

مقدمه‌ای بر برنامه‌نویسی بازی‌های 3D با

DirectX

12

مقدمه ای بر برنامه نویسی بازی های سه بعدی با Direct3D 12

نویسنده: علی بدیعی

تابستان ۱۴۰۲

قبل از شروع ...

کتابی پیش رو دارید، مباحث کتاب Introduction to 3d Game Programming With Directx 12 را در برمیگرد. در این کتاب از ترجمه‌ی محض خارج شده و موارد را به صورت گستردگرتر بررسی می‌کند و هر مرحله را قدم به قدم توضیح می‌دهد. در اوایل ترجمه‌ی کتاب، نظر به این بود تا کتاب ترجمه شود ولی با توجه به اینکه منابع فارسی برای مباحث کتاب بسیار کم بوده و حتی نایاب است، تصمیم بر آن شد که اطلاعات و مباحث اضافی نیز به آن اضافه گردد. همچنین در تمامی کتاب سعی شده تا موارد تا جای ممکن به اصل خود تشابه داشته باشند. در هر صورت تمامی منابع استفاده شده و همه موارد نیاز این منابع به صورت یکجا برای شما در لینک زیر قرار داده شده است. می‌توانید با اسکن QR Code زیر نیز، آن‌ها را مشاهده کنید.



<https://alibadiee.ir/Book/3d-DirectX12>

این کتاب به صورت رایگان منتشر می‌شود و بجز برای خرید نسخه‌ی چاپی آن، شما نیاز به

پرداخت هزینه‌ای ندارید. همچنین در صورتی که تمایل دارید، می‌توانید در بهتر کردن این اثر به ما کمک کنید. برای دریافت راهنمای لینک QR Code زیر استفاده کنید.



<https://alibadiee.ir/Book/3d-DirectX12/Help-Editing>

معرفی

Direct3D 12 یک کتابخانه رندر (render) برای نوشتن برنامه های گرافیکی سه بعدی با عملکرد بالا (high-performance) با استفاده از سخت افزار گرافیکی مدرن بر روی پلتفرم های مختلف ویندوز ۱۰ (Windows Desktop - Mobile - Xbox One) است. یک کتابخانه سطح پایین است، به این معنا که رابط برنامه نویسی کاربردی آن (API) مدل سازی سخت افزار گرافیکی زیرینی را که کنترل می کند، بر عهده دارد. بیشترین استفاده ای Direct3D صنعت بازی سازی است که در آن موتورهای رندرینگ سطح بالاتر بر روی Direct3D ساخته می شوند. با این حال، صنایع دیگر به گرافیک سه بعدی تعاملی با عملکرد بالا نیز نیاز دارند، مانند: تجسم پزشکی و علمی و بررسی های معماری. علاوه بر این، با مجهز شدن هر رایانه شخصی جدید به یک کارت گرافیک مدرن، برنامه هایی که سه بعدی نیستند، شروع به استفاده از GPU (واحد پردازش گرافیکی) برای سپردن محاسبات فشرده به کارت گرافیک کردند. این کار به عنوان *GPU general purpose computing* شناخته می شود و رابط برنامه نویسی کاربردی Direct3D shader، محاسباتی را برای نوشتن برنامه های GPU با اهداف عمومی ارائه می دهد. اگرچه Direct3D 12 معمولاً با C++ برنامه نویسی می شود، تیم SharpDX در حال کار بر روی بسته های .NET هستند تا بتوانند از برنامه های مدیریت شده به این API گرافیکی سه بعدی قدرتمند دسترسی داشته باشید.

این کتاب مقدمه ای بر برنامه نویسی گرافیک کامپیوتری تعاملی با استفاده از 12 Direct3D

و تأکید بر توسعه بازی است. اصول برنامه نویسی Direct3D و shader را آموزش می دهد و پس از آن خواننده آماده می شود تا تکنیک های پیشرفته تری را یاد بگیرد. کتاب به سه بخش اصلی تقسیم شده است. بخش اول ابزارهای ریاضی را توضیح می دهد که در این کتاب استفاده خواهد شد. بخش دوم نحوه پیاده سازی و ظایف اساسی در Direct3D مانند: مقداردهی اولیه، تعریف هندسه سه بعدی، راه اندازی دوربین ها، ایجاد رئوس، ایجاد پیکسل، ایجاد هندسه و محاسبه سایه زن، نور پردازی، بافت سازی، مخلوط کردن، شابلون سازی، و مفروش سازی را نشان می دهد. بخش سوم عمدهاً در مورد استفاده از Direct3D برای پیاده سازی انواع تکنیک های جالب و جلوه های ویژه است، مانند کار با مش شخصیت های متحرک، برداشتن، نگاشت محیط، نقشه برداری معمولی، سایه های بلادرنگ، و انسداد محیط.

افراد مبتدی، بهتر است این کتاب را از ابتدا به انتهایا به صورت کامل بخوانند. فصل ها به گونه ای مرتب شده اند که با گذر از هر فصل، میزان سختی به تدریج افزایش می یابد. به این ترتیب، هیچ جهش ناگهانی در پیچیدگی متنها وجود ندارد که خواننده را سردرگم کند. به طور کلی، برای هر فصل، از تکنیک ها و مفاهیمی که در فصل های قبل گفته شده اند استفاده خواهیم کرد. بنابراین، مهم است که بر مطالب هر فصل تسلط داشته باشید. خواننده کانی که تجربه ای استفاده از Direct3D را دارند، می توانند فصل های مورد علاقه خود را بخوانند. در نهایت، شاید از خود بپرسید که بعد از خواندن این کتاب چه نوع بازی هایی را می توانید توسعه دهید، پاسخ این سؤال را بهتر است با مرور این کتاب و مشاهده انواع برنامه های توسعه یافته به دست آورید. از این رو می توانید انواع بازی های را بر اساس تکنیک های آموزش داده شده در این کتاب و نبوغ خود تجسم کنید.

مخاطبان کتاب

این کتاب با در نظر گرفتن سه مخاطب زیر طراحی شده است:

۱. برنامه نویسان سطح متوسط C++ که می خواهند مقدمه ای بر برنامه نویسی سه بعدی با استفاده از آخرین نسخه Direct3D داشته باشند
۲. برنامه نویسان سه بعدی با تجربه که با API ای غیر از DirectX (به عنوان مثال، OpenGL) کار کرده اند و می خواهند مقدمه ای برای Direct3D 12 داشته باشند
۳. برنامه نویسان با تجربه Direct3D که مایل به یادگیری آخرین نسخه Direct3D هستند

پیش نیاز ها

لازم به ذکر است که این مقدمه ای بر 12 Direct3D، برنامه نویسی shader و برنامه نویسی بازی های سه بعدی است و نمی توان از آن برای یادگیری برنامه نویسی کامپیوتر به صورت کلی استفاده کرد. بنابراین خواننده باید شرایط زیر را رعایت کند:

۱. ریاضیات دبیرستان: مثلاً جبر، مثلثات و توابع ریاضی
۲. تجربه کار با Visual Studio: چگونگی ایجاد پروژه ها، اضافه کردن فایل ها و link کردن کتابخانه های خارجی
۳. C++ متوسط و مهارت های ساختار داده: به عنوان مثال: تجربه کار با اشاره گرها، آرایه ها، سربار اپراتورها، لیست های پیوندی، وراثت و چند ریختی
۴. آشنایی با برنامه نویسی ویندوز با Win32 API (مفید است اما لازم نیست)

ابزارها و سخت افزارهای توسعه مورد نیاز

برای برنامه نویسی برنامه های Direct3D 12 به حداقل موارد زیر نیاز است:

۱. Windows 10 .
۲. Visual Studio 2015 یا جدیدتر
۳. کارت گرافیکی که از 12 Direct3D پشتیبانی کند

نکته

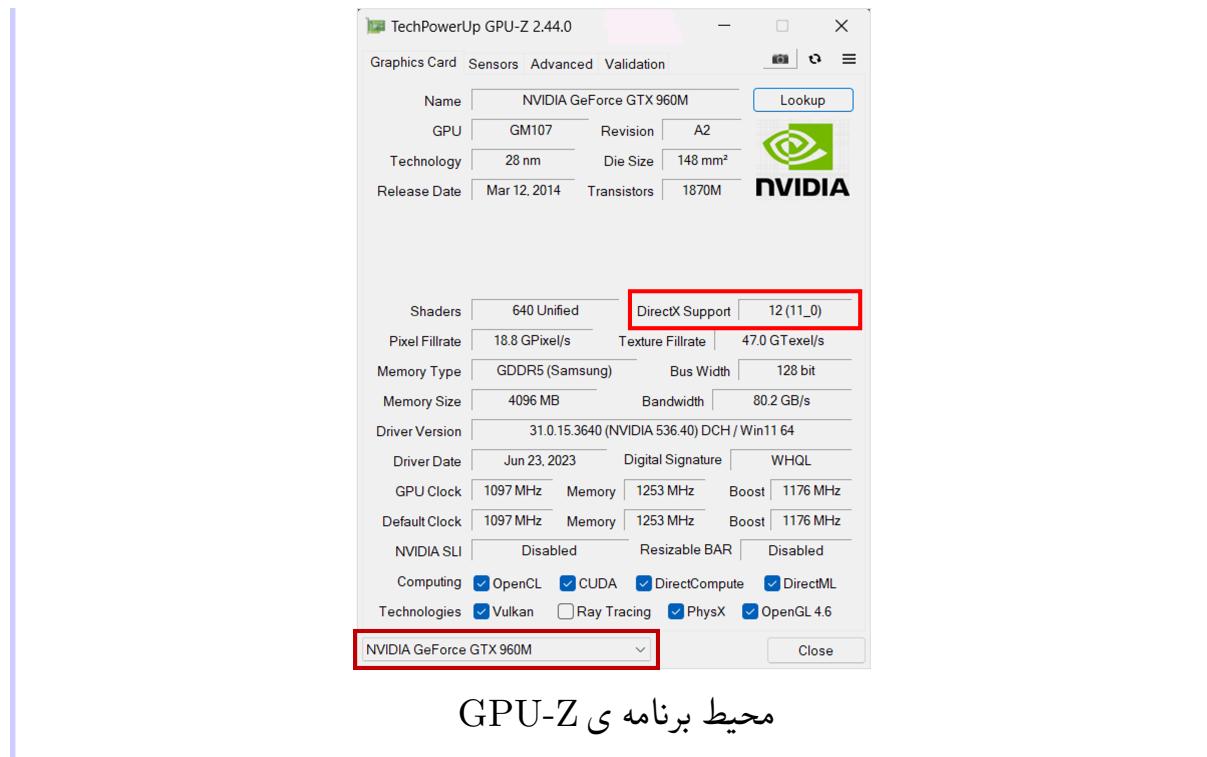
کانفیگ های استفاده شده از این کتاب طبق موارد زیر است:

1. Windows 11 Pro 22H2 Build 22621.1992
2. NVIDIA GeForce GTX 960m
3. Visual Studio 2022 (64-bit) Version 17.1.2

نکته

برای نصب و استفاده از 11 Windows و Visual Studio 2022 می توانید از آموزش های اینترنت استفاده کنید.

برای بررسی پشتیبانی کارت گرافیک از 12 DirectX می توانید مدل کارت گرافیک خود را در اینترنت جست و جو کنید یا از برنامه GPU-Z استفاده کنید. عکس زیر نحوه استفاده از برنامه را نشان می دهد.



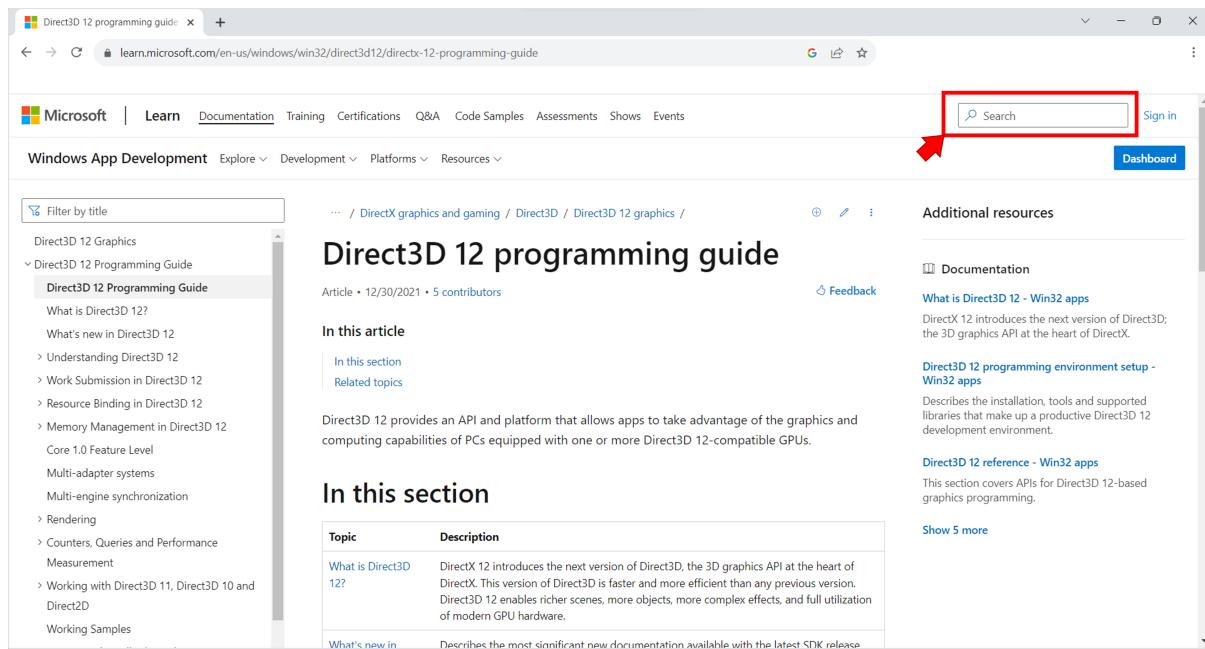
استفاده از اسناد SDK و نمونه‌های DIRECTX

SDK و نمونه‌های DirectX یک API بزرگ است و ما نمی‌توانیم تمام جزئیات آن را در این یک کتاب پوشش دهیم. بنابراین، برای به دست آوردن اطلاعات بیشتر، ضروری است نحوه استفاده از اسناد DirectX را یاد بگیرید. به روزترین اسناد در [MSDN](#) در دسترس خواهد بود.



<https://learn.microsoft.com/en-us/windows/win32/direct3d12/directx-12-programming-guide>

شکل ۱ تصویری از مستندات آنلاین را نشان می‌دهد. اسناد DirectX تقریباً همه‌ی بخش‌های DirectX را پوشش می‌دهد. بنابراین این اسناد می‌تواند یک مرجع بسیار مفید باشد، اما از



شکل ۱: راهنمای برنامه نویسی Direct3D در مستندات DirectX

آنچایی که مستندات آنچنان وارد جزئیات نمی‌شوند، می‌توان گفت که این اسناد بهترین ابزار یادگیری نیستند. با این حال، با هر نسخه جدید DirectX که منتشر شود، اسناد نیز بهتر و بهتر می‌شود.

همانطور که گفته شد، اسناد به عنوان یک مرجع قابل استفاده هستند. فرض کنید با یک نوع یا تابع مرتبط با DirectX مواجه شده‌اید که اطلاعات بیشتری در مورد آن می‌خواهید، به عنوان مثال تابع `ID3D12Device::CreateCommittedResource`. شما می‌توانید به سادگی در مستندات جستجو کنید و توضیحاتی از نوع آن شی یا در این مثال، توضیحاتی در مورد تابع دریافت کنید. شکل ۲ را ببینید.

نکته

در این کتاب هر از گاهی برای جزئیات بیشتر شما را به مستندات راهنمایی می‌کنیم.

همچنین برنامه های نمونه 12 Direct3D که به صورت آنلاین در دسترس هستند را میتوانید در لینک زیر مشاهده کنید:

The screenshot shows a Microsoft Learn API documentation page for the `ID3D12Device::CreateCommittedResource` method. The URL is learn.microsoft.com/en-us/windows/win32/api/d3d12/nf-d3d12-id3d12device-createcommittedresource. The page includes a sidebar with navigation links like Microsoft, Learn, Documentation, Training, Certifications, Q&A, Code Samples, Assessments, Shows, Events, and a search bar. The main content area displays the method's name, last updated date (10/13/2021), and a 'Feedback' link. It also lists related methods such as CreateComputePipelineState, CreateConstantBufferView, CreateDepthStencilView, CreateDescriptorHeap, CreateFence, CreateGraphicsPipelineState, CreateHeap, CreatePlacedResource, CreateQueryHeap, and CreateRenderTargetView. Below the methods list is a detailed description of the `CreateCommittedResource` method, which creates both a resource and an implicit heap. To the right, there is an 'Additional resources' section with links to 'Documentation', 'Creating and recording command lists and bundles - Win32 apps', 'D3D12_HEAP_TYPE (d3d12.h) - Win32 apps', and 'D3D12_HEAP_PROPERTIES (d3d12.h) - Win32 apps'. A 'Show 5 more' link is also present.

شکل ۲: دریافت مستندات یک تابع



<https://github.com/Microsoft/DirectX-Graphics-Samples>

میتوانید نمونه های Direct3D 12، NVIDIA، AMD و Intel را در وب سایت های مشاهده کنید. همچنان در آینده نمونه های بیشتری نیز خواهد آمد.

شفاف سازی

اگرچه ما تلاش می کنیم که کدهای کارآمدی بنویسیم و هم زمان از بهترین شیوه های برنامه نویسی Direct3D استفاده کنیم، اما هدف اصلی هر برنامه نمونه، نشان دادن مفاهیم Direct3D 12 تکنیک های برنامه نویسی گرافیکی است. هدف ما نوشتن بهینه ترین کد نیست زیرا به احتمال زیاد، این کارایدهایی را که تلاش می کردیم به تصویر بکشیم را مبهم می کرد. اگر از کد نمونه در پروژه های خود استفاده می کنید، این نکته را در نظر داشته باشید، زیرا باید برای کارایی بهتر دوباره روی آن کار کنید. علاوه بر این، به منظور تمرکز بر API Direct3D، زیرساخت های حداقلی را در بالای Direct3D ایجاد کرده ایم. به این معنی که ما مقادیر را هاردکد (hardcode) کردیم و چیزهایی را در کد منبع تعریف کردیم که ممکن است مبنی بر داده باشند. در یک برنامه بزرگ سه بعدی، احتمالاً یک موتور رندرینگ در بالای Direct3D پیاده سازی خواهد کرد. با این حال، موضوع این کتاب Direct3D API است، نه طراحی موتورهای رندرینگ.

نمونه برنامه ها و مکمل های آنلاین

وب سایت های این کتاب (www.merclearning.com و www.d3dcoder.net) نقش مهمی در استفاده حداکثری از این کتاب دارد. [متوجه: همه لینک ها و منابع در سایت من به صورت یکجا وجود دارند و نیازی نیست تمامی لینک های کتاب را ذخیره یا باز کنید.] در وب سایت شما کد منبع کامل و فایل های پروژه برای هر نمونه در این کتاب را خواهید یافت. در بسیاری از موارد، برنامه های DirectX آنقدر بزرگ هستند که نمی توانند به طور کامل در یک کتاب متنی جاسازی شوند. بنابراین، ما فقط قطعات کد مربوطه را بر اساس ایده هایی که نشان داده می شوند نشان می دهیم. اکیداً توصیه می شود که خواننده کد آزمایشی مربوطه را مطالعه کند تا برنامه را به طور کامل ببیند (هدف ما این است که آزمایشی ها (demos) را برای مطالعه آسان کوچک و متمرکز کنیم). به عنوان یک قاعده کلی، خواننده باید بتواند پس از خواندن فصل و گذراندن مدتی برای مطالعه کد نسخه آزمایشی، کد نسخه (ها)ی آزمایشی فصل را به تنها یی پیاده سازی کند. در واقع اینکه سعی

کنید نمونه ها را خودتان با استفاده از کتاب و کد نمونه (به عنوان مرجع) پیاده سازی کنید، یک تمرین بسیار خوب است.

راه اندازی پروژه آزمایشی در Visual Studio 2022

آزمایشی های این کتاب را می توانید به سادگی با دوبار کلیک کردن روی فایل پروژه مربوطه (vcxproj) یا فایل (.sln) باز کنید. این بخش نحوی باز کردن یا ایجاد و ساخت یک پروژه را از ابتدا با استفاده از چارچوب برنامه آزمایشی کتاب با استفاده از Visual Studio 2022 توضیح می دهد. به عنوان یک مثال کاربردی، نحوه بازآفرینی و build کردن نسخه آزمایشی "Box" از فصل ۶ را نشان خواهیم داد.

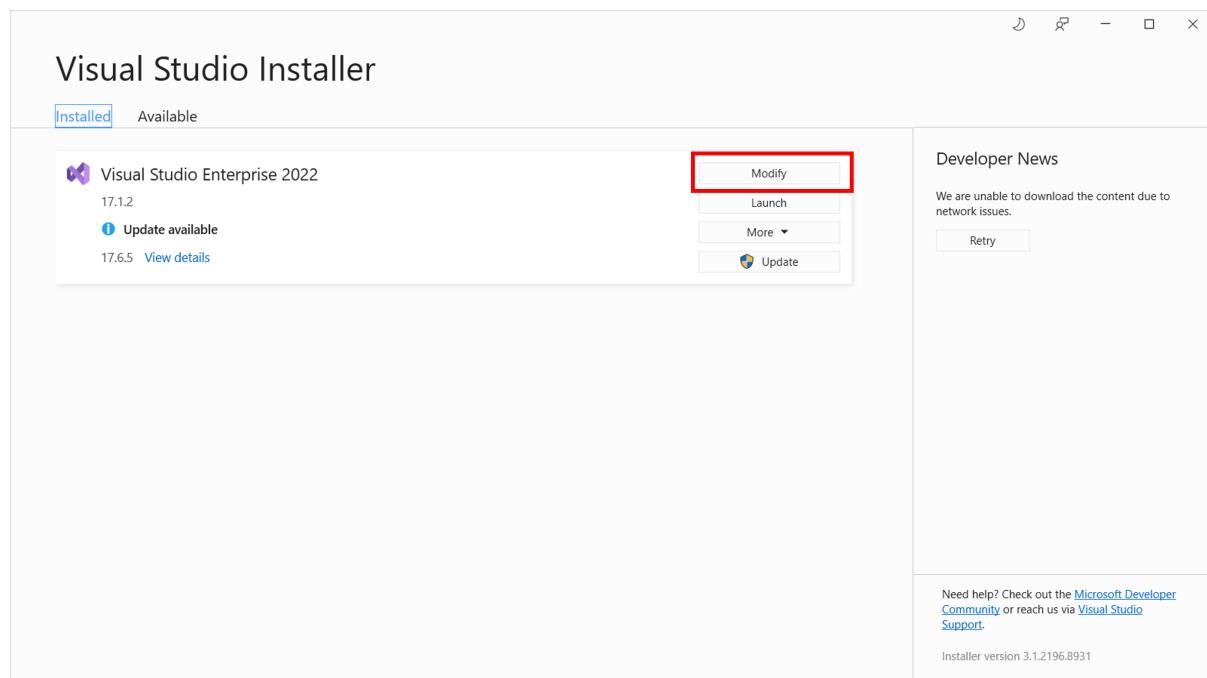
کد منبع کتاب را دانلود کنید

ابتدا کد منبع کتاب را در پوشه‌ای در هارد دیسک خود دانلود کنید. فرض می‌کنیم این پوشه C:\d3d12book است. در پوشه کد منبع، فهرستی از پوشه‌ها از هر فصل را خواهید دید. هر پوشه شامل پروژه‌های کد برای فصل داده شده است. همچنین به پوشه‌ای به نام Common توجه کنید. این پوشه حاوی کد منبع به اشتراک گذاشته شده است که در تمام پروژه‌های آزمایشی استفاده مجدد می‌شود. اکنون، در پوشه کد منبع، یک پوشه جدید ایجاد کنید که می‌خواهید آزمایشی‌های خود را در آن ذخیره کنید. برای مثال، C:\d3d12book\MyDemos. این پوشه جایی است که می‌توانید پروژه‌های جدید را بر اساس چارچوب نمونه‌ی کتاب ایجاد کنید.

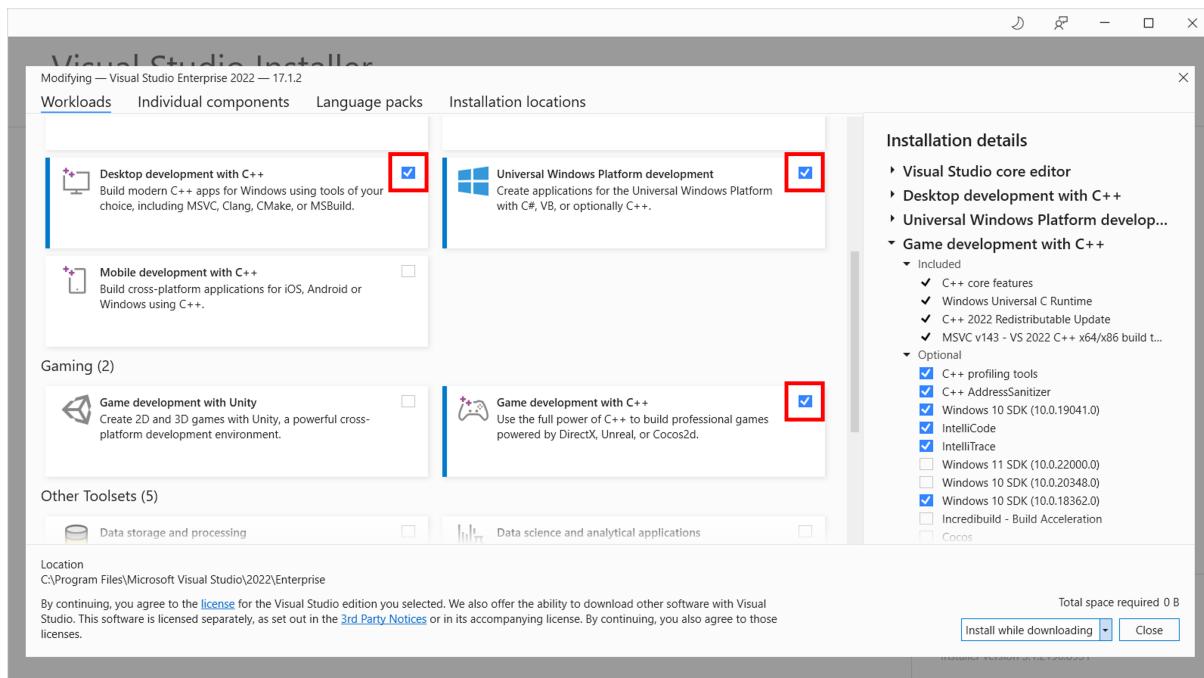
این ساختار پوشه‌بندی ضروری نیست، اما ساختاری است که آزمایشی‌های کتاب از آن پیروی می‌کنند. اگر با تنظیم مسیرهای اضافی راحت هستید، می‌توانید پروژه‌های آزمایشی خود را در هر مکانی قرار دهید و Visual Studio را تنظیم کنید که کد منبع را در دایرکتوری Common پیدا کند.

موارد مورد نیاز Visual Studio را نصب کنید

در صورتی که Visual Studio را نصب نکرده اید، ابتدا آن را با استفاده از راهنمای آنلاین نصب کنید. پس از نصب آن، ابتدا Visual Studio Installer را باز کنید. از پنجرهی باز شده، Install یا Modify را کلیک کنید (شکل ۳). در سمت چپ پنجرهی باز شده، از قسمت Universal Windows Platform development ، Desktop & Mobile تیک قسمتهای Desktop development with C++ و Game development with C++ نیز تیک + Gaming را بزنید (شکل ۴).



شکل ۳: محیط Visual Studio Installer



شکل ۴: مواردی که باید نصب شوند

بعد از انتخاب موارد بالا، باید موارد اضافه‌تری را نیز انتخاب کنید. این موارد شامل SDK‌های ویندوز و ورژن‌های جدید C++ است. در سمت راست پنجره‌ی باز شده، می‌توانید این موارد را ببینید. برای اجرای بهتر، سعی کنید موارد را مانند عکس‌ها رعایت کنید. پس از آن در پایین پنجره روی گزینه‌ی Install کلیک کنید تا موارد انتخاب شده نصب شوند.

The screenshots show the 'Optional' components section of the Visual Studio 2022 installer, highlighting specific items for each development path:

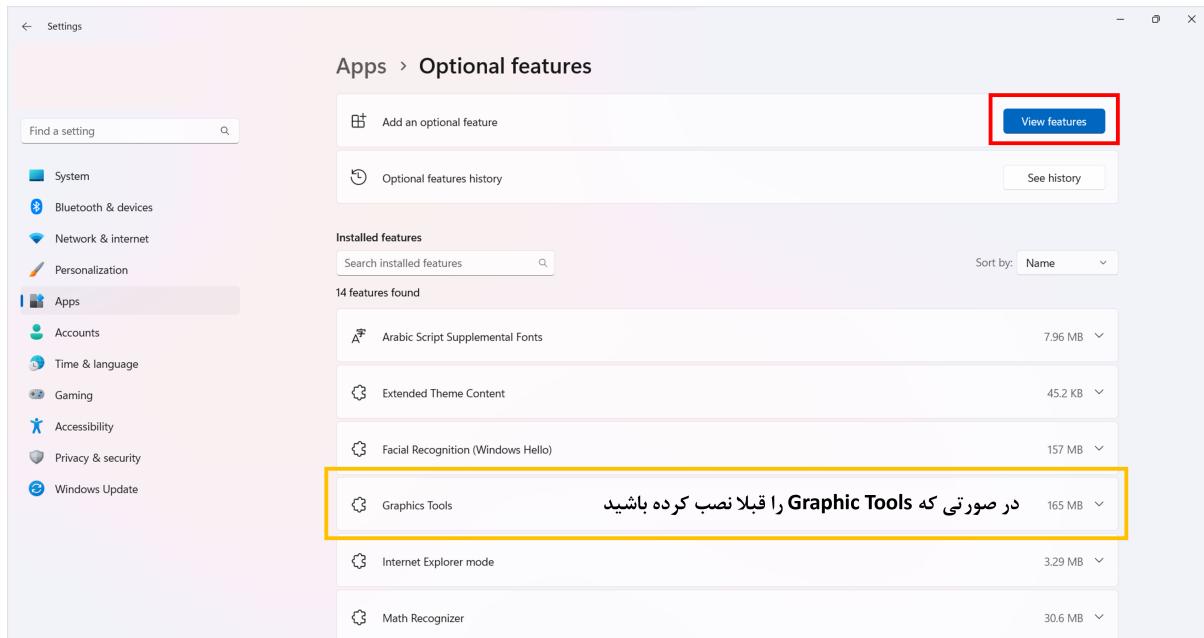
- Universal Windows Platform development with C++:** Shows checked boxes for MSVC v143, Windows 10 SDK (10.0.18362.0), and MSVC v141.
- Desktop development with C++:** Shows checked boxes for MSVC v143, Windows 10 SDK (10.0.19041.0), Just-In-Time debugger, C++ profiling tools, C++ CMake tools for Windows, C++ ATL for latest v143 build tools, Test Adapter for Boost.Test, Test Adapter for Google Test, Live Share, IntelliCode, C++ AddressSanitizer, IntelliTrace, and MSVC v143.
- Universal Windows Platform development with C++:** Shows checked boxes for IntelliCode, IntelliTrace, USB Device Connectivity, C++ (v143) Universal Windows Platform tools, C++ (v141) Universal Windows Platform tools, Graphics debugger and GPU profiler for DirectX, Windows 10 SDK (10.0.18362.0), and Windows 10 SDK (10.0.19041.0).
- Game development with C++:** Shows checked boxes for C++ profiling tools, C++ AddressSanitizer, Windows 10 SDK (10.0.19041.0), IntelliCode, IntelliTrace, Windows 10 SDK (10.0.18362.0), Windows 11 SDK (10.0.22000.0), Windows 10 SDK (10.0.20348.0), and Windows 10 SDK (10.0.18362.0).

مواردی که باید نصب شوند.

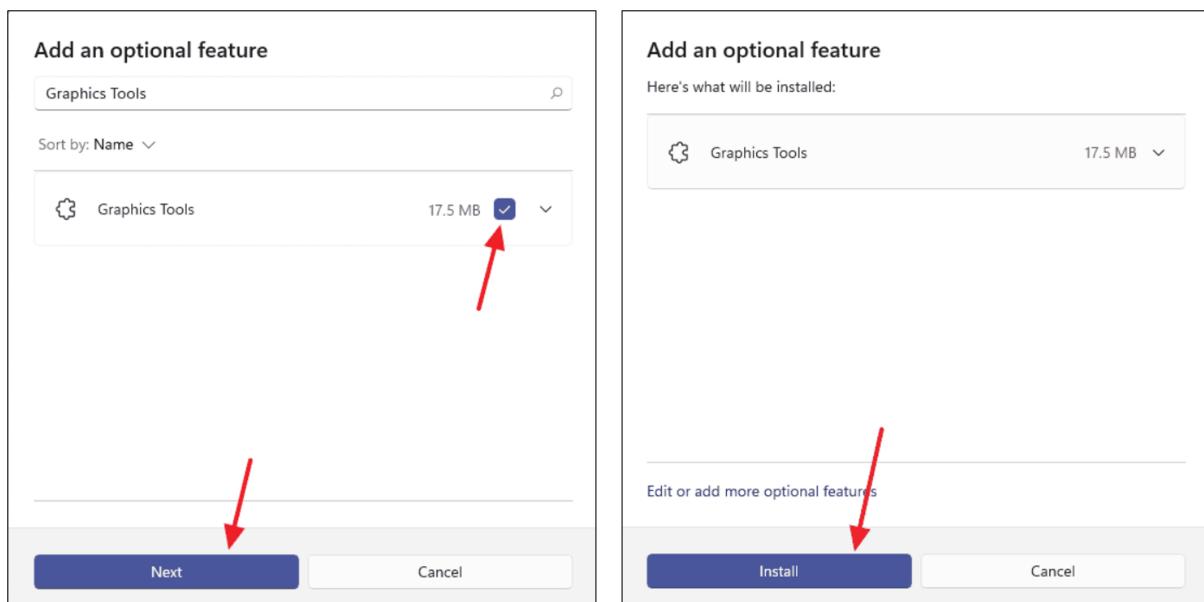
فعال کردن Graphic Tools

برای اجرای درست برنامه های آزمایشی DirectX 12، نیاز است که قابلیت Graphic Tools را فعال کنید. برای این کار می توانید به دو صورت عمل کنید:

روش اول) ابتدا در منوی استارت، Optional features را سرچ کرده و آن را باز کنید. اگر از قبل بر روی دستگاه شما نصب شده باشد، می توانید از قسمت پایین پنجره آن Graphic Tools را مشاهده کنید (کادر زرد رنگ) در غیر این صورت بر روی View features کلیک کرده و نام آن را سرچ کرده و نصب کنید (شکل ۵ - ۶).



شکل ۵: محیط Optional features



شکل ۶: نصب Graphic Tools

روش دوم) اگر به هر دلیلی روش اول برای شما قابل انجام نبود، می توانید پنجرهی Command Prompt را به صورت Run as administrator (CMD) باز کرده و با دستورات زیر آن را

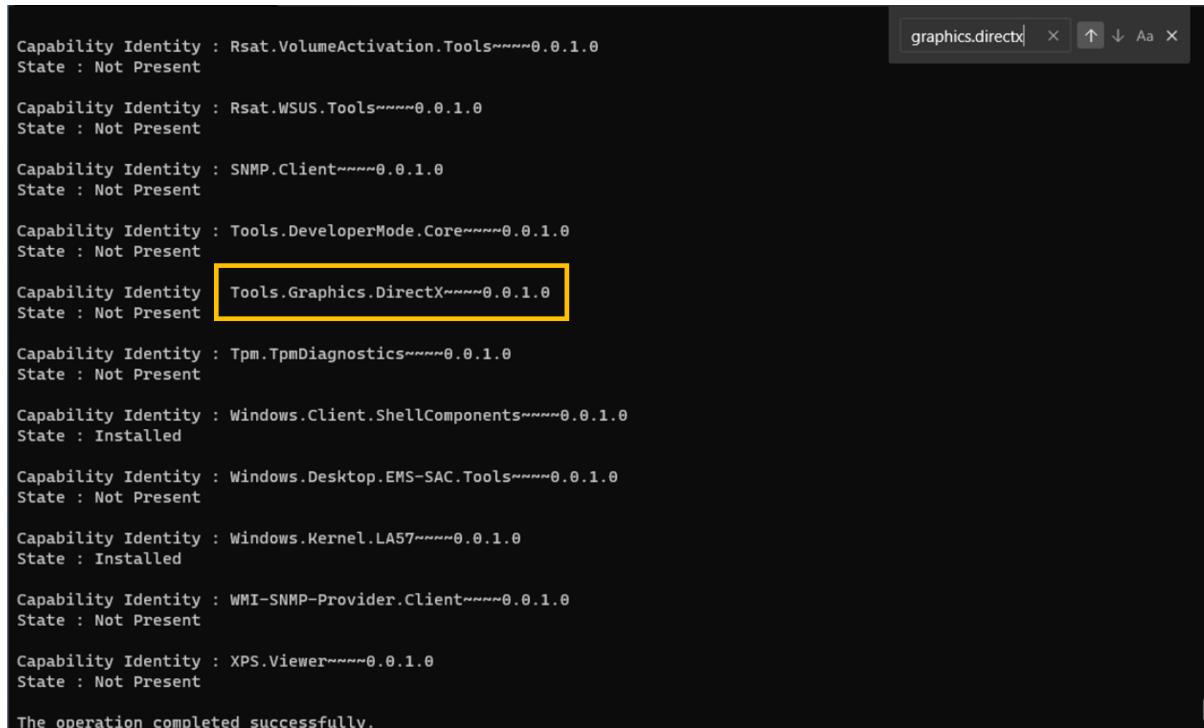
نصب کنید.

ابتدا دستور داخل CMD عبارت زیر را وارد کنید:

Dism /Online /Get-Packages /Format:Table

یا با دستور زیر:

dism /online /Get-Capabilities



```

Capability Identity : Rsat.VolumeActivation.Tools~~~~~0.0.1.0
State : Not Present

Capability Identity : Rsat.Wsus.Tools~~~~~0.0.1.0
State : Not Present

Capability Identity : Snmp.Client~~~~~0.0.1.0
State : Not Present

Capability Identity : Tools.DeveloperMode.Core~~~~~0.0.1.0
State : Not Present

Capability Identity : Tools.Graphics.DirectX~~~~~0.0.1.0
State : Not Present

Capability Identity : Tpm.TpmDiagnostics~~~~~0.0.1.0
State : Not Present

Capability Identity : Windows.Client.ShellComponents~~~~~0.0.1.0
State : Installed

Capability Identity : Windows.Desktop.Ems-Sac.Tools~~~~~0.0.1.0
State : Not Present

Capability Identity : Windows.Kernel.La57~~~~~0.0.1.0
State : Installed

Capability Identity : WMI-Snmp-Provider.Client~~~~~0.0.1.0
State : Not Present

Capability Identity : Xps.Viewer~~~~~0.0.1.0
State : Not Present

The operation completed successfully.
  
```

dism /online /Get-Capabilities

از لیست نشان داده شده، موردی را که با Tools.Graphics.DirectX شروع می شود را پیدا کرده و کامل نام آن را کپی کنید. (ممکن است اعداد مقابل آن با عکس متفاوت باشد)

سپس دستور زیر را وارد کنید و به جای <Name>، نامی که کپی کردید را قرار دهید.

Dism /Online /Add-Capability /CapabilityName:<Name>

به عنوان مثال:

```
Dism /Online /Add-Capability /CapabilityName:Tools.Graphics.DirectX~~~~0.0.1.0
```

```
PS C:\Users\xbox> dism /online /add-capability /capabilityname:Tools.Graphics.DirectX~~~~0.0.1.0
Deployment Image Servicing and Management tool
Version: 10.0.22000.1

Image Version: 10.0.22000.469

[=====100.0%=====]
The operation completed successfully.
```

Dism /Online /Add-Capability /CapabilityName:Tools.Graphics.DirectX~~~~0.0.1.0

نکته

دقت کنید که فرآیند نصب ممکن است زمان بر باشد. همچنین دستورات Dism به بزرگ و کوچک بودن حروف حساس نیستند.

یک پروژه جدید ایجاد کنید

ابتدا Visual Studio 2022 را اجرا کنید، میتوانید با انتخاب Create a new project را اجرا کنید، یک پروژه جدید بسازید (شکل ۷) یا به منوی اصلی بروید و File->New->Project را انتخاب کنید (شکل ۸). پنجره‌ی New Project ظاهر می‌شود (شکل ۹). از قسمت بالا C++ و Windows را انتخاب کنید و روی Empty Project مانند شکل کلیک کرده و Next را بزنید. در مرحله بعد به پروژه یک نام بدهید و مکانی را که می‌خواهید دایرکتوری پروژه در آن ذخیره شود را مشخص کنید (شکل ۱۰). حالا Create را بزنید.

Visual Studio 2022

Open recent

Get started

- ▷ Today
- ▷ This week
- ▷ This month



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder

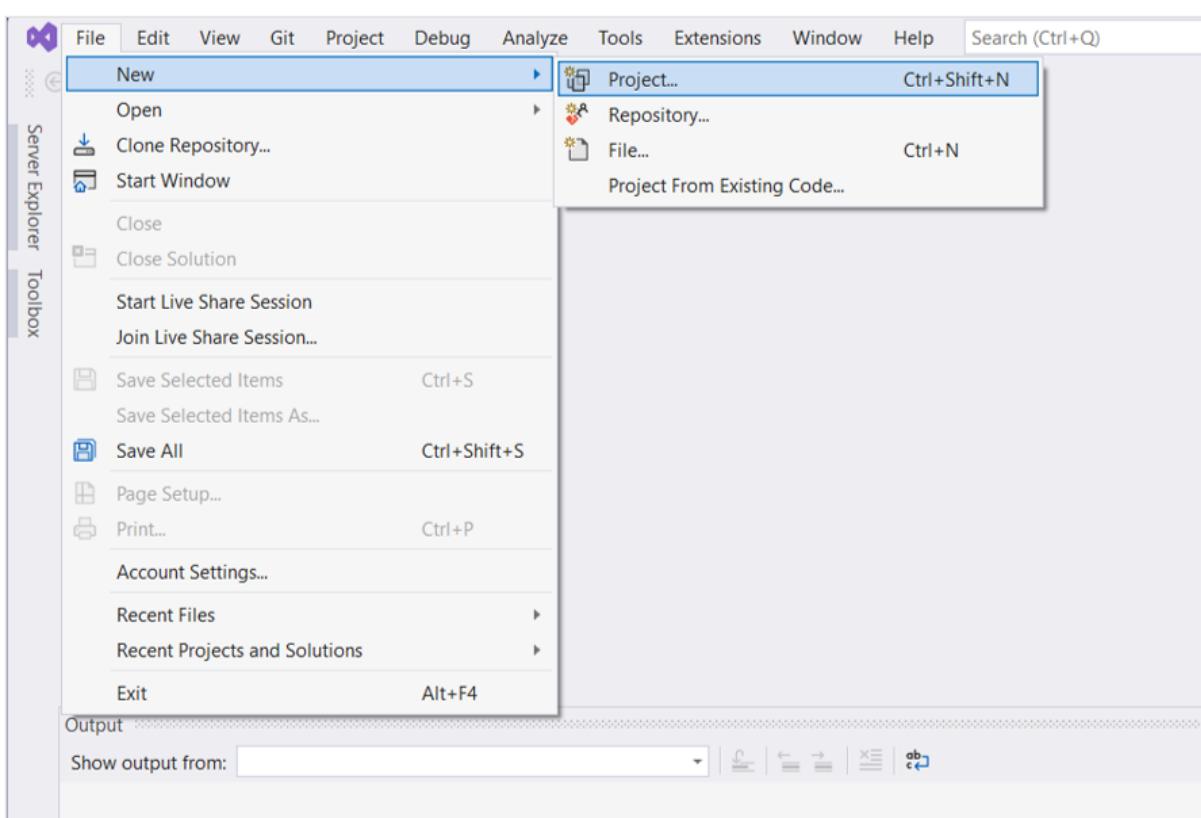


Create a new project

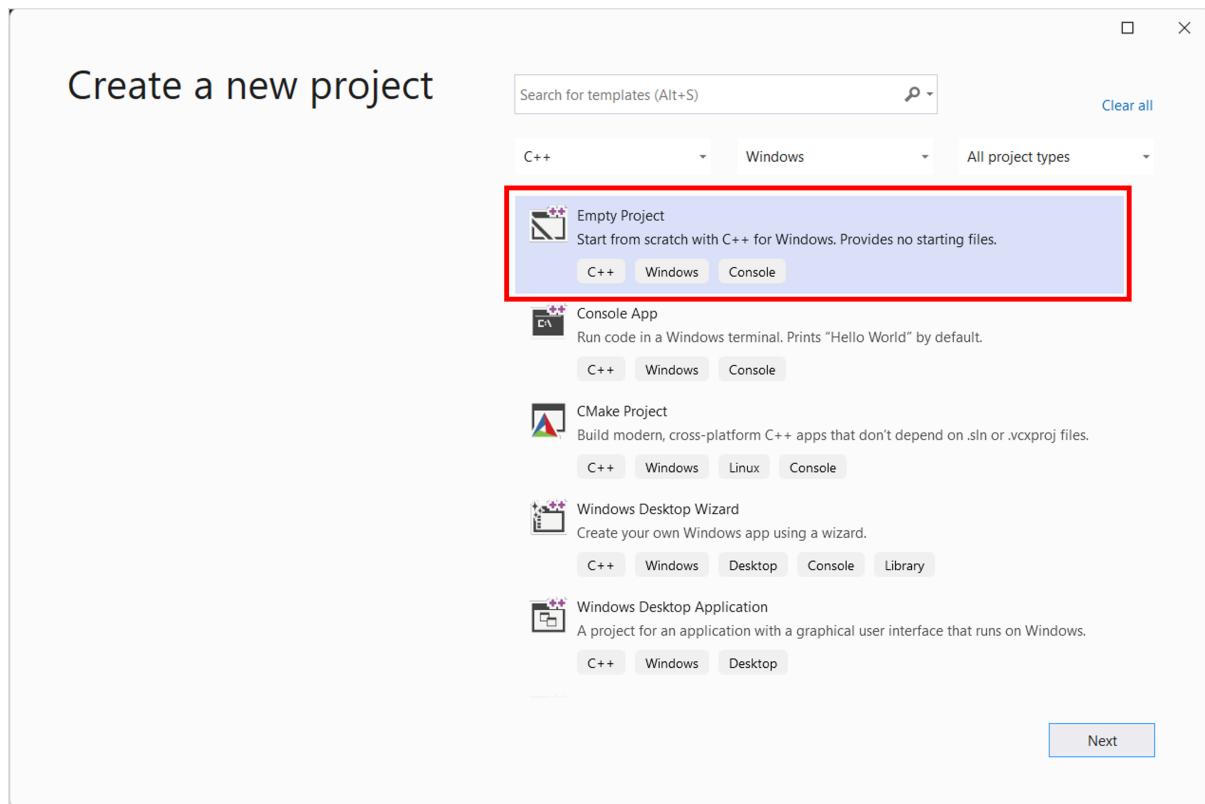
Choose a project template with code scaffolding to get started

[Continue without code →](#)

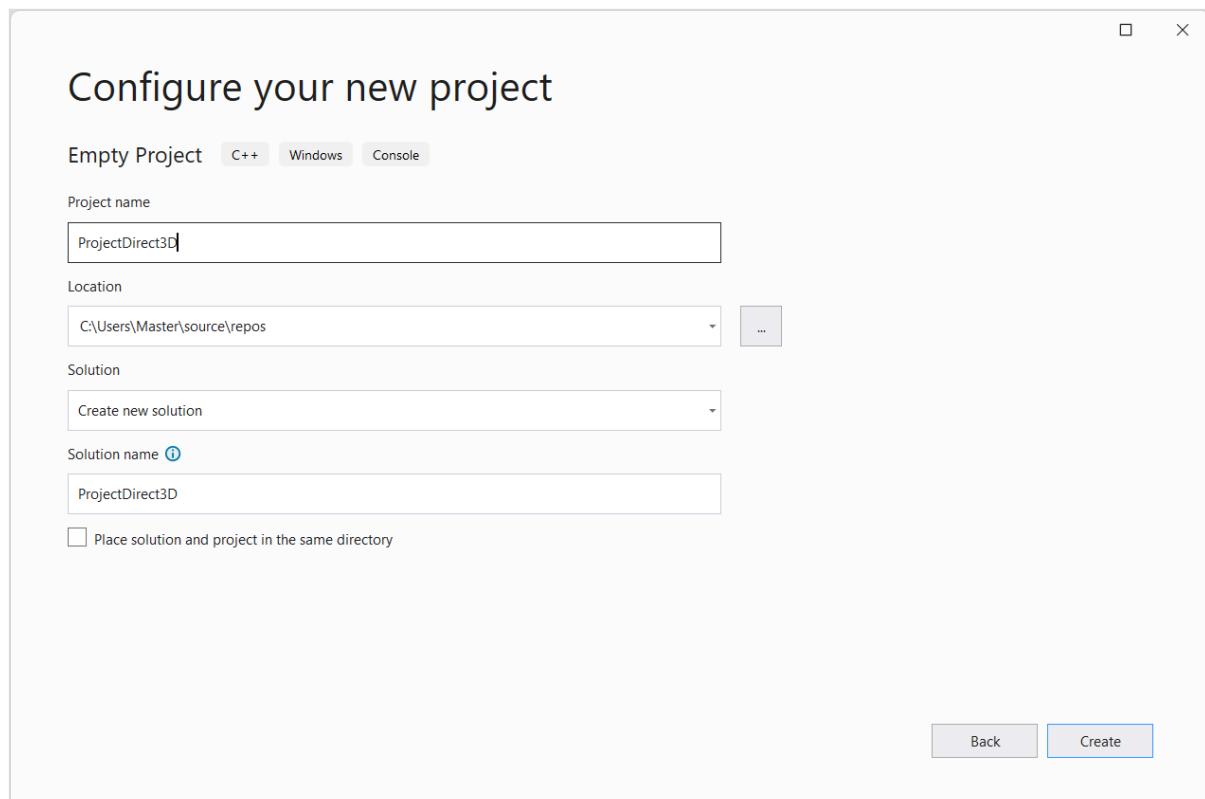
شکل ۷: انتخاب Create a new project



شکل ۸: انتخاب New Project



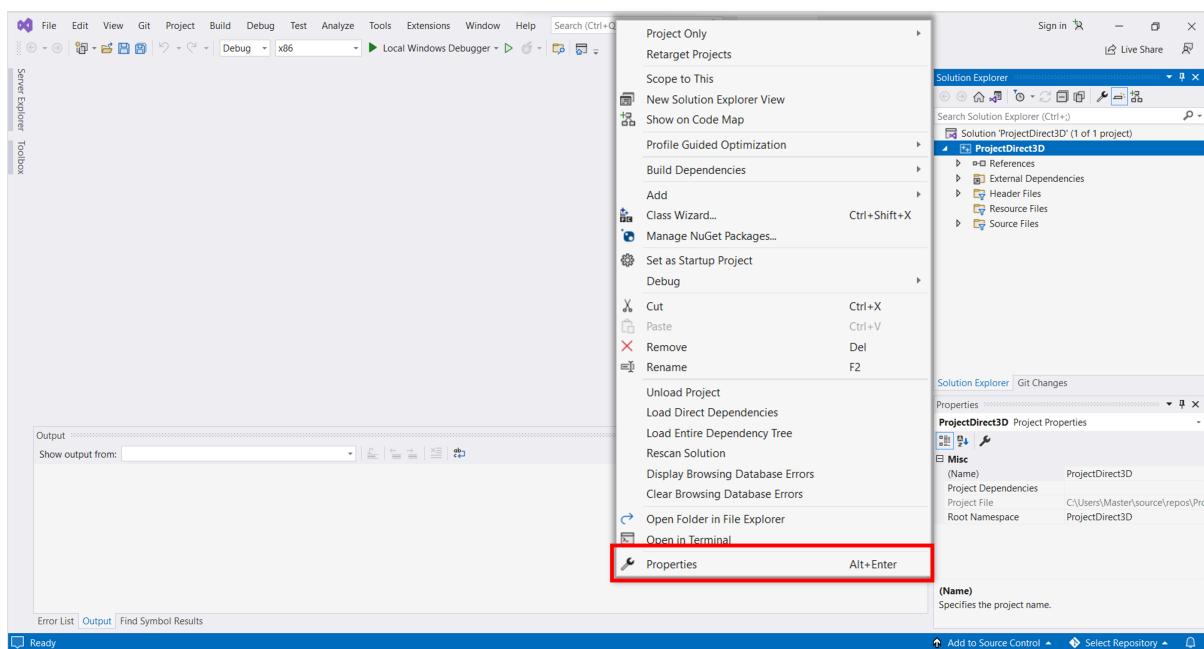
شکل ۹: انتخاب Empty C++ Project



شکل ۱۰: پنجره Configure new project

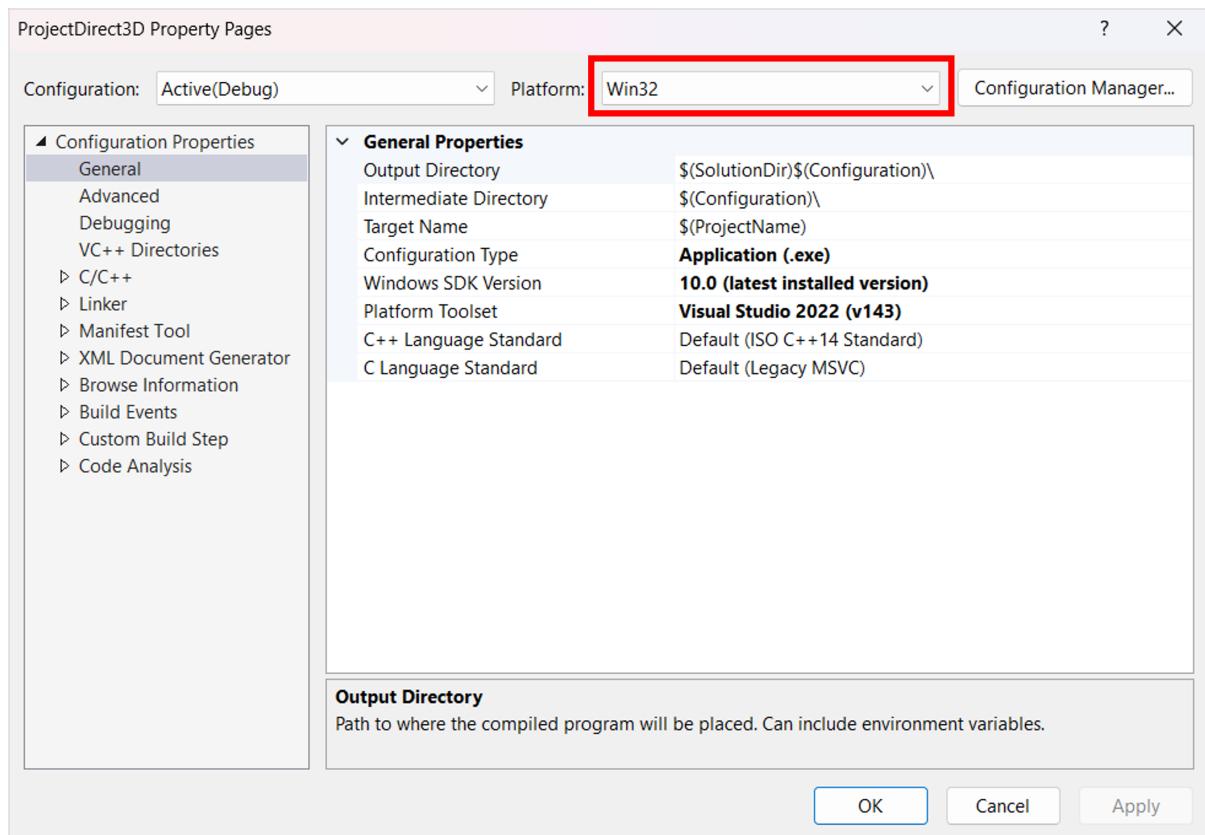
اکنون پروژه در پنجره‌ی جدیدی باز می‌شود. از قسمت Solution Explorer در سمت راست

، روی نام پروژه کلیک راست کرده و Properties را بزنید (شکل ۱۱). در پنجره‌ی باز شده باید تغییرات زیر را اعمال کنید.

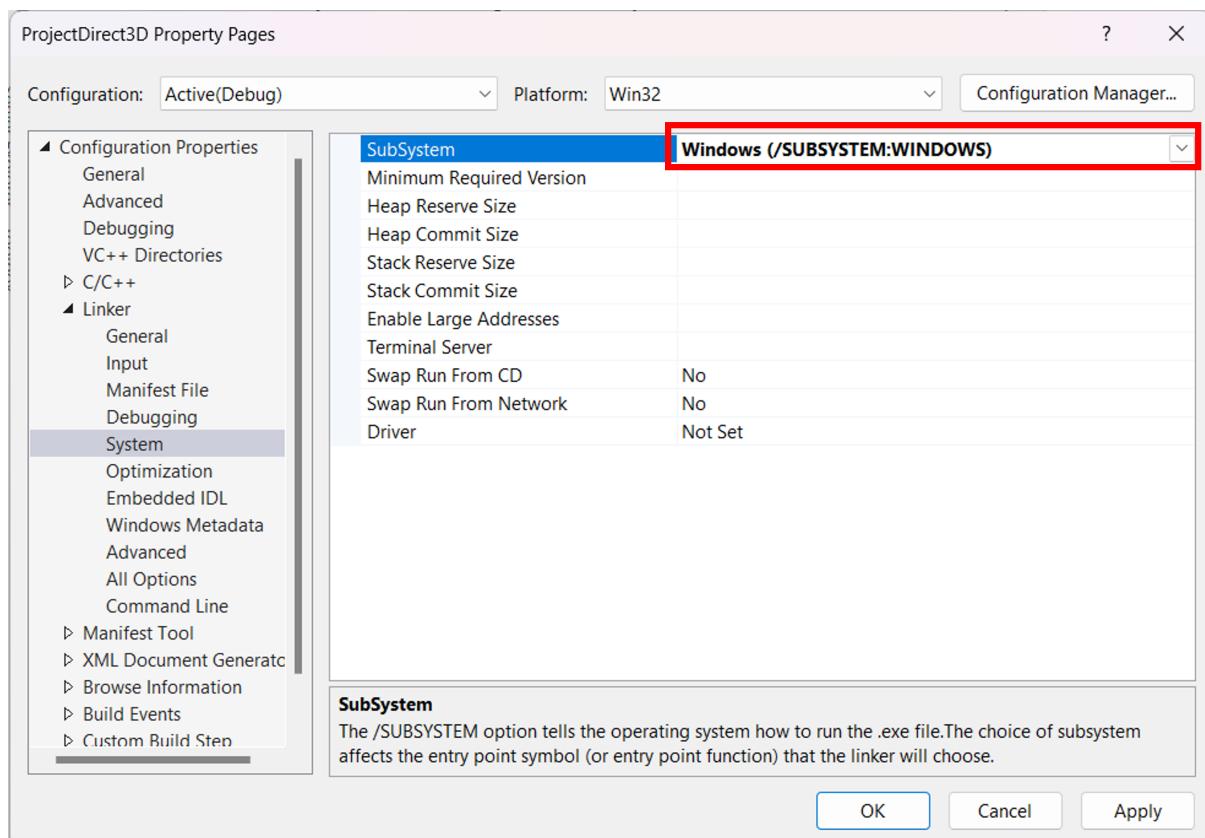


شکل ۱۱ Project Properties :

ابتدا از قسمت بالای پنجره Win32 را انتخاب کنید (شکل ۱۲) و بررسی که مقادیر پنجره‌ی شما با عکس یکسان است. سپس در قسمت سمت چپ پنجره زیر درخت Linker->System را انتخاب کنید و در سمت راست SubSystem:WINDOWS را روی (/SUBSYSTEM:WINDOWS) قرار دهید (شکل ۱۳).



شکل ۱۲ : پنجره Property Page

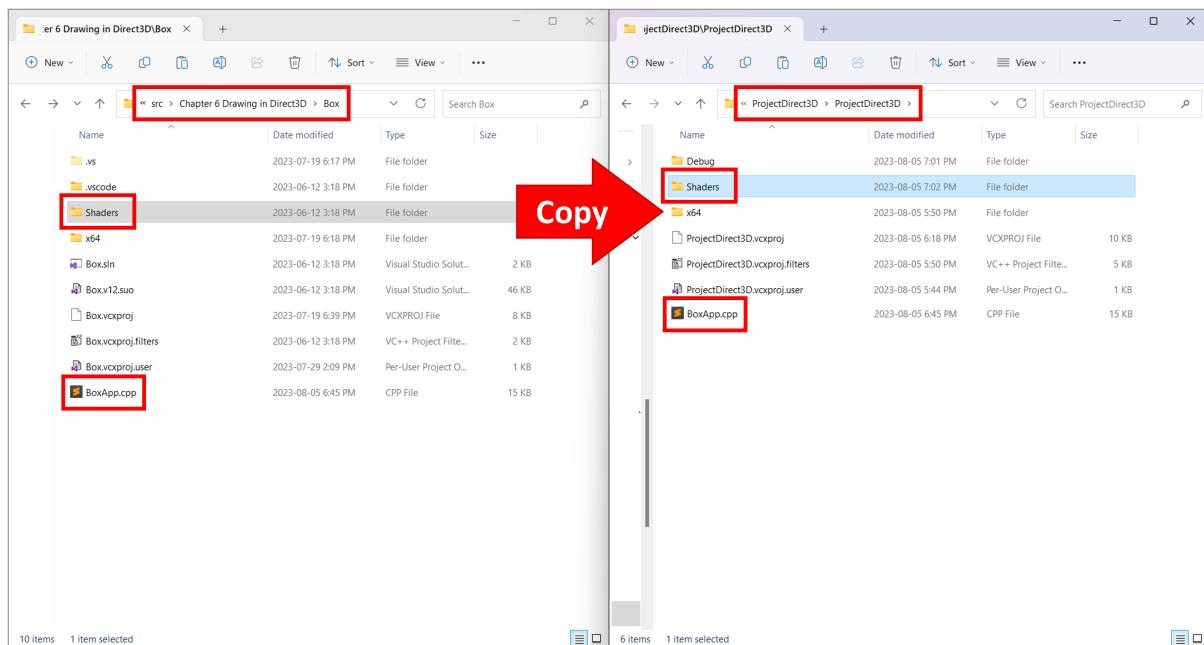


شکل ۱۳ : تنظیمات Linker System

اضافه کردن کد منبع و build کردن پروژه

بلاخره تنظيمات پروژه‌ي ما به اتمام رسيد. اکنون میتوانیم فایل‌های کد منبع خود را به پروژه اضافه کرده و آن build کنیم. ابتدا کد منبع آزمایشی BoxApp.cpp یعنی BoxApp.cpp و پوشش Shaders را در فهرست پروژه خود کپی کنید (شکل ۱۴). (فایل‌ها در این مسیر هستند: d3d12book\Chapter (۱۴)).

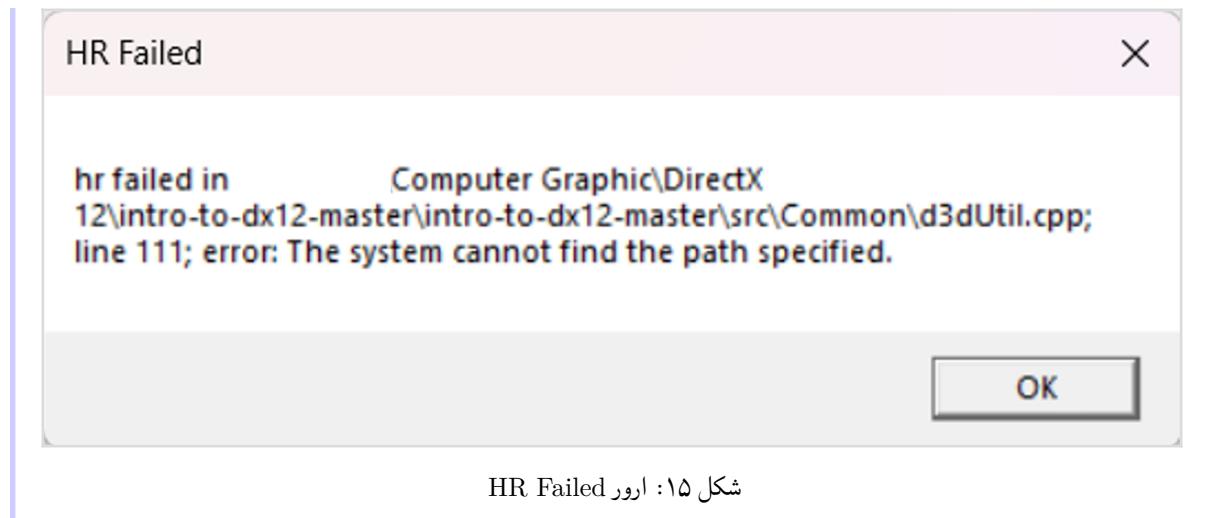
(6 Drawing in Direct3D\Box



شکل ۱۴ : کپی کردن فایل ها

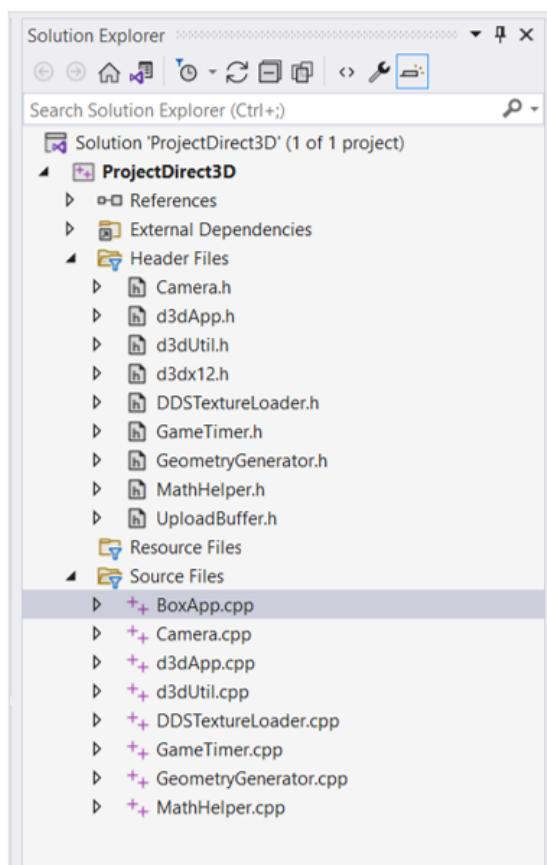
نکته

در صورتی که هنگام اجرا با ارور زیر مواجه شدید ، موارد بالا را به درستی انجام نداده اید و باید بار دیگر آن ها را بررسی کنید.



بعد از کپی کردن فایل ها، گام های زیر را برای اضافه کردن کد به پروژه دنبال کنید.

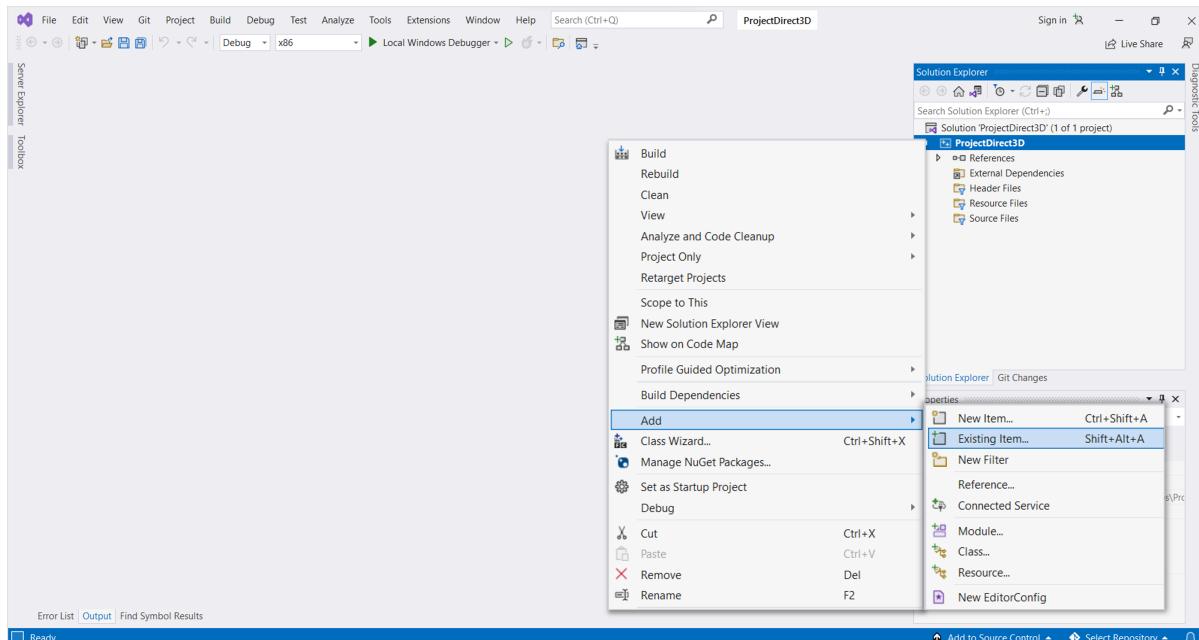
۱. در قسمت Add->Exiting Solution Explorer بر روی نام پروژه کلیک راست کنید و از پنجره ای باز شده BoxApp.cpp را انتخاب کنید.



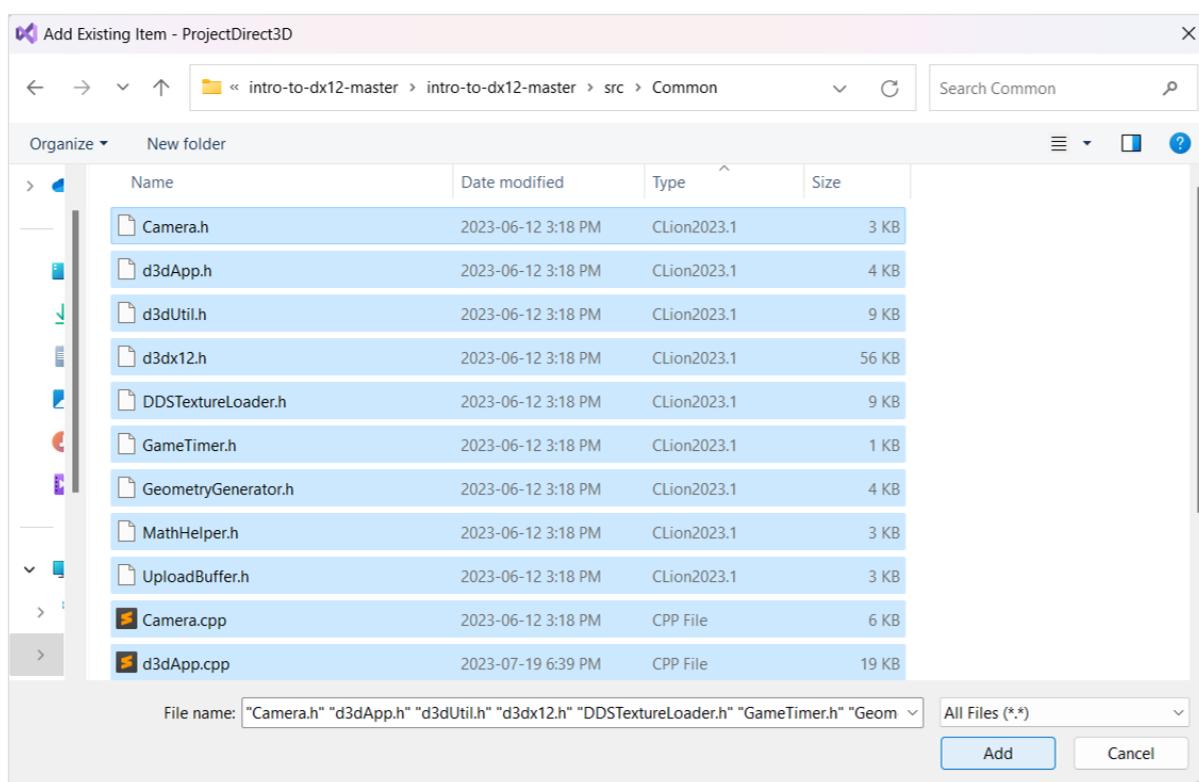
شکل ۱۶: قسمت Solution Explorer

۲. در قسمت Add->Exiting Solution Explorer بر روی نام پروژه کلیک راست کنید و

را انتخاب کنید. از پنجره‌ی باز شده به مسیر دایرکتوری Common بروید و تمامی فایل‌ها با پسوند .h. و .cpp. را انتخاب کنید (شکل ۱۷ - ۱۸). در انتهای باید مانند (شکل ۱۶) شود.



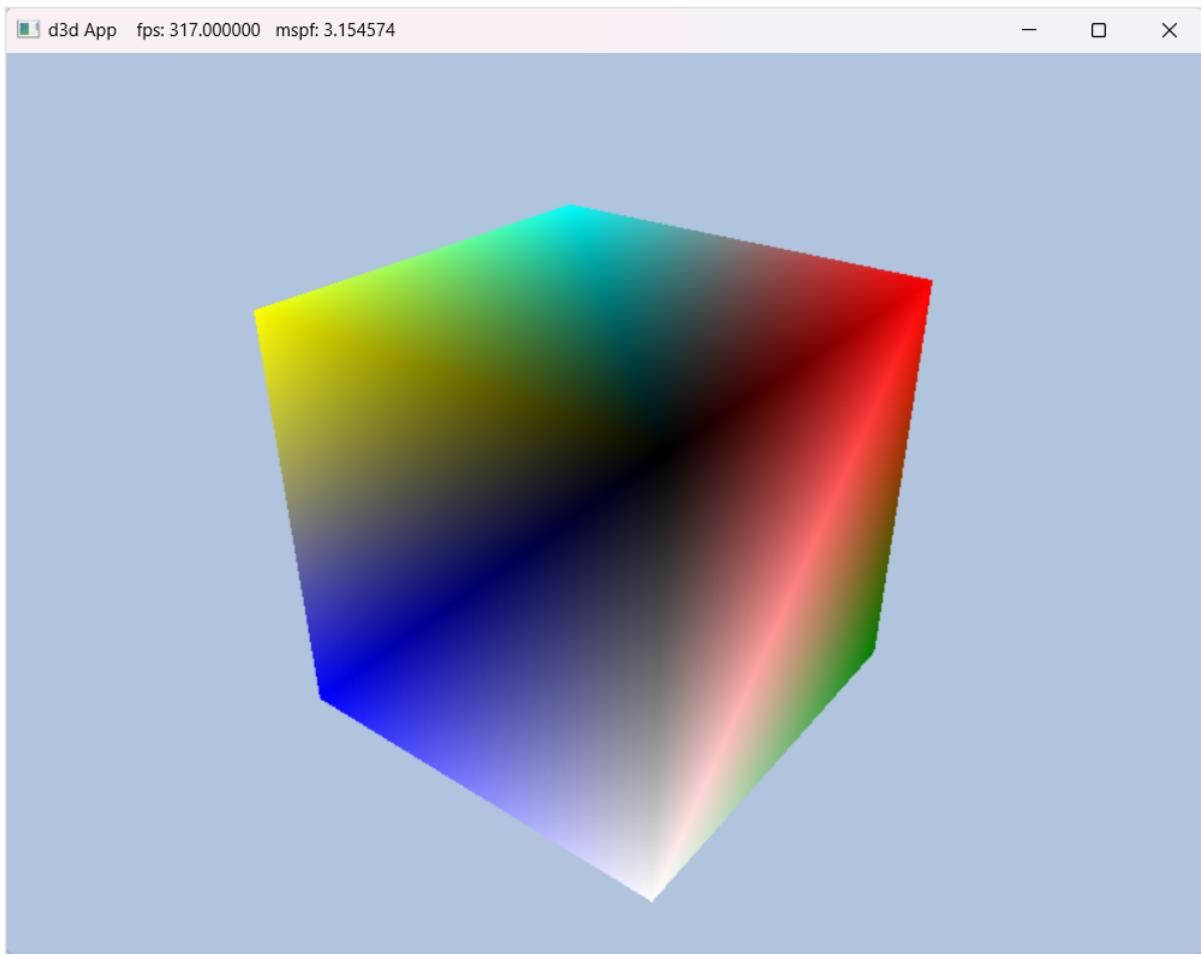
شکل ۱۷: اضافه کردن فایل به پروژه



شکل ۱۸: پنجره‌ی Add Existing Item

۳. اکنون فایل‌های کد منبع جزئی از پروژه بوده و میتوانید به منوی اصلی رفته و Debug->Start را انتخاب کنید.

Debugging را انتخاب کنید تا کد آزمایشی کامپایل ، لینک و اجرا شود (شکل ۱۹).



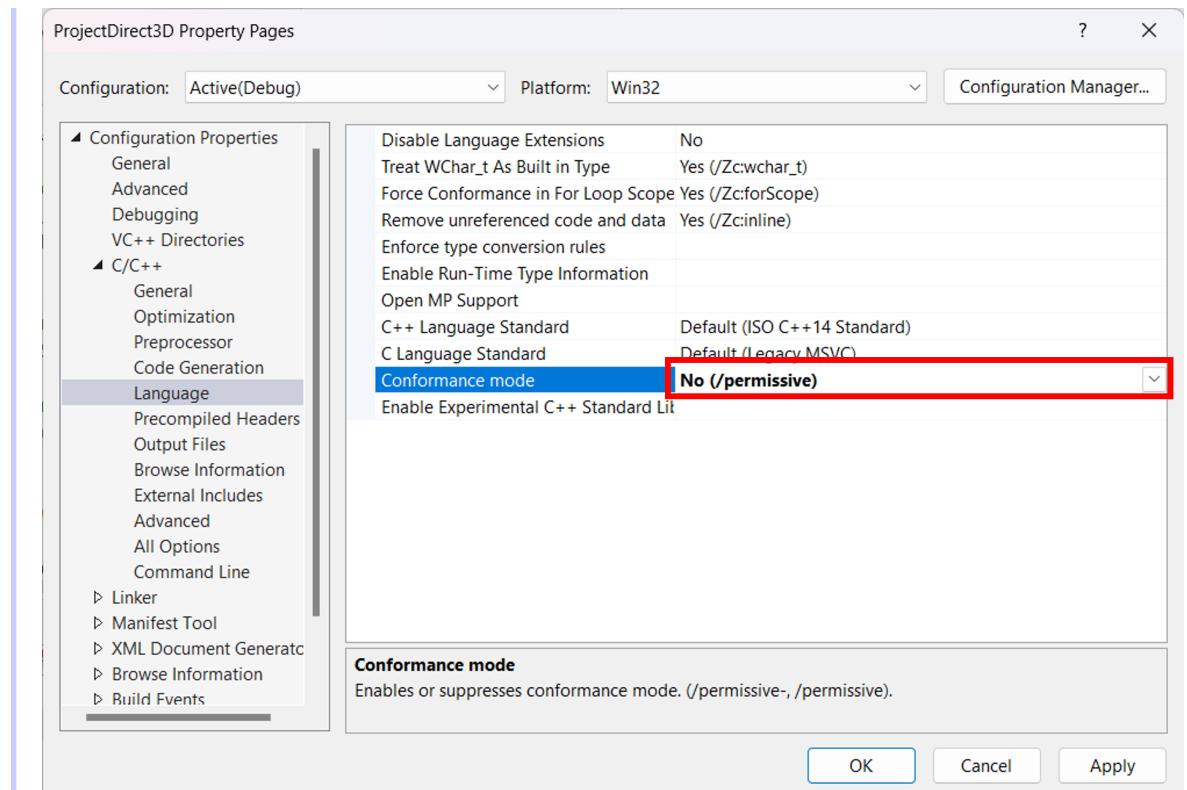
شکل ۱۹: خروجی برنامه

نکته

در صورتی که هنگام اجرا با ارور ‘&’ مواجه شدید (شکل ۲۰)، از پنجره Conformance C/C++->Language ، Property را انتخاب کنید و از قسمت (/permissive mode) بگذارید (شکل ۲۰).

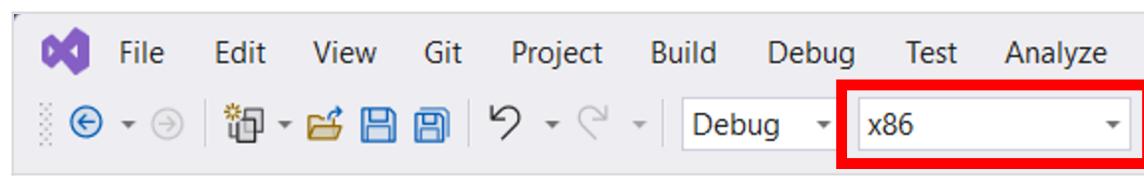
Error List		Build + IntelliSense	
Code	Description	Project	File
C2102	'&' requires l-value	ProjectDirect3D	BoxApp.cpp
C2102	'&' requires l-value	ProjectDirect3D	BoxApp.cpp
C2102	'&' requires l-value	ProjectDirect3D	BoxApp.cpp
C2102	'&' requires l-value	ProjectDirect3D	BoxApp.cpp
C2102	'&' requires l-value	ProjectDirect3D	d3dApp.cpp
C2102	'&' requires l-value	ProjectDirect3D	d3dApp.cpp

شکل ۲۰: ارور ‘&’ requires l-value



شکل ۲۱: تنظیمات C/C++->Language

در صورتی که ارور رفع نشد ، از قسمت بالا (شکل ۲۱) ، x86 را انتخاب کنید.



شکل ۲۲: نحوه اجرا به صورت x86