وب سایت آموزشی ساخت بازی‌های رایانه‌ای

سری آموزشی ساخت بازی‌های رایانه‌ای تیراندازی اول شخص (FPS)

با موتور بازی‌سازی یونیتی نگارش ۵

مجموعه‌های آموزشی ارایه شده در این وب سایت بر اساس منابع مختلفی از جمله کتاب‌ها، ویدیوهای آموزشی و توتریال‌های آموزشی در زمینه ساخت بازی‌های رایانه‌ای می‌باشد. مجموعه ارایه‌شده در این سری، آموزش ساخت بازی‌های شوتر اول شخص (FPS) با نرم‌افزار یونیتی است. سطح یادگیری در این سری آموزشی از مبتدی تا پیشرفته تنظیم گردیده است. کلیه حقوق مادی و معنوی این اثر متعلق به نویسنده بوده و هر گونه استفاده از آن بدون اجازه کتبی نویسنده ممنوع و موجب پی‌گرد قانونی است.

ترجمه و تالیف :

اکبر علی‌پور صیفار

info@gdlart.ir

www.gdlart.ir

۱۳۹۸

مقدمه

سری آموزشی کـه پـیش‌رو داریـد از مجموعـه آمـوزش‌هـای موتـور بازی‌سـازی یـونیتی و در مـورد سـاخت بـازی در سـبک شـوتر اول شـخص و تـک نفـره یـا بـه اصـطلاح (Single Player) می‌باشـد. اسـاس ایـن آموزش‌هـا برگرفتـه از آمـوزش‌هـای بازی‌سـازی موسـوم بـه GTGD S3 ، منـابع آموزشـی و توتریال‌هـای وب سـایت یـونیتی و تحقیقـات و تجربـه‌های عملـی نویسـنده اسـت. همـه ایـن سـری از آموزش‌هـا در قالب PDF نوشته شده تا بصـورت یـک مرجـع کـاملا عملـی در اختیـار کلیـه بازیسـازان و علاقمنـدان حـوزه سـاخت بـازی‌هـای رایانـه‌ای قـرار گیـرد. در گـردآوری ایـن مجموعـه آموزشـی سـعی شده‌اسـت تـا کلیـه سـطوح یـادگیری از مبتـدی تـا پیشـرفته در آموزش‌هـا لحـاظ گـردد. بنابراین کلیه قسـمت‌ها و فصـول مختلـف بصـورت کـاملا تشـریحی و عملـی بیان‌شـده و سعی‌شـده تـا تمـامی مفـاهیم و موضـوعات مـرتبط بـا ساخت یک بازی شوتر اول شخص (First Person Shooter) در آن گنجانده شود.

بدیهی‌است کـه آموزشـهای فـوق خـالی از اشـکال نبـوده و نیازمنـد اصـلاح و تکمیـل اسـت، لـذا کلیـه خواننـدگان و اسـتفاده کننـدگان از ایـن نوشتار می‌توانند در صورت برخورد با هرگونه ایراد و یا اشکالی از طریق پست الکترونیکی آن را ارسال فرمایند.

اکبر علی‌پور صیفار ۹۸ – ۱۳۹۵

فصل ششم

قسمت۱۰۱ – افکت برخورد گلوله

Gun Hit Effects

در این قسمت از فصل ششم آموزش ساخت بازی های تیراندازی اول شخص، به آموزش مبحث مربوط به نوشتن اسکریپتی جهت اجرای افکت گلوله می پردازم. هدف نوشتن اسکریپتی است که بتوان با آن هنگام برخورد گلوله به دشمن یا آبجکت های دیگر افکتی مخصوص به آن را اجرا نمود.

جهت پیاده سازی موارد فوق ابتدا در داخل پوشه Gun Scripts، اسکریپت جدیدی با نام Gun_HitEffects را ایجاد کرده و آن را در ویژوال استودیو باز می کنیم. طبق روال قبل ابتدا نیاز به تعریف چند متغیر جدید داریم که آنها را بصورت زیر می نویسیم.

```
4   namespace S3
5   {
6
    0 references
7   public class Gun_HitEffects : MonoBehaviour {
8
9       private Gun_Master gunMaster;
10
11      public GameObject defaultHitEffect;
12      public GameObject enemyHitEffect;
13
```

در کد های بالا دستور اول جهت دسترسی به رویداد های اصلی سیستم مدیریت اسلحه است. کد بعدی جهت تعریف افکت پیش فرض مربوط به برخورد گلوله از نوع گیم آبجکت بوده و کد آخر جهت تعریف افکت مربوط به برخورد گلوله به کاراکتر دشمن و از نوع گیم آبجکت است. به متد های Start() و Update() نیازی نداشته و لذا آنها را پاک می کنیم.

در این بخش متد های مربوط به افکت پیش فرض و همچنین افکت مربوط به برخورد گلوله به کاراکتر دشمن را بصورت زیر می نویسیم.

```
    0 references
29  void SpawnDefaultHitEffect(Vector3 hitPosition, Transform hitTransform)
30  {
31      if (defaultHitEffect != null)
32      {
33          Instantiate(defaultHitEffect, hitPosition, Quaternion.identity);
34      }
35  }
```

در کد بالا سطر اول مربوط به تعریف متدی برای ایجاد افکت پیش فرض مربوط به برخورد گلوله است. این متد دارای دو متغیر داخلی است که یکی برای تعیین نقطه برخورد گلوله و دیگری برای تعیین مختصات برخورد است. در سطر بعدی و در دستور شرطی بررسی می شود که افکت مربوط به برخورد گلوله به دشمن از قبل طراحی شده یا نه و اگر وجود داشته باشد، جواب شرط True شده و بقیه کد اجرا می گردد. در سطر بعدی با استفاده از دستور Instantiate افکت برخورد گلوله به کاراکتر دشمن با استفاده از اطلاعات نقطه برخورد (Hit Position) و چرخش به سمت کاراکتر دشمن (Hit Transform)، افکت مربوطه را ایجاد می کند. در ادامه کد مربوط به برخورد گلوله به کاراکتر دشمن را بصورت زیر می نویسیم.

```
36
    0 references
37  void SpawnEnemyHitEffect(Vector3 hitPosition, Transform hitTransform)
38  {
39      if (enemyHitEffect != null)
40      {
41          Instantiate(enemyHitEffect, hitPosition, Quaternion.identity);
42      }
43  }
```

در این بخش کد مربوط به متد SetInitialReferences() را بصورت زیر می نویسیم.

```
0 references
void SetInitialReferences()
{
    gunMaster = GetComponent<Gun_Master>();
}
```

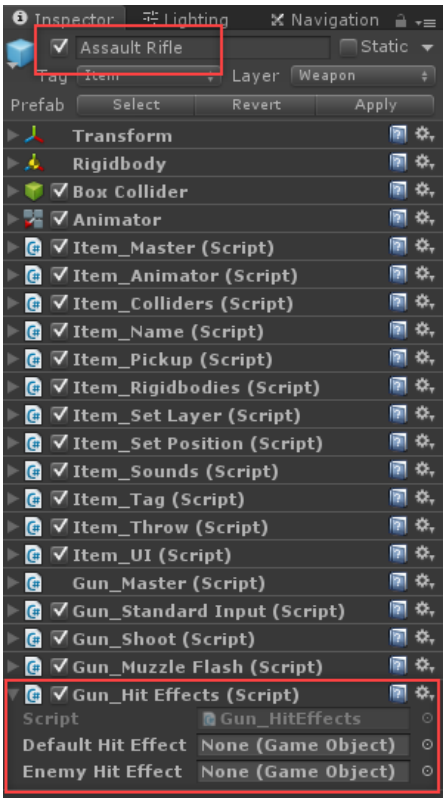
کد بالا جهت دسترسی به رویداد های اصلی سیستم مدیریت اسلحه است. متد فوق را در داخل متد OnEnable() فراخوانی می کنیم.

```
0 references
void OnEnable()
{
    SetInitialReferences();
    gunMaster.EventShotDefault += SpawnDefaultHitEffect;
    gunMaster.EventShotEnemy += SpawnEnemyHitEffect;
}
```

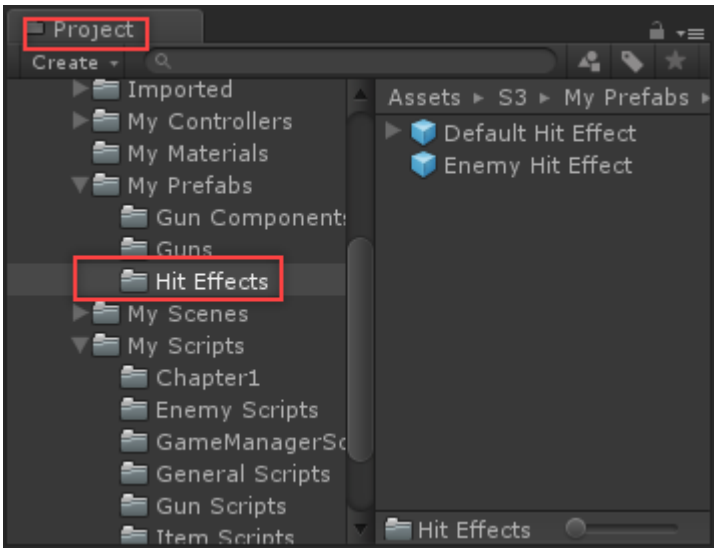
همانند قبل عکس کارهای فوق را نیز در داخل متد OnDisable() انجام می دهیم.

```
0 references
void OnDisable()
{
    gunMaster.EventShotDefault -= SpawnDefaultHitEffect;
    gunMaster.EventShotEnemy -= SpawnEnemyHitEffect;
}
```

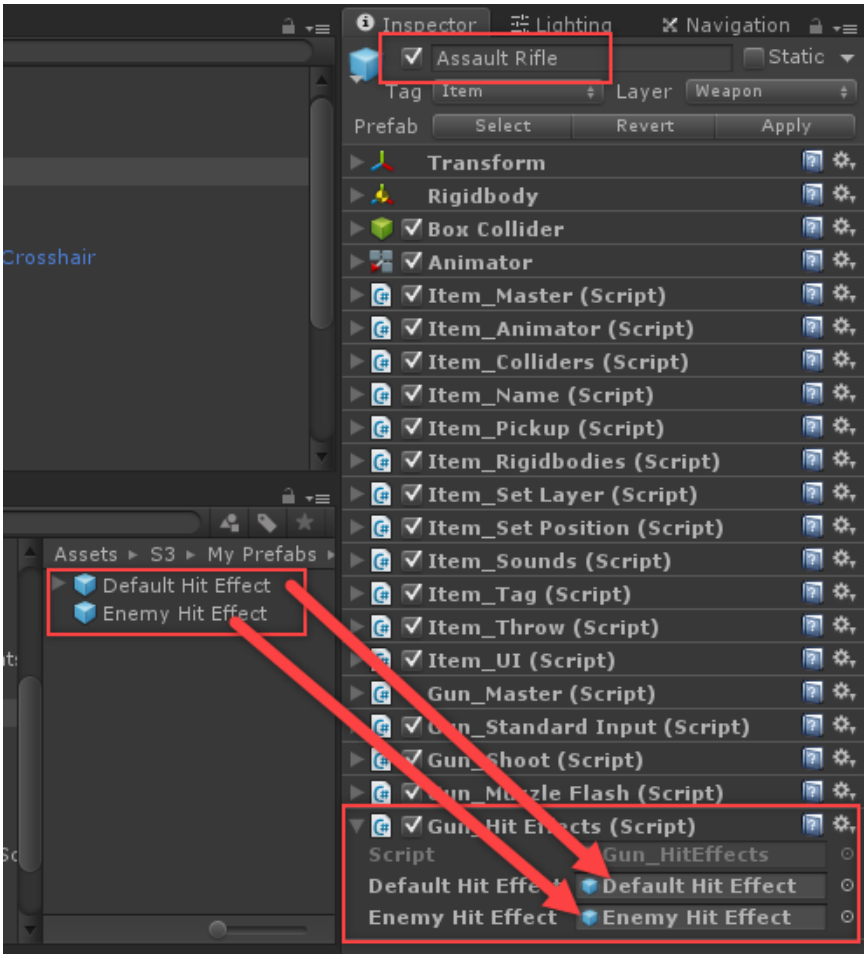
در این بخش به یونیتی رفته و از پنجره Hierarchy آبجکت اسلحه Assault Rifle را انتخاب می کنیم. از داخل پوشه Gun Scripts، اسکریپت Gun_HitEffects را کشیده و بر روی آبجکت اسلحه Assault Rifle می اندازیم تا به آن اضافه گردد.



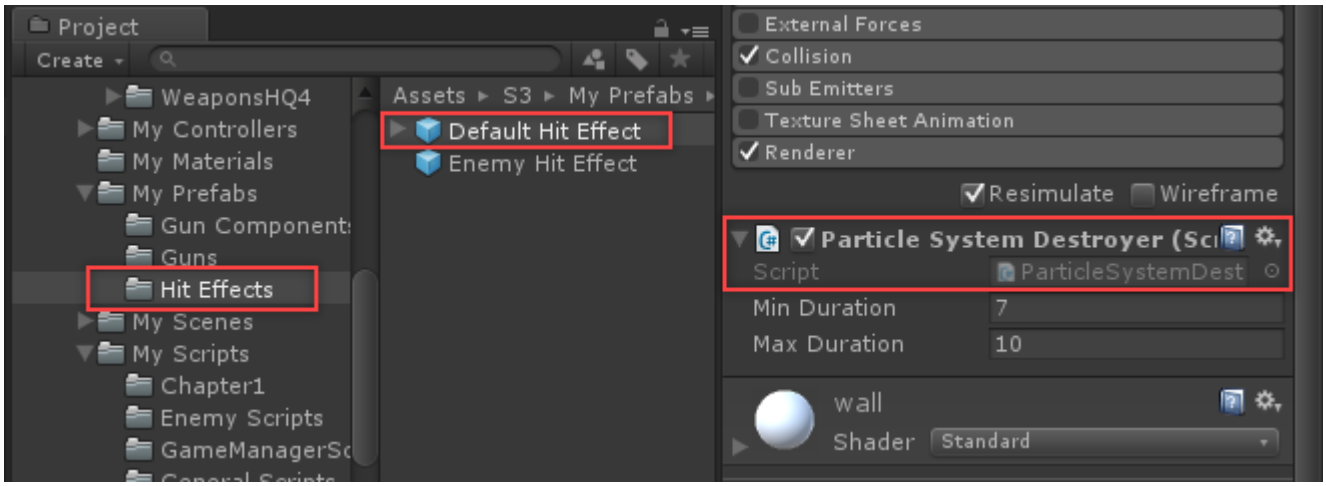
در پنجره Project به داخل پوشه MyPrefabs رفته و از داخل آن پوشه Hit Effects را انتخاب می کنیم. این همان پوشه ای است که در قسمت های آموزشی قبلی افکت های مربوط به برخورد گلوله پیش فرض و برخورد گلوله به کاراکتر دشمن را طراحی کرده و در آن ذخیره نمودیم. لذا جهت وصل کردن (Attach) این Prefab ها به اسکریپت مربوطه، مجددا از پنجره Hierarchy آبجکت اسلحه Assault Rifle را انتخاب می کنیم.



از پنجره Project و از داخل پوشه Hit Effects، افکت Default Hit Effect را انتخاب کرده، کشیده و در پنجره Inspector مربوط به اسلحه Assault Rifle در داخل اسکریپت Gun_HitEffects و داخل متغیر Default Hit Effect می اندازیم تا به آن اضافه گردد. همین کار را برای Prefab دوم یا Enemy Hit Effect نیز انجام داده و آن را کشیده و داخل متغیر Enemy Hit Effect واقع در اسکریپت Gun_HitEffects می اندازیم تا به آن اضافه گردد.



لازم به ذکر است که هر بازی ساز با توجه به نیاز خود و نوع بازی می تواند افکت های مخصوص به خود را طراحی کرده و داخل این دو متغیر قرار دهد. در این بخش بهتر است که بازی را اجرا کرده و نتیجه حاصل را مشاهده نماییم. با اجرای بازی به سمت کاراکتر دشمن رفته و به آن شلیک می کنیم. خواهیم دید که افکت مربوط به برخورد گلوله به کاراکتر دشمن اجرا شده و ذراتی به اطراف پرتاب می شود. همچنین می توانیم با شلیک به سمت دیوارها، افکت مربوط به برخورد گلوله پیش فرض را نیز مشاهده نماییم. موارد فوق کارکرد صحیح اسکریپت نوشته شده را به خوبی نشان می دهد. در آخر اگر به پنجره کنسول نگاه کنیم خواهیم دید که پیغام هایی مبنی بر فعال شدن یکی از افکت های طراحی شده هنگام برخورد گلوله به کاراکتر دشمن یا به دیوار ها مشاهده می شود. می توانیم این پیغام ها را از داخل اسکریپت (Particle System Destroyer) Comment Out کنیم تا غیر فعال گردند. برای دسترسی به این اسکریپت به داخل پوشه My Prefabs واقع در پنجره Project رفته و از داخل آن پوشه Hit Effects را انتخاب می کنیم. از داخل این پوشه نیز Prefab مربوط یا Default Hit Effect را انتخاب می کنیم. در پنجره Inspector مربوط به این Prefab، و در قسمت پایین بر روی اسکریپت Particle System Destroyer دو بار کلیک می کنیم تا در ویژوال استودیو باز شود. از داخل آن کد Debug آن را Comment Out می کنیم.




```
31          // wait for random duration
32
33          float stopTime = Time.time + Random.Range(minDuration, maxDuration);
34
35          while (Time.time < stopTime || m_EarlyStop)
36          {
37              yield return null;
38          }
39          //Debug.Log("stopping " + name);
40
41          // turn off emission
42          foreach (var system in systems)
43          {
44              var emission = system.emission;
45              emission.enabled = false;
46          }
47          BroadcastMessage("Extinguish", SendMessageOptions.DontRequireReceiver);
```

در آخر کد مربوط به اسکریپت Gun_HitEffects را برای بررسی بیشتر آورده ایم.

```
using UnityEngine;
using System.Collections;

namespace S3
{

public class Gun_HitEffects : MonoBehaviour {

    private Gun_Master gunMaster;

    public GameObject defaultHitEffect;
    public GameObject enemyHitEffect;

    void OnEnable()
    {
        SetInitialReferences();
        gunMaster.EventShotDefault += SpawnDefaultHitEffect;
        gunMaster.EventShotEnemy += SpawnEnemyHitEffect;
    }

    void OnDisable()
    {
        gunMaster.EventShotDefault -= SpawnDefaultHitEffect;
        gunMaster.EventShotEnemy -= SpawnEnemyHitEffect;
    }

    void SetInitialReferences()
    {
        gunMaster = GetComponent<Gun_Master>();
    }

    void SpawnDefaultHitEffect(Vector3 hitPosition, Transform hitTransform)
    {
        if (defaultHitEffect != null)
        {
            Instantiate(defaultHitEffect, hitPosition, Quaternion.identity);
        }
    }

    void SpawnEnemyHitEffect(Vector3 hitPosition, Transform hitTransform)
    {
        if (enemyHitEffect != null)
        {
            Instantiate(enemyHitEffect, hitPosition, Quaternion.identity);
        }
    }
}
}
```