Unreal Engine Projects for Non-coders

A guide to Unreal Engine's visual environment asset pipelines and basic VR concepts

Myron Mortakis

www.bpbonline.com

First Edition 2025

ISBN: 978-93-65896-350

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true and correct to the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but the publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My beloved wife ***Nicole***

My parents, ***Ioannis*** *and* ***Jennie***

and

The spirit of persistence and dedication

About the Author

A Greek-Brazilian-American developer and entrepreneur, **Myron** is HELM Systems' (a South Florida-based independent game development studio) founder, CEO, and lead developer, with more than 20 years of experience in interactive software development and an expert in Unreal Engine, which he has been working with since its first generation. He is knowledgeable in a vast multitude of development roles, making him the perfect individual for efficient project planning and management, from conceptualization to launching a finished qualitative product for a plethora of applications and industries, including but not limited to gaming, entertainment, aerospace, industrial, architectural, VR/AR/XR and more.

He has an excellently balanced approach between creative passion and realization of business necessities, with a strong emphasis on efficient and cost-effective development, with product quality and timely delivery being top priorities.

He has been a recipient of the US residency as an alien of extraordinary ability in the field of video game and interactive software development while has also received an IGF award nomination and an Epic Games Unreal DevGrant for work conducted in video game development, as well as press mentions both for his video game development work and an interactive VR end-product visualization application for the aerospace industry. He also served as a panelist and presenter at conferences. In his free time, he is an avid electric guitar player and musician who composes, plays, and produces original music, having an extensive knowledge and a lifelong passion for music production equipment and techniques.

About the Reviewer

Ilya's journey in game development began at a small startup, where the team focused on developing MVPs for presentations while continuously expanding their knowledge through books and online resources to master Unreal Engine 4. After 1.5 years, he accepted an offer from Xtrematic, a company specializing in Unreal Engine 4-based VR projects for arcade VR machines. There, he deepened his expertise in VR development, learning the intricacies of VR systems within Unreal Engine and optimizing both logic and graphics for immersive experiences.

During the COVID-19 period, the company nearly shut down, prompting Ilya to transition to AltwolfSoftware, where he joined an outsourcing team working for Flying Wild Hog on Space Punks. His contributions included gamepad support, UI improvements, and AI enhancements. After the game's release, he took on the role of a lead developer for a small outsourcing team on the decimated project, managing a team of four developers responsible for implementing player character features.

After nearly two years at AltwolfSoftware, Ilya moved to the UK to join Cloud Imperium Games and work on Star Citizen. For the past 2.5 years, he has been part of the mission features team, implementing mission-related functionalities and broader game improvements. In recent updates, he contributed to the UI implementation of mobiGlas and took part in gameplay improvements related to server meshing for the 4.0 update, enabling hundreds of players to interact within a single session.

In addition to his professional work, Ilya teaches Unreal Engine and C++ programming at an online school and runs his own YouTube channel, SikorskiPishet, where he shares his expertise with aspiring game developers.

Acknowledgement

I would like to thank my wife Nicole for standing by me and showing me continuous support while authoring this book as she has with every other aspect of life, and the choices I have made and continue to be making.

Much gratitude to all the people I have learned from or got inspired by in my 20+ year journey so far. I am immensely grateful to the teams I have worked with, as well as the projects I have undertaken and the challenges I have faced, as they have always driven me forward to keep improving, growing, and learning further. Furthermore, I want to give special thanks to certain people I have met along the game development road so far, specifically Christian Allen, Amanda Schade, Marc Brehme, David Jagneuax, James Lay, Nick Calandra, Ricardo Robles Ortiz, Lawrence Nembhard, Mike Herauf, Richard Cowgill, Doug Kennedy, Marc Diana, Mike Poropat, Andru Fratarcangeli and Alain Galvan.

I owe an immense gratitude to the team at BPB Publications for providing guidance and support, but most importantly for being patient with someone that engages in a book authoring endeavor for the very first time. I thank the entire team for their professionalism, communication and guidance, and for the opportunity to complete this endeavor.

Preface

This book is meant to start with very basic introductions to Unreal Engine 5, its core technologies, tools, and workflows. It targets mainly beginners and quickly accelerates their learning to an intermediate level, with a good general understanding of the overall Unreal Engine workflows.

It starts with small, simple, and easy-to-complete projects and then proceeds to a complex project that utilizes everything taught in the earlier chapters of the book.

The book is divided into 16 chapters. It starts with the most basic steps, such as downloading and installing the Unreal Engine and setting up a working environment, explaining the basics of the engine's UI, toolsets, and workflows. It continues with a very basic introductory project that serves as an introduction to the tools and workflows later projects will utilize in more depth, while offering brief and optional examinations of the workflow of external applications and their relation to the Unreal Engine, mainly for asset creation.

Since chapters on external application workflows are optional, the author ensured that all created and exported assets are readily available for the readers should they wish to retain their focus strictly on the Unreal Engine workflows and processes.

Ultimately, the readers are guided through completing a complex project that allows even a beginner user to advance to an intermediate level, while allowing an intermediate user to possibly freshen up or further their knowledge in specific workflows.

Lastly, the book closes with a brief introduction to virtual reality.

Chapter 1: Unleashing the Unreal Engine– This chapter provides the reader with a very brief introduction to the Unreal Engine's various applications and history, as well as the author's and proceeds to show in detail to first-time users how to download, install, and set up Unreal Engine 5 on their machines.

Chapter 2: Unreal Engine UI- Breaks down the basic UI elements of the Unreal Engine's default layout, facilitating first-time users' overcoming of the seemingly overwhelming interface and showing them how to navigate through 3D space.

Chapter 3: Unreal Engine's Building Blocks- Guides the reader through their first project by completing a simple scene. Thus, the book teaches the reader about level design, Static Meshes, Materials, lighting, and Blueprints in a very simple, introductory manner.

Chapter 4: Project Overview and Main Asset Creation for Statue Scene– Analyzes the book's first complex project and breaks it down into parts, proceeding then to an optional in-depth overview of the 3D modeling and overall asset creation workflow, utilizing external applications, namely Reallusion Character Creator 4, ZBrush and Substance 3D Painter, ending the chapter with the preparation and exporting process of the asset, for the Unreal Engine.

Chapter 5: Importing Assets and Setting Up the Statue Scene- Shows the reader how to properly import and organize the previously created assets in Unreal Engine 5. Then, it proceeds to utilize these assets for the creation of Materials and finalization of the Statue Scene within the engine.

Chapter 6: Lighting and Cinematic for the Statue Scene– Explains lighting in further detail, and guides the reader through the steps of lighting up the Statue Scene. Then, it proceeds with utilizing some of the Unreal Engine's tools for the rendering of a short cinematic sequence.

Chapter 7: Fantasy Castle Project Breakdown and Planning– Analyzes and breaks down the book's most complex project. Overall, an introduction to careful project planning of a complex project, which can result in smaller, manageable segments.

Chapter 8: Fantasy Castle Base Mesh Modeling- Is diving into the workflow of creating the base meshes that will later on make up a complete asset pack. This is achieved by looking into the workflow of an external application, namely 3ds Max.

Chapter 9: Fantasy Castle High Poly Mesh Sculpting– Will continue the workflow that Chapter 8 initiated by working on the previously produced base meshes in ZBrush and sculpting them to transform them into high-poly, detailed meshes.

Chapter 10: Fantasy Castle Texturing and Materials– Proceeds with Chapter 9's results, and finalizes their Materials and textures in Substance 3D Painter, ending with finalization and preparation for exporting to the Unreal Engine.

Chapter 11: Fantasy Castle Bringing It All in Unreal– Goes through a step by step process of importing and organizing the previously created assets into the Unreal Engine, then proceeds with finalizing the assets within Unreal, ending the chapter with level design, by placing the created assets in the scene and utilizing Unreal Engine's terrain and Foliage tools.

Chapter 12: Fantasy Castle Character and Interaction Blueprints– Goes deeper into Blueprint utilization, teaching the reader how to create an original playable character within the Unreal Engine, looking briefly into concepts such as retargeting and in-engine

animation, ending with assigning the playable character some interesting functions and abilities.

Chapter 13: Fantasy Castle Interactive Blueprint Actors– Continues with further utilization of the Unreal Engine Blueprints, by showing the reader how to create some interactive actors, such a firepit that can be toggled on and off, a teleportation portal.

Chapter 14: Fantasy Castle Mini Game Blueprints– Carrying on with the spirit of the previous two chapters, in this chapter, the reader goes through the steps of creating a mini game and its necessary components, all to be used in the Fantasy Castle Project.

Chapter 15: Fantasy Castle Level Finalization and Packaging– Goes through the steps of finalizing and completing the project, packaging it, and creating an executable format.

Chapter 16: Statue Scene Introduction to VR– Serves as a very brief introduction to virtual reality, by modifying the existing Statue Scene project into a VR project.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/c1m9l39

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Unreal-Engine-Projects-for-Non-coders.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Unleashing the Unreal Engine

Introduction

This chapter will give you a very brief introduction to the Unreal Engine history and the author's journey with it and then will guide you through downloading, installing, and running the Unreal Engine for the very first time. Most importantly, you will familiarize yourself with the Epic Games Launcher, the proper steps to installing the Unreal Engine, and the ability to access and select different engine versions, while you will receive an overview of the different types of project templates the engine offers from the beginning. In addition, you will learn how to properly set up your work environment and the very first project.

Structure

The topics covered in this chapter:

- History of the Unreal Engine and the author's experience with it
- Getting started with Unreal Engine

Objectives

By the end of this chapter, you will have a better understanding of the Unreal Engine's origins, evolution, its different applications to today's industries. You will also be

comfortable with downloading and installing the Unreal Engine, while being able to understand and select the different installation options, engine versions and Starter Content.

History of the Unreal Engine and the author's experience with it

The author's first interaction with the Unreal Engine came in 1999, playing *Epic Games Unreal Tournament*. It was a well-balanced, good-looking multiplayer **First Person Shooter** (**FPS**), released at a time when titles like *Doom* and *Quake* were dominating the FPS market. As a gamer would say, experiencing the game was thrilling, especially when playing it with friends and family at **local area network** (**LAN**) parties. There was an immediate interest on the author's behalf to find ways to expand his gaming experience and game time, which eventually led to downloading free, community-made levels for the game. This was quite exciting, as it meant that one could freely expand the game they were already enjoying and experience designs that further contributed to the game's enjoyment.

Having the ability to download custom levels was quite exciting, yet there was even more content to improve and expand one's experience with the default game. There were mods, modifications that would range from something relatively simple, such as introducing a new weapon with different weapon stats, to something complicated and radically transformative, such as a total conversion, which, in essence, would be an entirely new game, running within the original game.

A gamer would certainly be excited about all these possibilities to expand further and enjoy any experience with the game however, upon discovering more about all the available custom-made content, the author's creative curiosity started growing, pushing him towards obtaining more information on how one could go about creating such custom content. While information and learning resources were extremely limited, often vague, and quite difficult to find, determination to learn using the Unreal Engine and the Unreal Editor was at least enough to allow the author to express his creative visions for the original game, initially through custom-built maps, later through modding and total conversions.

At that time, the Unreal Engine was considered an FPS engine since it was widely used for the development of first-person shooter games and content. However, in 2000, *Human Head Studios* released *Rune*, a third-person hack-and-slash action game, which was using the Unreal Engine and offered the opportunity to witness the Unreal Engine's application on a product other than an FPS.

As someone who has always had a soft spot for ancient and medieval world history and an inclination towards fantasy themed IPs, Rune elevated the author's curiosity and interest in the Unreal Engine to the highest levels possible, resulting in him spending entire days, weeks, and months in learning as much as possible, building and releasing custom levels and content, and actively participating in the Rune's modding and Unreal Engine communities.

At the same time, the author was actively engaged in the Rune modding scene, eagerly learning and working hard to improve his skills, Epic Games made shockwaves in the gaming world with news of *Unreal Tournament 2003* (in essence, Unreal Tournament 2), which would be the first title to use the — then new — Unreal Engine 2. Early screenshots and sneak preview trailers made both gamers and enthusiast developers drool over the visuals and the overall possibilities of the second generation of the Unreal Engine. At this point, the author was strongly motivated and determined to start bringing his creative visions and aspirations of a dark fantasy world to life. While still a college student working to obtain *Bachelor of Science* degrees in *Liberal Studies* and *Computer Science*, the author proceeded with forming a small game development team of enthusiast and amateur game developers with the purpose of creating our own fantasy total conversion, which would utilize the Unreal Engine 2 and feature controllable dragons, horses, first-person ranged and spell casting combat, third person melee combat. An impossible feat, as described by many we consulted with at the time, and an ambitious project, to say the least, yet also a testament to the Unreal Engine's capabilities as well as our team's determination.

At this period, any custom content, such as a level, a player skin, a weapon pack, or an entire total conversion, could be released only as freely downloadable content and could not be monetized. Only Unreal Engine licensees were able to release and monetize products built with the Unreal Engine, but that did not deter the author and countless others from developing and releasing such projects, especially since they were all in the earlier learning stages of their careers and game development journeys.

In 2005, the author and his team successfully released the first iteration of their total conversion, *The SoulKeeper*, featuring all the mechanics they aimed to include: controllable dragons, horses, three different types of combat, all in a multi-player format, an add-on to Unreal Tournament 2003 and Unreal Tournament 2004. They received mainstream online and print media coverage, were featured on CD compilations, topped several *Most Downloaded* charts on popular gaming websites, and most notably received an *Independent Game Festival Nomination* in 2006 while also being invited by Epic Games and the **Independent Game Developer Association** (**IGDA**) to showcase their game at **Game Developers' Conference** (**GDC**) 2006.

All these achievements were great and satisfying motivators, but looking back at those days now, the most valuable takeaway was that they set out to do something challenging and unheard of at the time, and thanks to the Unreal Engine and, of course, thanks to the hard work and dedication, they were able to successfully overcome all the challenges and complete the technical goals they set out to.

While all this was happening, in 2006, Epic Games were heavily promoting and sharing news and information about their — then upcoming — Unreal Engine 3. At GDC 2006, Unreal Engine 3 was the talk of the show, and it was impossible to overlook how good the visuals were, how impressive the lighting, the visual effects, the overall quality, and the overall technology looked. With Unreal Engine 3, Epic Games also introduced an effective node-based method for visual scripting (known as *Kismet*) as well as material shader composition, vastly facilitating, expediting, and, therefore, improving development

workflows. Nonetheless, it was very expensive to license, therefore being a commercially feasible option only to AAA studios, backed by AAA publishers, who could afford the expensive licensing fees. Still, the Unreal Engine 3 was, as one would expect, a powerful technological marvel and a highly successful and widely used gaming engine, powering an overwhelming majority of commercially successful, blockbuster AAA games in the years that followed and undisputedly establishing it as an integral and leading technology within the gaming industry. In Unreal Engine 3's lifecycle, **Unreal Development Kit** (**UDK**) was also released as a freely available educational tool, essentially a lighter version of the fully licensed Unreal Engine 3.

By this point in time, any Unreal Engine announcement about a new generation had become an industry standard, an expected and most certainly anticipated event, so it was only natural that once Epic Games decided to share more information and news about Unreal Engine 4, once again shockwaves were sent out through the entire gaming industry. At GDC 2014, Epic Games launched Unreal Engine 4; however, this time, the innovation was not just at the technical level. Still, Epic Games had also announced a new licensing model, under which anyone who would be willing to become a subscriber for USD $19 per month would be able to license and commercially release products built with Unreal Engine 4. Among the many innovations Unreal Engine 4 introduced, it also shipped an updated and largely improved version of Unreal Engine 3's tools, introducing Unreal Blueprints (an updated and improved version of Unreal Engine 3's Kismet, the node-based, visual scripting tool). Eventually, Epic Games decided to drop the subscription-based licensing model. They made the Unreal Engine freely available and accessible, allowing virtually anyone to create and launch commercial products utilizing the Unreal Engine.

In addition to the innovations mentioned above brought by Unreal Engine's fourth generation, Unreal Engine 4 started finding its way into completely different industries and applications beyond the video gaming industry.

Briefly looking into the author's adventures in a long and still ongoing development journey, in 2015, *HELM Systems*, the author's development company, collaborated with a company in the aerospace industry, building a real-time **virtual reality** (**VR**) application for the customization and end-product visualization of the fuselage of private jets. They developed this application using Unreal Engine 4, and it was compatible with both the *Occulus Rift* and *HTC Vive* headsets. In 2017, they released an early access VR game, *The SoulKeeper VR*, which was an innovative and daring product in the uncertain and challenging VR gaming landscape at the time, being featured once again in popular mainstream VR gaming publications, as well as receiving an Epic Games Unreal Dev Grant. In the following years, the company utilized Unreal Engine 4 in several ways that went beyond the scope of game development, including architectural visualization applications, training applications, virtual production utilizations, and online convention applications, just to name a few.

The Unreal Engine 4 has made its way into several well-known and mainstream products and brands as well, such as popular car manufacturer user interfaces and vehicle visual customization applications, TV series and films, business and academic applications,

architectural visualization, engineering applications, and many more. To this book's authoring date, the Unreal Engine 4 is widely used and can be found in many popular products, even with Unreal Engine's fifth generation already available.

This brings us to this book's authoring present, the latest generation of the Unreal Engine, the main subject of this book's focus, the Unreal Engine 5. Unreal Engine 5 was released in early access in the Spring of 2021 and officially launched a year later, in the Spring of 2022. Once again, Unreal Engine 5 introduces several new and innovative technologies that truly help improve production workflow and creativity and achieve unparalleled levels of realistic, qualitative visuals. We will cover several of those in the following chapters, but some of the most notable innovations are *Nanite* and *Lumen*. Nanite allows, for the first time ever, the importation of high poly geometries in a real-time environment, while Lumen is a powerful dynamic global illumination solution.

Getting started with Unreal Engine

As long and varied as Unreal Engine's history has been thus far, it continues to grow, finding itself in more and more applications and industries, connected with emerging and innovative technologies, so rest assured there are many more chapters and developments yet to be written in this history. And this is the perfect opportunity to begin your journey and become part of this unwritten yet part of the Engine's history. But let us begin with small, conservative, and simple — for now — steps. What better starting point than to first download and install the Engine on your computer. Let us get started with the following steps:

1. First visit Unreal Engine's official documentation website to get detailed hardware and software system requirements to ensure you have the proper system setup that can run Unreal Engine 5: **https://docs.unrealengine.com/5.0/en-US/hardware-and-software-specifications-for-unreal-engine/**

   For your convenience, you can refer to the following three figures (*Figure 1.1, Figure 1,2,* and *Figure 1.3)* as well, all three taken straight from the official documentation website, displaying the minimum hardware requirements for Windows, macOS, and Linux, respectively:

Operating System	Windows 10 64-bit version 1909 revision .1350 or higher, or versions 2004 and 20H2 revision .789 or higher.
Processor	Quad-core Intel or AMD, 2.5 GHz or faster
Memory	8 GB RAM
Graphics Card	DirectX 11 or 12 compatible graphics card
RHI Version	▪ **DirectX 11:** Latest drivers ▪ **DirectX 12:** Latest drivers ▪ **Vulkan:** AMD (21.11.3+) and NVIDIA (496.76+)

***Figure 1.1**: Minimum hardware requirements for Windows*