

Orijinal Kaynak: <http://forum.unity3d.com/threads/unity-lesson-1-draft.103421/>

# Unity 3D İle Oyun Programlama

## Bölüm 12: Düşmanların Spawn Olması



**ÇEVİRİ: Süleyman Yasir KULA**

<http://yasirkula.com>

Ferdinand Joseph Fernandez

Chief Technological Officer, Dreamlords Digital  
Inc.

Admin and Co-founder, Unity Philippines Users  
Group

September 2011



All code snippets are licensed under CC0 (public domain)



This document except code snippets is licensed with  
Creative Commons Attribution-NonCommercial 3.0 Unported

Şu anda oyunda sadece üç zombi var. Onları öldürünce tüm olay bitiyor. Peki ya zamanla yeni zombiler spawn edersek (oluşturursak) nasıl olur?

## Düşmanın Spawn Olması

Basit bir script yazarak işe başlayalım. Her 10 saniyede bir sahnedeki bir spawn objesinden bir zombi spawn olsun. “EnemySpawnManager” adında yeni bir script oluşturun:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     void Update()
10     {
11         Instantiate(_enemyToSpawn);
12     }
13 }
14
```

Şimdi de sahnede “EnemySpawnManager” adında yeni bir empty gameobject oluşturun.

EnemySpawnManager scriptini objeye verin. Inspector'daki “Enemy To Spawn” a değer olarak Project panelinden Enemy prefab'ını verin. Oyunu çalıştırın (**bence çalıştırmayın, yoksa Unity birkaç saniye içinde takılmaya başlayacak**).

Düşmanların çok hızlı bir şekilde spawn olduğunu göreceksiniz.

Nasıl düşmanın iki saldırısı arasına Time.time kullanarak bir gecikme koyduysak spawn sistemine de aynısını yapalım.



Görsel 12.1: Bu kadarı da çok fazla!

Kodu şöyle güncelleyin:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     [SerializeField]
10     float _spawnDelay = 1.0f;
11
12     float _nextSpawnTime = -1.0f;
13
14     void Update()
15     {
16         if (Time.time >= _nextSpawnTime)
17         {
18             Instantiate(_enemyToSpawn);
19             _nextSpawnTime = Time.time + _spawnDelay;
20         }
21     }
22 }
23
```

Artık iki spawn arasında \_spawnDelay kadar saniye geçecek.

## Spawn Sistemini Geliştirmek

Şu anda zombilerin spawn olduğu yer belli ve hep oradan spawn oluyorlar. Bu çok güzel görünmüyor.

Sistemi biraz daha geliştirerek zombilerin bulunduğumuz konumun yakınında rastgele bir yerde spawn olmasını sağlayalım.

Aşama aşama gideceğiz. Önce kodu şöyle değiştirin:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     [SerializeField]
10     float _spawnDelay = 1.0f;
11
12     float _nextSpawnTime = -1.0f;
13
14     void Update()
15     {
16         if (Time.time >= _nextSpawnTime)
17         {
18             Vector3 placeToSpawn = new Vector3(0, 0.75f, 0);
19             Quaternion directionToFace = Quaternion.identity;
```

```

20         Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
21         _nextSpawnTime = Time.time + _spawnDelay;
22     }
23 }
24 }
25

```

Instantiate fonksiyonuna iki yeni parametre ekledik: placeToSpawn ile düşmanın 3 boyutlu uzayda hangi konumda (position) spawn olacağını belirlerken directionToFace ile de düşmanın spawn olduğu anda ki eğimini (rotation) belirledik.

Şu anda düşman (0, 0.75, 0) konumunda spawn oluyor. Bu değeri değiştirerek düşmanın rastgele bir yerde spawn olmasını sağlayacağız.

Eğim (rotation) olarak ise Quaternion.identity değerini kullanıyoruz. Bunun anlamı düşmanın doğduğu anda Euler açı cinsinden (0,0,0) rotasyonuna sahip olacağıdır.

Kodu güncellemeye devam edelim:

```

01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     [SerializeField]
10     float _spawnDelay = 1.0f;
11
12     float _nextSpawnTime = -1.0f;
13
14     void Update()
15     {
16         if (Time.time >= _nextSpawnTime)
17         {
18             Vector3 edgeOfScreen = new Vector3(1.0f, 0.5f, 8.0f);
19             Vector3 placeToSpawn = Camera.main.ViewportToWorldPoint(edgeOfScreen);
20             Quaternion directionToFace = Quaternion.identity;
21             Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
22             _nextSpawnTime = Time.time + _spawnDelay;
23         }
24     }
25 }
26

```

Oyunu çalıştırın. Zombiler ekranın sağ kenarının tam ortasından spawn olacaklar ve kamerayı döndürseniz bile zombiler yine ekranın sağından spawn olmaya devam edecekler.

Bu güzel sistemin nasıl işlediğini anlamak için önce viewport space'i anlamalıyız.

## Viewport Space

Daha önce local space (yerel uzay) ile world space'i (global uzay) gördük. Bir başka space türü ise viewport space. Diğer iki uzayın aksine viewport space 3D düzlemde değil 2D düzlemde çalışır.

Viewport space, ekranınızı temsil eden bir uzaydır. Bu space'in X ve Y koordinatları vardır. Bu uzaydaki bir vektörün X koordinatı 0.0 olunca ekranın en solu, 0.5 olunca ekranın ortası, 1.0 olunca ekranın en sağı kastedilir. Benzer şekilde, vektörün Y koordinatı 0.0 olunca ekranın en altı, 0.5 olunca ekranın ortası, 1.0 olunca ekranın en üstü kastedilir.

Yani Viewport Space'te (0.0, 0.0) ekranın sol alt köşesini, (0.5, 0.5) vektörü ekranın tam ortasını, (1.0, 1.0) vektörü ise ekranın sağ üst köşesini temsil eder. (1.0, 0.5) ise ekranın yatay ekseninde en sağını, dikey ekseninde de orta noktasını temsil eder. Yani ekranın sağ-orta noktasını temsil eder.

placeToSpawn'a değer verirken artık Camera class'ının ViewportToWorldPoint fonksiyonunu kullanıyoruz. Bu fonksiyon viewport space'indeki bir noktayı world space'indeki bir noktaya çevirir.

Peki ama 2D uzaydan 3D uzaya çeviri nasıl olur? Sonuçta 3D uzayda bir derinlik var. Bu derinlik için de fonksiyonun aldığı Vector3 parametresinin (edgeOfScreen) Z değeri kullanılıyor.

Vector3'ün X ve Y değerleri viewport space'teki bir noktayı temsil ederken Z değeri ise bu noktayı world space'e çevirirken elde ettiğimiz noktanın kameradan kaç birim uzakta olacağını belirliyor.

Fonksiyona parametre olarak verdiğimiz edgeOfScreen vektörünün değeri (1.0, 0.5, 8.0). Dediğim gibi, X ve Y koordinatları viewport space'teki bir noktayı temsil ediyor: (1.0, 0.5). Yani ekranın sağ-orta noktası. Z koordinatına değer olarak 8.0 verdik, bunun anlamı da elimizdeki noktayı viewport space'ten world space'e çevirince elde ettiğimiz noktanın kameraya uzaklığı 8.0 birim olacak.

Instantiate fonksiyonu düşmanı placeToSpawn noktasında oluşturduğu için de kamerayı ne kadar çevirirsek çevirelim yeni düşmanlar hep kameranın sağ-orta noktasında, kameradan 8.0 birim uzakta spawn ediliyor.

## Spawn Noktasını Geliştirmek

edgeOfScreen'in X koordinatı 1.0 olduğu için zombiler ekranın sağ kenarında spawn oluyor. Ancak böyle olunca onların orada \*puf\* diye spawn olduklarını kolayca görebiliyoruz. Bu X koordinatını öyle bir değiştirmeliyiz ki zombilerin \*puf\* diye spawn olduklarını görmemeliyiz. Yani zombiler ekranın dışında bir yerde spawn olmalılar.

Neyse ki bu çok basit. Viewport space'te X koordinatı 0.0 ile 1.0 olacak diye bir kural yok. Eğer X koordinatı 2.0 olursa bu ekranın sağ kenarından, ekranın genişliği kadar daha sağda olan görünmez bir noktayı temsil eder.

Eğer edgeOfScreen'in X koordinatını 1.25 olarak değiştirirsek bu zaman da zombilerin spawn olacağı nokta ekranın sağ kenarından, ekranın genişliğinin 1/4'ü kadar daha uzakta olacak ve onların \*puf\* diye aniden spawn olduğunu görmeyeceğiz:

```
18 Vector3 edgeOfScreen = new Vector3(1.25f, 0.5f, 8.0f);
```

Artık zombileri spawn olurken değil, spawn olduktan sonra bize doğru hareket ederken göreceğiz.

Böylesi çok daha iyi. Ama bununla yetinmek bize yakışmaz. edgeOfScreen'in Y koordinatı ile de biraz oynayalım:

```
18 Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
```

Hatırlayın, Random.value değeri 0.0 ile 1.0 arasında rastgele bir sayı döndürür. Bu sayıyı viewport space'in Y eksenine değer olarak veriyoruz ve böylece zombiler ekranın sadece sağ-orta noktasından değil ama sağ-alt ve sağ-üst noktalarından da (sağ kenardaki herhangi bir noktadan) spawn olabiliyor.

Bazen birkaç saniye beklediğiniz halde bir zombinin spawn olmadığını göreceksiniz. Çünkü kamera açısına bağlı olarak zombilerin spawn noktası bazen yerin altında oluyor ve zombiler spawn oldukları gibi aşağı düşmeye başlıyorlar. Veya zeminin bir kenarına gelip boşluğu ekranın sağ tarafına alırsanız da zombilerin boşlukta spawn olup aşağı düştüğünü görebilirsiniz. Bu gayet ciddi bir sorun. Zombilerin her zaman için zeminin üzerinde spawn olduklarından emin olmalıyız.

Bu işlem için raycast sistemine başvuracağız:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     [SerializeField]
10     float _spawnDelay = 1.0f;
11
12     float _nextSpawnTime = -1.0f;
13
14     void Update()
15     {
16         if (Time.time >= _nextSpawnTime)
17         {
18             Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
19             Ray ray = Camera.main.ViewportPointToRay(edgeOfScreen);
20             RaycastHit hit;
21             if (Physics.Raycast(ray, out hit))
22             {
23                 Vector3 placeToSpawn = hit.point;
24                 Quaternion directionToFace = Quaternion.identity;
25                 Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
26                 _nextSpawnTime = Time.time + _spawnDelay;
27             }
28         }
29     }
30 }
31
```

Viewport space'teki bir noktayı direkt world space'teki bir noktaya çevirmek yerine artık o noktadan çıkan bir raycast ışını (ray) oluşturuyoruz. Eğer bu raycast ışını bir yere temas ederse (if koşulunun içine girersek) zombiyi bu temas edilen noktada (hit.point) oluşturuyoruz. Yok eğer raycast ışını bir yere çarpmazsa düşman spawn etmiyoruz.

**ÇEVİRMEN EKLEMESİ:** Artık edgeOfScreen'in Z koordinatı olan 8.0 değeri önemsiz oldu. Çünkü bir Raycast ışınının "derinliği" gibi birşey söz konusu değil.

Düşmanlar hep ekranın sağ kenarından spawn oluyor. Bazen de ekranın sol kenarından spawn olmaları güzel olurdu. Çok ufak bir değişiklik ile bunu yapmak mümkün:

```

01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     [SerializeField]
10     float _spawnDelay = 1.0f;
11
12     float _nextSpawnTime = -1.0f;
13
14     void Update()
15     {
16         if (Time.time >= _nextSpawnTime)
17         {
18             Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
19             if (Random.value > 0.5f)
20             {
21                 edgeOfScreen.x = -0.25f;
22             }
23
24             Ray ray = Camera.main.ViewportPointToRay(edgeOfScreen);
25             RaycastHit hit;
26             if (Physics.Raycast(ray, out hit))
27             {
28                 Vector3 placeToSpawn = hit.point;
29                 Quaternion directionToFace = Quaternion.identity;
30                 Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
31                 _nextSpawnTime = Time.time + _spawnDelay;
32             }
33         }
34     }
35 }
36

```

Random.value değerini tekrar çağırıyoruz ve eğer bu değer 0.5'ten büyükse (%50 olasılık) edgeOfScreen'in X koordinatını -0.25 yapıyoruz. Bu da viewport uzayda ekranın sol kenarından ekranın genişliğinin 1/4'ü kadar soldaki görünmez bir noktayı temsil ediyor.



# Zombilerin Birbirinin Üzerinde Spawn Olmasını Önlemek

Bazen zombiler birbiri üzerinde spawn olabilmekte:



Eğer raycast ışını bir zombinin kafasına denk gelirse böyle istenmeyen bir sonuç oluşuyor. Bizim istediğimiz şey raycast ışınının zombileri es geçmesi; yani onlara temas etse bile onların içinden geçip yoluna devam etmesi.

Bunun için birkaç ufak değişiklik lazım:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     [SerializeField]
07     GameObject _enemyToSpawn;
08
09     [SerializeField]
10     float _spawnDelay = 1.0f;
11
12     float _nextSpawnTime = -1.0f;
13
14     [SerializeField]
15     LayerMask _spawnLayer;
16
17     void Update()
18     {
19         if (Time.time >= _nextSpawnTime)
20         {
21             Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
22             if (Random.value > 0.5f)
23             {
24                 edgeOfScreen.x = -0.25f;
25             }
26         }
27     }
28 }
```



```

26
27 Ray ray = Camera.main.ViewportPointToRay(edgeOfScreen);
28 RaycastHit hit;
29 if (Physics.Raycast(ray, out hit, Mathf.Infinity, _spawnLayer.value))
30 {
31     Vector3 placeToSpawn = hit.point;
32     Quaternion directionToFace = Quaternion.identity;
33     Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
34     _nextSpawnTime = Time.time + _spawnDelay;
35 }
36 }
37 }
38 }
39

```

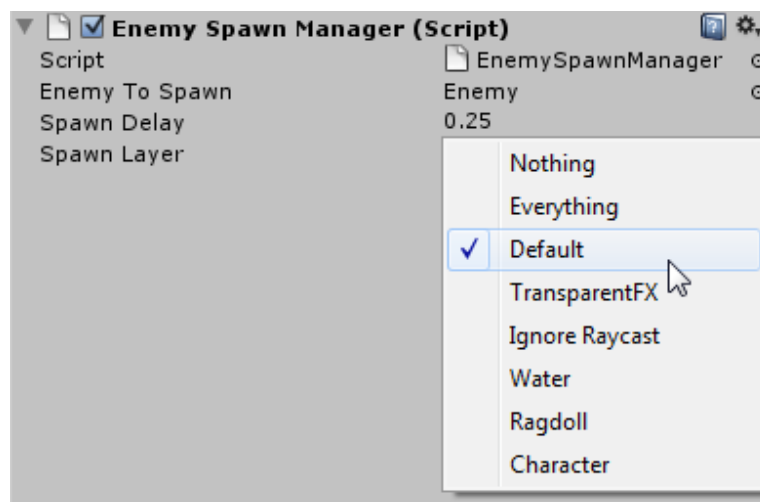
Unity'de *layer mask* olarak bilinen bir değişken türünden faydalaniyoruz. Bu değişkeni kullanarak raycast ışınının hangi layer'daki objeleri görmezden gelip hangilerine temas edebileceğini belirleyebiliyoruz.

15. satırda LayerMask değişkenimizi oluşturuyoruz.

29. satırda ise bu değişkeni kullanıyoruz. Maalesef Unity'de Raycast fonksiyonuna LayerMask verecekseniz öncesinde o raycast ışınının uzunluğunu parametre olarak girmelisiniz. Biz ışının 2-3 metre olmasını istemiyor, sonsuza kadar gitmesini istiyoruz. Bunun için ışının uzunluğunu `Mathf.Infinity` (sonsuz) olarak belirliyoruz.

EnemySpawnManager'ın Inspector'unda Spawn Layer adında bir değişken göreceksiniz. Bu bizim Layer Mask'ımız. Varsayılan değeri "Nothing" (hiçbir şey). Yani raycast ışını hiçbir layer ile temasa geçmiyor.

Bizim istediğimiz ise sadece zemin ile temasa geçmesi. Zemin objemiz (Ground) "Default" layer'ında yer alırken zombi objesi "Character" layer'ında yer alıyor. Yani Layer Mask'e değer olarak sadece "Default" verirsek raycast ışını sadece zemin ile temas edecek. Bu tam da istediğimiz şey!



## Zombi Sayısını Sınırlamak

Oyunu biraz oynadıktan sonra ekranda çok fazla zombi birikmekte ve bu da oyunun takılmasına yol açmakta. Oyunda aynı anda maksimum kaç tane zombinin yer alabileceğini sınırlamalıyız.



Görsel 12.2: Çok fazla zombi var, çok!

Örneğin zombi sayısı 30 ise daha fazla zombi spawn olmasın, ta ki bu 30 zombiden en azından birini öldürene kadar.

Elbette ki bunu başarmak için henüz ölmemiş olan zombilerin sayısını bir değişkende tutmalıyız.

EnemySpawnManager scriptini güncelleyin:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     static int _livingZombies = 0;
07
08     [SerializeField]
09     GameObject _enemyToSpawn;
10
11     [SerializeField]
12     float _spawnDelay = 1.0f;
```

```

13
14 float _nextSpawnTime = -1.0f;
15
16 [SerializeField]
17 LayerMask _spawnLayer;
18
19 void Update()
20 {
21     if (Time.time >= _nextSpawnTime && _livingZombies < 30)
22     {
23         Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
24         if (Random.value > 0.5f)
25         {
26             edgeOfScreen.x = -0.25f;
27         }
28
29         Ray ray = Camera.main.ViewportPointToRay(edgeOfScreen);
30         RaycastHit hit;
31         if (Physics.Raycast(ray, out hit, Mathf.Infinity, _spawnLayer.value))
32         {
33             Vector3 placeToSpawn = hit.point;
34             Quaternion directionToFace = Quaternion.identity;
35             Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
36             _nextSpawnTime = Time.time + _spawnDelay;
37             ++_livingZombies;
38         }
39     }
40 }
41 }
42

```

\_livingZombies adında static bir değişkenimiz var ve yeni zombi oluşturdukça bu değişkenin değerini 1 artırıyoruz: "++\_livingZombies". Eğer bu değişken 30'dan küçükse ancak o zaman spawn işlemini gerçekleştiriyoruz.

Şimdi bir zombi ölünce \_livingZombies değişkeninin değerini 1 azaltmalıyız:

```

01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     static int _livingZombies = 0;
07     static public void OnEnemyDeath()
08     {
09         --_livingZombies;
10     }
11
12     [SerializeField]
13     GameObject _enemyToSpawn;
14
15     [SerializeField]
16     float _spawnDelay = 1.0f;
17
18     float _nextSpawnTime = -1.0f;
19
20     [SerializeField]
21     LayerMask _spawnLayer;

```

```

22
23 void Update()
24 {
25     if (Time.time >= _nextSpawnTime && _livingZombies < 30)
26     {
27         Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
28         if (Random.value > 0.5f)
29         {
30             edgeOfScreen.x = -0.25f;
31         }
32
33         Ray ray = Camera.main.ViewportPointToRay(edgeOfScreen);
34         RaycastHit hit;
35         if (Physics.Raycast(ray, out hit, Mathf.Infinity, _spawnLayer.value))
36         {
37             Vector3 placeToSpawn = hit.point;
38             Quaternion directionToFace = Quaternion.identity;
39             Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
40             _nextSpawnTime = Time.time + _spawnDelay;
41             ++_livingZombies;
42         }
43     }
44 }
45 }
46

```

OnEnemyDeath fonksiyonu static bir fonksiyon ve çağırıldığı zaman \_livingZombies değişkenini 1 azaltmaya yarıyor. Artık yapmamız gereken bir zombi ölünce bu fonksiyonu çağırmak.

Health scriptini açıp şu şekilde güncelleyin:

```

01 using UnityEngine;
02 using System.Collections;
03
04 public class Health : MonoBehaviour
05 {
06     [SerializeField]
07     int _maximumHealth = 100;
08
09     int _currentHealth = 0;
10
11     override public string ToString()
12     {
13         return _currentHealth + " / " + _maximumHealth;
14     }
15
16     public bool IsDead { get { return _currentHealth <= 0; } }
17
18     void Start()
19     {
20         _currentHealth = _maximumHealth;
21     }
22
23     public void Damage(int damageValue)
24     {
25         _currentHealth -= damageValue;
26
27         if (_currentHealth < 0)

```

```

28     {
29         _currentHealth = 0;
30     }
31
32     if (_currentHealth == 0)
33     {
34         Animation a = GetComponentInChildren<Animation>();
35         a.Stop();
36
37         if (tag == "Player")
38         {
39             Destroy(GetComponent<PlayerMovement>());
40             Destroy(GetComponent<PlayerAnimation>());
41             Destroy(GetComponent<RifleWeapon>());
42         }
43         else // its an enemy
44         {
45             EnemySpawnManager.OnEnemyDeath();
46             Destroy(GetComponent<EnemyMovement>());
47             Destroy(GetComponentInChildren<EnemyAttack>());
48         }
49
50         Destroy(GetComponent<CharacterController>());
51
52         Ragdoll r = GetComponent<Ragdoll>();
53         if (r != null)
54         {
55             r.OnDeath();
56         }
57     }
58 }
59 }
60

```

Player objemize daha önceden tag olarak "Player" vermiştik. Scriptte bu tag'ı kontrol ediyoruz ve eğer Health scriptinin atandığı objenin Tag'ı "Player" ise Player'ın ilgili component'lerini siliyoruz; değilse (o halde bu bir zombi objesidir) zombinin ilgili component'lerini siliyoruz ve EnemySpawnManager scriptinin static olan OnEnemyDeath fonksiyonunu çağırıyoruz.

Hazır elimiz değmişken maksimum zombi sayısını direkt 30 olarak belirlemeyelim ama bir değişken ile kontrol edelim. EnemySpawnManager'a şu eklemeleri yapın:

```

01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemySpawnManager : MonoBehaviour
05 {
06     static int _livingZombies = 0;
07     static public void OnEnemyDeath()
08     {
09         --_livingZombies;
10     }
11
12     [SerializeField]

```

```

13  GameObject _enemyToSpawn;
14
15  [SerializeField]
16  float _spawnDelay = 1.0f;
17
18  [SerializeField]
19  int _enemyLimit = 30;
20
21  float _nextSpawnTime = -1.0f;
22
23  [SerializeField]
24  LayerMask _spawnLayer;
25
26  void Update()
27  {
28      if (Time.time >= _nextSpawnTime && _livingZombies < _enemyLimit)
29      {
30          Vector3 edgeOfScreen = new Vector3(1.25f, Random.value, 8.0f);
31          if (Random.value > 0.5f)
32          {
33              edgeOfScreen.x = -0.25f;
34          }
35
36          Ray ray = Camera.main.ViewportPointToRay(edgeOfScreen);
37          RaycastHit hit;
38          if (Physics.Raycast(ray, out hit, Mathf.Infinity, _spawnLayer.value))
39          {
40              Vector3 placeToSpawn = hit.point;
41              Quaternion directionToFace = Quaternion.identity;
42              Instantiate(_enemyToSpawn, placeToSpawn, directionToFace);
43              _nextSpawnTime = Time.time + _spawnDelay;
44              ++_livingZombies;
45          }
46      }
47  }
48 }
49

```

## Zombi Cesetlerini Ortadan Kaldırmak

Yaşayan zombi sayısını sınırladık ama ölen zombilerin cesetleri yerde birikiyor ve zamanla sahnede çok fazla 3D obje olduğu için oyun yine takılmaya başlıyor. Ölen zombilerin cesetlerini sahneden kaldırmalıyız.





Görsel 12.3: Cesetler oyunun performansını hiç de iyi etkilemiyor.

Health scriptini şöyle güncelleyin:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class Health : MonoBehaviour
05 {
06     [SerializeField]
07     int _maximumHealth = 100;
08
09     int _currentHealth = 0;
10
11     override public string ToString()
12     {
13         return _currentHealth + " / " + _maximumHealth;
14     }
15
16     public bool IsDead { get { return _currentHealth <= 0; } }
17
18     void Start()
19     {
20         _currentHealth = _maximumHealth;
21     }
22
23     public void Damage(int damageValue)
24     {
25         _currentHealth -= damageValue;
26
27         if (_currentHealth < 0)
28         {
29             _currentHealth = 0;
30         }
31     }
32 }
```



```

31
32     if (_currentHealth == 0)
33     {
34         Animation a = GetComponentInChildren<Animation>();
35         a.Stop();
36
37         if (tag == "Player")
38         {
39             Destroy(GetComponent<PlayerMovement>());
40             Destroy(GetComponent<PlayerAnimation>());
41             Destroy(GetComponent<RifleWeapon>());
42         }
43         else // its an enemy
44         {
45             EnemySpawnManager.OnEnemyDeath();
46             Destroy(GetComponent<EnemyMovement>());
47             Destroy(GetComponentInChildren<EnemyAttack>());
48
49             Destroy(gameObject, 8.0f);
50         }
51
52         Destroy(GetComponent<CharacterController>());
53
54         Ragdoll r = GetComponent<Ragdoll>();
55         if (r != null)
56         {
57             r.OnDeath();
58         }
59     }
60 }
61 }
62

```

Burada Destroy fonksiyonunun iki argümanlı versiyonunu kullanıyoruz. İlk argümana gameObject'i veriyoruz, bunun anlamı ölen zombi objesinin yok olacağı. İkinci argümana ise 8.0 değerini veriyoruz, bunun anlamı da bu satır çalıştırdıktan 8.0 saniye sonra objenin yok olacağıdır. 8 saniye, zombinin ragdoll'unun devreye girip zombinin yerde bir süre kalması için yeterli bir zaman dilimi.

## Cesetlerden Kurtulmanın Alternatif Yolu

Cesedin 8 saniye sonra \*puf\* diye ekrandan kaybolması garip durabilir. Onun yerine ceset kameranın görüş alanından tamamen çıktığında onu yoketmek daha sağlıklı bir çözüm.

Az önce eklediğimiz Destroy satırını silin ve onun yerine scriptte şu değişiklikleri yapın:

```

01 using UnityEngine;
02 using System.Collections;
03
04 public class Health : MonoBehaviour
05 {
06     [SerializeField]
07     int _maximumHealth = 100;
08
09     int _currentHealth = 0;
10

```

```

11  override public string ToString()
12  {
13      return _currentHealth + " / " + _maximumHealth;
14  }
15
16  public bool IsDead { get { return _currentHealth <= 0; } }
17
18  Renderer _renderer;
19
20  void Start()
21  {
22      _renderer = GetComponentInChildren<Renderer>();
23      _currentHealth = _maximumHealth;
24  }
25
26  public void Damage(int damageValue)
27  {
28      _currentHealth -= damageValue;
29
30      if (_currentHealth < 0)
31      {
32          _currentHealth = 0;
33      }
34
35      if (_currentHealth == 0)
36      {
37          Animation a = GetComponentInChildren<Animation>();
38          a.Stop();
39
40          if (tag == "Player")
41          {
42              Destroy(GetComponent<PlayerMovement>());
43              Destroy(GetComponent<PlayerAnimation>());
44              Destroy(GetComponent<RifleWeapon>());
45          }
46          else // its an enemy
47          {
48              EnemySpawnManager.OnEnemyDeath();
49              Destroy(GetComponent<EnemyMovement>());
50              Destroy(GetComponentInChildren<EnemyAttack>());
51          }
52
53          Destroy(GetComponent<CharacterController>());
54
55          Ragdoll r = GetComponent<Ragdoll>();
56          if (r != null)
57          {
58              r.OnDeath();
59          }
60      }
61  }
62
63  void Update()
64  {
65      if (IsDead && !_renderer.isVisible)
66      {
67          Destroy(gameObject);

```

```
68     }  
69 }  
70 }  
71
```

Daha önce bahsettiğim gibi, `Renderer component`'i 3D objeyi ekranda çizdirmeye yarar. Bu `component`'in `"isVisible"` adında bir değişkeni vardır. Bu değişken, objenin herhangi bir kamera tarafından görünüp görünmediğini depolar. Eğer değeri `true` ise obje bir kameranın görüş açısındadır, `false` ise obje herhangi bir kameranın görüş açısında değildir.

İlk önce 18. satırda, düşmanın `Renderer component`'ini depolayacak bir değişken oluşturuyoruz.

Zombie objesini inceleyecek olursanız `"zombie_lowres"` adındaki child objesinin `Skinned Mesh Renderer component`'ine sahip olduğunu görebilirsiniz. Bu, zombimizi ekrana çizdirmeye yarayan `component`. `Component`, `Zombie`'nin child objesinde olduğu için ona 22. satırda `GetComponentInChildren` fonksiyonu ile erişiyoruz.

Son olarak da `Update` fonksiyonunun içinde zombinin ölüp ölmediğini (`IsDead`) ve bir kamera tarafından görünüp görünmediğini (`_renderer.isVisible`) kontrol ediyoruz. Eğer zombi ölmüşse ve kameranın görüş alanının dışındaysa onu yok ediyoruz.

**ÇEVİRMEN EKLEMESİ:** Bazen zombi öldüğü ve kameranın dışında olduğu halde silinmeyebilir. Bunun sebebi Unity editöründe çalışırken eğer zombi cesedi `Scene` panelinde görünür vaziyette ise o zaman `_renderer.isVisible`'in `true` döndürmesidir. Ama eğer oyunu build ederseniz ya da tam ekran olarak test ederseniz (`Game` panelinin sağ üstündeki `"Maximize on Play"` seçeneğini seçerek) bu sorunu yaşamazsınız.

## Özet Geçecek Olursak...

Oyunumuz gerçekten birşeye benzemeye başladı. Artık bizi sadece üç zombi kovalamıyor ve zombiler öldükçe yenileri gelmeye devam ediyor. Bu bölümde `viewport space`'i, `layer mask` değişkenini ve bir objenin kameraların görüş alanında olup olmadığını kontrol etmeyi gördük.