



TECHNOLOGY IN ACTION™

Beginning 3D Game Development with Unity 4

All-in-One, Multi-Platform Game Development



Sue Blackman

SECOND EDITION

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.



Contents at a Glance

About the Author	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
Introduction	xxv
■ Chapter 1: Introduction to Game Development.....	1
■ Chapter 2: Unity UI Basics—Getting Started	23
■ Chapter 3: Scripting: Getting Your Feet Wet	63
■ Chapter 4: Terrain Generation: Creating a Test Environment.....	85
■ Chapter 5: Navigation and Functionality	139
■ Chapter 6: Cursor Control	189
■ Chapter 7: Imported Assets	223
■ Chapter 8: Action Objects	263
■ Chapter 9: Managing State	299
■ Chapter 10: Exploring Transitions	345
■ Chapter 11: Physics and Special Effects	391
■ Chapter 12: Message Text	451
■ Chapter 13: Inventory Logic	483
■ Chapter 14: Managing the Inventory	511
■ Chapter 15: Dialogue Trees.....	551

■ **Chapter 16: Mecanim and Characters581**

■ **Chapter 17: Game Environment.....637**

■ **Chapter 18: Setting Up the Game679**

■ **Chapter 19: Menus and Levels725**

Index.....767

Introduction

Why Write This Book

Real-time 3D games have been around for well over ten years now. We've all played them, created assets in the style of our favorites, and maybe even "mod"ed a few of them. But with the Unity game engine leaping ahead in its effort to provide a free or low-cost means of authoring for desktop, mobile, or console games, the only barrier left to creating your own games is your level of commitment and the number of hours you are willing or able to devote.

Times have changed. 3D has become affordable not only in the movie industry, as seen by the number of titles featuring CG (computer graphics), but also in the game industry, where we've seen a shift in casual games from 2D to a 3D format. With Unity's bold move to offer a robustly featured free version of its engine, a radical change in the pricing models of the high-end engines has rocked the industry. The cost of the engine is no longer a barrier to taking your game from a nebulous idea to a working prototype and even on to a marketable product.

Whether your interest is in casual games or you have more ambitious aims, if you have no previous scripting experience but are eager to bring your art assets and story to life, this may be just the book to get you under way. In today's modern game engines, the heavy programming is handled by the engine itself, so the logic and game play can be scripted by those with more creativity than traditional programming knowledge.

In this book, I will approach game creation and design decisions from a 3D artist's view, taking the logic and scripting in small pieces, while introducing artists, budding game designers, and novice programmers to real-time game engine concepts at the same time.

This book is written in a project-based format, so you will not only end up with a playable game and scripting resources you can reuse for other games, but you will experience typical design decisions that have to be addressed throughout the process. You will create your game by starting with the basics and refining it as you add functionality; I will explain the logic behind certain choices and real-time game concepts as you go along.

The project for this book is based on a first-person point-and-click adventure game, complete with inventory, state management, load/save functionality, and a strong emphasis on the visual aspects of game creation. Even if you are more of a first-person-shooter-type game enthusiast, you will be able to see how including entertaining or interesting tasks and features can enhance your favorite genre.

The aim of this project is to introduce you to a logical progression of design decisions and problem solving that will be of value well beyond the scope of the adventure game genre. It provides a framework and a methodology for creating and, more important, *finishing* your own game. You will be going beyond the basic Unity functionality and use of standard assets to investigate topics rarely covered in online tutorials or books. All of the necessary art assets to complete the project are provided.

Author's Note

Several years ago, after taking my class through a race game, a first-person shooter, and a platform jumper, I decided that the last mini-project would be a classic adventure game. Much to my surprise, the class got quite motivated with all of the design decisions and logic involved. As there were no existing tutorials on this genre at the time, we created the game from the ground up, borrowing scripts from the previous projects, creating several of our own, and drawing heavily on the knowledge base of the Unity community. This book grew out of that experience, both to fill a need and to share the adventure with others.

Contemporary Adventure Games

For this project, you will be creating a variation on the classic point-and-click adventure game. The adventure genre, of which there are many variations, is an ideal starting place for artists and others with little or no scripting experience. The story, art assets, and animation are major components of this type of game, and the slower pace allows more freedom from script optimization, as split-second player reaction time is not required. If your only goal is to create a first-person-shooter-type game, this book may not be for you. Conversely, if your tastes are more eclectic, you may find all sorts of information that can be applied to your current genre of choice.

With Unity 4.0, the addition of Mecanim provides an ideal opportunity to introduce another common feature of the adventure genre—characters and dialogue trees. That you can do so without being an expert on character animation makes it even more fun. The more substantial incorporation of physics and various special effects this time around has also contributed to pushing this edition of the book's project from “classic” to “contemporary.”

One of the most enjoyable components of the adventure game is the collection and use of an odd assortment of objects. Because of the importance of inventory and state management, several chapters are dedicated to their design and implementation. Neophyte Unity developers often ask how to implement these features in community forums, but they rarely receive answers, owing to the scope of the topic. By the end of this book, you will be armed with the scripts, concepts, and experience to be able to take the knowledge beyond this genre.

Interactive adventure games are also ideal for indie developers, and they appeal to a broad range of players. FireProof Games' *The Room*, authored in Unity, was one of the top-rated mobile games of the year, proving that Unity has taken its commitment to the mobile and casual gaming community very seriously.

About the Unity Game Engine

Unity is the perfect choice for small studios, indie developers, and those of us who have always wanted to make our own games. Its large user base (more than 1.2 million in the summer of 2013) and extremely active user community allows everyone from newbies to seasoned veterans to get answers and share information quickly.

Unity provides an excellent entry point into game development, balancing features and functionality with price point. The free version of Unity allows people to experiment, learn, develop, and sell games before committing any of their hard-earned cash. Unity's very affordable, feature-packed Pro version is royaltyfree, allowing people to make and sell games with the very low overhead essential to the casual games market.

The market for multi-platform games—especially casual games for iPhone and Android—is extremely popular at the moment, and Unity's commitment to cross-platform delivery is well proven. Originally a Mac-based authoring application that could publish to Mac and Windows, Unity unveiled its Windows version in spring 2009. As expected, it has opened opportunities for PC-based developers and artists. Since that time, Unity has continued to add support for iPhone, Android, iPad, and Wii and is developing support for Xbox 360 and PS3. In spring 2013, Unity Technologies revealed its latest surprise: for Unity free users, the license for iOS, Android, Windows 8 mobile, and Blackberry is now, or will shortly be, free as well.

Early adapters of the Unity engine tended to migrate from Flash and Director, making the scripting environment easily adoptable. While many Unity users have an ActionScript background in making Flash games, it is by no means a prerequisite. There is a lot of material for creating games in Unity, including first-person shooters, racing games, platform jumpers, and the like. Even if your interest lies elsewhere, there are plenty of helpful tips and tricks to be gleaned from even the most unlikely sources. Don't be afraid to take advantage of the resources available on the Unity web site (www.Unity3D.com), the Unity Forum (forum.unity3d.com), Wiki (www.unifycommunity.com/wiki), UnityAnswers (answers.unity3d.com), and numerous private-party web sites devoted to the Unity engine.

Unity documentation also contains a wealth of valuable information, but, as with any technology that has a unique vocabulary, it's sometimes hard to find just what you're looking for. Prefacing your query with Unity or Unity3D and searching the Internet is often the easiest way to find elusive concepts or functionality. You can make use of them via the Help menu, but it is often quicker to take advantage of the online version.

Will I Have to Learn to Script?

You don't have to be a programmer to make your own game with Unity, but you *will* have to be able to understand enough of what the scripts do to know what can be tweaked to your advantage or decide if a particular script will suit your needs.

Most game play has to be scripted in Unity, but there are hundreds of scripts already available that can be readily reused. Unity ships with several of the most useful. More can be found by searching the Unity Forum, Wiki, or UnityAnswers. Many forum members will even write bits of script for less adept users. In the Collaboration section of the forum, you can even find scripters looking to trade for art assets. By the end of this book, you should know enough to be able to take advantage of the wealth of material available from the Unity community.

Games, by their very definition, are about interaction; even in games that are largely controlled by physics, logic-driven cause and effect is what differentiates them from linear plot-driven passive media. Even the most “artist friendly” game engines require scripting to move beyond simple environmental walk-throughs. This book's aim is to familiarize you with scripting a few lines at a time, while providing visual feedback as often as possible. The assumption is that you have not yet decided to *learn* to script but are happy enough to participate in its creation in a more passive manner.

Scripting Is More About Logic Than Syntax

While many people worry that learning a new language will be difficult and intimidating, think of it this way: most people under the age of 35 are quite fluent in texting, which is a valid subset of our native language. To a certain extent, it has its own vocabulary and syntax, and it is very similar to scripting languages, in that it is more of a subset than an entirely new language.

The difference mainly lies in the method used to acquire the “language.” In texting, as with our native language, one doesn't set out to study and *learn* the language. Instead, one absorbs, experiments, and eventually becomes competent through repeated contact, trial and error, and a host of other passive means, rather than rote memorization and stressful examination. This book, because it's about the *logic* behind game design and creation, treats scripting as an immersive experience that is a sideline to the game-development process. You are free to choose your own amount of involvement with the scripting. Whatever you choose, the main scripts developed in the book will enable you to create or expand upon the final output of the book, the game, with your own ideas and art assets.

That being said, there are only a few important concepts and a handful of keywords you need to know in order to get an idea of what's being done in a script. Fortunately, most scripters are quite good at including comments explaining what the code is doing, thus making complicated scripts much less daunting to investigate.

Scripts developed in this book will be provided on a per-chapter basis, and the logic behind them explained in the chapters, but I hope that you will display classic adventurer curiosity and take advantage of the scripting explanations to do some experimenting on your own.

What About Math?

One of the most common things we hear people say in the 3D industry is, “If I'd known math was going to be so useful, I would have paid more attention in class.” Be that as it may, most artists and designers are not going to want to invest any time in brushing up on their math skills. Don't worry! My primary goal is to help you create a game. Some of the functionality you will scavenge for your game is easy to use without even knowing how it works. Most of us are quite happy to drive cars without having extensive knowledge of the internal combustion engine. Don't be afraid to treat scripting the same way!

Assumptions and Prerequisites

This book assumes that you are at least somewhat familiar with 3D assets and 3D space, but it does have a short review of the concepts and plenty of tips and tricks throughout.

It assumes that you will not have had much scripting experience (if any at all) but that you'll be willing to work through it in order to bring your stories to life.

It assumes that, as of yet, you have little or no experience with using the Unity game engine.

It also assumes that you are fairly new to the game-making process but have a strong desire to create your own real-time 3D game.

Additionally, this book assumes that if you want to explore genres other than classic point-and-click adventure games, you will work through the book with a goal of thinking about how to apply the various techniques to get results related to the other genre. In the casual game market, combining elements of adventure games, first-person shooters, and other genres is not only acceptable but makes for some very entertaining results.

What This Book Doesn't Cover

This book is not about conventional game design; it is more of a precursor, getting you into the habit of analyzing needs and weighing choices. Not only is creating a massive design document intimidating when you are the one who will have to implement everything, but it is likely to be unrealistic until you are more familiar with the engine and your own capabilities. You're going to build your game up a little bit at a time, prototyping ideas and functionality as you go along.

This is not a book on how to become a programmer, still less on programming best practices. The scripting in this book is designed to ease a non-programmer into the process by providing instant visual feedback as often as possible. While there are usually several ways to attain the same goal, the scripting choices made in this book are the easiest to read and understand from an artist's or designer's point of view. In this book, scripting is presented in the way a native speaker learns his or her own language. He or she is surrounded by it, immersed in it, and allowed to tinker with it to slowly gain a basic working knowledge of it. Don't worry about remembering it all. Some things you will use throughout the project, and others you will be encouraged to take note of for future reference.

Conventions Used in This Book

1. Instructions look like this.

■ **Tip** Follow this format.

Code looks like this.

Platform

This book was written using Unity 4.x in a Windows 7 and a Windows 8 environment. Differences for shortcut keys and operating system file storage with Unity on a Mac will be noted throughout the book and are also available through the help files.



Introduction to Game Development

In the first edition of this book, the classic point-and-click adventure genre provided not only a forgiving environment in which to create a game, but it also allowed readers to explore tips, tricks, and techniques often ignored in other beginning Unity game books. This edition updates the classic adventure game to its modernized counterpart, the contemporary adventure game. A 3D world, physics, real-time special effects, and 3D characters enhance the player's experience as he goes through the world acquiring objects of questionable use and solving puzzles, both physical and logic-based.

Coupled with the fact that it appeals to a wide range of players, including many who have never played a first-person-shooter type game, it also becomes an ideal vehicle for the casual games market. And in case your idea of adventure still includes gratuitous use of weaponry, you will even get an introduction to that as well, as the book progresses. As with any project, how to begin is with a bit of research. Critical thinking and investigation at the start will save time in the long run.

The Adventure Genre

If you are old enough to have been around during much of the adventure game's history, you've probably gotten ideas for your own game, based on the most successful and entertaining features of the classics. If you are younger, you may have discovered them through various web sites dedicated to their preservation. Either way, there are lots of good ideas to be found, especially as changes in technology open new avenues for game content to take advantage of.

Text Adventure Games

"You are standing at the end of a road..."

The granddaddy of all adventure games is arguably *Adventure*, the text-based game originally designed by Will Crowther in the mid-1970s. In an era where computer games were dominated by *Pong* and *PacMan*, the text-based game that catered to those with a dexterous mind rather than dexterous fingers was a revelation. In the earliest version, also known as *Colossal Cave*, Crowther set the scene for intrepid adventurers to explore a great underground cave system, collecting loot and dealing with the occasional monster. It was reportedly fashioned after the Mammoth Cave system in Kentucky, where Crowther had developed a vector-based map in conjunction with his own explorations and existing surveys.

Adventure set the stage for the genre, where the prose was always beautifully descriptive and often highly entertaining, due in part to the vocabulary and parsing of the users' input.

Infocom's *Zork* series, introduced us to the Great Underground Empire, the Flathead dynasty, and the coin of the realm, the zorkmid. They spawned several memorable lines, such as "Your lamp is getting dim," "You can't get there from here," and many running jokes. Anyone who knows what a grue is can tell you that when your lamp goes out, you are in danger of being eaten by one.

Graphical Adventure

With the advent of computer graphics, the text-based adventure-game genre waned, as graphic quality and resolution slowly improved. Eventually, the text-based predecessor gave way to the still-image format, or graphical adventure genre, pioneered by Sierra Online's *King's Quest*, a host of LucasArts offerings, and Infocom's later *Zork* titles.

The graphical format spelled the end of players typing in their instructions or questions, relying instead on a short predefined list of verbs activated by mouse picks. Gone was a large part of the charm of the early text adventures, where one could type in just about anything to see if the authors had allowed for it, no matter how ridiculous or risqué.

As far as creating material for the new genre, it now required more than just a writer and programmer. It introduced the artist as a major part of the production pipeline. As resolution increased, so did the art assets required. Unlike levels in today's first-person shooters, where the player faces enemies that are increasingly more difficult to overcome in each successive level, the worlds in adventure games continue to be strongly differentiated by theme, visual style, color scheme, and music. The reward for gaining access to the various worlds is the discovery of new and intriguing environments, designed to stimulate the senses and draw the player into the fantasy.

In the early 1990s, Rand and Robyn Miller's *Myst* twisted the usual format to introduce the concept of using game play to reveal the story itself. Acquisition and inventory was practically nonexistent, but interaction and task- or puzzle-solving was visually breathtaking for the times (Figure 1-1). Among the first to incorporate 3D graphics instead of traditional artwork, they introduced another shift in the genre. Now, not only did one require artists to produce a game, a good number of them had to be able to create artwork in the fledgling 3D programs.



Figure 1-1. *Myst*, one of the first adventure games to make use of 3D graphics. (*Myst* [TM] is the sole property of Cyan Worlds, Inc. Copyright 1993, 2001. Cyan Worlds, Inc. All rights reserved. Used with permission.)

LucasArts Titles

Undoubtedly a force in the early days of the graphical adventure game genre was LucasArts. With an army of professional film personnel from which to draw, LucasArts titles gained a huge reputation for outstanding storytelling, top-notch graphics, and marvelous soundtracks. Of note is the *Monkey Island* series. As with several other titles, such as *Sam and Max* and *Day of the Tentacle*, the entertainment is heavily driven by humor, references to pop culture, and a hearty sense of the ridiculous.

Monkey Island III, *The Curse of Monkey Island* was one of the last LucasArts titles to use traditional hand-painted backdrops and cell animation. Apart from low resolution by today's standards, its style and execution continue to stand the test of time.