



Beginning Game Development with Amazon Lumberyard

Create 3D Games Using Amazon Lumberyard and Lua

Jaken Chandler Herman

Apress®

Beginning Game Development with Amazon Lumberyard

**Create 3D Games Using
Amazon Lumberyard and Lua**

Jaken Chandler Herman

Apress®

Beginning Game Development with Amazon Lumberyard: Create 3D Games Using Amazon Lumberyard and Lua

Jaken Chandler Herman
Conroe, TX, USA

ISBN-13 (pbk): 978-1-4842-5072-3
<https://doi.org/10.1007/978-1-4842-5073-0>

ISBN-13 (electronic): 978-1-4842-5073-0

Copyright © 2019 by Jaken Chandler Herman

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Spandana Chatterjee

Development Editor: Matthew Moodie

Coordinating Editor: Shrikant Vishwakarma

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-5072-3. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my amazing wife, family,
and friends for supporting me
along this journey.*

Table of Contents

About the Author	ix
About the Technical Reviewer	xi
Chapter 1: Install and Setup of Amazon Lumberyard	1
Required Components Outside of the Lumberyard Engine	1
Installing Visual Studio	2
Installing the Amazon Lumberyard Engine	4
Express Install	5
Custom Install.....	6
Running the Lumberyard Editor	12
Summary.....	12
Chapter 2: Navigating the Engine	13
Open the Engine.....	13
Pane Navigation	17
Introduction to the Viewport.....	18
Snapping Panes.....	21
Introduction to the Asset Browser	23
Time-Saving Key Bindings	35
Chapter 3: Creating Your First Project.....	39
Opening the Project Configurator.....	39
Creating Your First Project	41
Creating a Project Through the Project Configurator.....	42
Creating a Project Through the CLI.....	46

TABLE OF CONTENTS

Setting Default Projects	48
Gems	49
Enabling Gems.....	49
Creating Gems	55
Advanced Project Settings	56
Levels.....	59
Creating a New Level.....	59
Deleting a Level.....	62
Changing Levels	63
Chapter 4: Terrain	65
Getting Started.....	66
Heightmaps.....	67
Importing Heightmaps.....	68
Changing Heightmap Properties.....	70
Painting Your Own Heightmap	72
Adding a Pop of Color.....	79
Texture Layers	80
Terrain Texture Layers Editor	81
Painting Texture Layers onto Your Terrain.....	87
Using Megatextures.....	91
Vegetation	93
Time of Day	98
Chapter 5: Entities and Slices.....	103
Verifying the Project Structure.....	103
Entities and the Entity Outliner	104
Child and Parent Entities.....	110

The Pinned Inspector	112
Slices	114
Instantiating Slices	116
Creating Your Player	117
Chapter 6: Components	121
Adding Components to an Entity	121
The AnimGraph and Actor Components	124
The Attachment Component.....	128
The Fog Volume Component	133
The Wind Volume Component	137
The Particle Component.....	140
The Transform Component.....	142
Chapter 7: Creating the Enemy	145
Creating an Enemy That Always Exists	145
Creating an Enemy AI Trigger	146
Creating an Enemy AI Navigation Area.....	150
Creating an Enemy AI Spawn Point.....	155
Creating Enemy Patrol Waypoints	158
Creating More AI Characters	162
Tidying Up Our AI Entities.....	163
Chapter 8: Beginning Lua Scripting.....	167
Comments.....	168
Variables	169
Conditionals	172
Operators	173

TABLE OF CONTENTS

Loops	176
While Loop.....	177
For Loop.....	177
Repeat Until Loop	179
Arrays.....	179
Functions	182
Classes.....	184
Custom Inputs.....	187
Chapter 9: Polishing Your Game	191
Lens Flare	191
Particle Manager.....	195
Audio and Sound Effects.....	200
Chapter 10: Setting Up a User Interface	205
Creating a Canvas	206
Creating a Script Canvas.....	209
Adding the Script Canvas to Our Game	217
Chapter 11: Exporting the Game.....	219
Creating a PC Build	219
Creating a Console Build.....	223
Xbox.....	223
PlayStation	224
Final Thoughts.....	226
Index.....	229

About the Author



Jaken Chandler Herman is a software engineer living in Texas. Jaken holds a bachelor of science in computer science with a focus on software engineering from Sam Houston State University. Outside of university, Jaken worked as a software engineering contractor at NASA. He has worked with many different and new technologies and has an ever-expanding wealth of information on all topics related to programming. In game development, he has academic experience as well as hands-on hobbyist experience having created many mobile-based games, a side-scrolling platformer, and a racing game.

About the Technical Reviewer



Dominique Regalado is a computer science graduate from Sam Houston State University with interests in full stack architecture, cloud platform technologies, and game development. She currently works as an associate software engineer, so providing insight for those new to software development is one of her passions. She has attended hackathons, career fairs, and even the renowned Grace Hopper convention to spread knowledge. Assisting in the authoring of this book was just another great opportunity to learn about a new engine while passing along the knowledge discovered.

CHAPTER 1

Install and Setup of Amazon Lumberyard

You are about to embark on an amazing journey. A journey in which you will be creating your very first video game in Amazon Lumberyard. This will be an extremely rewarding and likely humbling experience. Through my many failed attempts at creating video games when I first began working on them, I often found myself having so much information thrown at me that it was seemingly impossible to figure out where to start, what pieces of the information were *actually* important to my specific project, and where to look for advice and questions along the way.

Lucky for you, however, this book will walk you through everything, from the beginning all the way to the *game over* screen. Before we can begin, we need to go through the setup process.

Required Components Outside of the Lumberyard Engine

Amazon Lumberyard requires a few components outside of those that come packaged with it. One of the great things about Amazon Lumberyard is that the required components are completely free, so you do not need to worry about any additional costs in your game development endeavors. You can develop a video game on a AAA engine completely and 100% for free.

Note AAA, pronounced “Triple-A,” is an informal classification used for video games typically to denote higher budgets and higher quality of development. Think of AAA as a way of saying “Blockbuster,” as you would with movies. While in this book we likely won’t build a AAA game, Amazon Lumberyard easily supports developing such games.

There are a few *optional* components, however, that do have a price tag attached to them, but for the purposes of this book, we will ensure that everything is free of charge along the way.

Installing Visual Studio

The first component we will install and set up is going to be Visual Studio, an IDE made by Microsoft.

Note IDE stands for “Integrated Development Environment,” which is just a software application that aids programmers (like you) in creating their own software. These integrated development environments typically consist of tools like a source-code editor (like a notepad), debugging system, and a build automation suite, to name a few.

Installing Visual Studio will allow us to select certain C++ development packages that Amazon Lumberyard will require for compiling the game code. Don’t worry, we are not going to be writing any C++ in this project; the engine just requires C++ compilers and libraries to be installed.

On your web browser of choice, navigate to <https://visualstudio.microsoft.com>. Note that there will be many options for Visual Studio. Microsoft typically releases a new edition every other year, so at the time of your reading this, there may be a new edition out. For this book, we will be

using Visual Studio 2017 although any future editions will still suffice when the 2017 edition becomes antiquated. Do not get too hung up on editions of external software; if Amazon Lumberyard supports it, just take the latest version.

Be careful not to download *Visual Studio Code*. While both are made by Microsoft, Visual Studio and Visual Studio Code are not the same application. Visual Studio Code is a more lightweight text editor that has some debugging features, and Visual Studio is a fully fledged IDE packed with features for developing, testing, and deploying projects built in many different frameworks and languages. Visual Studio Code will not come bundled with the necessary development components that Visual Studio provides.

Under the area labeled “Visual Studio IDE,” click “Download for Windows,” and you will be greeted with three options: Community, Professional, and Enterprise. Any of these will work, but for the purposes of this text, we will be selecting the Community edition (as it is the free option). Professional and Enterprise editions are more feature-rich and suitable for more advanced software development outside of the scope for development of a simple game. For new developers, the environment may be more overwhelming than necessary. Select “Community,” and your download will begin.

Run the .exe executable that you just downloaded in order to install the Visual Studio Installer. If prompted by a window asking, “Do you want to allow this app to make changes to your device?,” select “Yes.” You will now be looking at the main GUI for the Visual Studio Installer.

Note GUI stands for “Graphical User Interface,” and it is exactly that – an interface with graphics that allows users to interact with software systems through means of visual indicators like graphical icons, buttons, and photos.

This is where we will select what we need, where to install what we need on our machine, and what we feel comfortable leaving behind. Select the tiles “Desktop development with C++,” “Game Development with C++,” and “Linux Development with C++,” pictured in Figure 1-1, then select “Install” in the bottom-right corner of the window.

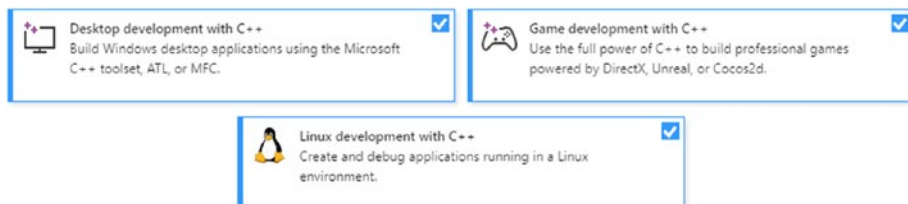


Figure 1-1. These tiles represent the required C++ components needed to compile game code in Amazon Lumberyard

You will now have a working copy of Visual Studio as well as the required C++ additional components installed on your machine.

Installing the Amazon Lumberyard Engine

Now that we have Visual Studio Community 2017 installed, it is time to install and set up the main software program we will be using throughout the book – Amazon Lumberyard. Navigate to <https://aws.amazon.com/lumberyard/downloads>. Click the button that says “Download Now” under the tile currently titled as “Lumberyard Beta 1.17,” but just remember that version numbers do change, so the page title may be different at the time of your reading this. Download the latest version of Amazon Lumberyard and launch the executable when prompted, keeping in mind that it is required that you will be using a Windows machine running Windows operating system 7 or higher.

Currently, Amazon Lumberyard is not wholly supported on Macintosh systems, despite Lumberyard Mac Support Files available on the download

page. This download will include some source code and tools required to build and run the Lumberyard game engine for developers creating macOS, iOS, and Android games on a Macintosh. However, the Asset Processor, Lumberyard Editor, and Remote Shader Compiler will require Windows 7 or later in order to build game assets and edit levels. If you want to run commands from your Macintosh's terminal, you must have access to a computer with a version of Amazon Lumberyard installed.

Note Any hardships you may encounter along the way of your environment setup can likely be resolved by navigating to the Amazon Lumberyard documentation currently available at <https://docs.aws.amazon.com/lumberyard>. If a solution still cannot be found, try developer forums like <https://gamedev.amazon.com/forums> or even the popular Game Development Stack Exchange QA site, <https://gamedev.stackexchange.com>, searching via the “lumberyard-engine” question tag.

Once you have downloaded the LumberyardInstaller[versionnumber].exe file, navigate to it in your file system and double-click it in order to launch the executable. After the initial install process, click the button that says “Launch Lumberyard Setup Assistant.” The Lumberyard Setup Assistant will greet you with two options: express install and custom install.

Express Install

Express install will give you everything you need to get started with the Amazon Lumberyard Editor, and it will allow you the option to come back to the Setup Assistant later to add in any plugins you may want, as well as SDKs that you may need to add to your project that were previously unconsidered.

Note SDK stands for “Software Development Kit,” which is a set of development tools that allows developers to create applications for certain software frameworks, platforms, or packages. For example, the development of an Android app on a Java platform requires developers to install the Java SDK.

Custom Install

Custom install allows you to select any plugins or SDKs at the time of your installing Amazon Lumberyard. For the purpose of the project, we will want to select this installation type. Click the orange *Customize* button in the *Custom Install* tile in the Lumberyard Setup Assistant, shown in Figure 1-2.

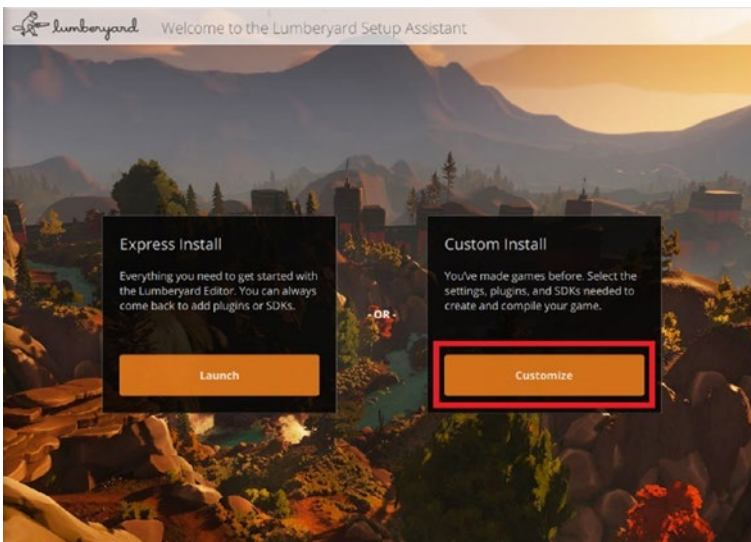


Figure 1-2. The first interaction with Amazon Lumberyard Setup Assistant we will have – selecting a custom install

You will be presented with a few groups of checkboxes. Each item on the left submenu is a group, and each group will have a selection of checkboxes that relate to that group's categorical text.

Get Started

First the *Get Started* tab will pop up, and there will be a subtitle *What do you want to do with Lumberyard?* displayed. By default, the “Run your game project” and “Run the Lumberyard Editor and tools” options will be selected. If these options are not selected, go ahead and select them now, along with the “Compile the game code,” “Compile the engine and asset pipeline,” and “Compile the Lumberyard Editor and tools” options, as these will be the five options we will need. We are not going to select any other options for now. Under the subtitle *Visual Studio Version* in the “Get Started” group we are already on, we are going to choose the “Visual Studio 2017” option (deselect any other version if chosen by default).

Again, keep in mind that because Visual Studio versions will change, you should select the option that corresponds to the Visual Studio platform you installed earlier in the chapter. Figure 1-3 shows which selections we will have made in the “Get Started” group, assuming “Run your game project” and “Run the Lumberyard Editor and tools” options were already selected. Click the orange *next* button in the bottom right corner to advance to the next group of the Amazon Lumberyard Setup Assistant, titled *Install Software*.

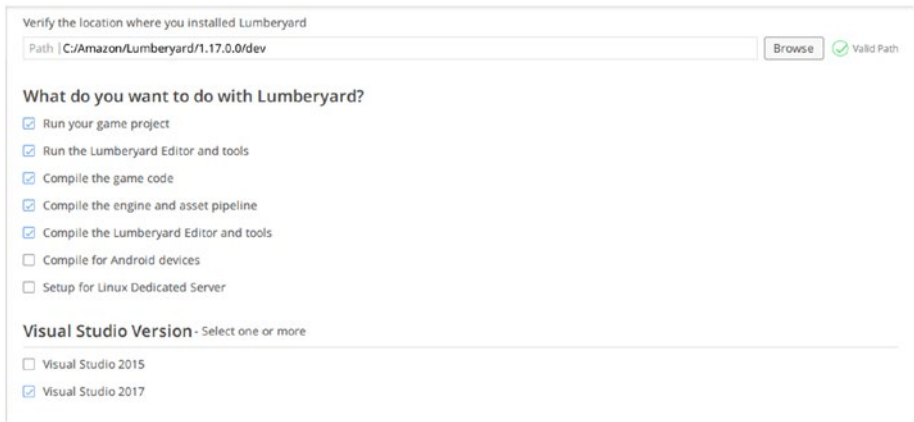


Figure 1-3. The necessary selections to make in the “Get Started” group of the Amazon Lumberyard Setup Assistant

Install Software

The required software for Amazon Lumberyard includes *Microsoft DirectX Redistributable*, which allows multimedia-rich applications to run on Windows-based machines, as well as *Visual C++ Redistributable for Visual Studio 20XX* and *Visual Studio 20XX*. Note here that I put “20XX” because you will need the Redistributable packages for whichever Visual Studio platform you chose to use. This package will install runtime components that are required to run C++ applications, which Amazon Lumberyard will be building in the background.

Ensure that all fields under the required software subtitle are installed, by checking the “Status” column of those fields. If the software is installed, you will see a green check mark in this column, and if the software is not installed, you will see a red X. If the software is partially installed, or perhaps the version of the software is out of date or deprecated, you will see a yellow warning sign. All three of these symbols are shown in Figure 1-4.



Figure 1-4. *The status column of required and optional software fields shows a green check mark if the software is installed, a red “X” if the software is not installed, or a yellow warning indicator if the software is partially installed or requiring an update*

Optional software includes *Audiokinetic Wwise LTX Authoring Tool*, which is an advanced, interactive sound engine for games, *FFmpeg*, which is video encoding software, and *SpeedTree for Lumberyard*, which is a program for modeling and real-time rendering of trees and plants. We will not be opting to install any of the optional software for our game; thus, we will just leave these programs uninstalled and ignore this section of the Amazon Lumberyard Setup Assistant.

Once your required software packages are all installed and showing the green check-mark indicator, click the orange *Next* button in the bottom right-hand corner of the screen to move on to the *Install Required SDKs* group of the Setup Assistant.

Install Required SDKs

Need a break? This is one of the easiest steps of the entire setup of Amazon Lumberyard. This is where we will install all of our required SDKs, including SQLite, RapidJSON, MD5, and, of course, Lua. The Amazon Lumberyard Setup Assistant has made this step a breeze by adding a “Install all” button at the top of this group. You will see a button titled “Browse” above the “Install all” button next to an entry field titled “Third-party path.” Click the “Browse” button, and select the directory in which you would like to install all these SDKs. If you are okay installing them in the default location, do not worry about this step. Go ahead and click the “Install all” button now, shown in Figure 1-5, in order to install all the required SDKs for our game development process.



Figure 1-5. *The required SDKs are the easiest part to install when setting up Amazon Lumberyard, due to this “Install all” button*

Grab some popcorn, this may take a while. You can see the progress of your SDK downloads in the bottom left-hand side of the Lumberyard Setup Assistant window, shown in Figure 1-6. Like the “Install Software” group earlier, the Required SDKs table has a column labeled “Status” that will show the green check mark if the SDK has been installed, a yellow warning indicator if the SDK has only been partially installed or requiring an update, or a red “X” if the SDK is not installed.



Figure 1-6. *The SDK download progress bar shows the percentage of required SDKs that have successfully been downloaded*

Once all your required SDKs have been installed, click the orange Next button in the bottom right-hand corner of the screen to move on to the Install optional SDKs group of the Setup Assistant.

Install Optional SDKs

Like the “Install Required SDKs” group, there will be an entry field labeled “Third-party path” and a “Browse” button in order for you to select where you’d like any optional SDKs to be installed; however, unlike the “Install Required SDKs” group, there is no convenient “Install all” button. That is okay, as there is only one of the optional SDKs we will need. At the time of writing this, the Clang SDK is considered optional in the Lumberyard Setup Assistant, but it is required to create new projects and open them in the Lumberyard Editor. Select the orange “Install SDK” button next to the

Clang option, shown in Figure 1-7, if you do not already have a version of Clang on your computer. Again, you will see the blue progress bar in the bottom left-hand corner of your screen while the SDK installs. As stated earlier, we will not be using any other optional SDKs, although, after you have finished the book and have become an Amazon Lumberyard wizard, I would highly recommend you come back and install some of these SDKs to implement some more advanced functionality into your project.

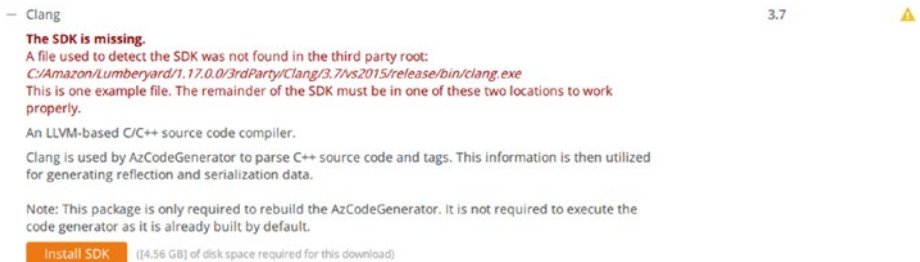


Figure 1-7. While the Lumberyard Setup Assistant claims that Clang is an optional SDK, it is required to build your project and engine

One of the more interesting optional SDKs is the Twitch Commerce SDK, which provides access to social functions, login, chat, and other APIs associated with the famous livestreaming service Twitch.

Because we will not be utilizing anymore of the available optional SDKs, select the orange Next button in the bottom right-hand corner of the screen to continue to the “Install Plugins” group.

Install Plugins

In the “Install Plugins” group, we have the option to install popular content creation software like various versions of Autodesk Maya, Adobe Photoshop, and Autodesk Max, as well as RC Shell Commands. We also have the option, instead of installing these programs, to locate them on our hard drive if we were to have them already installed on our machines.

We are not going to need any of these plugins for the purposes of this project, although I would again recommend that you come back to this page and experiment once you are more familiar with the engine.

Click the orange *Next* button in the bottom right-hand corner of the screen to continue to the final group of the Amazon Lumberyard Setup Assistant – “Summary.”

Install Optional Software The final group of the Amazon Lumberyard Setup Assistant will give you one final option to install any optional software you may want. Do not worry, you will be able to come back to the Setup Assistant even after it is closed in order to install these again later in the future. You will now be able to launch the editor, configure a new project, or enable gems, which are extensions of Amazon Lumberyard.

For now, let’s click the orange *Close* button in the bottom right-hand corner of the screen to finalize our setup.

Running the Lumberyard Editor

Next, we will run the Lumberyard Editor. Press the Windows key, type “Lumberyard Editor,” and select the Lumberyard Editor Icon to launch the software. You may, depending on your user status on your personal machine, be greeted with a Windows Defender Firewall Security alert saying that some features of the application have been blocked. In order to allow all editor features to work as intended, select the “Allow Access” button if this happens.

Summary

Congratulations, you have just completed the setup of the Amazon Lumberyard engine. In the next chapter, we will learn to navigate the engine, focusing on helpful keyboard shortcuts, pane navigation, and other general terminology related to the program.

CHAPTER 2

Navigating the Engine

Now that you have Amazon Lumberyard installed on your computer, this chapter will walk you through the layout of the engine. We are going to talk about pane navigation, some useful hot keys to help you stay more efficient in your work, and some terminology of engine-related topics. We still will not be creating any game elements or logic yet, as it will be more beneficial for us to navigate through the environment and learn the ins and outs of the Amazon Lumberyard engine before we try to jump into development mode.

Open the Engine

The first, most obvious step we will need to take is opening up the engine. While it sounds easy enough, there are a few different ways to do it, and there are some conflicts that arise if you have multiple versions of Visual Studio on your computer. Because Amazon Lumberyard is in beta, it is possible that these conflicts will be resolved within the next year or two, but for now, let us walk through the process of opening the engine in a few different ways.

Note Beta is the second phase of the software development life cycle. The beta phase generally begins when software is seemingly complete but likely contains a few unknown bugs. The benefit of beta testing is that it allows users to find bugs by using the software in a way the developers previously had not thought of.

The first method, and likely the easiest if your computer only has one version of Visual Studio, is through the start menu. Click the Windows icon or press the Windows key on your keyboard to open the start menu. Type “Lumberyard Editor” and select the gray Amazon Lumberyard logo, shown in Figure 2-1.

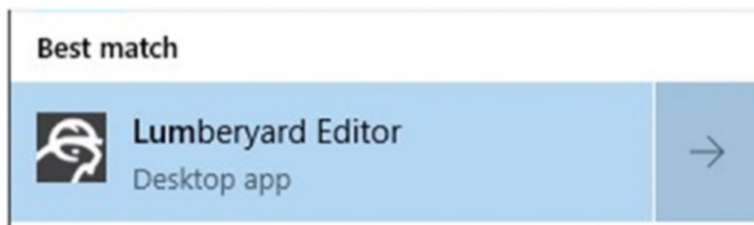


Figure 2-1. While the *Lumberyard Setup Assistant* claims that *Clang* is an optional SDK, it is required to build your project and engine

If all goes well, you may automatically be signed in to your Amazon account. If you are not automatically signed in, the startup of the Lumberyard engine may prompt an Amazon Login at this point. If you have made it to this stage, you have started up the engine, and you are ready to get to work. If this fits your case, move on to the next section, titled “INSERT_TITLE_HERE.” If this did not work for you, keep reading, and we will come to a solution.

If you got an error message saying, “An error occurred while loading gems ... please build the current project before attempting to run Asset Processor again”, shown in Figure 2-2, followed by an error about the