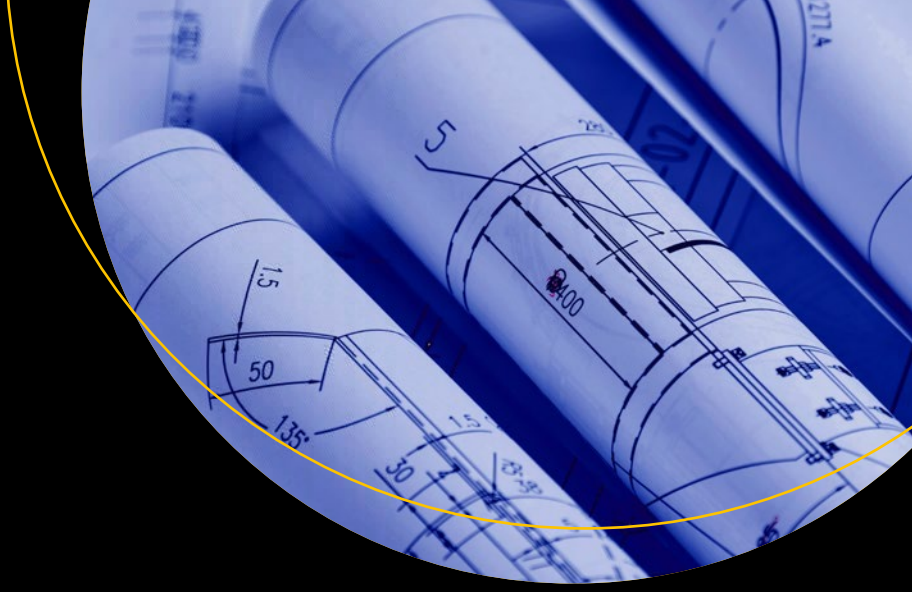

Beginning Unreal Engine 4 Blueprints Visual Scripting

Using C++: From Beginner to Pro

Satheesh Pv

Apress®

Beginning Unreal Engine 4 Blueprints Visual Scripting

Using C++: From Beginner to Pro

Satheesh Pv

Apress®

Beginning Unreal Engine 4 Blueprints Visual Scripting

Satheesh Pv
Mumbai, India

ISBN-13 (pbk): 978-1-4842-6395-2 ISBN-13 (electronic): 978-1-4842-6396-9
https://doi.org/10.1007/978-1-4842-6396-9

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Spandana Chatterjee
Development Editor: Matthew Moodie
Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. 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 booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

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/978-1-4842-6395-2. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Thanks to Jesus for His guidance and to my wife, mother, and brother for their continuous support, and a big thanks to Apress for giving me the opportunity to write this book

Table of Contents

About the Author

Satheesh Pv is a game programmer living in Mumbai, India. He started his career as a game developer in 2012 by making a first-person multiplayer game with his brother and close friend using the Unreal Development Kit. Satheesh created Unreal X-Editor, an IDE developed for UnrealScript, the native scripting language of Unreal Engine 3. He was selected by Epic Games as one of the closed beta-testers for Unreal Engine 4 before its public release. He is also a moderator at Unreal Engine forums and a spotlight member and engine contributor.

About the Technical Reviewer

Pranav Paharia is a game developer who has worked on game technologies like Cocos2dx, Unity3D, and Unreal Engine 4. He has a bachelor's degree in information technology and a postgraduate degree in game development. After realizing his die-hard interest in games, he started his career in game development by working for Indie Game Studios making mobile games in many genres. One of the projects he worked on, *Song of Swords*, won the NASSCOM 2013 People's Choice of the Year Award. He has worked on a variety of systems for games, including gameplay, multiplayer, data pipelines, and cinematics. He is proficient in C++ and C# and can work on any game technology to create mind-boggling simulations. He is a self-taught programmer and designer.

Since 2013, Pranav has created simulations for single-player games, multiplayer games, card games, VR games, AR simulations, serious games, training simulations, and learning games. He has also worked on a few game development books. With his vast experience in creating simulations, he is now involved in solving real-life problems using the latest technologies, such as creating architectural visualizations, VR training systems, and medical data imaging for clients like DRDO India, Zaha Hadid, Line Creative, and MediaMonks.

Apart from developing graphical simulations, Pranav is currently working on creating virtual productions tech using Unreal Engine. He is an avid gamer who loves Dota 2. He also has keen interest in photography, reading books with philosophical context, and riding his bike on long road trips. He is grateful to Krsna for guiding him through his purpose in life. You can contact him at pranavpaharia@gmail.com; also check out his website at www.pranavpaharia.com.

Introduction

This book covers the basics of Unreal Engine, including Blueprints, materials, and C++. It starts with downloading Unreal Engine using Epic Games Launcher and using the GitHub version. From there, it moves forward to Blueprint classes and the common classes, such as Game Mode, Game State, Game Instance, and Player Controller. You learn about how to add C++ and get a brief introduction to Unreal C++, materials, and physics. At the end of the book, you make a small demo game extended from a first-person template using Blueprints. In this demo game, you learn how to add ammo and ammo pickup.

This book is primarily aimed at beginners who want to learn more about the Engine, how a project is structured, Unreal Blueprints, and C++.

CHAPTER 1

Introduction to Unreal Engine 4

Hello there, and welcome to this beginner's guide to Unreal Engine 4. Throughout this book, you learn about different aspects of Unreal Engine 4, and you learn to create a sample game with the knowledge you gain. In this chapter, you learn how to download Unreal Engine through Epic Games Launcher and GitHub. After that, you learn how a project is structured and become familiar with the Unreal Editor interface.

Getting Unreal Engine

This chapter looks at how you acquire Unreal Engine. You can download it through either Epic Games Launcher or GitHub. Either way, you need to create an account at www.unrealengine.com, which is free.

First, let's look at the differences between Epic Games Launcher and GitHub.

S. Pv, *Beginning Unreal Engine 4 Blueprints Visual Scripting*,
https://doi.org/10.1007/978-1-4842-6396-9_1

- The Epic Games Launcher version (a.k.a. the binary version or vanilla version) of Unreal Engine 4 comes with the engine prebuilt, and you can select the platforms you need. You can also select the engine source, starter template, feature packs, and so forth, if you need them. The binary version does not support creating dedicated servers for your game, so if you are planning to develop a multiplayer game with a dedicated server, you must use the source version.
- The GitHub version (a.k.a. the source version) gives you the entire source code of the engine without any binaries, so you need to compile it manually. The source code version of the engine is typically used by developers who want to fix the engine's bugs or add new features. This version is also required if your game relies on a dedicated server. The prerequisites for the source code version of the engine are Visual Studio 2019 (or higher) on Windows or Xcode on macOS.

Download from Epic Games Launcher

If you don't have an Epic Games account, you need to create one at `www.unrealengine.com/id/register`.

If you do have an Epic Games account, then head over to `www.unrealengine.com/en-US/get-now` and select your license to download and install Epic Games Launcher for your platform. After installation, open the launcher, and log in using your credentials. You should see the screenshot shown Figure 1-1.

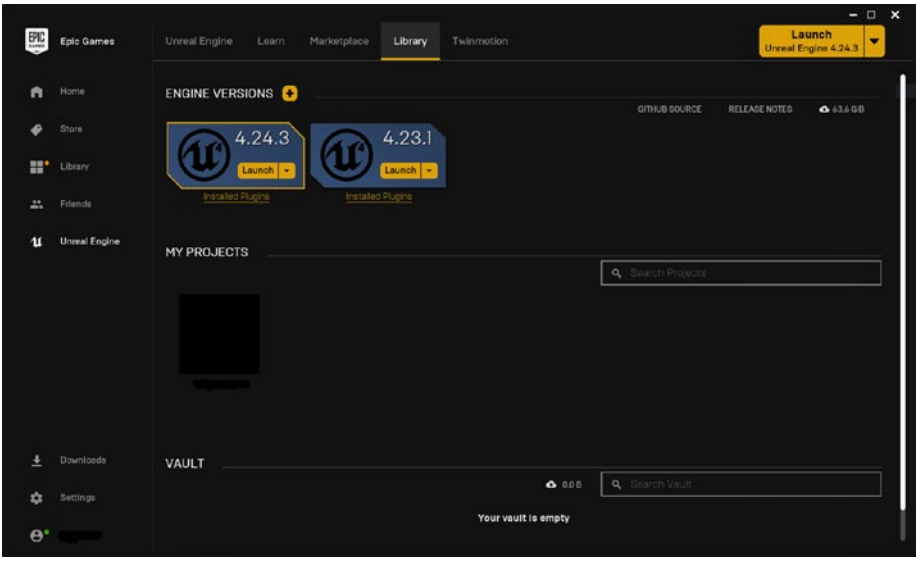

Figure 1-1. *Epic Games Launcher with engine version4.24 installed*

Near the ENGINE VERSIONS tab, you can see a + button, which allows you to download and install any engine version you want.

Download from GitHub

If you prefer to work with the source version of the engine, you can do so by downloading the engine source code and compiling it yourself, but you must have Visual Studio 2019 (with C++ support enabled) installed if you use Windows or Xcode if you are on macOS.

First, you must create a GitHub account (it's free) and log in to your Epic Games account. Once logged in, go to your account dashboard in Epic Games and link your GitHub account. After this, you are ready to download the full source code for Unreal Engine 4.

Downloading Source Code

Once you have access to the Unreal Engine repository, you can click the **Clone or download** button and select the **Download ZIP** button (as seen in Figure 1-2).

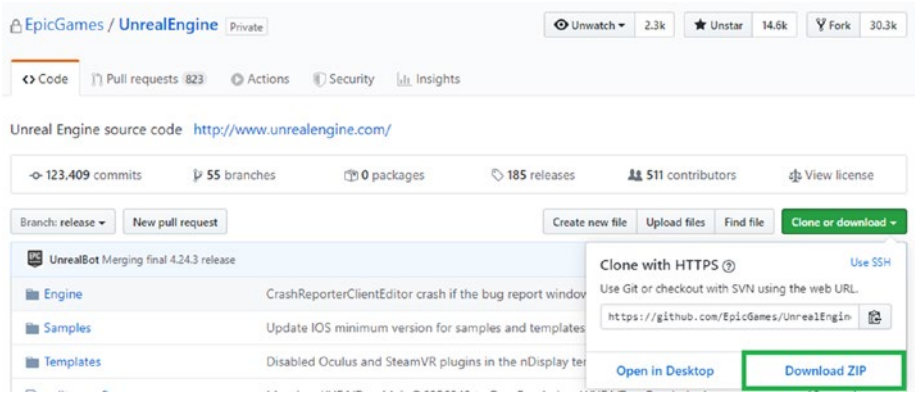

Figure 1-2. *Download ZIP button in Unreal Engine Git repository*

Cloning the Unreal Engine Repository

To clone a repository, you need to have a Git client installed. Cloning is the process of downloading or copying a repository into an empty folder in your working machine, including the full Git history, so you can use Git commands. You only download the source code without any Git files, so you won't track changes or have any information about previous commits.

I use SourceTree from Atlassian.

Note If you prefer other tools, please visit `https://git-scm.com/download/gui/windows` for Windows or `https://git-scm.com/download/gui/mac` for macOS.

After installing SourceTree, open the application. In the new tab, select **Add an account**. In the new window, switch the hosting service to GitHub and select the **Refresh OAuth Token** button. Once SourceTree has access to your repos, you can select **Unreal Engine repo** from your repositories list and then select **Clone**. This lets you choose a path to save the files. Under **Advanced Options,** select the **release** branch and click the **Clone** button.

Once cloning is done or after downloading the ZIP file, go to the directory and double-click the **Setup.bat** file. (If you downloaded the ZIP file, extract it first). You can include or exclude specific platforms by passing the necessary flags in the Setup.bat file. For example, to exclude Mac and iOS platforms on a Windows machine, you can run Setup.bat like this:

```
Setup.bat --exclude=Mac --exclude=iOS
```

This ensures that any dependencies and files required for the Mac and iOS platforms are skipped. Once Setup.bat finishes, run **GenerateProjectFiles.bat**, which generates the UE4 solution file that you can open in Visual Studio. After opening the solution file, you can see UE4 under the Engine folder in Solution Explorer. Right-click UE4 and select Build. This starts the build process, which might take an hour or more to compile, depending on your hardware.

Getting to Know Unreal Editor

Now that you have installed (or compiled) your engine, let's start it up. Throughout this book, we only work with the binary version of the engine, which is 4.24. You create a blank project and learn about the aspects of the engine. To start the engine, click the Launch button for 4.24.3. This opens the **Unreal Project Browser** window, where you can select an existing project or create a new one from scratch or a template (see Figure 1-3).

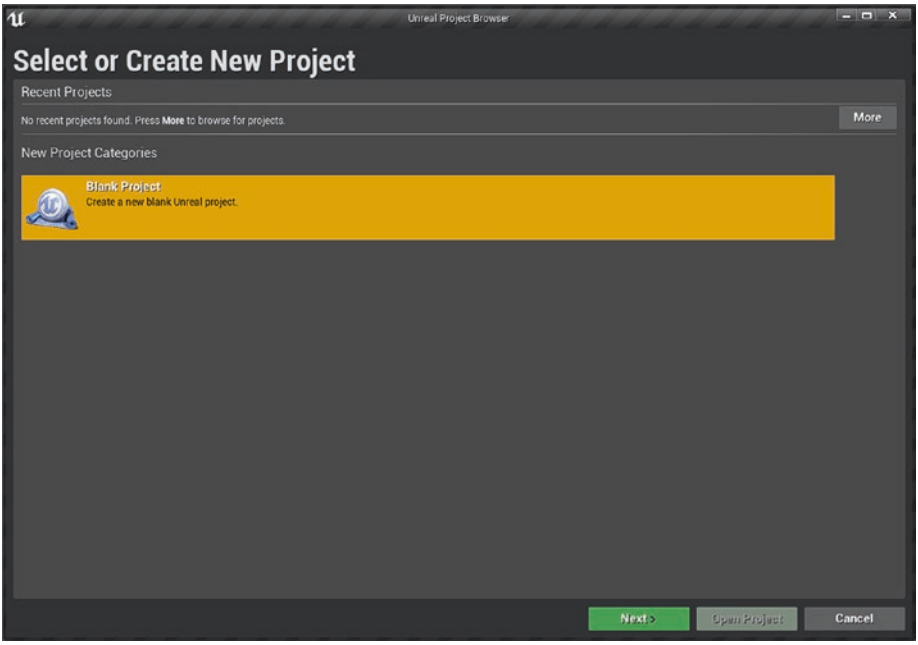

Figure 1-3.

Let's select **Blank Project** and click Next. On the next page, you are prompted to either start a blank project or create one based on a template. For our purposes, let's select a blank template and click Next. This gives you a project with no code or content and with the default settings. Finally, the last page allows you to do basic configuration and name your project.

Let's go through the Project Settings page shown in Figure 1-4.

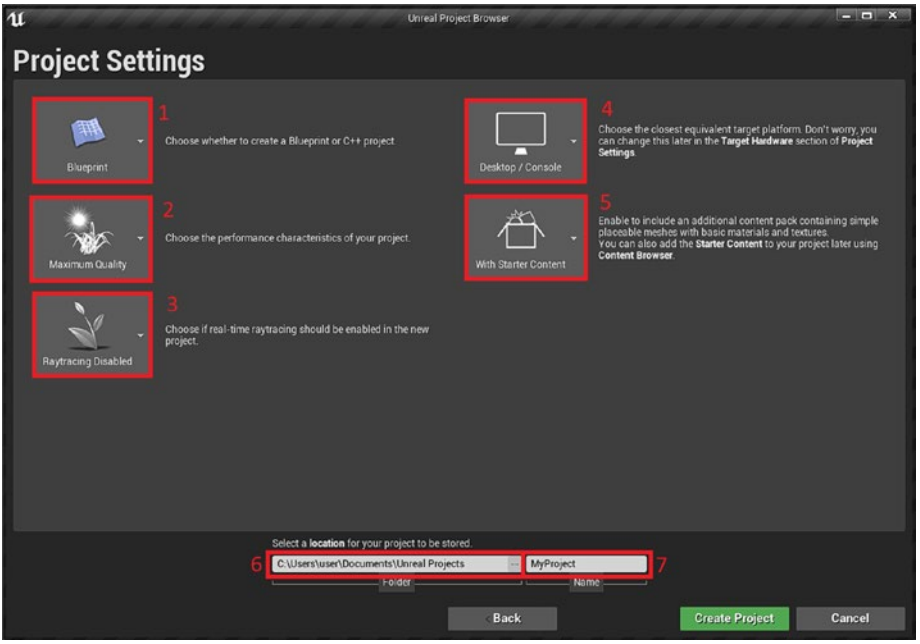

Figure 1-4.

- **Blueprint** (labeled 1 in the screenshot) lets you choose whether your project is based on Blueprints or C++. If you start in Blueprints, you can later add C++ code to your project.
- Depending on your project, you can change **Maximum Quality** (labeled 2 in the screenshot) to **Scalable 3D/2D**. The first option is suitable for PCs/consoles, and the second option is suitable for mobile.
- If you target high-end PC games and own an Nvidia RTX graphics card, you can enable **raytracing** features (labeled 3 in the screenshot) for your game.
- **Desktop/Console** (labeled 4 in the screenshot) lets you select the closest equivalent target platform.