

HAFTA 7 – RAPOR

DERS : OYUN PROGRAMLAMA

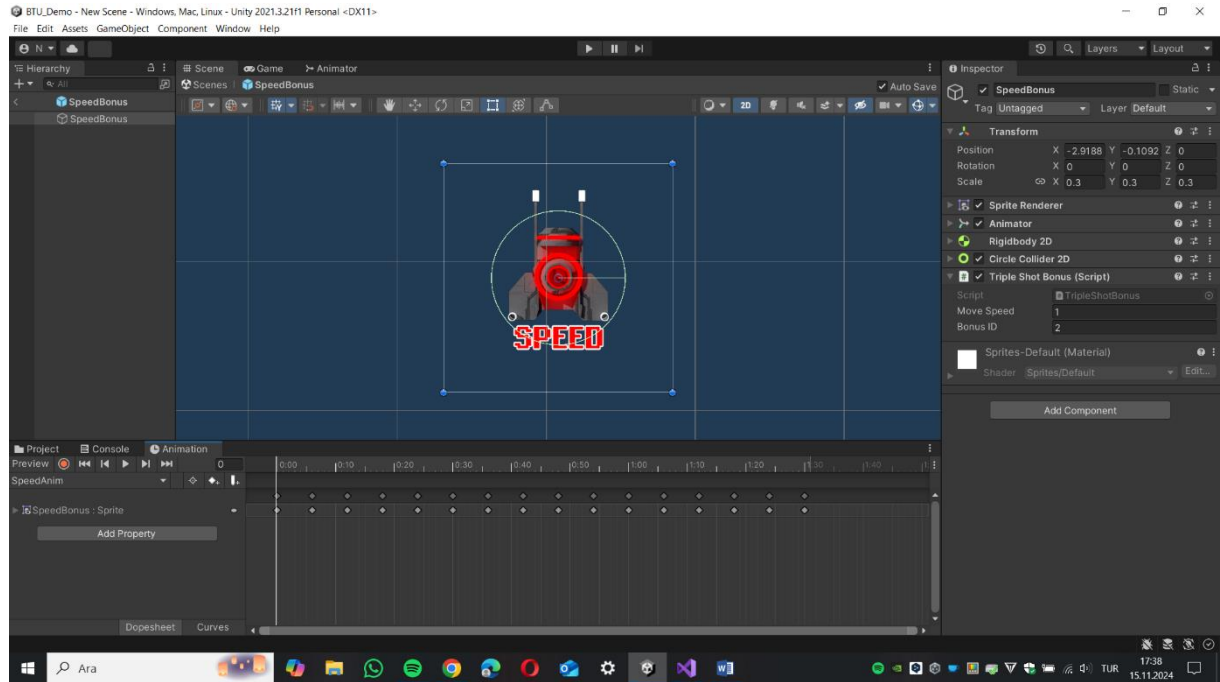
AD SOYAD : LÜTFÜ BEDEL

ÖĞRENCİ NO : 21360859030

GITHUB : https://github.com/lutfubedel/BTU_Oyun_Programlama_Kodlari/tree/main/Hafta_7

BÖLÜM 1 : Hız Bonusunun Oluşturulması

İndirilen assetler içerisinde bulunan hız sprite kullanılarak “SpeedBonus” adında bir obje oluşturulur. Bu objenin tüm boyutlardaki scale değerleri 0.3 olarak ayarlanır. Ardından çarpışma ve fizik işlemleri için gerekli olan “Rigidbody2D” ve “CircleCollider2D” componentleri eklenir. Son olarak da spritenin daha güzel görünmesi için hız spriteları ile animasyon oluşturulur ve bu animasyonu oynatmak için objeye “Animator” componenti eklenir.



BÖLÜM 2 : Bonus Scriptinin Güncellenmesi

Daha önceden üçlü atış bonusu için oluşturulan "TripleShotBonus" scriptinin ismini "Bonus" olarak güncellenir ve tüm bonuslar için ortak bir script haline getirilecektir. Öncelikle bu scriptin hangi bonus prefabının içerisinde olduğunu tutmak için "bonusID" isminde bir int değişken oluşturulur. bonusID üçlü atış bonusu için 1 hız bonusu için ise 2 değerini alacaktır. OnTriggerEnter2D fonksiyonu içerisinde if else-if yapıları kullanılarak her bonus için ayrı çalışacak aksiyonlar kodlanır.

Hız bonusu için "bonusID" değişkeni 2 ye eşit ise ve bonusun çarptığı objenin tagı "Player" a eşit ise playerın "PlayerMovement" scriptine ulaşır ve içerisindeki "isSpeedBonusActive" değişkenini true ya eşitler ve kendini yok eder.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Bonus : MonoBehaviour
6 {
7
8     public float moveSpeed;
9     public int bonusID;
10
11
12     private void Update()
13     {
14         transform.Translate(new Vector3(0, -1 * moveSpeed * Time.deltaTime, 0));
15
16         if (transform.position.y < -4.5f)
17         {
18             Destroy(this.gameObject);
19         }
20     }
21
22
23     private void OnTriggerEnter2D(Collider2D collision)
24     {
25         if(bonusID == 1)
26         {
27             if (collision.gameObject.CompareTag("Player"))
28             {
29                 collision.gameObject.GetComponent<Fire>().isTripleLaserActive = true;
30                 Destroy(this.gameObject);
31             }
32         }
33
34         else if (bonusID == 2)
35         {
36             if (collision.gameObject.CompareTag("Player"))
37             {
38                 collision.gameObject.GetComponent<PlayerMovement>().isSpeedBonusActive = true;
39                 Destroy(this.gameObject);
40             }
41         }
42     }
43 }
44
45
```

BÖLÜM 3 : PlayerMovement Scriptinin Güncellenmesi

Öncelikle PlayerMovement scripti içerisinde hız bonusunun aktif olup olmadığını kontrol eden “isSpeedBonusActive” isminde bir değişken tanımlanır. Playerın hızını Update fonksiyonu içerisinde ternary yapısı ile eğer isSpeedBonusActive değişkeni true ise 100 değilse 50 olarak güncellenmesi sağlanır.

```
moveSpeed = isSpeedBonusActive ? 100 : 50;
```

Eğer isSpeedBonusActive true ise yine Update fonksiyonunda değişken aktif olduktan 5sn sonra değişkeni tekrar false yapacak “ChangeSpeedBonusActive” fonksiyonu Invoke fonksiyonu ile çağırılır.

```
private void ChangeSpeedBonusActive()
{
    isSpeedBonusActive = false;
}
```

```
© Unity Betiği (2 varlık başvurusu) | 2 başvuru
public class PlayerMovement : MonoBehaviour
{
    Rigidbody2D rb;
    SpawnManager spawner;

    public float moveSpeed;
    public float life;

    public bool isSpeedBonusActive;

    © Unity İletisi | 0 başvuru
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        spawner = GameObject.FindWithTag("SpawnManager").GetComponent<SpawnManager>();
    }

    © Unity İletisi | 0 başvuru
    void Update()
    {
        moveSpeed = isSpeedBonusActive ? 100 : 50;

        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");

        Vector3 movement = new Vector3(horizontal, vertical, 0);
        rb.MovePosition(transform.position + movement * moveSpeed * Time.deltaTime);

        if(life <=0)
        {
            spawner.StopAllCoroutines();
        }

        if (isSpeedBonusActive)
            Invoke(nameof(ChangeSpeedBonusActive), 5f);
    }

    1 başvuru
    private void ChangeSpeedBonusActive()
    {
        isSpeedBonusActive = false;
    }
}
```

BÖLÜM 4 : SpawnManager Scriptinin Güncellenmesi

Öncelikle GameObject türünde oluşturulacak olan hız bonusunu tutan “speedBonus” ve hız bonusu objesini oluşturan coroutine ni kontrol eden bool türünde “speedSpawnerWorking” isminde bir değişken tanımlanır.

Ardında zaman ile ilgili işlemleri yapabilmek için “SpeedBonusSpawner” adında ve IEnumerator türünde bir coroutine oluşturulur. Hız bonusunun oluşturulma işlemi tüm oyun boyunca devam edeceği için öncelikle bir while döngüsü oluşturulur ve “speedSpawnerWorking” değişkeninin değeri true olduğu sürece enemy objesinin oluşturulması sağlanır Ardından Instantiate fonksiyonu ile speedBonus prefabı ekranın üst kısmında x konumu rastgele olacak şekilde ve parent objesi “parentObj” olacak şekilde oluşturulur.

```
Instantiate(speedBonus, new Vector3(Random.Range(-8.5f, 8.5f), 6.5f, 0), Quaternion.identity);
```

Son olarak aynı anda birden fazla hız bonusu objesi oluşturmasını önlemek adına “WaitForSeconds” ile bir sonraki bonus objesini oluşturulana kadar 10 saniye beklemesi sağlanır.

```
yield return new WaitForSeconds(10);
```

```
public GameObject enemy;
public GameObject tripleShotBonus;
public GameObject speedBonus;

public bool enemySpawnerWorking;
public bool tripleShotSpawnerWorking;
public bool speedSpawnerWorking;

@ Unity İletisi | 0 başvuru
private void Start()
{
    enemySpawnerWorking = true;
    tripleShotSpawnerWorking = true;

    StartCoroutine EnemySpawner();
    StartCoroutine TripleShotSpawner();
    StartCoroutine SpeedBonusSpawner();
}
```

```
1 başvuru
IEnumerator SpeedBonusSpawner()
{
    while (tripleShotSpawnerWorking)
    {
        Instantiate(speedBonus, new Vector3(Random.Range(-8.5f, 8.5f), 6.5f, 0), Quaternion.identity);
        yield return new WaitForSeconds(10);
    }
}
```

```
0 başvuru
public void StopSpeedSpawner()
{
    speedSpawnerWorking = false;
}
```

