

BLM0364 OYUN PROGRAMLAMA – HAFTA 6 RAPORU

Yasin Ekici

21360859029

Rapor İçeriği

- Üçlü atış bonusu: Laser'den iki kopya daha oluşturun ve bu üç laser'in konumlarını ayarlayın.
- Boş bir oyun nesnesi oluşturun ve üç laser'i bunun child'ı olarak atayın
- Bu oyun nesnesinden bir prefab oluşturun
- Üçlü bonusun hangi durumlarda aktive edileceğinin mantığını geliştirin ve kodlayın
- Üçlü bonus sprite'ı toplandığında belirli bir süre için üçlü bonus'u etkinleştirin
- FireLaser fonksiyonunda gerekli güncellemeleri yapın
- Üçlü bonus sprite'ından bir nesne oluşturun, ölçeğini ayarlayın, circle collider ve rigidbody ekleyin, sorting layer'ı düzenleyin
- Script oluşturarak üçlü bonus nesnesi ile ilişkilendirin, script ile nesnenin belirli bir hızla hareket etmesini sağlayın. Ekran dışına çıktığında nesneyi yok edin. Çarpışma kontrolü ekleyin.
- Player kodu içinden üçlü bonus koduna erişin ve üçlü bonus'u etkinleştirin
- Üçlü atış bonusu nesnesi için animasyon ekleyin
- SpawnManager'ı üçlü atış bonusunu da dikkate alacak şekilde düzenleyin

Bu raporda anlatılan kodlar için: <https://github.com/YasinEkici/game-programming>

Asset'ler için:

<https://e.pcloud.link/publink/show?code=XZhFVJZuzLw5aWEuizh3kDHE43iNVyak6l7>

1. Üçlü Atış Bonusu

Triple Shot Prefab Oluşturma:

- Hierarchy'de bir Laser ekleyip iki tane daha kopyalayın.
- Aynı y değerine sahip olacak şekilde kanatları negatif x değerinde konumlandırın.
- Yeni bir Triple_Shot adlı boş nesne oluşturun; 3 laser nesnesini bunun çocuğu yapın.
- Prefab haline getirip Hierarchy'den silin.



2. Üçlü bonusun hangi durumlarda aktive edileceği:

- Özel bir sprite toplandığında, kullanıcının triple shot'ın belli bir süre için aktif olduğunu bilmesini sağlayın.
- Triple Shot'ı belirli bir süre sonra devre dışı bırakmak için Boolean değişkeni tanımlayın.

```
private bool isTripleShotActive = false;
```

- Triple shot prefab'i için bir game object değişkeni tanımlayın.

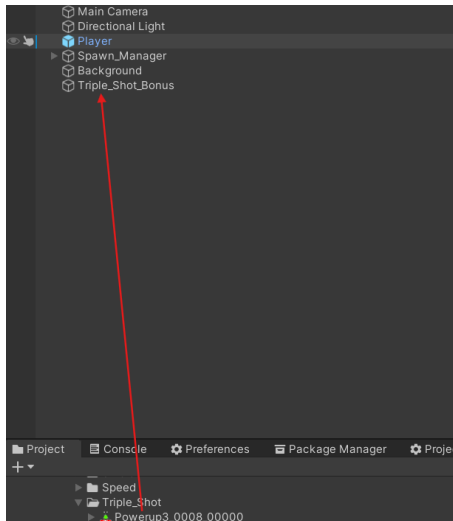
```
private GameObject tripleShotPrefab;
```

- FireLaser fonksiyonunu triple shot'ı dikkate alacak şekilde güncelleyin.

```
public void FireLaser(){  
    nextFire = Time.time + fireRate;  
  
    if(isTripleShotActive == true){  
        Instantiate(tripleShotPrefab, transform.position, Quaternion.identity);  
    }  
  
    else{  
        Instantiate(laserPrefab, transform.position + new Vector3(0, 1.15f, 0), Quaternion.identity);  
    }  
}
```

3. Üçlü Bonus Sprite'ı

- Triple shot'ın ilk karesini sprites klasöründen alıp Hierarchy penceresine sürükleyin.



- Nesnenin adını "Triple_Shot_Bonus" olarak değiştirin.
- Ölçeklendirmeyi 0.5, 0.5, 0.5 olarak ayarlayın.
- Toplanabilir olması için bir Circle Collider ekleyin.
- isTrigger özelliğini true olarak ayarlayın.

- Rigidbody2D ekleyin.
 - Gravity Scale değerini 0 yapın.
- Sorting Layer'ı "Foreground" olarak ayarlayın.
- Yeni bir C# script oluşturun: Bonus_sc.
- Bu script'i nesneye ekleyin.
- Nesneyi prefab haline getirin.
- Bonus nesnesi aşağı doğru 3 hızında hareket etmeli, ekran dışına çıktığında yok olmalı, ve oyuncu ile çarpıştığında triple shot'ı etkinleştirip kendini yok etmeli. Bonus_sc scriptine şu kodları ekleyin;

```
float speed = 3f;
void Update()
{
    transform.Translate(Vector3.down * speed * Time.deltaTime);

    if(transform.position.y < -5.83f){
        Destroy(this.gameObject);
    }
}

void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        Destroy(this.gameObject);
    }
}
```

4. Player kodundaki değişiklikler;

- Player scripti için, triple shot'ı aktif hale getirmek üzere bir public metod tanımlayın.

```
public void TripleShotActive(){
    isTripleShotActive = true;

    StartCoroutine(TripleShotBonusDisableRoutine());
}

IEnumerator TripleShotBonusDisableRoutine(){
    yield return new WaitForSeconds(5.0f);

    isTripleShotActive = false;
}
```

- Triple shot'ı etkinleştirmek ve süre dolduktan sonra devre dışı bırakmak için script iletişimini kurun. Tekrardan Bonus_sc'deki kodda şu değişiklikleri yapın;

```

void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        Player_sc player = other.transform.GetComponent<Player_sc>();
        if(player != null){
            player.TripleShotActive();
        }
        Destroy(this.gameObject);
    }
}
}
}

```

5. Parent Triple Shot'ı Yok Etme:

- Nesnenin bir parent'ı olup olmadığını kontrol edin (Triple shot lazerlerinde var olan bir parent var).
- Parent'ı silin. Laser_sc scriptindeki Update kodunu güncelleyin;

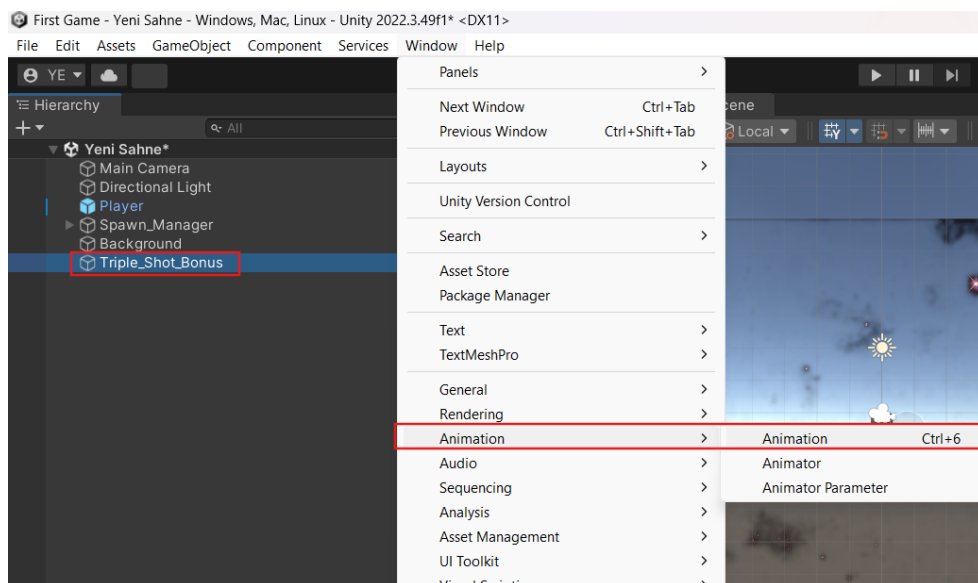
```

if(transform.position.y > 7f){
    if(transform.parent != null){
        Destroy(transform.parent.gameObject);
    }
    Destroy(this.gameObject);
}
}

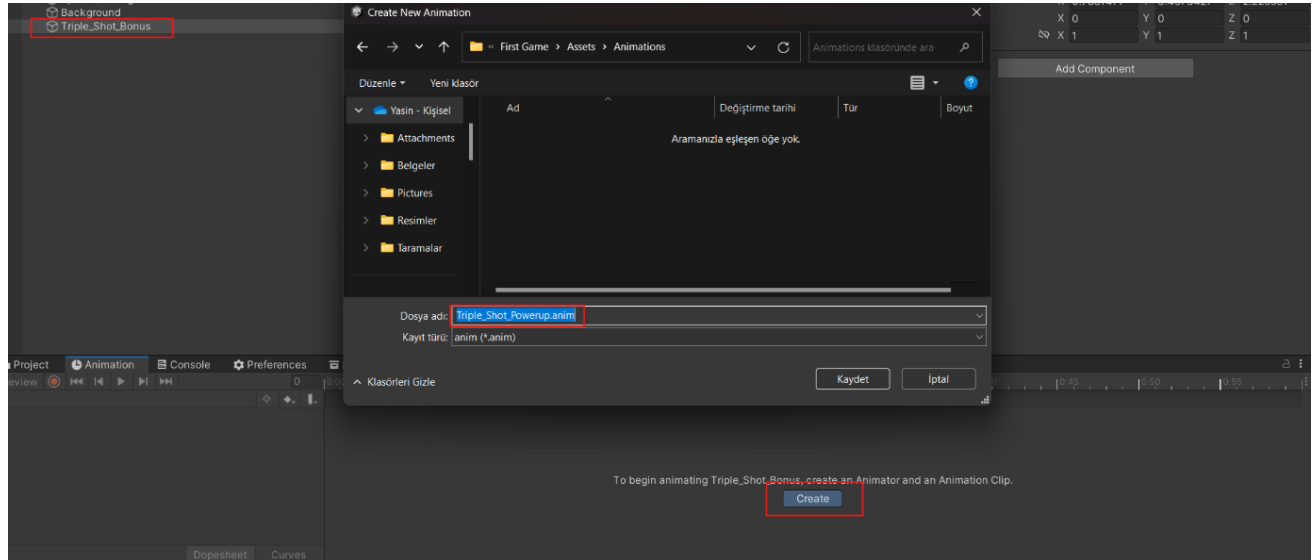
```

6. Animasyon

- Triple_Shot_Bonus nesnesini seçin.
- Window -> Animation -> Animation yolunu izleyin.



- Animation penceresini konsolun yanına sabitleyin.
- Bir animator ve bir animasyon klibi oluşturun.
- Assets altında animasyonlar için yeni bir klasör oluşturun: Animations.
- Yeni animasyonu Animations klasörüne kaydedin: Triple_Shot_Bonus_anim.
- Nesnenin seçili olduğundan emin olun.



- Key frame kaydı modunu etkinleştirin.
- Triple shot powerup klasöründeki kareleri seçip animasyon karesine sürükleyin.
- Animasyonu test etmek için oynatın.
- Kayıt modunu durdurun.
- Animasyonu prefab'e atayın.

7. Bonus'un spawnlanması

- Triple Shot Bonus'u dahil etmek için Spawn Manager'ı düzenleyin.
- Triple Shot Bonus prefab'i için bir değişken oluşturun.

GameObject tripleShotBonusPrefab;

- Mevcut rutinin adını SpawnRoutine'den SpawnEnemyRoutine olarak değiştirin.
- Bonus için yeni bir rutin oluşturun: SpawnBonusRoutine.

IEnumerator SpawnBonusRoutine(){

while(stopSpawning == false){

Vector3 position = new Vector3(Random.Range(-9.8f, 9.8f), 7.4f, 0);

Instantiate(tripleShotBonusPrefab, position, Quaternion.identity);

yield return new WaitForSeconds(Random.Range(5,10));

}

}

- **SpawnBonusRoutine'i başlatın;**

void Start()

```
{  
    StartCoroutine(SpawnEnemyRoutine());  
    StartCoroutine(SpawnPowerupRoutine());  
}
```