



THIRD EDITION

آموزش یونیتی

Unity in action



Unity
IN ACTION



Unity IN ACTION

THIRD EDITION

Multiplatform game development in C#



Joseph Hocking

Foreword by Jesse Schell



Unity Hub

در هنگام دانلود و شروع به نصب شما در واقع یک مدیر نصب سبک وزن را به جای برنامه اصلی Unity دانلود خواهید کرد. این برنامه مدیریتی که Unity Hub نام دارد برای ساده کردن نصب و استفاده از چندین نسخه یونیتی به طور همزمان وجود دارد. همانطور که در شکل ۱.۹ نشان داده شده است، نصب ویرایشگر اولین چیزی است که هنگام راه اندازی Unity Hub اتفاق می افتد.

When Unity Hub runs for the first time, it will start on the tab for installing a version of Unity. Click the Add button and then install the default Recommended release.

The next several screenshots use the Karting microgame. To install that project, click the Learn tab and then select that project in the list.

After the first launch, Unity Hub will start on the Projects list.

Return to Installs to see which versions of Unity are currently installed and to install new versions.

Use these buttons to Add existing projects (e.g., this book's code samples) to the list or to create a New project.

The Unity version that a given project will launch in. If you need to change this (e.g., you want to upgrade a project), click in this column for a menu of versions installed on this machine.

Project Name	Unity Version	Target Platform	Last Modified
ch12	2020.3.12f1	Current platform	a few seconds ago
ch11	2020.3.12f1	Current platform	14 days ago
ch10	2020.3.12f1	Current platform	14 days ago
MyFirstGame	Select a version	Current platform	15 days ago

Figure 1.9 Unity Hub on first launch versus subsequently



زمانی که این مطلب را بخوانید، احتمالاً نسخه های جدیدتر Unity منتشر شده اند. ویژگی های پیشرفته تغییر خواهند کرد، و احتمالاً حتی ظاهر رابط نیز می تواند متفاوت باشد، اما مفاهیم اساسی که در این اسلایدها پوشش داده شده است، همچنان صادق خواهند بود. توضیحات ارائه شده در این کتاب به طور کلی همچنان در مورد هر نسخه ای از Unity که در آینده موجود باشد، اعمال می شود.

در پروژه ها دخیره می شود که در کدام نسخه از Unity ایجاد شده اند و اگر بخواهید آنها را در نسخه دیگری باز کنید، خطاری صادر می کنند.





رابط کاربری در Unity به بخش‌هایی تقسیم می‌شود: تب صحنه، تب بازی، نوار ابزار، برگه سلسله مراتب، بازرس، برگه پروژه و تب کنسول.

Scene and Game are tabs for viewing the 3D scene and playing the game, respectively.

The whole top area is the Toolbar. To the left are buttons for looking around and moving objects, and in the middle is the Play button.

The Inspector is on the right side. This displays information about the currently selected object (a list of components mostly).

Hierarchy shows a text list of all objects in the scene, nested according to how they're linked together. Drag objects in the hierarchy to link them.

Project and Console are tabs for viewing all files in the project and messages from the code, respectively.

Navigate folders on the left, and then double-click MainScene.

The Tutorials list appears only in the starter micro-games and won't be here normally.

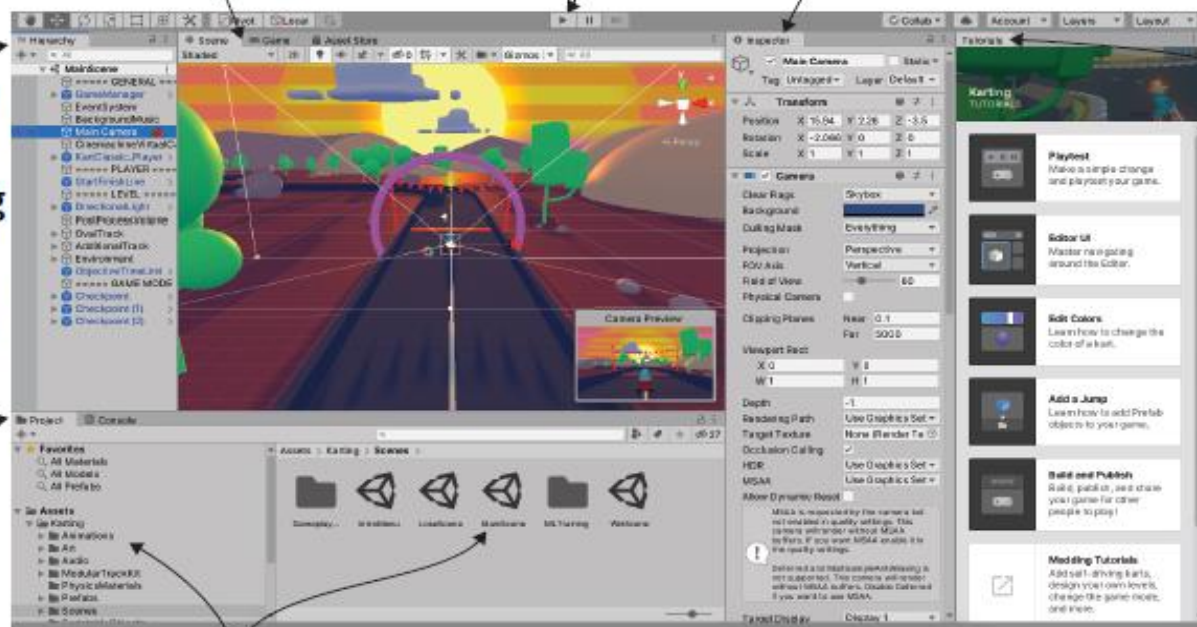


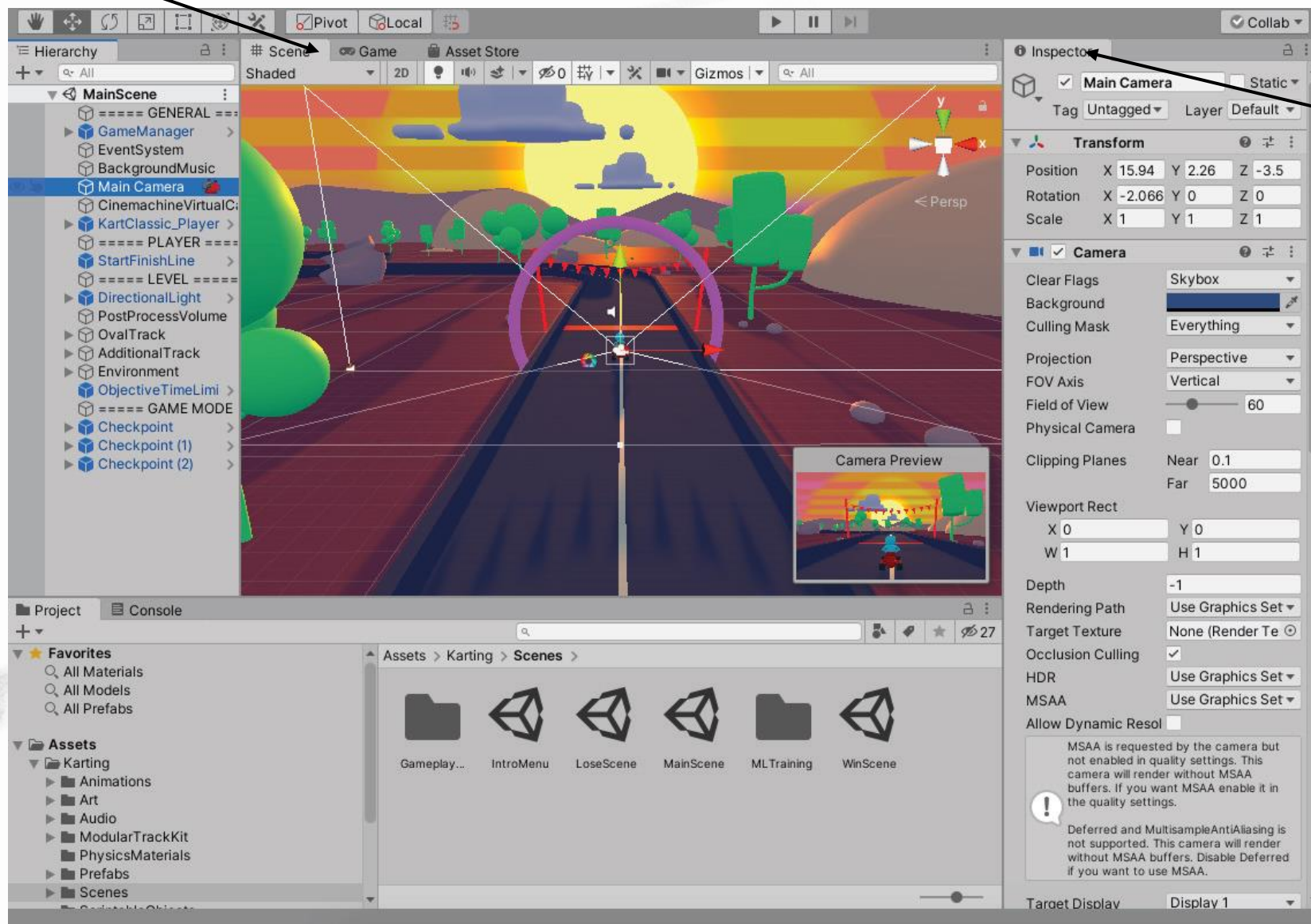
Figure 1.10 Parts of the interface in Unity



Scene و Game به ترتیب تب هایی برای مشاهده صحنه سه بعدی و اجرای بازی هستند.

کل قسمت بالا نوار ابزار است. در سمت چپ دکمه هایی برای نگاه کردن به اطراف و جابجایی اشیاء و در وسط آن دکمه Play قرار دارد.

این اطلاعات مربوط به شیء انتخاب شده فعلی را نمایش می دهد



سلسله مراتب یک لیست متنی از تمام اشیاء در صحنه را نشان می دهد که بر اساس نحوه پیوند آنها به یکدیگر تودرتو شده اند.

Project و Console به ترتیب تب هایی برای مشاهده تمام فایل های پروژه و پیام های کد هستند.



- ما می توانید تمام فایل های موجود در تب **Project** را مرور کنید.
- با استفاده از تب **Scene** می توانید اشیا را در صحنه فعلی قرار دهید.
- نوار ابزار دارای کنترل هایی برای کار با صحنه است.
- می توانید روابط شی را در تب **Hierarchy** بکشید و رها کنید.
- **Inspector** اطلاعات مربوط به اشیاء انتخاب شده، از جمله کد پیوند شده را فهرست می کند.
- می توانید هنگام تماشای خروجی خطا در تب **Console**، بازی را در نمای بازی آزمایش کنید.



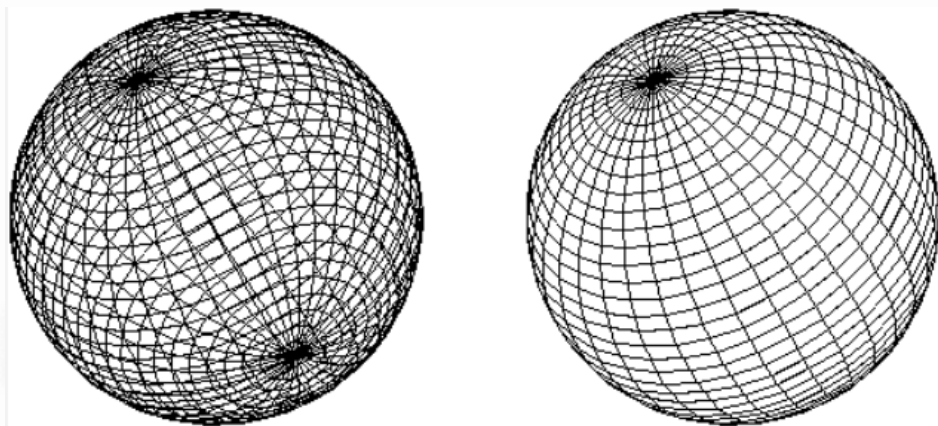


Scene view

برجسته ترین قسمت رابط، نمای صحنه است. اینجا است که می توانید ببینید که دنیای بازی چگونه است و اشیاء را به اطراف حرکت دهید. اشیاء مش در صحنه به عنوان مشبک و توری شکل ظاهر می شوند

همچنین می توانید اشیاء دیگری را در صحنه مشاهده کنید که با نمادها و خطوط رنگی نشان داده شده اند: دوربین ها، چراغ ها، منابع صوتی، مناطق برخورد و غیره. توجه داشته باشید که نمایی که در اینجا می بینید با نمای بازی در حال اجرا یکسان نیست — شما می توانید به دلخواه خود به اطراف صحنه نگاه کنید بدون اینکه محدود به دید بازی باشید.

یک شی مشبک یک شی بصری در فضا است. تصاویر در گرافیک سه بعدی از خطوط و شکل های به هم پیوسته زیادی ساخته می شوند - از این رو کلمه مش استفاده می شود.



مش / Mesh

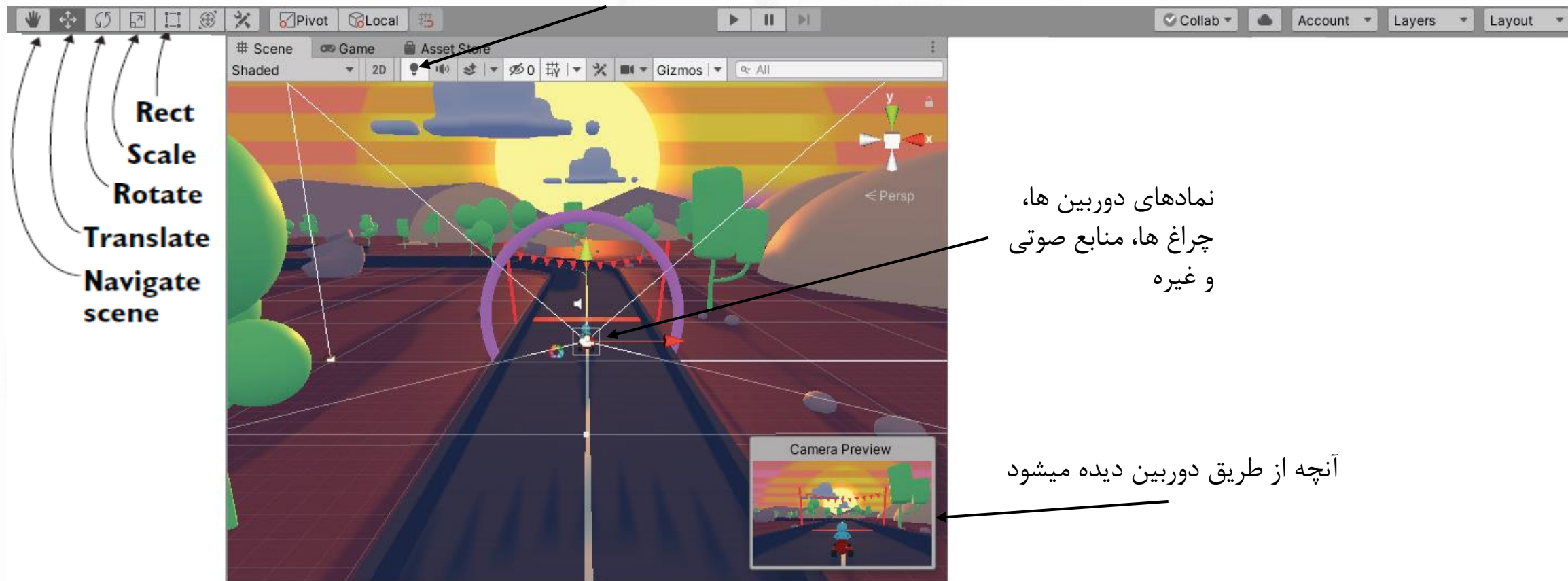
شاید یکی از اصطلاحاتی که در همان ابتدا با آن برخورد کنید اصطلاح مش / Mesh است. در حالت ساده هر حجمی که ما در دنیای سه بعدی ایجاد می کنیم در اصل شبکه های به هم دوخته شده ی سیمی است که در نهایت حجم مورد نظر را می سازند. اگر دقت کرده باشید وظیفه ی مش ها ساختن اسکلت مدل مورد نظر ماست و ما می توانیم با اصلاح مش های یک مدل شکل نهایی آن را تغییر دهیم.



هنگامی که بازی در حال اجرا است، آنچه در این نما می بینید همان بازی است. لازم نیست هر بار که بازی را اجرا می کنید، تب ها را به صورت دستی تغییر دهید، زیرا با شروع بازی، نمای به طور خودکار به بازی تغییر می کند.

در حالی که بازی در حال اجرا است، می توانید به نمای صحنه برگردید و به شما امکان می دهد اشیاء را در صحنه در حال اجرا بررسی کنید. این قابلیت برای دیدن آنچه در حین اجرای بازی در حال انجام است بسیار مفید است و یک ابزار عیب یابی مفید است که در اکثر موتورهای بازی موجود نیست.

گزینه هایی برای نمایش جنبه های صحنه
(به عنوان مثال، دکمه جابه جایی برای نمایش نور)





تبدیل اجسام نیز از طریق سه عمل اصلی انجام می شود و سه حرکت ناوبری صحنه عبارتند از

- Translate
- Rotate
- Scale

(شکل 1.12 تبدیل ها را در یک مکعب نشان می دهد.)

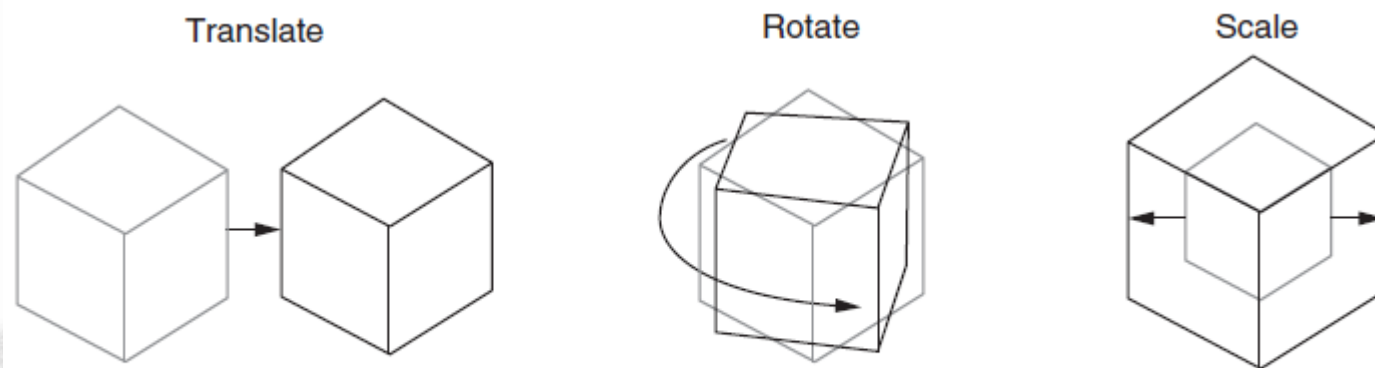
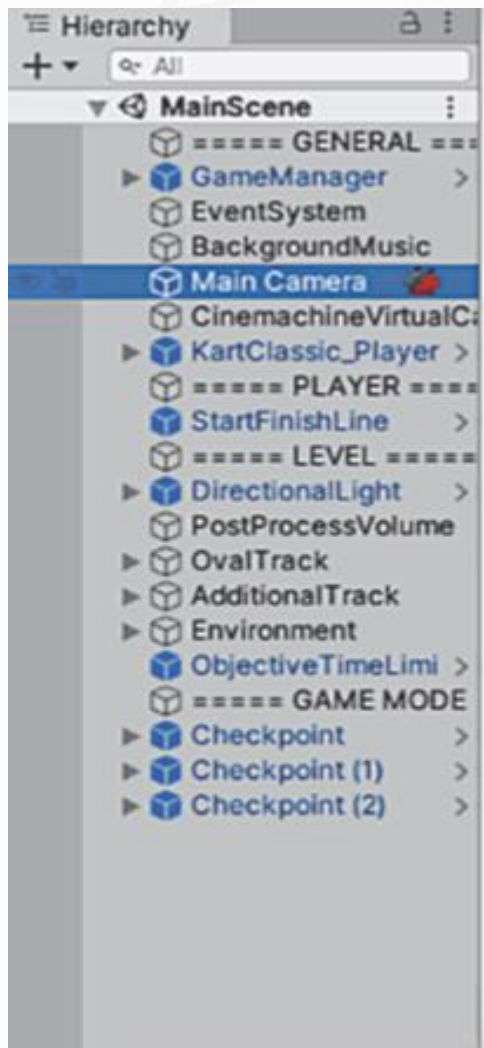


Figure 1.12 Applying the three transforms: Translate, Rotate, and Scale. (The lighter lines are the previous state of the object before it was transformed.)



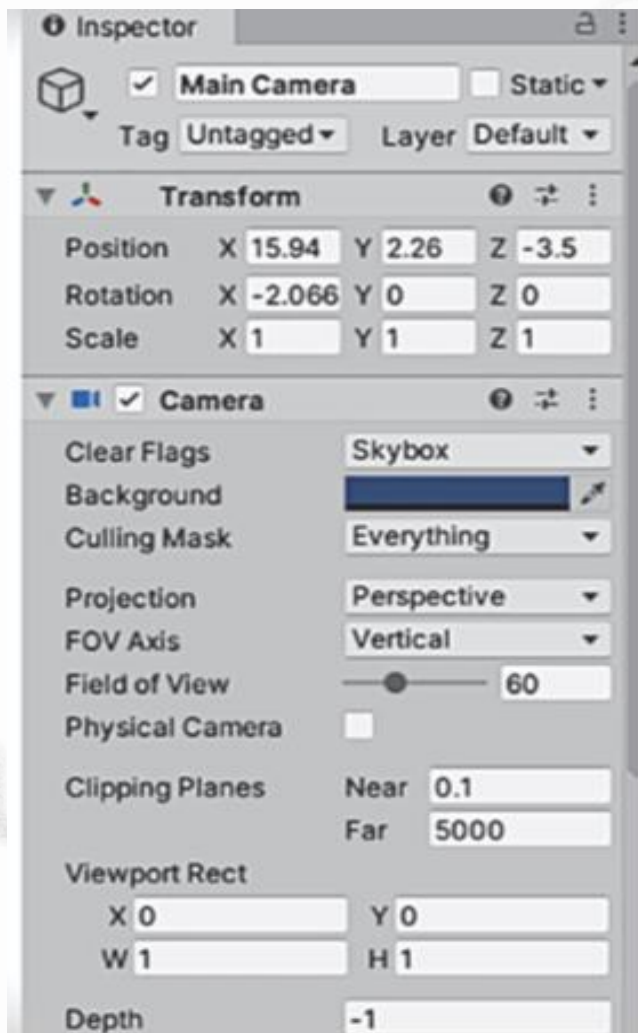
لیست سلسله مراتب (*Hierarchy*): نام هر شی را در صحنه فهرست می کند و نام ها را با توجه به پیوندهای سلسله مراتبی آنها در صحنه در کنار هم قرار می دهد. اساساً، این روشی است برای انتخاب اشیاء با نام به جای کلیک کردن روی آنها در نمای صحنه (*Scene*) است. پیوندهای سلسله مراتبی، اشیاء را به صورت بصری با هم گروه بندی می کند، مانند پوشه ها، به شما این امکان را می دهد که کل گروه را به عنوان یک واحد حرکت دهید.





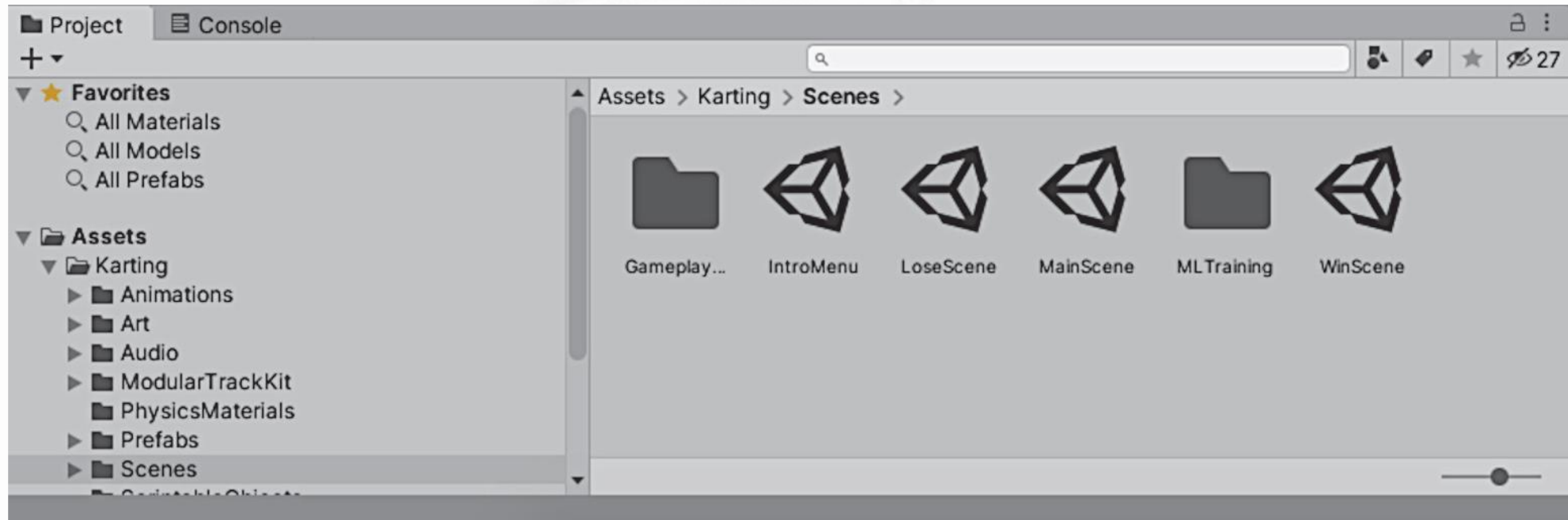
پنجره کنترل Inspector : اطلاعاتی را درباره شی انتخاب شده فعلی به شما نشان می دهد.

- اطلاعات نشان داده شده تقریباً فهرستی از مؤلفه ها (Components) است.
- شما می توانید مؤلفه هایی را از اشیا متصل یا حذف کنید.
- همه اشياء بازی حداقل یک جزء دارند، Transform، بنابراین شما همیشه حداقل اطلاعاتی در مورد موقعیت و چرخش در Inspector خواهید دید.





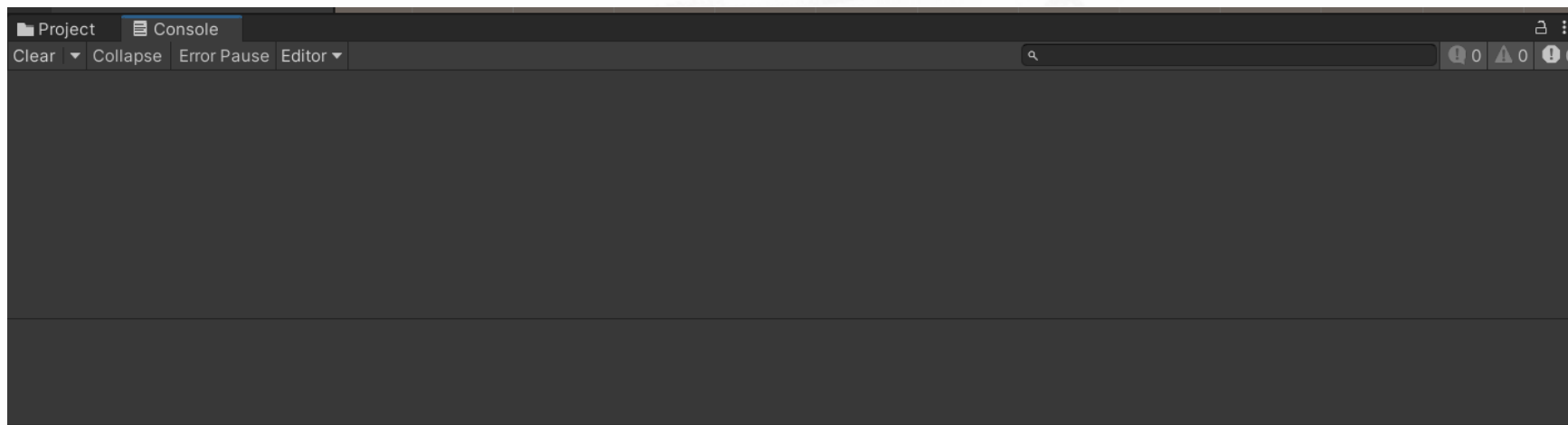
- **نمای پروژه** تمام دارایی ها (فایل گرافیکی، کد و غیره) را در پروژه نشان می دهد.
به طور خاص، در سمت چپ نما فهرستی از دایرکتوری های پروژه وجود دارد. هنگامی که یک دایرکتوری را انتخاب می کنید، سمت راست نمای فایل های جداگانه در آن دایرکتوری را نشان می دهد.



نمای پروژه منعکس کننده دایرکتوری **Assets** روی دیسک است، اما به طور کلی، شما نباید فایل ها را مستقیماً با رفتن به پوشه **Assets** در کاوشگر فایل سیستم عامل خود منتقل یا حذف کنید. اگر این کارها را در نمای پروژه انجام دهید، یونیتی با آن پوشه همگام خواهد بود.



تب Console جایی است که پیام های کد نمایش داده می شود. برخی از این پیام ها خروجی اشکال زدایی هستند که شما عمداً قرار داده اید، اما یونیتی همچنین اگر در اسکریپتی که نوشته اید با مشکل مواجه شود، پیام های خطا منتشر می کند.



راه اندازی و اجرا با برنامه نویسی یونیتی

Unity را راه اندازی کنید و یک پروژه جدید ایجاد کنید: در Unity Hub گزینه New را انتخاب کنید، یا اگر Unity در حال اجرا است، File > New Project را انتخاب کنید.

- یک نام برای پروژه تایپ کنید، الگوی سه بعدی پیش فرض را انتخاب کنید و سپس محل ذخیره پروژه را انتخاب کنید.
- پروژه یونیتی به سادگی یک دایرکتوری (پوشه) پر از فایل های دارایی و تنظیمات مختلف است، بنابراین پروژه را در هر نقطه از رایانه خود ذخیره کنید.
- در زمان تحویل پروژه در کلاسی این پوشه را باید به همراه داشته باشید.



Running code in Unity: Script components

کدهایی در یونیتی اجرا می شوند که به اشیاء موجود در صحنه متصل باشد.

- به این کدها کامپوننت گفته می شود
- یک شیء از اجزا مختلف تشکیل شده است این قسمت ها را می توان در پنجره کنترل مشاهده کرد
- ترتیب اجرای کدهای متصل به یک شیء به ترتیبی است که در پنجره کنترل مشاهده می کنید

Unity به فایل های کد به عنوان اسکریپت اشاره می کند، با استفاده از تعریفی از اسکریپت که معمولاً با جاوا اسکریپت در حال اجرا در مرورگر با آن مواجه می شویم: کد در موتور بازی Unity اجرا می شود، برخلاف کد کامپایل شده که به عنوان فایل اجرایی خودش اجرا می شود.

- اسکریپت ها در Unity بیشتر شبیه به کلاس های برنامه نویسی شیء گرا هستند و اسکریپت های متصل به اشیاء در صحنه نمونه های شیء هستند.





همه کامپوننت ها در یونیتی کامپوننت نیستند و فقط کلاس هایی که از کلاس **MonoBehaviour** ارث بری کرده اند کامپوننت متصل به شی میشوند کلاس **MonoBehaviour** ابزار و امکانات لازم برای اتصال یک کد به یک شی را فراهم می آورد دو تابع مهم که از این کلاس به ارث برده می شوند عبارتند از:

Start: که یکبار با فعال شدن شیء فراخوانی می شود (که معمولاً به محض بارگذاری صحنه با آن شیء است)

Update: در هر فریم فراخوانی میشود یعنی کدهای که در آن می نویسید در هر فریم فراخوانی می شود

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

← **Include namespaces for Unity and .NET/Mono classes.**

```
public class HelloWorld : MonoBehaviour {
```

← **The syntax for inheritance**

```
void Start() {
    // do something once
}
```

← **Put code here that runs once.**

```
void Update() {
    // do something every frame
}
}
```

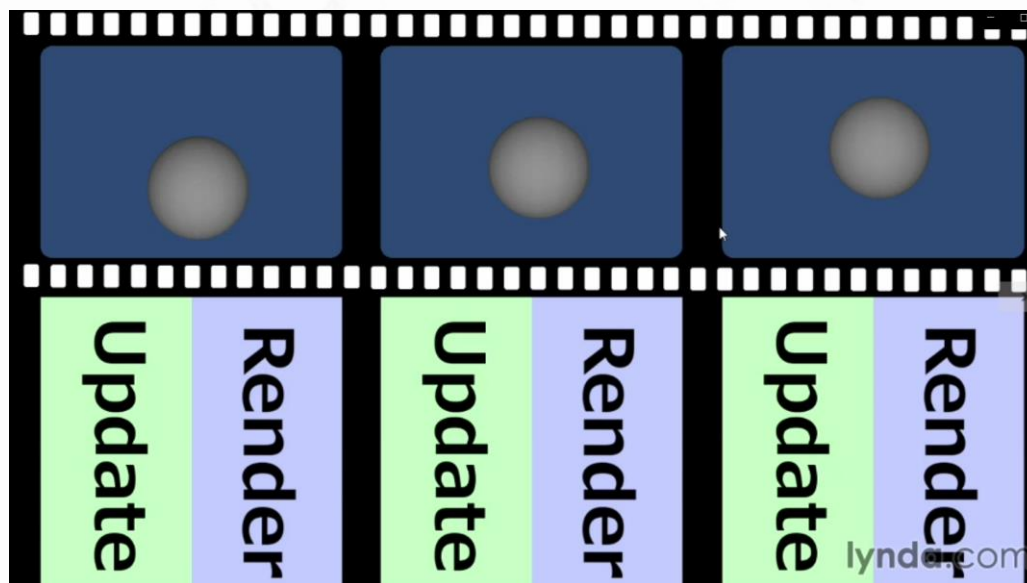
← **Put code here that runs every frame.**



فریم یک چرخه از حلقه کد بازی است. تقریباً همه بازی‌های ویدیویی (نه تنها در Unity، بلکه بازی‌های ویدیویی به طور کلی) دارای یک حلقه تکرار هستند.

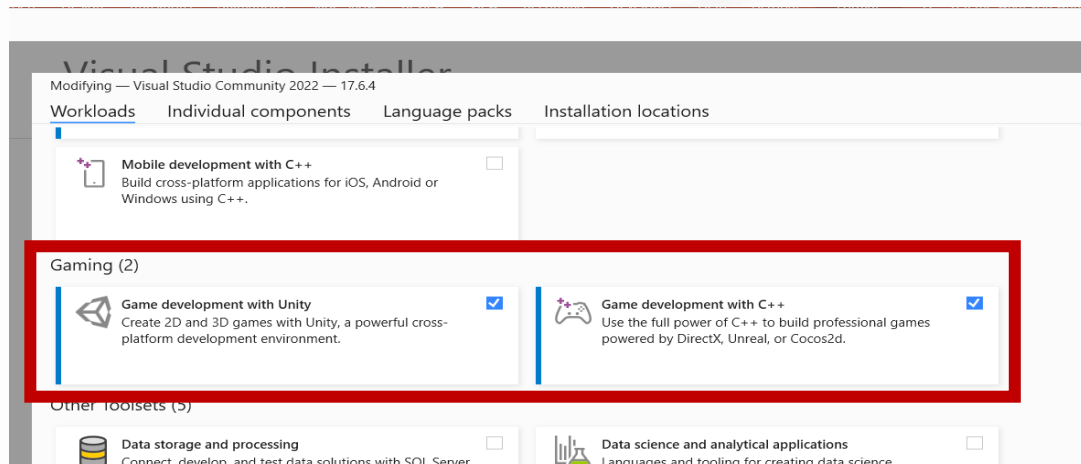
هر چرخه شامل ترسیم صفحه است - محاسبات انجام میشود و رندر تصویر خروجی انجام خواهد شد

Delta time and the update loop in Unity





Using Visual Studio, the included IDE



نصب ابزار یونیتی در ویژوال استودیو

تنظیم ویژوال استودیو به عنوان IDE پیش فرض در یونیتی

Go to Preferences > External Tools > External Script Editor to select an IDE.

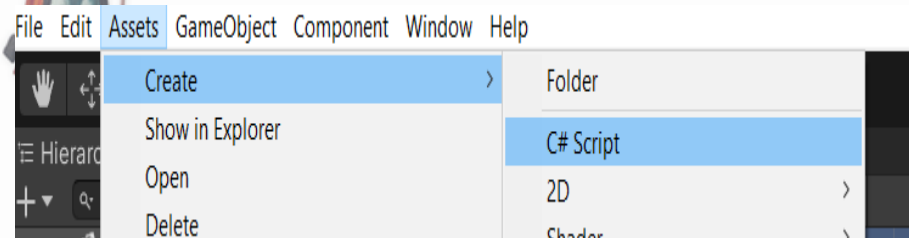
ویژوال استودیو فایل ها را در گروه هایی به نام راه حل (solution) سازماندهی می کند. یونیتی به طور خودکار راه حلی (solution) تولید می کند که همه فایل های اسکریپت را دارد، بنابراین معمولاً نیازی به نگرانی در مورد آن نیست.

همیشه به خاطر داشته باشید که اگرچه کد در ویژوال استودیو نوشته شده است، اما کد در آنجا اجرا نمی شود. IDE تقریباً یک ویرایشگر متن فانتزی است و زمانی که روی Play در یونیتی کلیک کنید، کد اجرا می شود.



تهیه نخستین اسکریپت

Printing to the console: Hello World!



ساخت فایل اسکریپت:

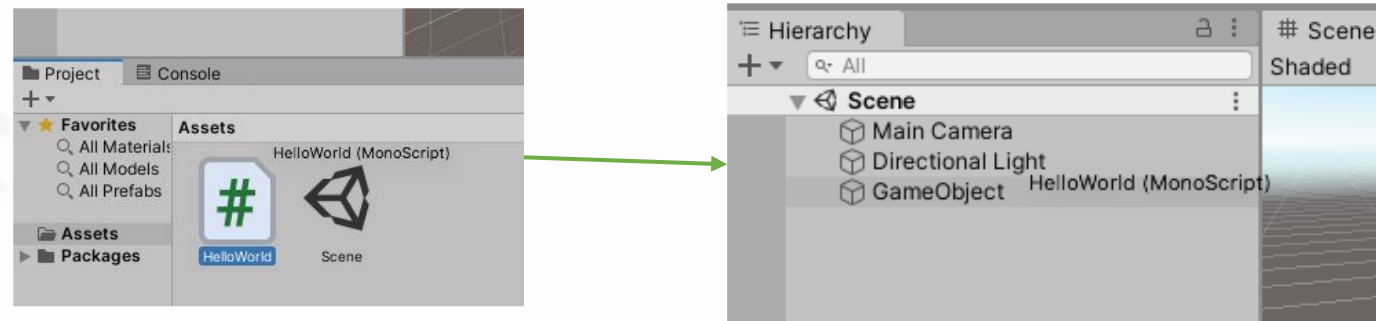
- برای ایجاد یک اسکریپت، **C# Script** را از منوی **Create** انتخاب کنید، که یا در منوی **Assets** به آن دسترسی دارید یا با کلیک راست در نمای پروژه می توانید به آن دسترسی داشته باشید.

(توجه داشته باشید که **Assets** و **GameObjects** هر دو فهرست‌هایی برای **Create** دارند، اما منوهای متفاوتی هستند)

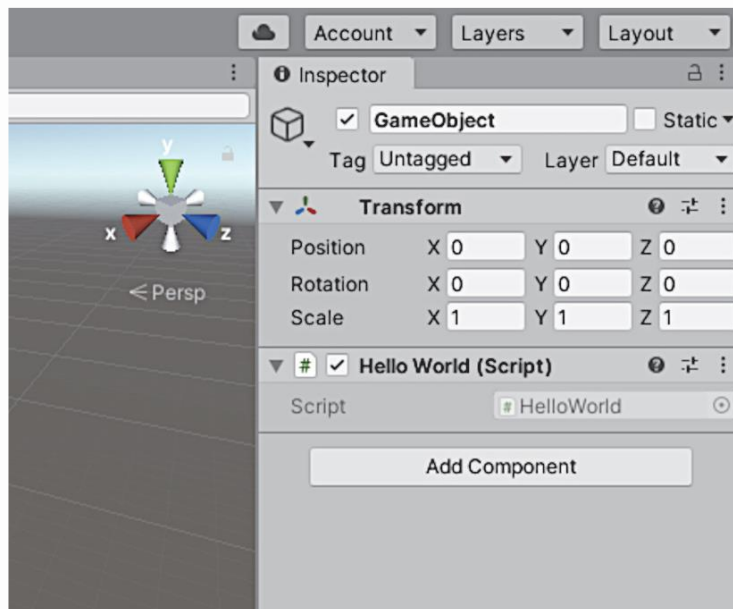
- یک نام برای اسکریپت جدید تایپ کنید، مانند **HelloWorld**

بسیار خوب، شما در حال حاضر یک اسکریپت خالی در پروژه دارید، اما برای پیوست کردن اسکریپت به یک شی در صحنه نیز نیاز دارید.

Choose **GameObject > Create Empty**, and a blank **GameObject** will appear in the **Hierarchy** list.



روی اسکریپت کلیک کرده و از نمای پروژه به سمت نمای سلسله مراتبی بکشید و در **GameObject** رها کنید.



هنگامی که یک اسکریپت به یک شی پیوند داده می شود، چیزی شبیه شکل را می بینید که اسکریپت به عنوان یک جزء در Inspector نشان داده می شود.

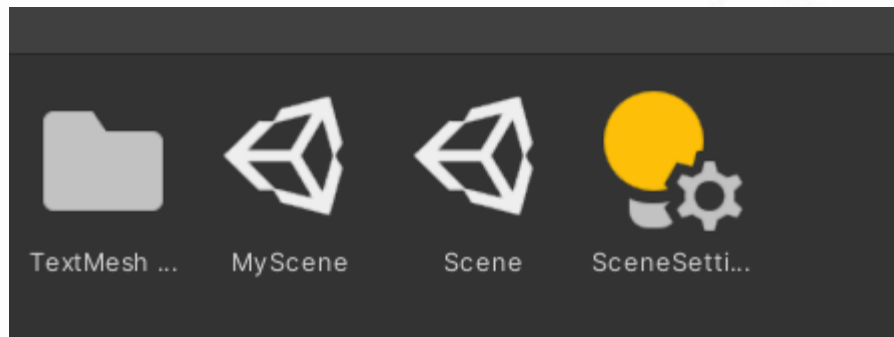
Add the logging command here.

```
void Start() {  
    Debug.Log("Hello World!");  
}
```

همیشه به یاد داشته باشید که پس از انجام تنظیمات روی یک اسکریپت، فایل را ذخیره کنید! یک اشتباه بسیار رایج این است که کد را تنظیم کنید و بلافاصله روی Play در Unity بدون ذخیره کلیک کنید، در نتیجه بازی همچنان از کدهای قبل از تنظیم آن استفاده می کند.



اکنون زمان ذخیره صحنه است. این یک فایل **unity** با نماد **Unity** ایجاد می کند. فایل صحنه یک عکس فوری از همه چیزهایی است که در حال حاضر در بازی بارگذاری شده است تا بتوانید بعداً این صحنه را دوباره بارگذاری کنید. ذخیره این صحنه ممکن است به سختی ارزشمند به نظر برسد زیرا بسیار ساده است (یک **GameObject** خالی) - اما اگر صحنه را ذخیره نکنید، پس از خروج از یونیتی، دوباره آن را خالی خواهید یافت.



برای اینکه ببینید **Unity** چگونه خطاها را نشان می دهد، عمداً یک اشتباه تایپی در اسکریپت **HelloWorld** قرار دهید. به عنوان مثال، اگر نماد پرانتز اضافی را تایپ کنید، یک پیام خطا در تب **Console** با نماد خطای قرمز ظاهر می شود.

