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 ۹۹ – ۱۳۹۵

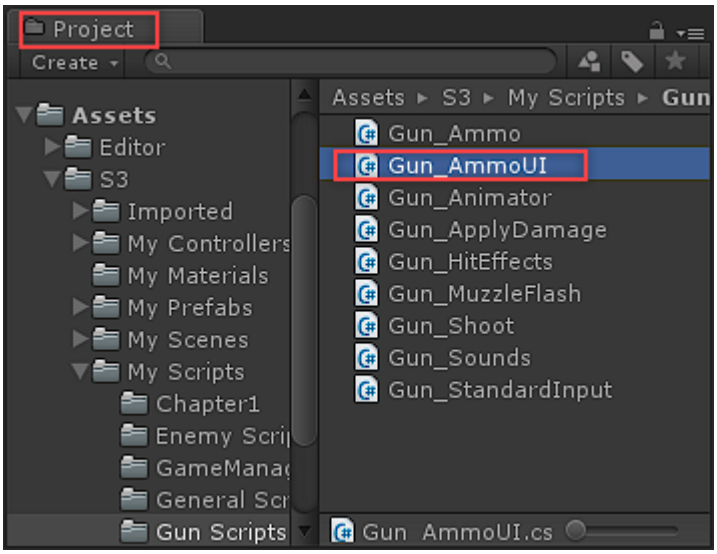
فصل ششم

قسمت۱۰۷ – رابط کاربری مهمات اسلحه

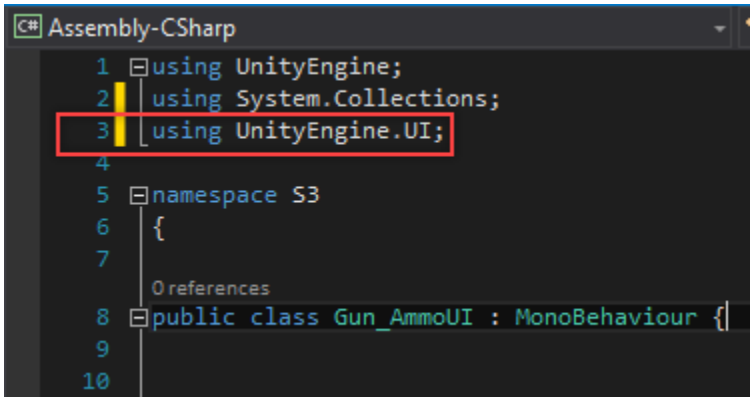
Gun Ammo UI

در این قسمت از فصل ششم سری آموزشی ساخت بازی های تیراندازی اول شخص به آموزش مبحث نوشتن اسکریپتی مربوط به نشان دادن میزان مهمات یا (Ammo) برای کاربر بازی، می پردازم. هدف نوشتن اسکریپتی است که بتوان با آن میزان مهمات فعلی و میزان مهمات کل را در هر لحظه و با هر بار شلیک به روز رسانی کرده و برای کاربر از طریق رابط کاربری یا UI، نشان داد.

برای پیاده سازی موارد فوق ابتدا اسکریپت جدیدی با نام Gun_AmmoUI را در داخل پوشه Gun Scripts ایجاد کرده و آن را در ویژوال استودیو باز می کنیم.



در ابتدا بدلیل اینکه می خواهیم از رابط کاربری UI استفاده کنیم باید با استفاده از دستور Using، UI مربوط به نرم افزار یونیتی را فراخوانی نماییم. جهت این کار کد زیر را در زیر کد Using System. Collections می نویسیم.



لازم به ذکر است که در هر زمان بخواهیم که با رابط کاربری اطلاعاتی را به کاربر نمایش دهیم، باید کد بالا را وارد نماییم. در ادامه طبق روال قبلی ابتدا نیاز به تعریف چند متغیر جدید داریم که آنها را بصورت زیر می نویسیم.

```
5   namespace S3
6   {
7
    0 references
8   public class Gun_AmmoUI : MonoBehaviour {
9
10      public InputField currentAmmoField;
11      public InputField carriedAmmoField;
12
13      private Gun_Master gunMaster;
14
```

در کد های بالا، متغیر اولی جهت تعریف یک محدوده برای نمایش دادن مقدار مهمات فعلی اسلحه است. متغیر دومی جهت تعریف یک محدوده برای نمایش دادن مقدار کل مهمات اسلحه است. متغیر بعدی جهت استفاده از رویداد های اصلی سیستم مدیریت اسلحه است.

به متد های Start() و Update() نیازی نداشته و لذا آنها را پاک می نماییم.

در این بخش متد مربوط به، به روز رسانی مقادیر UI مربوط به مقدار مهمات فعلی و مقدار مهمات کل را بصورت زیر می نویسیم.

```
    0 references
31  void UpdateAmmoUI(int currentAmmo, int carriedAmmo)
32  {
33      if (currentAmmoField != null)
34      {
35          currentAmmoField.text = currentAmmo.ToString();
36      }
37      if (carriedAmmoField != null)
38      {
39          carriedAmmoField.text = carriedAmmo.ToString();
40      }
41  }
```

در کد های بالا و در دستور شرطی اول، بررسی می شود که محدوده ای برای نمایش مقدار مهمات فعلی وجود داشته باشد (لازم به ذکر است که این محدوده در قسمت های آموزشی قبلی طراحی شده است). در خط بعدی، مقدار متغیر Current Ammo که بصورت عددی و از نوع Integer یا عدد صحیح است، برای نمایش به عنوان یک رشته یا Text باید به نوع String تبدیل شده و در داخل متغیر Text مربوط به محدوده نمایش مقدار مهمات فعلی قرار گیرد تا در رابط کاربری، نمایش داده شود. در دستور شرطی دوم، مقدار مهمات کل به نوع String تبدیل شده و در داخل متغیر Text مربوط به محدوده نمایش مقدار مهمات کل، جهت نمایش ذخیره می گردد.

در ادامه متد SetInitialReferences() را نیز بصورت زیر می نویسیم.

```
    0 references
26  void SetInitialReferences()
27  {
28      gunMaster = GetComponent<Gun_Master>();
29  }
30
```

متد فوق را در داخل متد OnEnable() فراخوانی می کنیم.

```
0 references
16  void OnEnable()
17  {
18      SetInitialReferences();
19      gunMaster.EventAmmoChanged += UpdateAmmoUI;
20  }
```

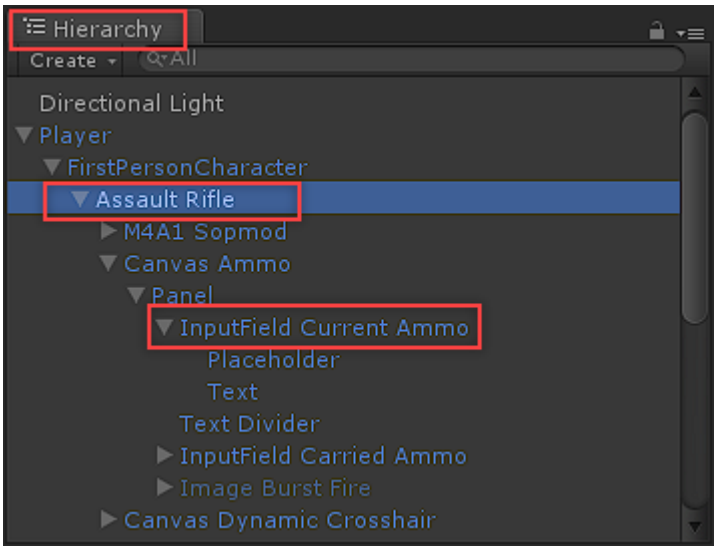
در کد بالا و در سطر دوم آن، ابتدا رویداد تغییرات میزان مهمات فراخوانی شده و توسط آن متد به روز رسانی میزان مهمات در UI، اجرا می گردد.

عکس کار فوق را نیز در داخل متد OnDisable() انجام می دهیم.

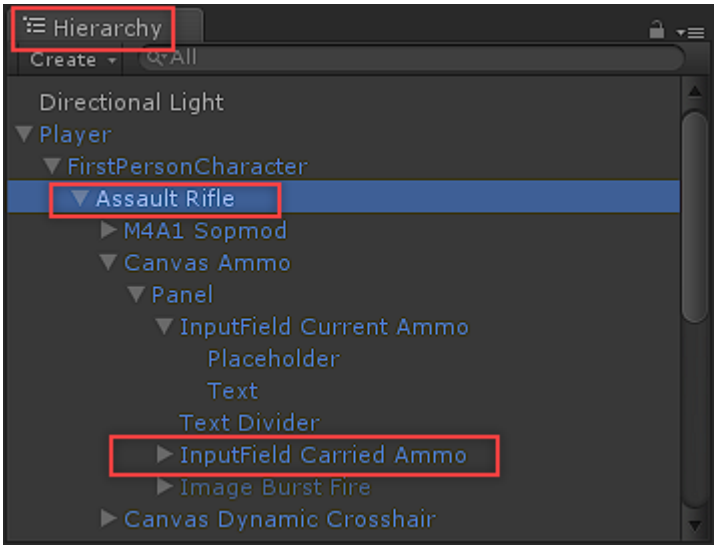
```
21
    1 references
22  void OnDisable()
23  {
24      gunMaster.EventAmmoChanged -= UpdateAmmoUI;
25  }
26
```

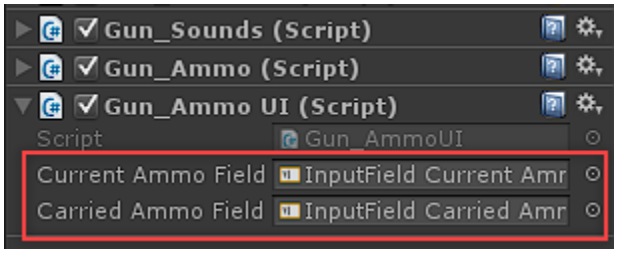
در این مرحله به یونیتی رفته و از پنجره Hierarchy آبجکت اسلحه یا Assault Rifle را انتخاب می کنیم. از داخل پوشه Gun Scripts واقع در پنجره Project، اسکریپت Gun_AmmoUI را کشیده و بر روی اسلحه می اندازیم تا به آن اضافه گردد.

جهت مشخص نمودن محدوده نمایش میزان مهمات فعلی و کلی، در پنجره Hierarchy بر روی فلش کوچک کنار نام اسلحه Assault Rifle کلیک می کنیم تا کلیه مولفه های آن نمایش داده شود. از میان آنها مولفه Input Field Current Ammo را کشیده و در پنجره Inspector مربوط به اسلحه Assault Rifle و در قسمت مربوط به اسکریپت Gun_AmmoUI، بر روی متغیر Current Ammo Field می اندازیم تا به آن اضافه گردد.

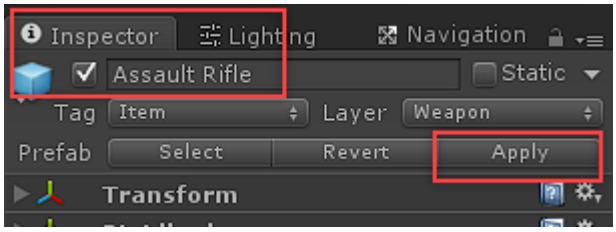


همین کار را برای مولفه دیگر اسلحه Assault Rifle یعنی مولفه Input Field Carried Ammo نیز انجام داده و آن را بر روی متغیر Carried Ammo Field می اندازیم تا به آن اضافه گردد.





در پنجره Inspector مربوط به آبجکت اسلحه، کلید Apply را کلیک می کنیم تا تغییرات فوق اعمال گردند.



در این بخش بازی را اجرا می کنیم. اگر به سمت جعبه مهمات یا همان شکل کره (Sphere) رفته و آن را بر می داریم. خواهیم دید که میزان کل مهمات در UI زیر اسلحه افزایش یافته و مقدار فعلی آن به روز رسانی می شود. به طرف کاراکتر دشمن رفته و به آن شلیک می کنیم. خواهیم دید که در هر بار شلیک، به میزان ۱ واحد از مقدار مهمات فعلی کم شده و در قسمت UI مربوطه به آن در هر لحظه به روز رسانی شده و آخرین مقدار موجود نمایش داده می شود. کلید مربوط به خشاب گذاری (R) را کلیک می کنیم، خواهیم دید که انیمیشن مخصوص به آن اجرا شده و مقدار مهمات فعلی به حداکثر آن که در قبل تعیین کرده بودیم، رسیده و از مقدار مهمات کل موجود کم می گردد که این موارد نشان دهنده کارکرد صحیح اسکریپت این قسمت آموزشی است.

در آخر کد نهایی اسکریپت Gun_AmmoUI جهت مطالعه بیشتر آورده شده است.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

namespace S3
{

public class Gun_AmmoUI : MonoBehaviour {

    public InputField currentAmmoField;
    public InputField carriedAmmoField;

    private Gun_Master gunMaster;


    void OnEnable()
    {
        SetInitialReferences();
        gunMaster.EventAmmoChanged += UpdateAmmoUI;
    }

    void OnDisable()
    {
        gunMaster.EventAmmoChanged -= UpdateAmmoUI;
    }

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

    void UpdateAmmoUI(int currentAmmo, int carriedAmmo)
    {
        if (currentAmmoField != null)
        {
            currentAmmoField.text = currentAmmo.ToString();
        }
        if (carriedAmmoField != null)
        {
            carriedAmmoField.text = carriedAmmo.ToString();
        }
    }
}
}
```