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

Unity 3D İle Oyun Programlama

Bölüm 15: Görsel ve Ses Efektleri Eklemek

ÇEVİRİ: Süleyman Yasir KULA

http://yasirkula.com

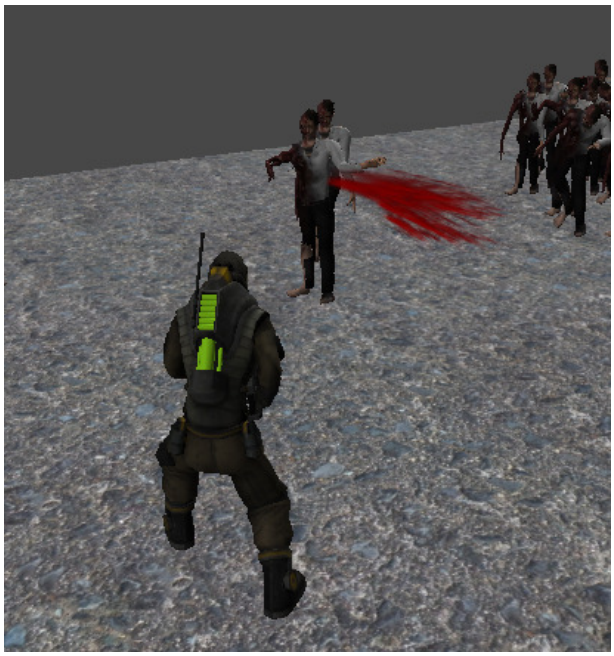
Ferdinand Joseph Fernandez

Chief Technological Officer, Dreamlords Digital Inc.

Admin and Co-founder, Unity Philippines Users Group

September 2011

Bu bölümde oyunumuza partikül efektleri ve ses efektleri ekleyeceğiz.



Zombiyi Vurunca Kan Efekti Çıkması

Winrar arşivinin olduğu yerdeki "Packages" klasörünün içinde “BloodHit.unitypackage” adında bir Unitypackage var. Bu package'ın içinde kullanıma hazır bir kan efekti bulunmakta. Biz oyunumuzda bunu kullanacağız.

Unity'de Assets > Import Package > Custom Package... yolunu izleyin ve BloodHit paketini projenize import edin. TheGame/Prefabs/ klasörüne "BloodHit" adında bir prefab gelecek.

RifleWeapon scriptini açın. Düşmana vurup vurmadığımızı bu scriptte kontrol ediyoruz. Kodu güncelleyin:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class RifleWeapon : MonoBehaviour
05 {
06       [SerializeField]
07       int _damageDealt = 10;
08
09       [SerializeField]
10       GameObject _hitEffectPrefab;
11
12       void Start()
13       {
14             Screen.lockCursor = true;
15       }
16
17       void Update()
18       {
```

```
19            if (Input.GetKey(KeyCode.Escape))
20            {
21                  Screen.lockCursor = false;
22            }
23
24            if (Input.GetButtonDown("Fire1"))
25            {
26                  Screen.lockCursor = true;
27                  Ray mouseRay = Camera.main.ViewportPointToRay(new  Vector3(0.5f, 0.
28                                    5f, 0));
29                  RaycastHit hitInfo;
30
31                  if (Physics.Raycast(mouseRay, out hitInfo))
32                  {
33                        Health enemyHealth = hitInfo.transform.GetComponent<Health>();
34                        if (enemyHealth != null)
35                        {
36                              enemyHealth.Damage(_damageDealt);
37
38                              Vector3 hitEffectPosition = hitInfo.point;
39                              Quaternion hitEffectRotation = Quaternion.identity;
40                              Instantiate(_hitEffectPrefab, hitEffectPosition,
41                                          hitEffectRotation);
42                        }
43                  }
44            }
45      }
46 }
```

Yaptığımız şey basit: Düşmana her vurduğumuzda, ona vurduğumuz noktada (hitInfo.point) BloodHit prefab'ının bir klonunu oluşturuyoruz (Instantiate).

Inspector'dan Player'ın RifleWeapon scriptindeki “Hit Effect Prefab”a "BloodHit" prefabını verin.

Oyunu çalıştırın. Zombiye vurunca zombiden kan fışkıracak.

Kan Efektinin Hep Aynı Yöne Doğru (Rotation) Çıkmasını Değiştirmek

Zombiye vurunca kan, her zaman vurduğumuz noktadan dışarı doğru fışkırmıyor. Bazen tam tersi yönde bazense garip bir yönde fışkırıyor. Kodu biraz daha güncelleyelim:

```
                                                .
                                                .
                                                .
33                  if (enemyHealth != null)
34                  {
35                        enemyHealth.Damage(_damageDealt);
36
37                        Vector3 hitEffectPosition = hitInfo.point;
38                        Quaternion hitEffectRotation = Quaternion.FromToRotation(Vector3.forward,
                              hitInfo.normal);
39                        Instantiate(_hitEffectPrefab, hitEffectPosition, hitEffectRotation);
40                  }
41            }
42      }
43      }
44 }
```

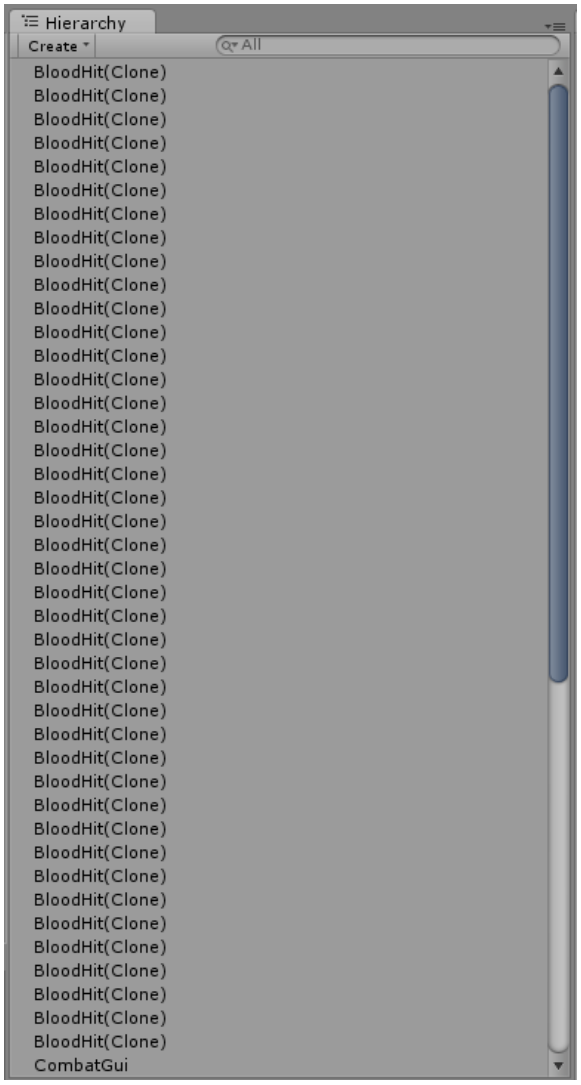
Artık kan efekti doğru yöne doğru fışkıracak.

"hitInfo.normal" komutu, raycast zombinin hangi noktasına isabet etmişse o noktada zombinin yüzeyine dik doğrultuda bir vektör (normal) döndürür.

Bildiğiniz üzere Unity'de bir objenin eğimi Quaternion türünde depolanıyor. Quaternion class'ının ise FromToRotation adında hazır bir fonksiyonu var. Bu fonksiyon bize öyle bir eğim (rotation)(quaternion) döndürür ki bu eğime sahip olan bir objenin birinci parametrede girdiğimiz ekseni, ikinci parametrede girdiğimiz vektör yönünde olur. Bizim örneğimizde birinci parametre Vector3.forward ve ikinci parametre ise "hitInfo.normal". Bunun anlamı Instantiate ettiğimiz kan efektinin rotation'ı öyle olacak ki kan efektinin Z ekseni (forward), "hitInfo.normal" yönünde, yani kurşunun zombiye temas ettiği noktadan dışarı doğru olacak ve böylece kan dışarı yönde fışkıracak.

Kan Efekti Klonlarını Temizlemek

Oyunu çalıştırıp da zombileri vurmaya başlayınca Hierarchy panelinde kan efekti klonlarının "BloodHit(Clone)" biriktiğini göreceksiniz:



Çünkü BloodHit prefab'ı iki child objeden oluşuyor. Bu iki child obje de birer partikül sistemi. Bu partikül sistemleri işleri bitince otomatik olarak yok oluyorlar ama parent obje olan BloodHit(Clone) sahnede kalmaya devam ediyor.

Yapmamız gereken parent objenin de silinmesini sağlamak. Peki nasıl? Child obje olan partikül sistemlerinin çalışmayı bitip bitmediğini kodla kontrol edip çalışmaları bittiğinde parent objeyi de elle silerek mi? Hayır, bunu yapmak zor olur. Daha basit bir çözüm yolu var: sahnede kan efekti klonu oluşturduğumuz anda bu klonun iki child objesini de child obje olmaktan çıkaralım (artık kan efekti klonu onların parent'ı olmasın) ve ardından tek başına kalan parent objeyi elle silelim.

“EffectsCleanup” adında yeni bir C# scripti oluşturun:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EffectsCleanup : MonoBehaviour
05 {
06     void Start()
07     {
08         foreach(Transform child in transform)
09         {
10             child.parent = null;
11         }
12         Destroy(gameObject);
13     }
14 }
```

Bir for döngüsü yardımıyla scriptin yazıldığı objenin tüm child objeleri üzerinden tek tek geçiyoruz ve bu child objenin artık child obje olmamasını sağlıyoruz (child.parent = null;). Ardından scriptin yazıldığı objeyi (parent objeyi) Destroy fonksiyonunu kullanarak siliyoruz.

Project panelinden BloodHit prefab'ını seçin ve EffectsCleanUp scriptini bu prefaba verin.

Oyunu test ettiğinizde artık Hierarchy'de klonların birikmediğini göreceksiniz.

Player Hasar Alınca Kan Efekti Oluşması

EnemyAttack scriptinde biraz düzenleme yaparak bir zombi bize hasar verirse player objesinden kan fışkırmasını sağlayacağız

Zombiler saldırırken raycast kullanmadığı için bu sefer hasarın tam olarak hangi noktada oluştuğunu bilmiyoruz. Bu yüzden kanın fışkıracağı noktayı farklı bir yöntemle hesaplayacağız.

EnemyAttack scriptini düzenleyin:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class EnemyAttack : MonoBehaviour
05 {
06     float _nextTimeAttackIsAllowed = -1.0f;
07
08     [SerializeField]
09     float _attackDelay = 1.0f;
10
11     [SerializeField]
```

```
12      int _damageDealt = 5;
13
14      [SerializeField]
15      GameObject _hitEffectPrefab;
16
17      void OnTriggerStay(Collider other)
18      {
19          if (other.tag == "Player" && Time.time >= _nextTimeAttackIsAllowed)
20          {
21              Health playerHealth = other.GetComponent<Health>();
22              playerHealth.Damage(_damageDealt);
23
24              Vector3 hitDirection = (transform.root.position - other.transform.position).normalized;
25              Vector3 hitEffectPosition = other.transform.position + (hitDirection * 0.01f) + (Vector3.up * 1.5f);
26              Quaternion hitEffectRotation = Quaternion.FromToRotation(Vector3.forward, hitDirection);
27              Instantiate(_hitEffectPrefab, hitEffectPosition, hitEffectRotation);
28
29              _nextTimeAttackIsAllowed = Time.time + _attackDelay;
30          }
31      }
32 }
```

Öncelikle 24. satırda, yönü player'dan zombiye doğru olan bir vektör oluşturup bunu hitDirection değişkeninde tutuyoruz.

Hatırlarsanız player objesinin pivot noktası ayak hizasında. Eğer kan efektini oradan çıkarırsak ayağımız kanıyor gibi garip bir görüntü oluşur. 25. satırda kan efektinin çıkacağı pozisyonu hesaplarken player'ın pozisyonunun (other.transform.position) 1.5 birim yukarısından (Vector3.up * 1.5f) zombinin olduğu yöne doğru 0.01 birim dışarıda (hitDirection * 0.01f) bir nokta alıyoruz.

26. satırda ise kan efektinin eğimini hesaplarken Quaternion.FromToRotation yardımıyla kan efektinin player'dan zombinin olduğu yöne doğru fışkırmasını sağlıyoruz.

Inspector'dan, Enemy prefab'ının EnemyAttack child objesindeki "Enemy Attack" component'inde yer alan "Hit Effect Prefab"a "BloodHit"i verip oyunu test edebilirsiniz. Düşman size hasar verince player objesinden düşmana doğru kan fışkıracak.

Ses Efektleri

Player veya bir zombi hasar aldıkça bir ses efekti oynatacağız.

Health scriptine şu kodu ekleyin:

```
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      PlayerStats _playerStats;
21
22      [SerializeField]
23      AudioClip _hitSound;
24
25      void Start()
26      {
27          _renderer = GetComponentInChildren<Renderer>();
28          _currentHealth = _maximumHealth;
29
30          GameObject player = GameObject.FindGameObjectWithTag("Player");
31          _playerStats = player.GetComponent<PlayerStats>();
32      }
33
34      public void Damage(int damageValue)
35      {
36          _currentHealth -= damageValue;
37
38          if (_currentHealth < 0)
39          {
40              _currentHealth = 0;
41          }
42          else
43          {
44              if (_hitSound != null)
45              {
46                  audio.clip = _hitSound;
47                  audio.Play();
48              }
49          }
50
51          if (_currentHealth == 0)
52          {
                                                    .
                                                    .
                                                    .
```

Audio Source componenti ses dosyaları çalmaya yarar. Audio Source component'inin Play() fonksiyonunu kullanarak _hitSound değişkeninde depoladığımız ses dosyasını oynatıyoruz. Tabi öncesinde Audio Source component'inin çalacağı ses klibini (audio.clip) belirliyoruz.

Scriptin çalışması için Health scriptinin atandığı obje(ler)de (player ve enemy) Audio Source componenti olması lazım. Eğer yoksa Component > Audio > Audio Source ile componenti bu objelere ekleyin.

Project panelini kullanarak "TheGame" klasörünün içinde "Sounds" adında bir klasör oluşturun. Bu klasörde oyunda kullanacağımız ses dosyalarını depolayacağız.

Şimdi ise Winrar arşivini çıkardığınız yerdeki "Sounds" klasöründe yer alan “player_killed_1.wav” ve “ZombieDeath.wav” ses dosyalarını projenizdeki "Sounds" klasörüne import edin.

Ardından Zombie prefab'ındaki Health scriptinde yer alan "Hit Sound"a değer olarak "ZombieDeath.wav" asset'ini, Player'daki "Hit Sound"a da "player_killed_1.wav" asset'ini değer olarak verin ve oyunu çalıştırın. Hasar aldığınızda "player_killed_1" sesi, zombiye hasar verdiğinizde "ZombieDeath" sesi çalacak.

Player Hasar Alınca Rastgele Bir Hasar Alma Sesi Çalmak

Winrar arşivinin oradaki Sounds klasöründe birden çok "player_killed_x.wav" ses dosyası olduğunu farketmişsinizdir. Bence player hasar alınca hep aynı ses dosyasını çalmak yerine Sounds klasöründeki ses dosyaları içinden rastgele bir tanesini çalalım.

Bunun için diğer "player_killed_x.wav" ses dosyalarını da projenizdeki Sounds klasörüne import edin.

Sonrasında Health scriptini şöyle değiştirin:

```
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       PlayerStats _playerStats;
21
22       [SerializeField]
23       AudioClip[] _hitSounds;
24
25       void Start()
26       {
27             _renderer = GetComponentInChildren<Renderer>();
28             _currentHealth = _maximumHealth;
29
30             GameObject player = GameObject.FindGameObjectWithTag("Player");
31             _playerStats = player.GetComponent<PlayerStats>();
32       }
33
34       public void Damage(int damageValue)
35       {
36             _currentHealth -= damageValue;
37
38             if (_currentHealth < 0)
39             {
```

```
40                  _currentHealth = 0;
41              }
42              else
43              {
44                  if (_hitSounds != null && _hitSounds.Length > 0)
45                  {
46                      AudioClip soundToUse = _hitSounds[Random.Range(0, _hitSounds.Length)];
47                      audio.clip = soundToUse;
48                      audio.Play();
49                  }
50              }
51
52              if (_currentHealth == 0)
53              {
                                                        .
                                                        .
                                                        .
```

Yaptığımız şey _hitSounds adında bir AudioClip array'i bulundurmak ve player ya da enemy hasar alınca bu array'in içindeki ses kliplerinden rastgele bir tanesini oynatmak. Array'den rastgele bir eleman çekmek için Random.Range fonksiyonunu kullanıyoruz.

Player objesini seçin ve "Hit Sounds"un Size'ını artırıp import ettiğiniz "player_killed_x.wav" ses dosyalarını array'e değer olarak ekleyin. Ardından Enemy prefab'ını seçin ve onun "Hit Sounds"unun Size'ını 1 yapıp "Element 0"ına değer olarak "ZombieDeath.wav" asset'ini verin.

Player Ölünce Bir Müziğin Çalması

Hasar alma sesiyle aynı mantığı kullanıyoruz. Bu sefer ses dosyasını sağlığımız dibe vurunca çaldırıyoruz:

```
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     PlayerStats _playerStats;
21
22     [SerializeField]
```

```
23      AudioClip[] _hitSounds;
24
25      [SerializeField]
26      AudioClip _deathSound;
27
28      void Start()
29      {
30          _renderer = GetComponentInChildren<Renderer>();
31          _currentHealth = _maximumHealth;
32
33          GameObject player = GameObject.FindGameObjectWithTag("Player");
34          _playerStats = player.GetComponent<PlayerStats>();
35      }
36
37      public void Damage(int damageValue)
38      {
39          _currentHealth -= damageValue;
40
41          if (_currentHealth < 0)
42          {
43              _currentHealth = 0;
44          }
45          else
46          {
47              if (_hitSounds != null && _hitSounds.Length > 0)
48              {
49                  AudioClip soundToUse = _hitSounds[Random.Range(0, _hitSounds.Length)];
50                  audio.clip = soundToUse;
51                  audio.Play();
52              }
53          }
54
55          if (_currentHealth == 0)
56          {
57              if (_deathSound != null)
58              {
59                  audio.clip = _deathSound;
60                  audio.Play();
61              }
62
63              Animation a = GetComponentInChildren<Animation>();
64              a.Stop();
                                                    .
                                                    .
                                                    .
```

Winrar arşivindeki Sounds klasöründe yer alan “you are dead dead dead 2.wav” ses dosyasını projenizdeki Sounds klasörüne import edin ve ardından Player'daki "Death Sound"a değer olarak verin. Bu müziğin sadece Player ölünce çalmasını istiyoruz. O yüzden Enemy'de bir değişiklik yapmayacağız.

İpucu

Winrar arşivinde “23289__progic35__resocopter-fastE.wav” adında bir ses dosyası var. Bu, helikopterin pervanesi için ideal bir ses. Kendiniz uğraşarak helikopterin bu ses dosyasını sürekli olarak (loop halinde) çalmasını sağlayın.

ÇEVİRMEN EKLEMESİ: Bunu script yazmadan da yapabilirsiniz. Dikkat etmeniz gereken şey helikoptere vereceğiniz Audio Source component'inin "Audio Clip", "Play On Awake" ve "Loop" değerleri.

Özet Geçecek Olursak...

Oyuna bazı görsel efektler ve ses efektleri ekledik. Bunlar oyun mekaniklerini doğrudan değiştirmiyor; sadece oyunun daha hoş durmasını sağlıyor. Oyunlarınıza efektler eklemeniz, oyuncunun dikkatini çekmenize çok yardımcı olabilir.