

BLM0364 OYUN PROGRAMLAMA – HAFTA 4 RAPORU

Yasin Ekici

21360859029

Rapor İeriği;

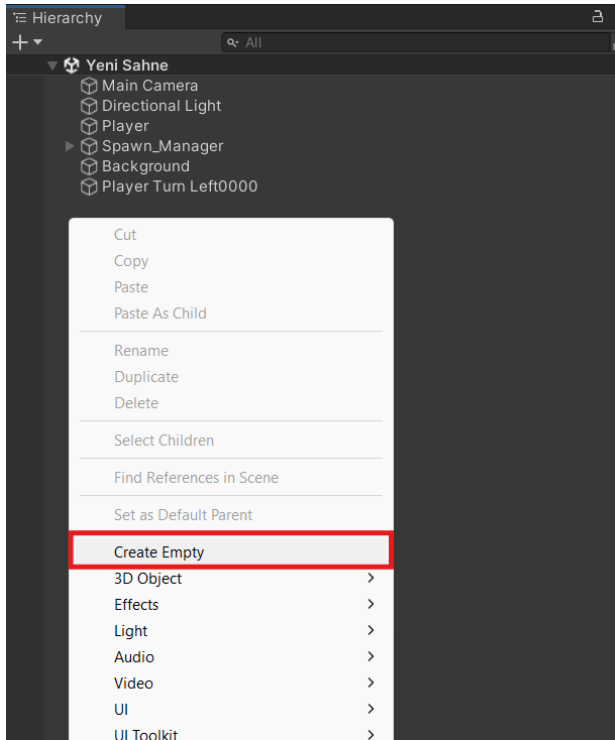
- Spawn Manager boş oyun nesnesi ve script'i oluşturma
- Spawn Manager script'i ile her 5 saniyede bir enemy üretme (spawn)
- CoRoutine kullanımı
- Çok sayıda Enemy oluşacağı için bunların bir parent container oyun nesnesi için toplanması
- Oyuncu yok olduğunda spawn işleminin durdurulması için fonksiyon yazılması
- Bu fonksiyonun Player_sc sınıfından oyuncu yok olduğunda çağırılması

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

1. Spawn Manager boş oyun nesnesi ve script'i oluşturma:

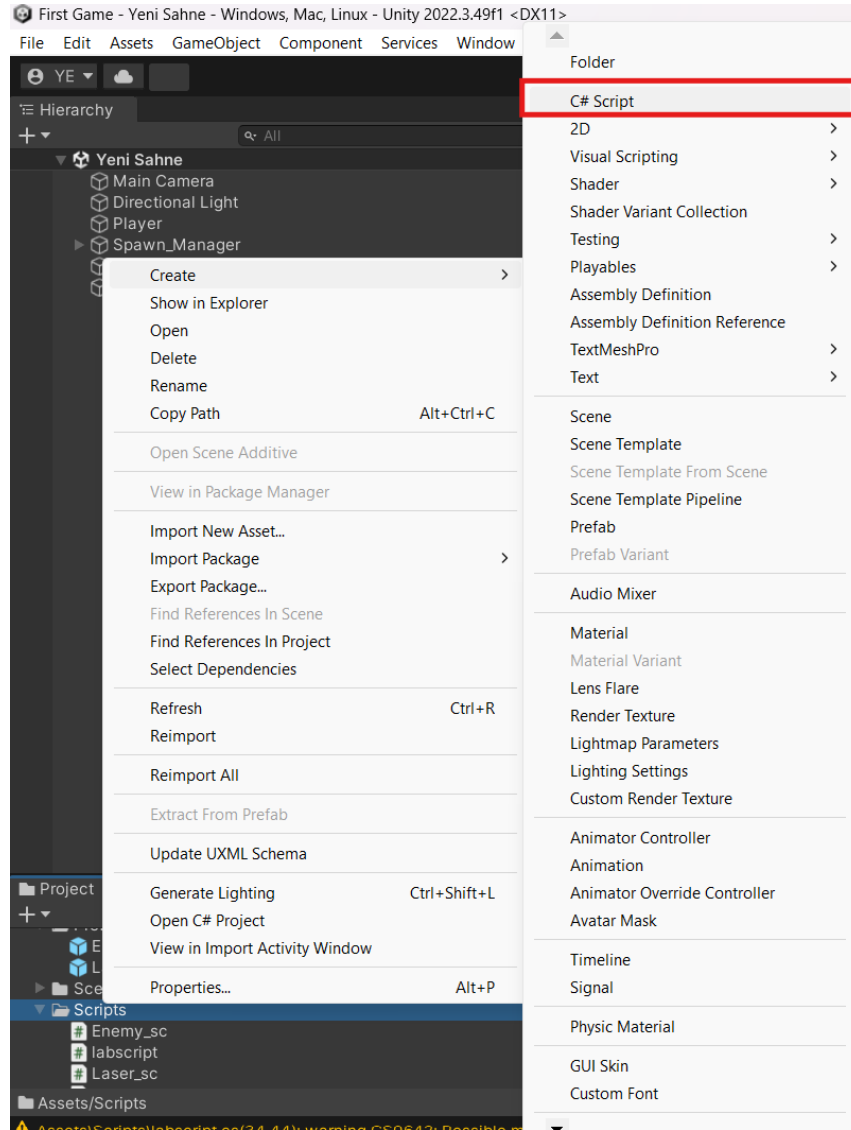
Spawn_Manager Nesnesi Oluşturma:

- Unity'de Hierarchy sekmesine sağ tıklayın, Create Empty seçeneğine tıklayın.
- Bu nesneye "Spawn_Manager" adını verin. Bu nesne, düşman karakterlerinin (Enemy) oyunda üretileceği yönetici nesne olacak.



SpawnManager_sc Script'i Oluřturma:

- Project sekmesinde Scripts klasörüne sağ tııklayın ve Create > C# Script seçeneğine tııklayın.
- Script'in adını "SpawnManager_sc" olarak belirleyin.
- Script'i oluřturduktan sonra, bu script'i "Spawn_Manager" oyun nesnesine sürükleyip bırakın.
- Not: Bu script, düşmanları belli aralıklarla üretme (spawn) işini yürütecek.



2. Spawn Manager ile enemy üretme ve Coroutine kullanımı:

- Unity'de olayları nasıl geciktirebiliriz?
- Bir fonksiyon çağırdığınızda, işlemi tamamlanana kadar çalışır ve sonrasında geri döner.
- Bir fonksiyonda gerçekleşen herhangi bir işlem, tek bir kare güncellemesi içinde gerçekleşmelidir.
- Bir nesnenin opaklığını (şeffaflığını) nasıl kademeli olarak azaltabiliriz?

Coroutine

- Bir fonksiyon gibi çalışır ancak yürütmeyi duraklatıp kontrolü Unity'e geri verme yeteneğine sahiptir.
- Sonraki karede kaldığı yerden devam eder.

Spawn Manager

- Düşmanlar "Enemy" prefab'ından instantiate edilir.
- "SpawnManager_sc" içinde prefab değişkenini tanımlayın.
- Hierarchy penceresinden "Enemy" nesnesini silin.
- "SpawnManager_sc" script'inin "Enemy" prefab değişkenini ayarlayın.

[SerializeField]

`GameObject enemyPrefab;`

Enemy pozisyonunu belirleyin:

- x eksenini -8 ile 8 arasında rastgele bir değere ayarlayın.
- y eksenini ekranın üst kısmına ayarlanır.
- "Enemy" nesnesini instantiate edin.
- Başka bir nesne oluşturmadan önce 5 saniye bekleyin.

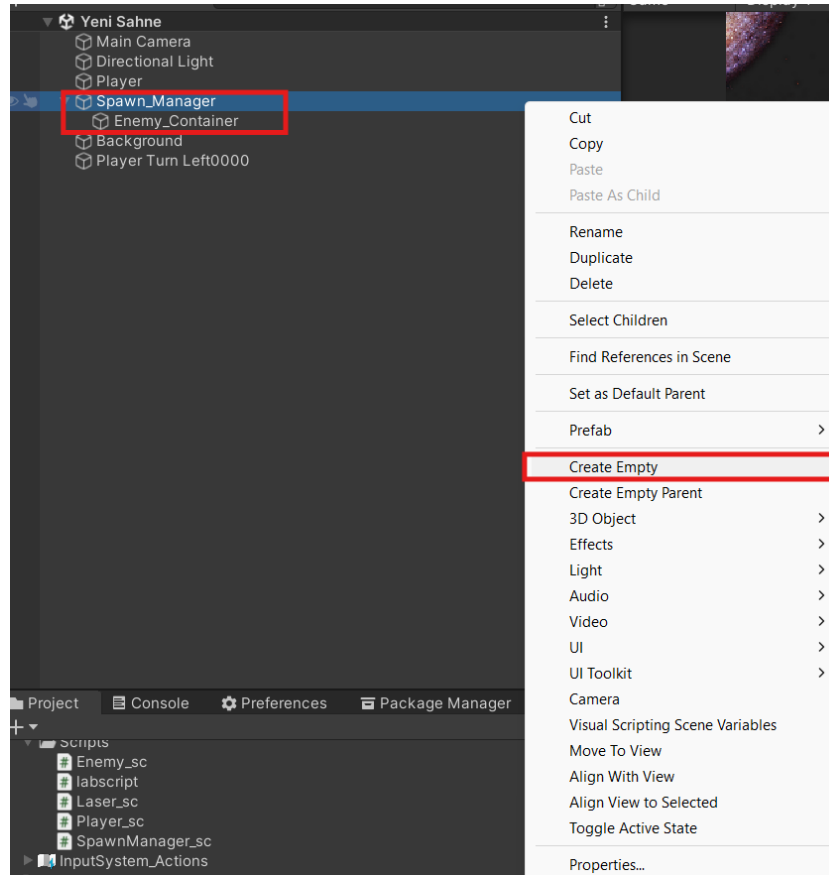
```
IEnumerator SpawnRoutine(){  
    while(true){  
        Vector3 position = new Vector3(Random.Range(-9.4f, 9.4f), 7, 0);  
        Instantiate(enemyPrefab, position, Quaternion.identity);  
        yield return new WaitForSeconds(5.0f);  
    }  
}
```

Coroutinesleri çağırmanın bir yolu:

- Start fonksiyonu içinde;
`StartCoroutine(SpawnRoutine());`

3. Enemy nesnelerinin sayısı Hierarchy'de artıyor:

- Daha iyi debugging için Enemy nesnelerini düzenleyin.
- Enemy nesnelerini Spawn_Manager'ın alt nesnesi yapın.
- Spawn_Manager altında, Enemy nesneleri için bir container olarak boş bir nesne oluşturun, adı Enemy_Container olsun.



- SpawnManager_cs içinde bu container için bir değişken oluşturun.
GameObject enemyContainer;
- Enemy_Container nesnesini bu değişkenin üzerine sürükleyin.
- Enemy nesnelerini Enemy_Container'ın alt nesnesi olarak ayarlayın.
- Yeni Enemy nesnelerinin parent'ını Enemy_Container olarak atayın.

```
IEnumerator SpawnRoutine(){
```

```
while(true){
```

```
Vector3 position = new Vector3(Random.Range(-9.4f, 9.4f), 7, 0);
```

```
GameObject newEnemy = Instantiate(enemyPrefab, position,  
Quaternion.identity);
```

```
newEnemy.transform.parent = enemyContainer.transform;
```

```
yield return new WaitForSeconds(5.0f);
```

```
}  
}
```

Player öldüğünde yeni düşman nesnelerinin üretilmesini durdurun:

- Player_sc ile SpawnManager_sc arasında script iletişimi kurun.
- Üretim işlemini durdurmamız gerekip gerekmediğini kontrol etmek için SpawnManager_sc içinde bir Boolean değişken oluşturun.

```
bool stopSpawning = false;
```

- Spawn Coroutine içindeki kodu güncelleyin;

```
IEnumerator SpawnRoutine(){  
    while(stopSpawning == false){  
        Vector3 position = new Vector3(Random.Range(-9.4f, 9.4f), 7, 0);  
        GameObject newEnemy = Instantiate(enemyPrefab, position,  
            Quaternion.identity);  
        newEnemy.transform.parent = enemyContainer.transform;  
        yield return new WaitForSeconds(5.0f);  
    }  
}
```

- Boolean kontrol değişkenini true olarak ayarlamak için bir public fonksiyon tanımlayın.
- Böylece sonsuz while döngüsünden çıkın.

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

4. Player_sc'nin iletişim kurabilmek için SpawnManager_sc'yi bulması gerekiyor:

- Player_sc içinde spawn manager için bir değişken oluşturun.
- Find fonksiyonunu kullanarak SpawnManager_sc script'ini bulun.
- İlgili bileşeni bulmak için GetComponent fonksiyonunu kullanın.
- Player_sc içinde;

```
SpawnManager_sc spawnManager_sc;
```
- Şimdi start fonksiyonu içerisinde;

```
spawnManager_sc=GameObject.Find("Spawn_Manager").GetComponent<SpawnM  
anager_sc>();  
    if(spawnManager_sc == null){  
        Debug.LogError("Spawn_Manager oyun nesnesi bulunamadi");  
    }
```

- Şimdi Player_sc içerisinde Damage fonksiyonunu güncelleyelim:

```
public void Damage(){  
    lives--;  
    if(lives < 1){  
        if (spawnManager_sc != null){  
            spawnManager_sc.OnPlayerDeath();  
            Destroy(this.gameObject);  
        }  
    }  
}
```