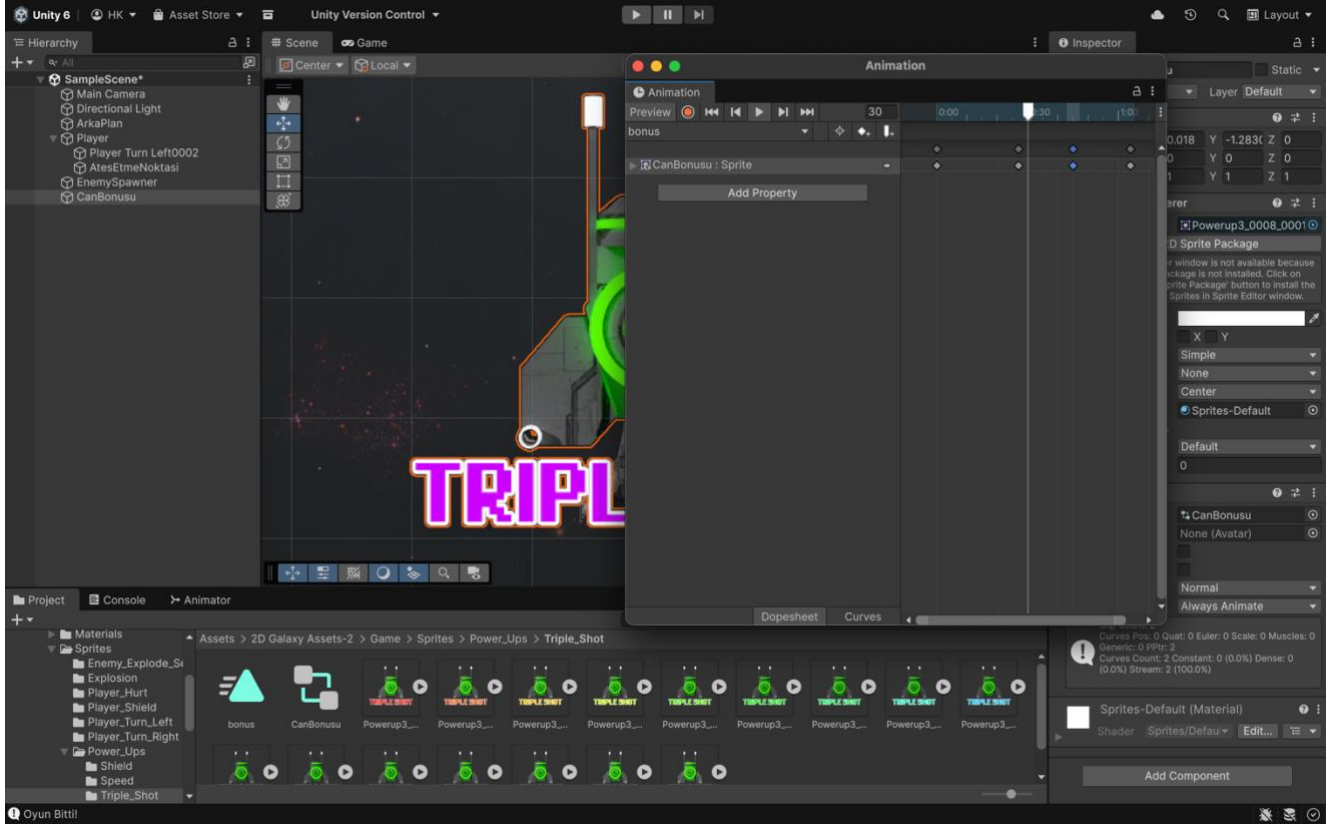
Oyun başlatıldığında enemyler aşağı doğru inip kaybolurken player nesnesi sadece yatay eksenli hareket yapabilecek ve lazer fırlatabilecektir.



Daha sonra CanBonusu adında bir boş nesne oluşturup Sprite Renderer komponentini ekleyerek istediğimiz görseli sprite kısmına sürükleriz.



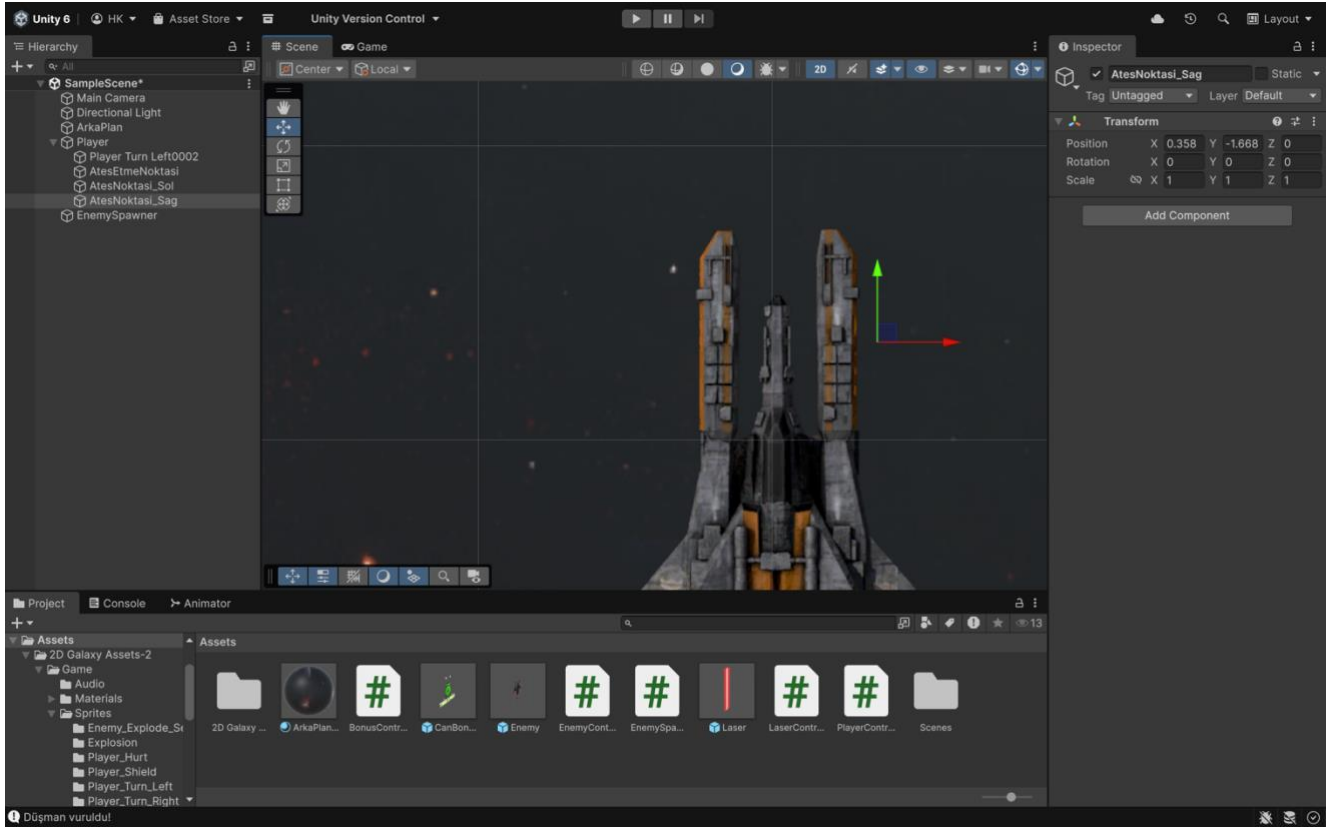
Biz CanBonusu kısmını animasyonlu yapmak istiyoruz. Bu yüzden şu adımları izleyeceğiz: Window -> Animation -> Animation dedikten sonra create butonuna basarak tekrar etmesini istediğimiz görselleri zaman çizelgesine sürükleriz.



Daha sonra CanBonusu nesnemize de diğerleri gibi gerekli komponentleri (rigidbody, box collider) ekledikten sonra box collider altında edit collider seçeneğiyle temas sınırlarını belirleriz. Ardından tag'ini de Bonus olarak ayarlarız.



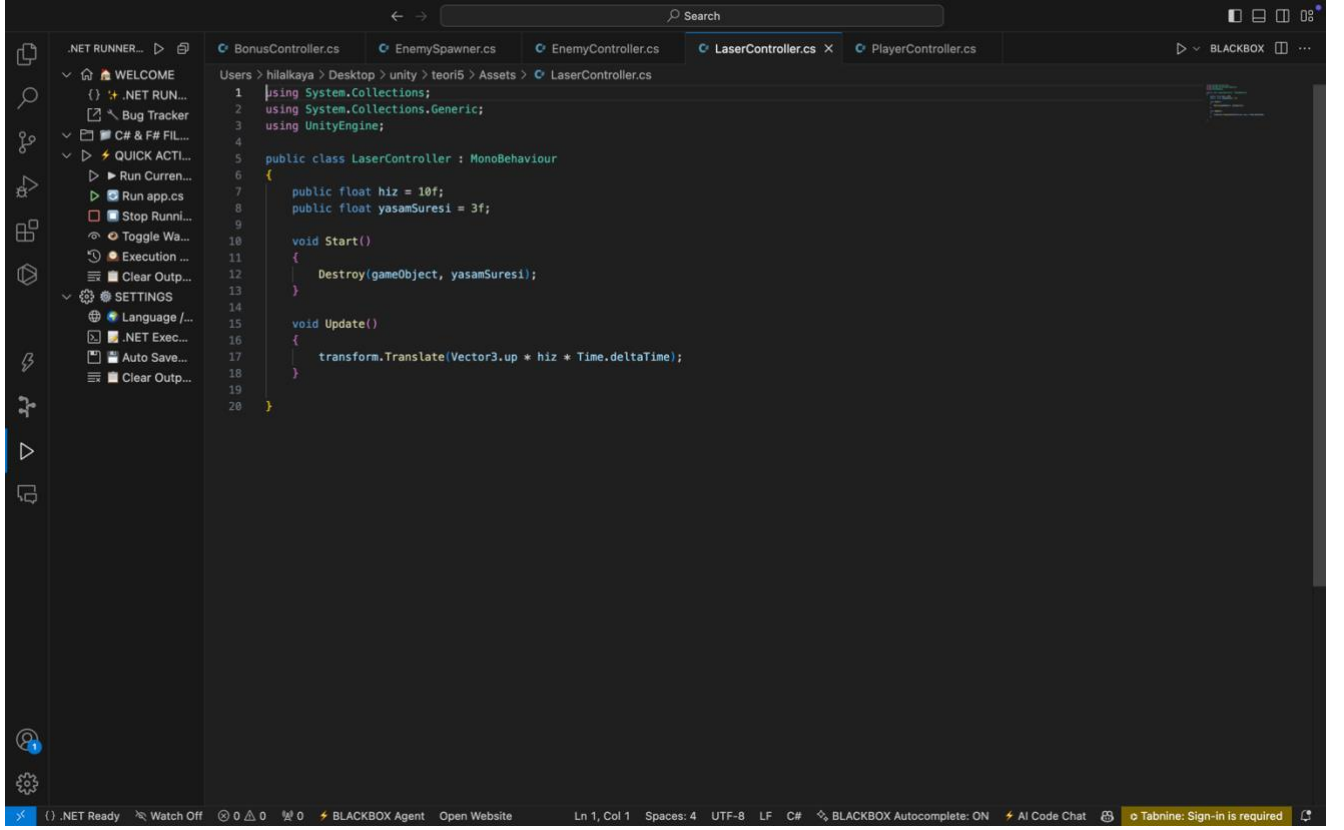
Tüm bunları yaptıktan sonra biz player'ın 3'lü atış yapmasını istiyoruz. Bunun için de sağ ve sola iki tane daha ateş etme noktası ekliyoruz.



Bunları da player nesnemizin scriptinde ateş etme noktaları listesine ekleriz.

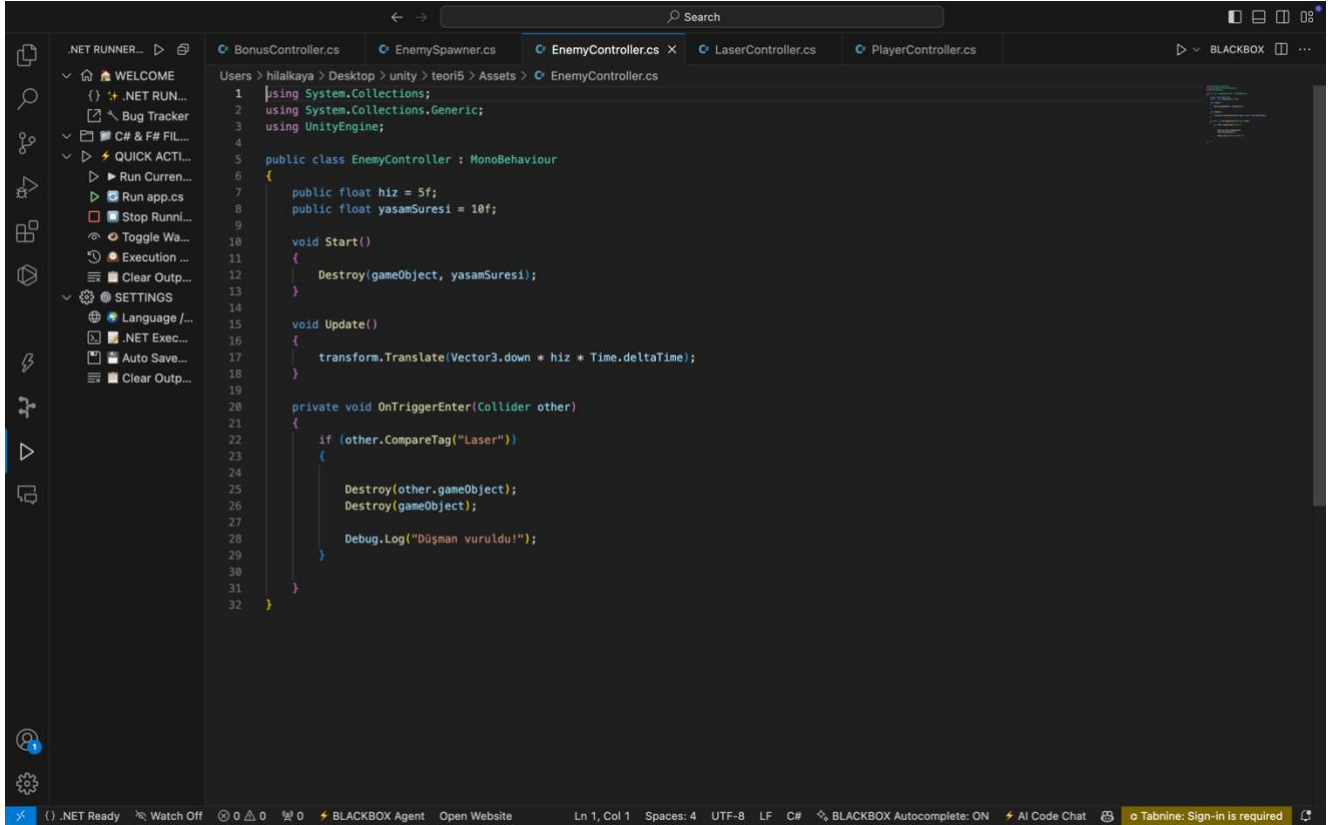


Şimdi sırada bunların davranışlarını kontrol ettiğimiz scriptlerin içeriğine göz atmak var.



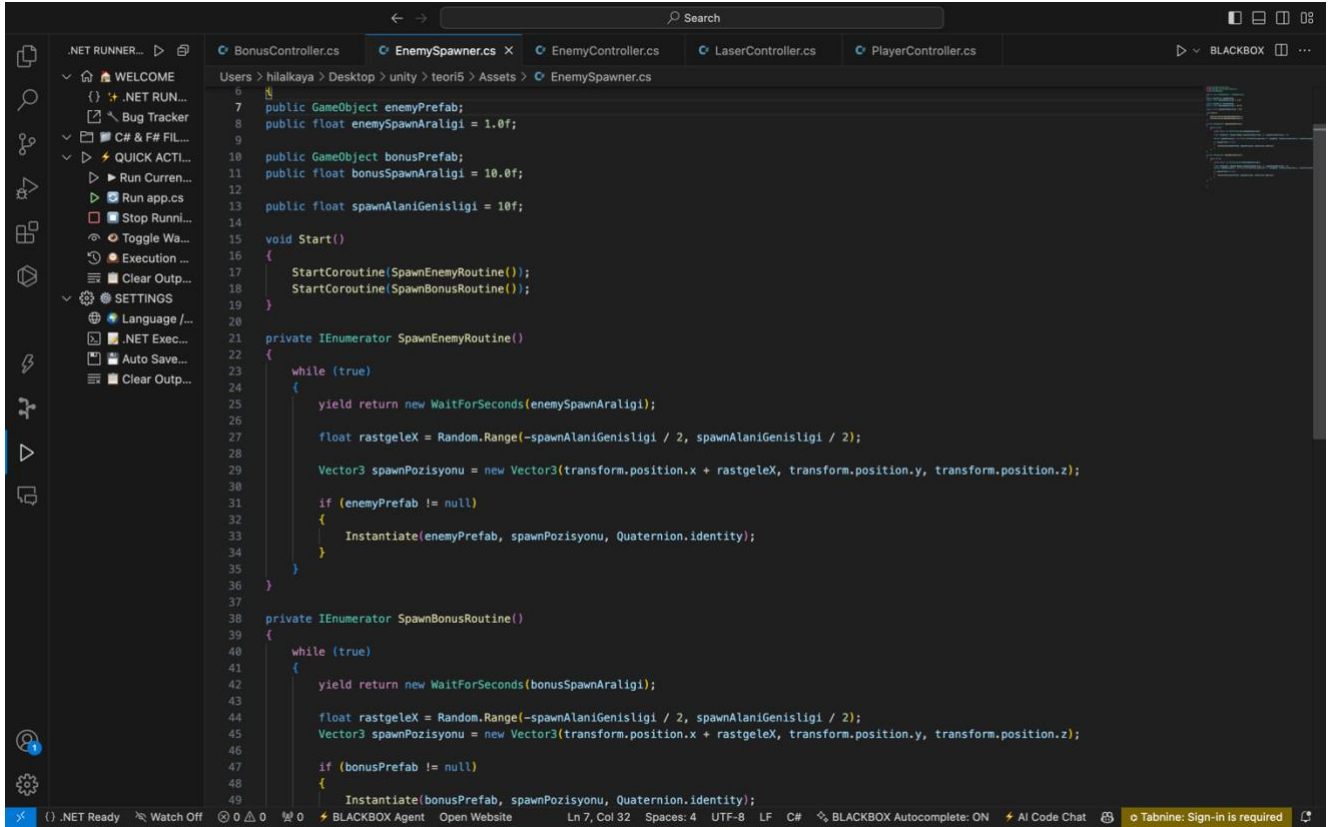
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class LaserController : MonoBehaviour
6 {
7     public float hiz = 10f;
8     public float yasamSuresi = 3f;
9
10    void Start()
11    {
12        Destroy(gameObject, yasamSuresi);
13    }
14
15    void Update()
16    {
17        transform.Translate(Vector3.up * hiz * Time.deltaTime);
18    }
19 }
20
```

Laser ve Enemy nesneleri için hareket yönlerini ve ve ekranda ne kadar süre kalacaklarını belirleyen script kodlarını yazdık.



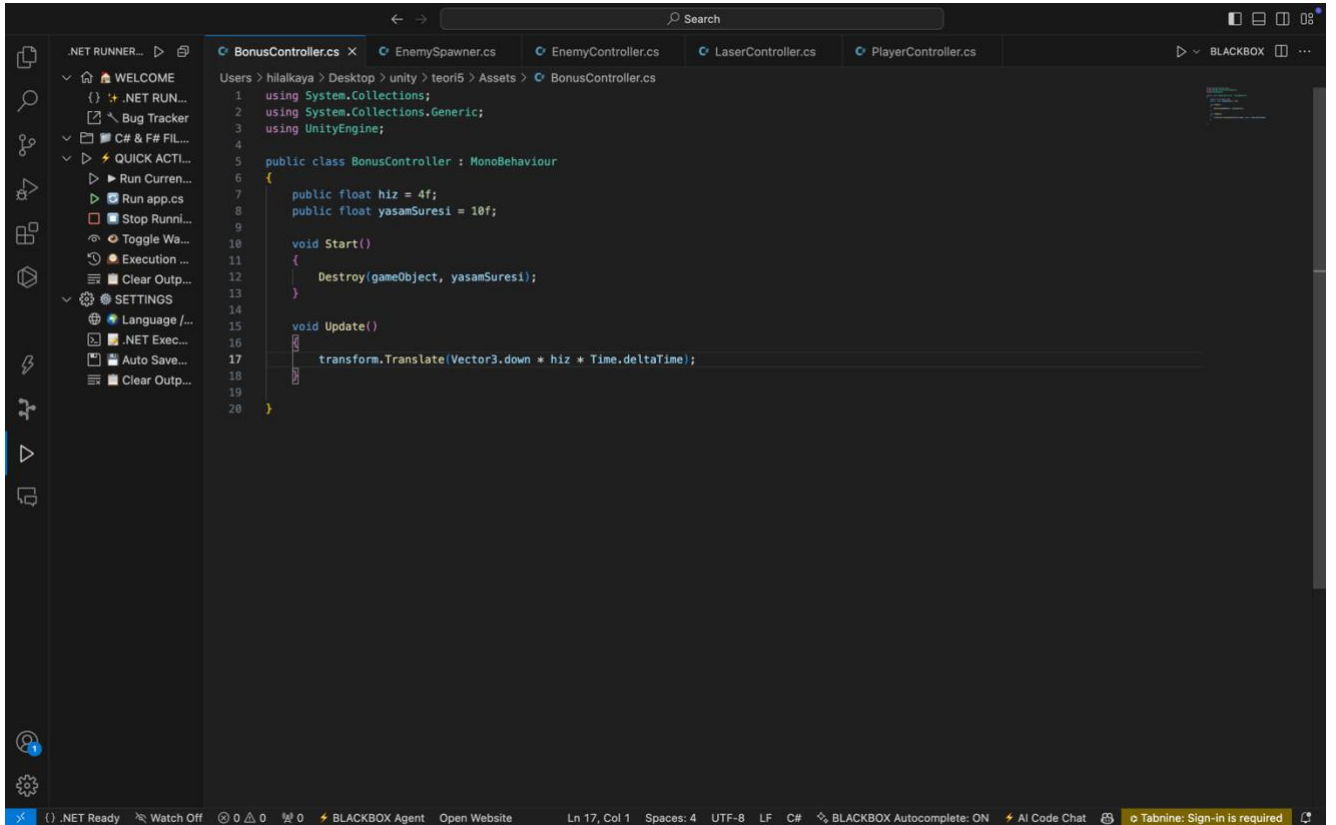
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnemyController : MonoBehaviour
6 {
7     public float hiz = 5f;
8     public float yasamSuresi = 10f;
9
10    void Start()
11    {
12        Destroy(gameObject, yasamSuresi);
13    }
14
15    void Update()
16    {
17        transform.Translate(Vector3.down * hiz * Time.deltaTime);
18    }
19
20    private void OnTriggerEnter(Collider other)
21    {
22        if (other.CompareTag("Laser"))
23        {
24            Destroy(other.gameObject);
25            Destroy(gameObject);
26            Debug.Log("Düşman vuruldu!");
27        }
28    }
29 }
30
31
32
```

EnemySpawner'da ise sonsuz döngü ile rastgele x pozisyonundan enemy ve bonus nesnelerinin üretilmesini sağladık.



```
6 public GameObject enemyPrefab;
7 public float enemySpawnAraligi = 1.0f;
8
9
10 public GameObject bonusPrefab;
11 public float bonusSpawnAraligi = 10.0f;
12
13 public float spawnAlaniGenisligi = 10f;
14
15 void Start()
16 {
17     StartCoroutine(SpawnEnemyRoutine());
18     StartCoroutine(SpawnBonusRoutine());
19 }
20
21 private IEnumerator SpawnEnemyRoutine()
22 {
23     while (true)
24     {
25         yield return new WaitForSeconds(enemySpawnAraligi);
26
27         float rastgeleX = Random.Range(-spawnAlaniGenisligi / 2, spawnAlaniGenisligi / 2);
28
29         Vector3 spawnPozisyonu = new Vector3(transform.position.x + rastgeleX, transform.position.y, transform.position.z);
30
31         if (enemyPrefab != null)
32         {
33             Instantiate(enemyPrefab, spawnPozisyonu, Quaternion.identity);
34         }
35     }
36 }
37
38 private IEnumerator SpawnBonusRoutine()
39 {
40     while (true)
41     {
42         yield return new WaitForSeconds(bonusSpawnAraligi);
43
44         float rastgeleX = Random.Range(-spawnAlaniGenisligi / 2, spawnAlaniGenisligi / 2);
45         Vector3 spawnPozisyonu = new Vector3(transform.position.x + rastgeleX, transform.position.y, transform.position.z);
46
47         if (bonusPrefab != null)
48         {
49             Instantiate(bonusPrefab, spawnPozisyonu, Quaternion.identity);
50         }
51     }
52 }
```

BonusController ile de bonusun hareket yönlerini ve ve ekranda ne kadar süre kalacaklarını belirledik.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class BonusController : MonoBehaviour
6 {
7     public float hiz = 4f;
8     public float yasamSuresi = 10f;
9
10     void Start()
11     {
12         Destroy(gameObject, yasamSuresi);
13     }
14
15     void Update()
16     {
17         transform.Translate(Vector3.down * hiz * Time.deltaTime);
18     }
19 }
20
```



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public float hareketHizi = 7f;
    public float ekranKenarBoslugu = 1f;
    private float minX, maxX;

    public int can = 3;

    public GameObject laserPrefab;

    public Transform[] atesEtmeNoktalari;

    public float atesEtmeAraligi = 0.5f;
    private float atesEtmeZamani;

    void Start()
    {
        EkranSinirlariniHesapla();
        Debug.Log("Oyuncu Canı: " + can);
    }

    void EkranSinirlariniHesapla()
    {
        Camera kamera = Camera.main;
        Vector3 solAlt = kamera.ViewportToWorldPoint(new Vector3(0, 0, 10));
        Vector3 sagUst = kamera.ViewportToWorldPoint(new Vector3(1, 0, 10));
        minX = solAlt.x + ekranKenarBoslugu;
        maxX = sagUst.x - ekranKenarBoslugu;
    }

    void Update()
    {
        if (can <= 0) return;
        HareketiYonet();
        AtesEtmeyiYonet();
    }
}
```

```

}

void HareketiYonet()
{
    float yatayGiris = Input.GetAxis("Horizontal");
    Vector3 pozisyon = transform.position;
    pozisyon.x += yatayGiris * hareketHizi * Time.deltaTime;
    pozisyon.x = Mathf.Clamp(pozisyon.x, minX, maxX);
    transform.position = pozisyon;
}

void AtesEtmeyiYonet()
{
    if (Input.GetKey(KeyCode.Space) && Time.time > atesEtmeZamani)
    {
        atesEtmeZamani = Time.time + atesEtmeAraligi;
        AtesEt();
    }
}

void AtesEt()
{
    foreach (Transform nokta in atesEtmeNoktalari)
    {
        Instantiate(laserPrefab, nokta.position, Quaternion.identity);
    }
}

private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Enemy"))
    {
        can--;
        Debug.Log("Düşman çarptı! Kalan Can: " + can);
        Destroy(other.gameObject);

        if (can <= 0)
        {

```

```

        OnPlayerDeath();
    }
}

else if (other.CompareTag("Bonus"))
{
    can++;
    Debug.Log("Bonus alındı! Kalan Can: " + can);
    Destroy(other.gameObject);
}
}

private void OnPlayerDeath()
{
    Debug.Log("Oyun Bitti! (OnPlayerDeath çağrıldı)");
    SpriteRenderer sr = GetComponent<SpriteRenderer>();
    if (sr != null)
    {
        sr.enabled = false;
    }
    Collider col = GetComponent<Collider>();
    if (col != null)
    {
        col.enabled = false;
    }
}

```

PlayerController'in uzuluğu biraz fazla olduğu için onun kodlarını buraya kopyaladım. Şimdi işlevlerini anlatacağım.

Start fonksiyonunun içinde Player'ın canını konsola yazdıran kodu yazdık. Player'ın yatay hareketini ve ekranın dışına çıkmasını önleyecek fonksiyonları da verdik. Boşluk tuşuna basıldığında laser üretimini gerçekleyecek fonksiyonları ve enemyler ile çarpıştığında canını artıracak, bonuslar ile çarpıştığında ise canını artıracak komutları da verdik. Canı bittiğindeyse onu yok ettik.

Github Linki :