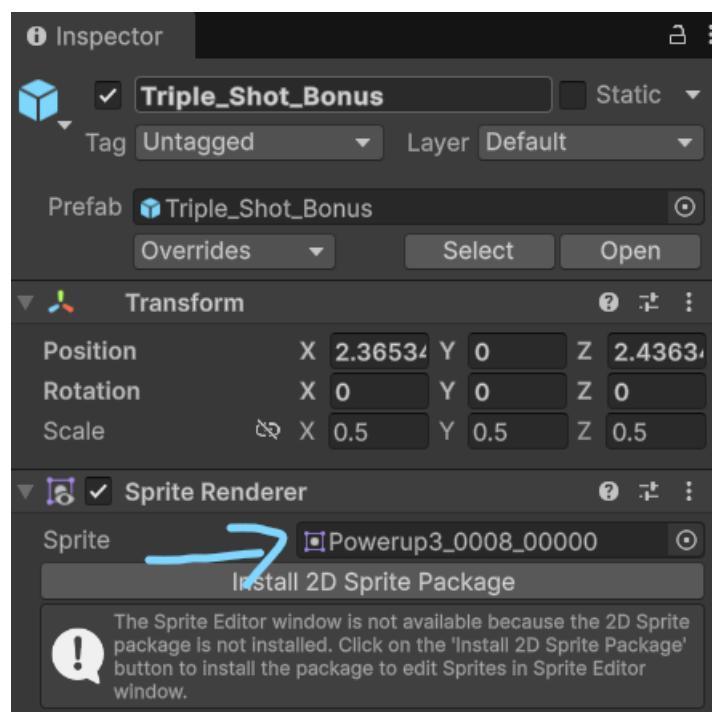


Oyun Programlama Ödev 5

1) Üçlü Atış Bonusu

Hierarchy penceresinde sağ tık yapıp CreateEmpty ile oluşturduğumuz nesneye Triple_Shot_Bonus adı veririz. Add component ile eklediğimiz Sprite Renderer componentinde Sprite kısmına daha önce Assets dosyasına eklediğimiz Sprites assetlerinden Power_ups'ın içindeki en üst asseti sürükleyip bırakırız.



Daha sonra Add Component kısmından Circle Collider 2D ve Rigidbody 2D componentlerini ekleriz. Circle Collider 2D içinde Is Trigger seçeneğini işaretleriz. Bu sayede oyuncu bu nesneyle temas ettiğinde çarpışma yerine bir tetiklenme olayı olur. Edit Collider ile de temas edeceği alanın boyutunu ayarlarız RigidBodyde ise Gravity Scale değerini 0 yaparız. Sonrasında ise Assets klasöründe daha önce oluşturduğumuz Prefabs klasörüne sürükleyip bırakırız ve böylece Prefab haline getiririz.

2)Bonus_sc

Assets dosyasının içindeki Script klasörüne yeni bir script oluşturur ve adına Bonus_sc adını veririz. Kodlarımız ise şu şekildedir.

```
1  using UnityEngine;
2
3  public class Bonus_sc : MonoBehaviour
4  {
5
6      [SerializeField]
7      float speed = 3 ;
8
9      // Update is called once per frame
10     void Update()
11     {
12         this.transform.Translate(Vector3.down * speed * Time.deltaTime);
13
14         if(this.transform.position.y < -5.8f){
15
16             Destroy(this.gameObject);
17         }
18     }
19
20
21
22     void OnTriggerEnter2D(Collider2D other)
23     {
24         if(other.tag == "Player"){           // Player ile mi çarpışıyor?
25
26             //Üçlü atış bonusunu aktifleştir
27             Player_sc player_sc = other.transform.GetComponent<Player_sc>();
28             if(player_sc != null)
29             {
30                 player_sc.TripleShotActive();
31             }
32
33             Destroy(this.gameObject);           //Bonus nesnesini yok et
34
35         }
36     }
}
```

this.transform.Translate(Vector3.down * speed * Time.deltaTime); kodu ile bonus aşağı yönde hareket eder.

```
if(this.transform.position.y < -5.8f)
```

Destroy(this.gameObject); ile ekrandan çıkarsa nesne yok edilir.

```
void OnTriggerEnter2D(Collider2D other)
```

fonksiyonu ile de oyuncu bonusla çarpışlığında player_sc'den TripleShotActive() fonksiyonu çağrılır ve bonus yok edilir.

Triple_Shot_Bonus nesnesi seçiliğinde sürükle bırak ile Bonus_sc eklenir.

3) SpawnManager_Sc İçindeki Değişiklikler

Daha önce kullandığımız SpawnManager_sc içinde değişiklikler yapılır.

```
IEnumerator SpawnBonusRoutine()
{
    while(stopSpawning == false)
    {

        Vector3 position = new Vector3(Random.Range(-9.18f, 9.18f),
                                         7.7f,
                                         0);
        GameObject TripleShotBonus = Instantiate(TripleShotBonusPrefab, position, Quaternion.identity);
        int waitTime = Random.Range(3,8);
        Debug.Log("Üçlü atış bekleme süresi: " +waitTime);
        yield return new WaitForSeconds((float) waitTime);
    }
}
```

while(stopSpawning == false) ile oyuncu ölmemiş sürece bonus spawn olmaya devam eder.

Instantiate(TripleShotBonusPrefab, position, Quaternion.identity);

Bonus prefab'ını sahneye ekler. Bu prefab, Bonus_sc script'i sayesinde aşağı doğru hareket eder ve oyuncuya çarptığında üçlü atıştı aktif eder.

int waitTime = Random.Range(3,8); Bir sonraki bonusun kaç saniye sonra geleceğini belirler.

yield return new WaitForSeconds(waitTime); Coroutine burada belirtilen süre kadar bekler.

Coroutine, Start() fonksiyonundaki StartCoroutine(SpawnBonusRoutine()); ile başlar.

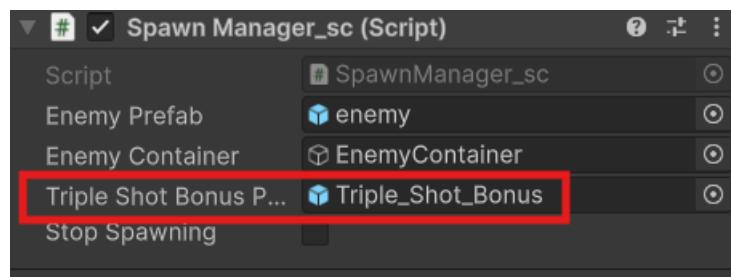
```
void Start()
{
    startCoroutine(SpawnEnemyRoutine());
    StartCoroutine(SpawnBonusRoutine());
}
```

Oyuncu öldüğünde ise

```
public void OnPlayerDeath(){
    stopSpawning = true;
}
```

kodu ile spawn durur.

Hierarchy içinde SpawnManager nesnesinin Inspector penceresinde gerekli prefabların sürüklendiğinde bırakılması unutulmamalıdır.



4) Player_Sc İçindeki Değişiklikler

Daha önce kullandığımız Player_sc içinde de bazı değişiklikler yapılır.

```
void FireLaser()
{
    if (!isTripleShotActive)
    {
        Instantiate(laserPrefab, this.transform.position + new Vector3(0,1.05f,0), Quaternion.identity);
    }
    else
    {
        Instantiate(tripleLaserPrefab, this.transform.position + new Vector3(0,1.05f,0), Quaternion.identity);
    }
}
```

isTripleShotActive false ise normal tek mermi

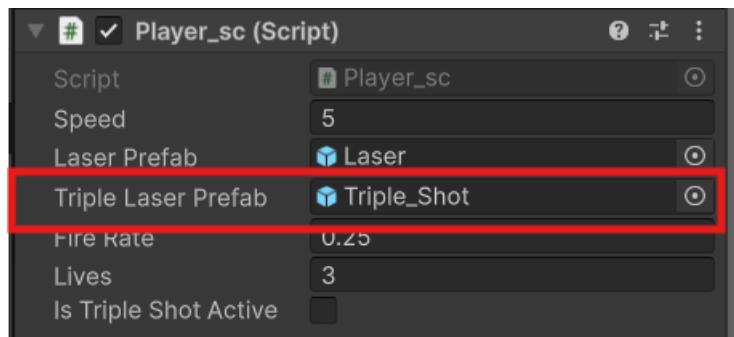
true ise 3 mermi birden yani TripleShot.

```
public void TripleShotActive()          // başka bir scriptten erişecez o yüzden public fonksiyon
{
    isTripleShotActive = true;
    StartCoroutine(TripleShotCancelRoutine());
}

IEnumerator TripleShotCancelRoutine()
{
    yield return new WaitForSeconds(5.0f);
    isTripleShotActive = false;
}
```

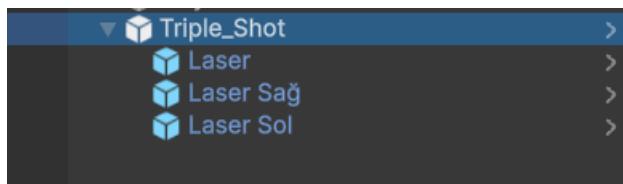
Bonus ile çarpışınca bu fonksiyon çağrırlır. 5 saniye boyunca üçlü atış aktif olur, sonra eski normale döner.

Hierarchy içinde Player nesnesinin Inspector penceresinde gerekli prefabların sürüklendiğinde bırakılması unutulmamalıdır.

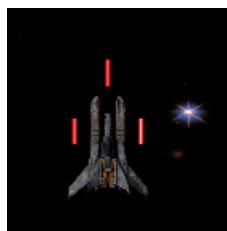


5)TripleShot

Hierarchy penceresinde sağtık yapıp oluşturduğumuz CreateEmpty nesnesine TripleShot adını veririz. TripleShot nesnesinin üçlü atış yapacağımız için 3 tane laser prefabı atarız.



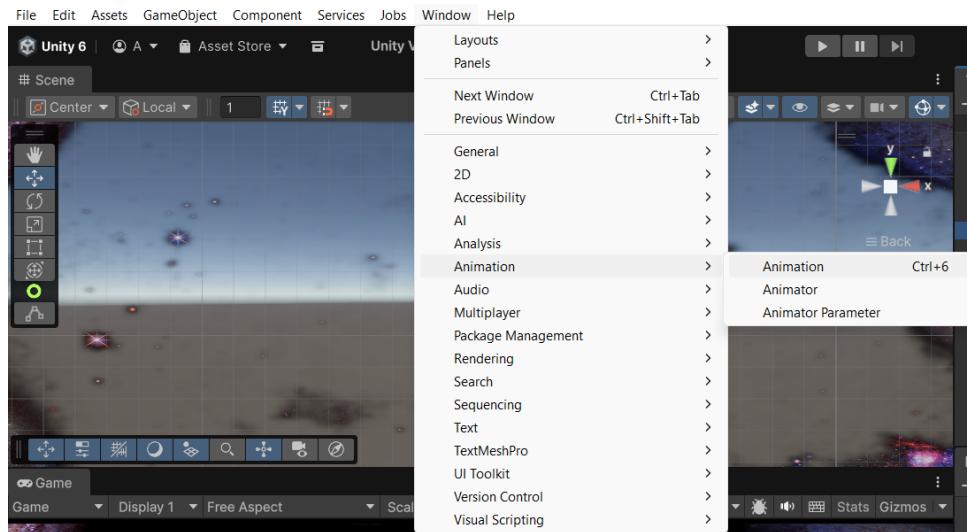
Eklediğimiz Laserler, laser prefabı özelliği taşıdığı için zaten Box Collider 2D componenti vardı ayrıca eklememize gerek yoktur. Daha sonra Player nesnesinde laserleri playerın neresinden çıkışmasını istiyorsak ona göre konumunu güncelleriz.



6)Animasyon Ekleme

Triple_Shot_Bonus prefabini Hierarchy kismına rürkleyip bırakılır.

Triple_Shot_Bonus seçiliyken Window → Animation → Animation menüsüne tıklanır.

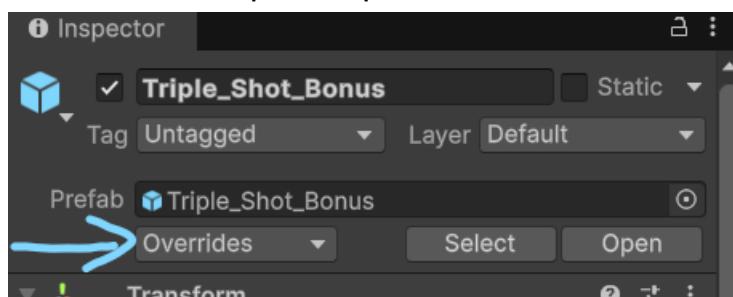


Animation paneli açıldığında prefab seçiliyken Create butonuna basın.

Animasyon dosyasını Assets > Animations klasörüne kaydedilir.

Animation panelinde artık bir timeline var. Assets panelinden animasyonda kullanmak istediğiniz sprite'ları sürükleyip timeline'a bırakın.

Daha sonra Inspector penceresinde



Overrides seçenekinde

Apply all dememiz lazımlı aksi halde Sadece seçili değişikliği Prefab dosyasına uygular. Apply All ile sahnedeki tüm değişiklikleri Prefab dosyasına uygular.

Script Dosyalarına Github Hesabımıdan Ulaşabilirsiniz:

<https://github.com/AsmBrk/Oyun-Programlama-Lab/tree/main/%C3%96dev%205>