

INTRODUCTION TO

# 3D GAME PROGRAMMING

## WITH DIRECTX® 11



INCLUDES DVD



FRANK D. LUNA

*Introduction to*  
**3D GAME PROGRAMMING**  
**WITH DIRECTX® 11**

**Frank D. Luna**



**MERCURY LEARNING AND INFORMATION**

Dulles, Virginia

Boston, Massachusetts

Copyright ©2012 by MERCURY LEARNING AND INFORMATION LLC. All rights reserved.

*This publication, portions of it, or any accompanying software may not be reproduced in any way, stored in a retrieval system of any type, or transmitted by any means, media, electronic display or mechanical display, including, but not limited to, photocopy, recording, Internet postings, or scanning, without prior permission in writing from the publisher.*

Publisher: David Pallai  
MERCURY LEARNING AND INFORMATION  
22841 Quicksilver Drive  
Dulles, VA 20166  
info@merclearning.com  
www.merclearning.com  
1-800-758-3756

This book is printed on acid-free paper.

Frank D. Luna. *Introduction to 3D GAME PROGRAMMING WITH DIRECTX 11*

ISBN: 978-1-9364202-2-3

The publisher recognizes and respects all marks used by companies, manufacturers, and developers as a means to distinguish their products. All brand names and product names mentioned in this book are trademarks or service marks of their respective companies. Any omission or misuse (of any kind) of service marks or trademarks, etc. is not an attempt to infringe on the property of others.

Library of Congress Control Number: 2012931119

12131432

Our titles are available for adoption, license, or bulk purchase by institutions, corporations, etc. For additional information, please contact the Customer Service Dept. at 1-800-758-3756 (toll free).

The sole obligation of MERCURY LEARNING AND INFORMATION to the purchaser is to replace the disc, based on defective materials or faulty workmanship, but not based on the operation or functionality of the product.

*Introduction to*  
**3D GAME PROGRAMMING**  
**WITH DIRECTX® 11**

## LICENSE, DISCLAIMER OF LIABILITY, AND LIMITED WARRANTY

By purchasing or using this book (the “Work”), you agree that this license grants permission to use the contents contained herein, but does not give you the right of ownership to any of the textual content in the book or ownership to any of the information or products contained in it. *This license does not permit uploading of the Work onto the Internet or on a network (of any kind) without the written consent of the Publisher.* Duplication or dissemination of any text, code, simulations, images, etc. contained herein is limited to and subject to licensing terms for the respective products, and permission must be obtained from the Publisher or the owner of the content, etc., in order to reproduce or network any portion of the textual material (in any media) that is contained in the Work.

MERCURY LEARNING AND INFORMATION LLC (“MLI” or “the Publisher”) and anyone involved in the creation, writing, or production of the companion disc, accompanying algorithms, code, or computer programs (“the software”), and any accompanying Web site or software of the Work, cannot and do not warrant the performance or results that might be obtained by using the contents of the Work. The author, developers, and the Publisher have used their best efforts to insure the accuracy and functionality of the textual material and/or programs contained in this package; we, however, make no warranty of any kind, express or implied, regarding the performance of these contents or programs. The Work is sold “as is” without warranty (except for defective materials used in manufacturing the book or due to faulty workmanship).

The author, developers, and the publisher of any accompanying content, and anyone involved in the composition, production, and manufacturing of this work will not be liable for damages of any kind arising out of the use of (or the inability to use) the algorithms, source code, computer programs, or textual material contained in this publication. This includes, but is not limited to, loss of revenue or profit, or other incidental, physical, or consequential damages arising out of the use of this Work.

The sole remedy in the event of a claim of any kind is expressly limited to replacement of the book, and only at the discretion of the Publisher. The use of “implied warranty” and certain “exclusions” vary from state to state, and might not apply to the purchaser of this product.

To my nieces and nephews,  
Marrick, Hans, Max, Anna, Augustus, and Presley

# *ACKNOWLEDGMENTS*

I would like to thank Rod Lopez, Jim Leiterman, Hanley Leung, Rick Falck, Tybon Wu, Tuomas Sandroos, and Eric Sandegren for reviewing earlier editions of the book, and I would like to thank Jay Tennant and William Goschnick for reviewing select chapters of the current edition. I want to thank Tyler Drinkard for building some of the 3D models and textures used in the demo programs of this book. I also want to thank Dale E. La Force, Adam Hoult, Gary Simmons, James Lambers, and William Chin for their assistance. Lastly, I want to thank the staff at Mercury Learning and Information, in particular, David Pallai.

# INTRODUCTION

Direct3D 11 is a rendering library for writing high performance 3D graphics applications using modern graphics hardware on the Windows platform. (A modified version of DirectX 9 is used on the XBOX 360.) Direct3D is a low-level library in the sense that its application programming interface (API) closely models the underlying graphics hardware it controls. The predominant consumer of Direct3D is the games industry, where higher level rendering engines are built on top of Direct3D. However, other industries need high performance interactive 3D graphics as well, such as medical and scientific visualization and architectural walkthrough. In addition, with every new PC being equipped with a modern graphics card, non-3D applications are beginning to take advantage of the GPU (graphics processing unit) to offload work to the graphics card for intensive calculations; this is known as *general purpose GPU computing*, and Direct3D 11 provides the compute shader API for writing general purpose GPU programs. Although Direct3D is usually programmed from native C++, stable .NET wrappers exist for Direct3D (e.g., <http://slimdx.org/>) so that you can access this powerful 3D graphics API from managed applications. Finally, at their 2011 BUILD conference (<http://www.buildwindows.com/>), Microsoft recently showed that Direct3D 11 will play the key role in writing high performance 3D “Metro” applications in Windows 8. All-in-all, the future looks bright for Direct3D developers.

This book presents an introduction to programming interactive computer graphics, with an emphasis on game development, using Direct3D 11. It teaches the fundamentals of Direct3D and shader programming, after which the reader will be prepared to go on and learn more advanced techniques. The book is divided into



three main parts. Part I explains the mathematical tools that will be used throughout this book. Part II shows how to implement fundamental tasks in Direct3D, such as initialization, defining 3D geometry, setting up cameras, creating vertex, pixel, geometry, and compute shaders, lighting, texturing, blending, stencil, and tessellation. Part III is largely about applying Direct3D to implement a variety of interesting techniques and special effects, such as working with meshes, terrain rendering, picking, particle systems, environment mapping, normal mapping, displacement mapping, real-time shadows, and ambient occlusion.

For the beginner, this book is best read front to back. The chapters have been organized so that the difficulty increases progressively with each chapter. In this way, there are no sudden jumps in complexity leaving the reader lost. In general, for a particular chapter, we will use the techniques and concepts previously developed. Therefore, it is important that you have mastered the material of a chapter before continuing. Experienced readers can pick the chapters of interest.

Finally, you may be wondering what kinds of games you can develop after reading this book. The answer to that question is best obtained by skimming through this book and seeing the types of applications that are developed. From that you should be able to visualize the types of games that can be developed based on the techniques taught in this book and some of your own ingenuity.

## INTENDED AUDIENCE

---

This book was designed with the following three audiences in mind:

1. Intermediate level C++ programmers who would like an introduction to 3D programming using the latest iteration of Direct3D.
2. 3D programmers experienced with an API other than DirectX (e.g., OpenGL) who would like an introduction to Direct3D 11.
3. Experienced Direct3D 9 and Direct3D 11 programmers wishing to learn the latest iteration of Direct3D.

## PREREQUISITES

---

It should be emphasized that this is an introduction to Direct3D 11, shader programming, and 3D game programming; it is *not* an introduction to general computer programming. The reader should satisfy the following prerequisites:

1. High School mathematics: algebra, trigonometry, and (mathematical) functions, for example.

2. Competent with Visual Studio: should know how to create projects, add files, and specify external libraries to link, for example.
3. Intermediate C++ and data structure skills: comfortable with pointers, arrays, operator overloading, linked lists, inheritance, and polymorphism, for example.
4. Familiarity with Windows programming with the Win32 API is helpful, but not required; we provide a Win32 primer in Appendix A.

## REQUIRED DEVELOPMENT TOOLS AND HARDWARE

---

To program Direct3D 11 applications, you will need the DirectX 11 SDK; the latest version can be downloaded from <http://msdn.microsoft.com/en-us/directx/default.aspx>. Once downloaded, follow the instructions given by the installation wizard. At the time of this writing, the latest SDK version is the June 2010 DirectX SDK. All of our sample programs were written using Visual Studio 2010.

Direct3D 11 requires Direct3D 11 capable hardware. The demos in this book were tested on a Geforce GTX 460.

## USE OF THE D3DX LIBRARY

---

Since version 7.0, DirectX has shipped with the D3DX (Direct3D Extension) library. This library provides a set of functions, classes, and interfaces that simplify common 3D graphics related operations, such as math operations, texture and image operations, mesh operations, and shader operations (e.g., compiling and assembling). That is to say, D3DX contains many features that would be a chore to implement on your own.

We use the D3DX library throughout this book because it allows us to focus on more interesting material. For instance, we would rather not spend pages explaining how to load various image formats (e.g., .bmp, .jpeg) into a Direct3D texture interface when we can do it in a single call to the D3DX function `D3DX11CreateTextureFromFile`. In other words, D3DX makes us more productive and lets us focus more on actual content rather than spending time reinventing the wheel.

Other reasons to use D3DX:

1. D3DX is general and can be used with a wide range of different types of 3D applications.
2. D3DX is fast, at least as fast as general functionality can be.
3. Other developers use D3DX. Therefore, you will most likely encounter code that uses D3DX. Consequently, whether you choose to use D3DX or not, you should become familiar with it so that you can read code that uses it.

4. D3DX already exists and has been thoroughly tested. Furthermore, it becomes more improved and feature rich with each iteration of DirectX.

## USING THE DIRECTX SDK DOCUMENTATION AND SDK SAMPLES

Direct3D is a huge API and we cannot hope to cover all of its details in this one book. Therefore, to obtain extended information it is imperative that you learn how to use the DirectX SDK documentation. You can launch the C++ DirectX online documentation by executing the *windows\_graphics.chm* file in the *DirectX SDK\Documentation\DirectX9* directory, where *DirectX SDK* is the directory you installed DirectX to. In particular, you will want to navigate to the Direct3D 11 section (see Figure 1).

The DirectX documentation covers just about every part of the DirectX API; therefore it is very useful as a reference, but because the documentation doesn't go into much depth, or assumes some previous knowledge, it isn't the best learning tool. However, it does get better and better with every new DirectX version released.

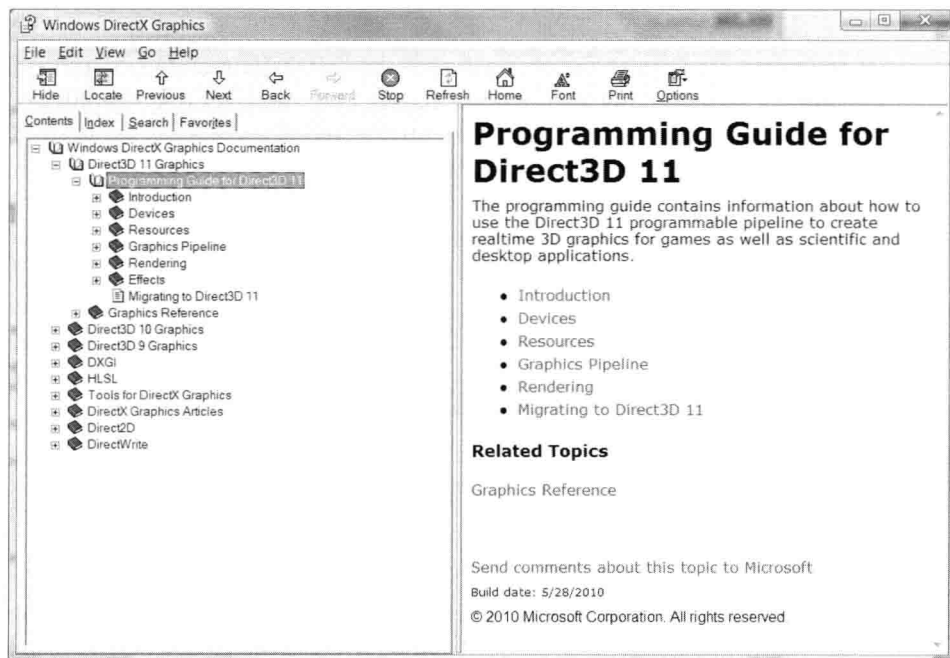
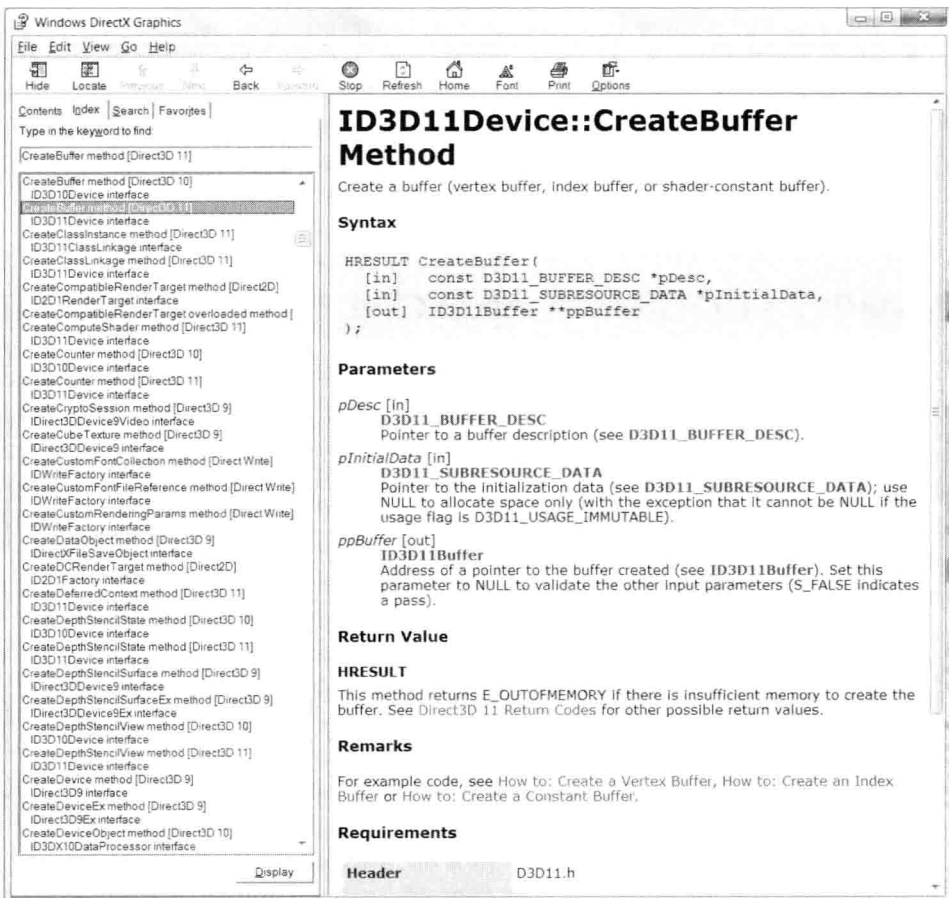


Figure 1. Direct3D Programming Guide in the DirectX documentation.

As said, the documentation is primarily useful as a reference. Suppose you come across a DirectX related type or function, say the function `ID3D11Device::CreateBuffer`, for which you would like more information. You simply do a search in the documentation index and you get a description of the object type, or in this case function; see Figure 2.

**Note:** *In this book we may direct you to the documentation for further details from time to time.*

We would also like to point out the available Direct3D sample programs that ship with DirectX SDK. The C++ Direct3D samples are located in the *DirectX SDK\Samples\C++\Direct3D10* and *DirectX SDK\Samples\C++\Direct3D11* directories. Each sample illustrates how to implement a particular effect in Direct3D. These



**Figure 2.** Index of the DirectX documentation.

samples are fairly advanced for a beginning graphics programmer, but by the end of this book you should be ready to study them. Examination of the samples is a good “next step” after finishing this book. Note that we mentioned both the Direct3D 10 and Direct3D 11 samples. Direct3D 11 extends Direct3D 10 with new features, so Direct3D 10 techniques still apply when making a Direct3D 11 application; hence, it is still worthwhile to study the Direct3D 10 samples to see how a particular effect can be implemented.

## CLARITY

---

We want to emphasize that the program samples for this book were written with clarity in mind and not performance. Thus, many of the samples may be implemented inefficiently. Keep this in mind if you are using any of the sample code in your own projects, as you may wish to rework it for better efficiency. Moreover, in order to focus on the Direct3D API, we have built minimal infrastructure on top of Direct3D. In a large 3D application, you will likely implement a rendering engine on top of Direct3D; however, the topic of this book is the Direct3D API, not rendering engine design.

## SAMPLE PROGRAMS AND ONLINE SUPPLEMENTS

---

The companion DVD and Web sites for this book ([www.d3dcoder.net](http://www.d3dcoder.net) and [www.merclearning.com](http://www.merclearning.com)) play an integral part in getting the most out of this book. On the DVD and Web site you will find the complete source code and project files for every sample in this book. In many cases, DirectX programs are too large to fully embed in a textbook; therefore, we only embed relevant code fragments based on the ideas being shown. It is highly recommended that the reader study the corresponding demo code to see the program in its entirety. (We have aimed to make the demos small and focused for easy study.) As a general rule, the reader should be able to implement a chapter’s demo(s) on his or her own after reading the chapter and spending some time studying the demo code. In fact, a good exercise is trying to implement the samples on your own using the book and sample code as a reference.

In addition to sample programs, the website also contains a message board. We encourage readers to communicate with each other and post questions on topics they do not understand or on topics for which they need clarification. In many cases, getting alternative perspectives and explanations to a concept speeds up the time it takes to comprehend it. And lastly, additional program samples and

tutorials are planned to be added to the web site on topics that we could not fit into this book.

## DEMO PROJECT SETUP IN VISUAL STUDIO 2010

The demos for this book can be opened simply by double-clicking the corresponding project file (.vcxproj) or solution file (.sln). This section describes how to create and build a project from scratch using the book's demo application framework using Visual Studio 2010 (VS10). As a working example, we will show how to recreate and build the "Box" demo of Chapter 6.

It is assumed that the reader has already successfully downloaded and installed the latest version of the DirectX SDK (available at <http://msdn.microsoft.com/directx/>), which is needed to program DirectX applications. The installation of the SDK is straightforward, and the installation wizard will walk you through it.

### Create a Win32 Project

First, launch VS10, then go to the main menu and select **File > New > Project**, as shown in Figure 3.

The New Project dialog box will appear (Figure 4). Select **Visual C++ > Win32** from the Visual C++ Project Types tree control on the left. On the right, select **Win32 Project**. Next, give the project a name and specify the location you wish to store the project folder. Also uncheck **Create directory for solution**, if it is initially checked by default. Now hit **OK**.

A new dialog box will appear. On the left, there are two options: Overview and Application Settings. Select **Application Settings**, which produces the dialog box shown in Figure 5. From here, be sure that **Windows application** is chosen, and the **Empty project** box is checked. Now press the **Finish** button. At this point, you have successfully created an empty Win32 project, but there are still some things to do before you can build a DirectX project demo.

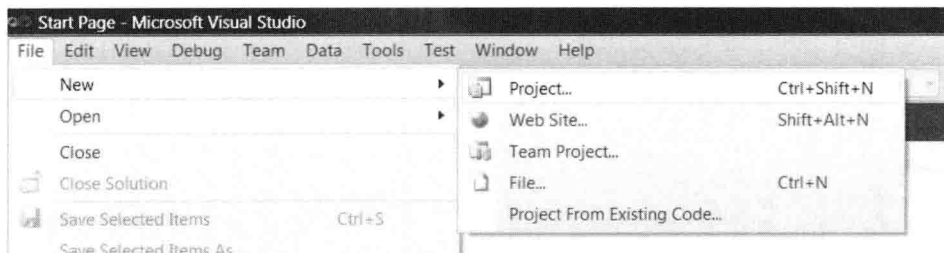


Figure 3. Creating a new project.

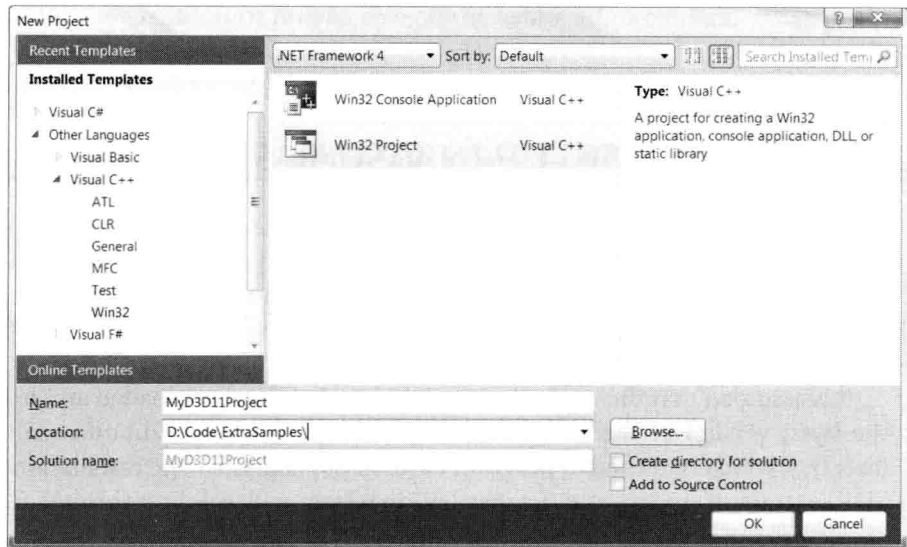


Figure 4. New Project settings.

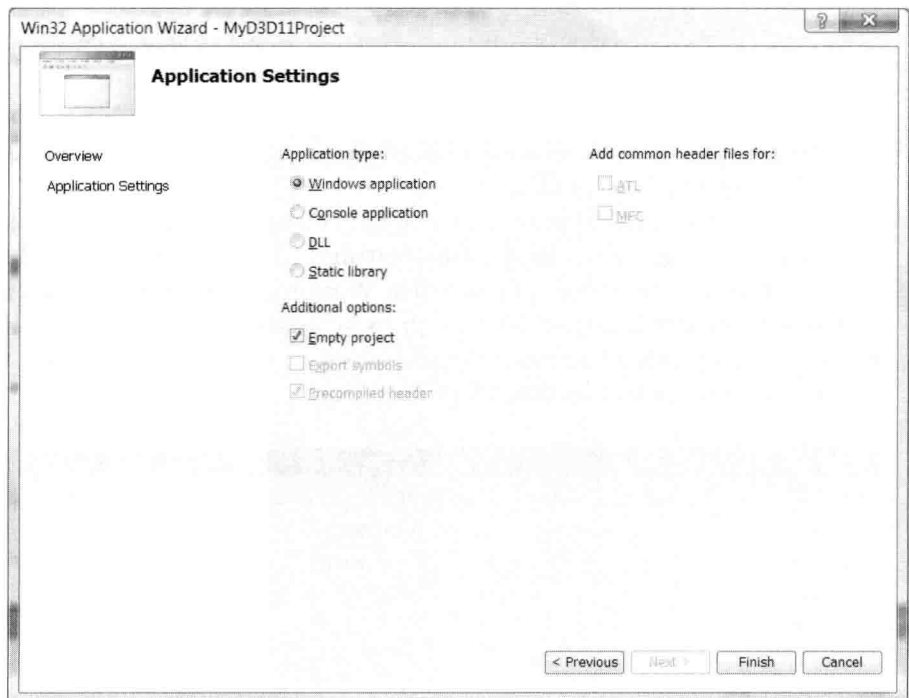


Figure 5. Application settings.

## Linking the DirectX Libraries

We now need to link the DirectX libraries to our project. For a debug build, add the additional libraries:

```
d3d11.lib;  
d3dx11d.lib;  
D3DCompiler.lib;  
Effects11d.lib;  
dxerr.lib;  
dxgi.lib;  
dxguid.lib;
```

For a release build, add the same libraries above, but remove the 'd' character at the end of `d3dx11d.lib` and `Effects11d.lib`, so that they are just `d3dx11.lib` and `Effects11.lib`.

To link the library files, right-click on the project name under the Solution Explorer and select **Properties** from the dropdown menu (Figure 6). This launches the dialog

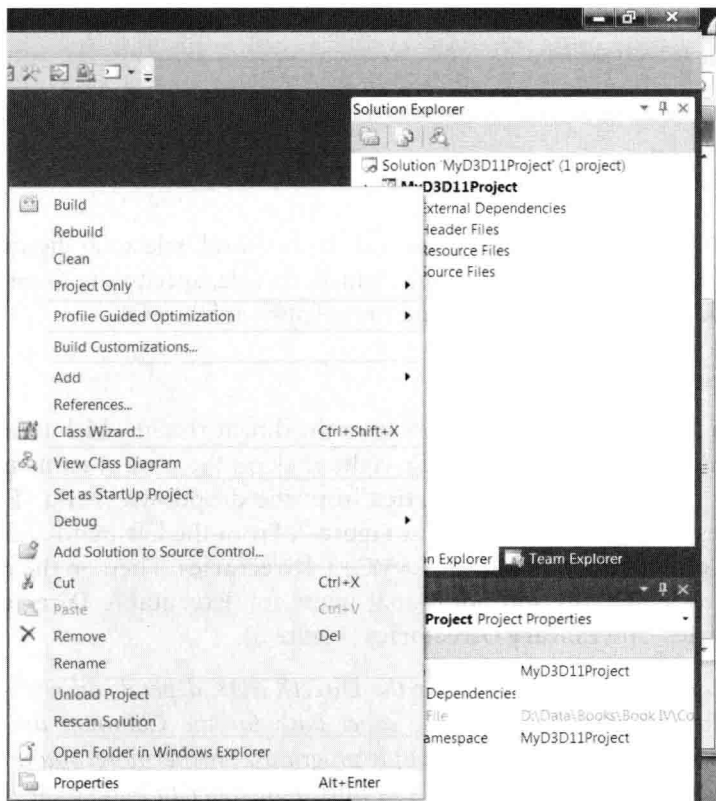


Figure 6. Right-click on the project and select Properties from the context menu.



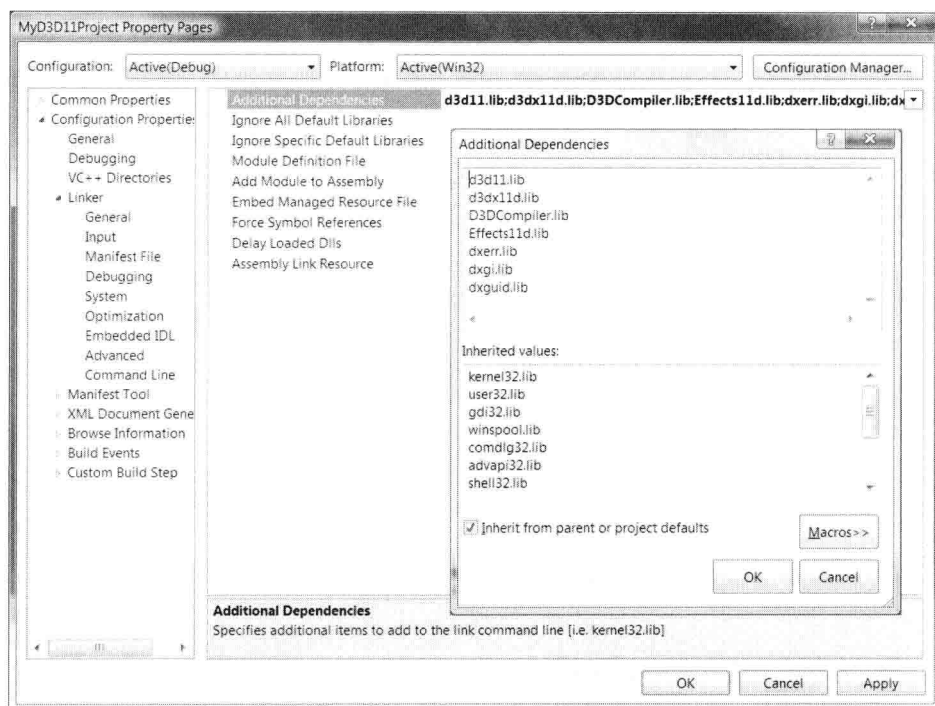


Figure 7. Link the DirectX libraries.

box shown in Figure 7. From the left-hand-side tree control, select **Configuration Properties > Linker > Input**. Then on the right-hand-side, specify the library file names on the **Additional Dependencies** line. Press **Apply** and then **OK**.

## Setting up the Search Paths

We now need to make sure Visual Studio knows the directories in which to search for DirectX header and library files. Again, right-click on the project name under the Solution Explorer and select **Properties** from the dropdown menu (Figure 6). This launches the dialog box shown in Figure 7. From the left-hand-side tree control, select **Configuration Properties > VC++ Directories**. Then on the right-hand-side, you will need to add additional items for **Executable Directories**, **Include Directories**, and **Library Directories** (Figure 8).

**Note:**

*Keep in mind that the exact path for the DirectX SDK depends on where you installed the DirectX SDK, and the exact path for the Common directory depends on where you extract the sample programs. Furthermore, you are free to move the Common directory, but you must remember to update the search paths in Visual Studio accordingly.*