



C o m m u n i t y E x p e r i e n c e D i s t i l l e d

Building an FPS Game with Unity

Create a high-quality first person shooter game using the Unity game engine and the popular UFPS and Probuilder frameworks

Foreword by Calle Lundgren, Creator of UFPS

John P. Doran

[PACKT]
PUBLISHING

Building an FPS Game with Unity

Create a high-quality first person shooter game using the Unity game engine and the popular UFPS and Probuilder frameworks

John P. Doran



BIRMINGHAM - MUMBAI

Building an FPS Game with Unity

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: October 2015

Production reference: 1271015

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78217-480-6

www.packtpub.com

Cover image by John P. Doran

Credits

Author

John P. Doran

Copy Editor

Akshata Lobo

Reviewers

Schuyler L. Acosta

Alex Madsen

Tony Pai

Chittersu Raghu Vamsi

Nevin Vu

Project Coordinator

Sanchita Mandal

Proofreader

Safis Editiing

Indexer

Hemangini Bari

Commissioning Editor

Edward Bowkett

Graphics

Disha Haria

Acquisition Editor

Vivek Anantharaman

Production Coordinator

Shantanu N. Zagade

Content Development Editor

Riddhi Tuljapurkar

Cover Work

Shantanu N. Zagade

Technical Editor

Shivani Kiran Mistry

Foreword

My first love affair with an FPS game was in 1995. I was an intern at a local radio station and someone had installed the shareware version of *DOOM* on the CD database computer. A fast, sprawling ballet of violence unfolded before my eyes. This was what a computer game was supposed to be like! Running around 3D dungeons, guns blazing, blood splattering, and demons growling and scaring the bejeezus out of me before they were being blown to bits. We didn't get a whole lot of work done that summer. And my fate was sealed; I was going to be a game developer.

It took me a good 4 years of modding, scripting, and 3D modeling to land a job at a small startup game studio. Thrilled, I found myself working on a real multiplayer FPS game as a part of a team of 15 people. Coming from a hobbyist, do-it-all-by-yourself mindset, I remember my jaw hitting the floor as the project manager told me some numbers over lunch. He estimated that for one single person to create the whole game, it would take 65 years. 65 years!

In the following months, Moore's law and a relentless push for realism saw budgets and team sizes skyrocket. Soon it wasn't uncommon for an FPS project to have a head count in the hundreds. The would-be-indie developer inside me mourned these figures as I pondered my secret indie ambitions and sensed those already impossible 65 years stretching into 650.

Of course, back then, everybody was building their own game engine from scratch. Game programming books would explain in great detail how to construct your code from the bare metal up, going into hardware specifics, the basics of rasterizing polygons, brutal 3D math, and communicating with different brands of audio cards. You could license a game engine, but it would set you back hundreds of thousands of dollars. Besides, game coders loved to do everything from scratch back then (and as a result rarely got around to finishing their games).

Then, something happened. During the early 2000s, affordable and free engines such as Torque, Auran Jet, Crystal Space, and Ogre started popping up. Around the same time, the idea of "gap games" revitalized the indie movement. They were of real high quality, but were limited in scope; not your multimillion dollar production, but no scrawny "bedroom programmer" games either. They were fantastic looking games that could realistically be created by a small team with a good off-the-shelf engine in a reasonable amount of time. The dream was revived.

The Unity engine was first built for the Mac game *GooBall*. As the story goes, the team realized that, in the end, their game didn't show as much promise as their game engine, and Unity3d was announced at the 2005 Apple Worldwide Developers Conference. Initially, what Unity had going for it was the ability to run high-definition 3D games on a web browser. When the iOS and Android support was added, it became the engine of choice for mobile game development, and everything just exploded. Today, Unity3d is a free, extremely popular, powerful, and multiplatform AAA game engine. It has triggered an incredible surge in indie game development and spawned untold indie game successes. The addition of the Unity Asset Store allows thousands of pros and hobbyists to share and trade high-quality scripting, art, sound, design, and services.

UFPS started out as my side project dubbed by Ultimate FPS Camera. I released it as a small script pack in the Asset Store just to see what would happen. The response was overwhelming. Three years later, the system has grown into a full blown FPS solution. My team has assisted many hundreds of indies in pursuing their game ideas. We've seen many awesome and original games take shape; some released to critical acclaim. I've also had the privilege of working with the authors of several amazing Unity assets, including Gabriel and Karl, the developers of ProBuilder, two incredibly dedicated and talented guys who have put innumerable hours of hard work into their tool suite (so you won't have to). It's with a sense of joy and excitement that I learned of this book being written and featuring ProBuilder along with UFPS.

In this book, John has summarized not only how to take advantage of the awesome power of Unity and Asset Store. In a casual and direct way, he explains how to arrive at a small, complete FPS in the shortest amount of steps possible. He doesn't go into the nitty gritty details of programming camera systems, level editors, or a combat AI from scratch. Instead, he helps you free up time for the core activities that make your game fun with creative game- and level-design. If you're prototyping a game or just starting out as a game developer, the power available to you through this book, Unity, and its Asset Store would have been unthinkable just a few years ago.

Good luck with your dream game!

Calle Lundgren
Creator of UFPS

About the Author

John P. Doran is a technical game designer, who has been creating games for over 10 years. He has worked on an assortment of games in teams from just himself to over 70 students, mod, and professional projects.

He previously worked at LucasArts on *Star Wars 1313* as a game design intern. He later graduated from DigiPen Institute of Technology in Redmond, WA, with a bachelor of science in game design.

John is currently a designer in DigiPen's research and development branch in Singapore. He is also the lead instructor of the DigiPen-Ubisoft Campus Game Programming Program, instructing graduate-level students in an intensive, advanced-level game programming curriculum. In addition to this, he also tutors and assists students on various subjects while giving lectures on C#, C++, Unreal, Unity, game design, and more.

In addition to this title, he is the author of *Unreal Engine Game Development Cookbook*, *Unity Game Development Blueprints*, *Getting Started with UDK*, *UDK Game Development*, and *Mastering UDK Game Development*, and also the co-author of *UDK iOS Game Development Beginner's Guide*, all by Packt Publishing. More information on him can be found on his website (<http://johnpdoran.com/>).

Acknowledgment

Firstly, I would like to thank my family for being patient with me while I took yet another challenge that reduced the amount of time I could spend with them. Especially Hien, my wife, who has taken a big part of that sacrifice, and also Chris, my brother, who encourages me in his particular way.

This book also couldn't have been written at all without the amazing support from the game development community. Most of all, I'd like to thank Calle Lundgren at VisionPunk for being so supportive of the project and providing his very valuable insights for it. I'd also want to thank Gabriel Williams from ProCore3D and Nick Canafax at Rival Theory for their assistance.

In addition, the reason the projects in the book look so good is due to the artistic talents of the guys at *GameTextures.com*, who provided some amazing textures to work with as well as Paul Blackham for letting me use his awesome gun model.

On this same note, I also want to thank Samir Abou Samra and Elie Hosry for their support and encouragement while working on this book, as well as the rest of the DigiPen Singapore staff.

Thanks again to the wonderful guys at Packt, who were a pleasure to work with, including Vivek Anantharaman, who approached me about the project in the first place, and Riddhi Tuljapurkar, who worked with me as the book was being written.

Last, but not least, I want to thank my parents Sandra and Joseph Doran, who took me seriously when I told them that I wanted to make games for a living.

About the Reviewers

Schuyler L. Acosta is a graduate from the Art Institute of California, San Francisco, with a bachelor of science in game art and design. In his free time, he enjoys working on 2D/3D portfolio projects. This is his first book review. His other passion besides art is music, playing the piano and electric/acoustic guitar.

I would like to thank Pooja Mhapsekar, Sanchita Mandal, and Riddhi Tuljipurkar for their feedback and support in reviewing this book. I'd also like to thank my parents, Bonnie and Edwin Acosta, and my sister, Angel Acosta, for their continued support and encouragement.

Alex Madsen is a Gameplay Programmer at Pure Arts Ltd., Shanghai. He is a tech enthusiast that loves finagling with anything that is based on computers: Arduino, Linux, and the like. He started his career programming Excel spreadsheets for a rural Alberta tree farm, and got an opportunity to work in Shanghai.

He works at Pure Arts Ltd., Shanghai.

I would love to thank my parents and beloved friends, and Stark, the big dumb dog.

Tony Pai is an indie game developer from Taiwan. He learned Unity for about a year and a half and is now working with his friends to make games. They released two games to date, *The Guys* and *Elpis*. This is his first book review.

I want to thank Sanchita Madal, project coordinator, and everyone at Packt for their help in producing this book.

Chittersu Raghu Vamsi is a professional programmer, game developer, analyst, and designer. He has a computer science background, which he pursued at one of the most reputed colleges in India, BITS Pilani. He has an experience of over 3 years in the field of game development. He has done projects on machine learning and artificial intelligence. His other interests include reading novels, making short films, and writing articles.

I would like to thank my family and friends for their encouragement and support. I also thank Packt Publishing for providing me such a great opportunity.

Nevin Vu graduated with a diploma in game design. While he was out to complete his diploma, he gained valuable experience by interning at Panasonic Avionics, Singapore. He is currently pursuing a degree in computer science to ensue his passion in programming. Upon graduation, he hopes to work in the video game industry to contribute to the gaming development community with his knowledge in programming. He has developed various games from platforms to FPS games, which are available on his website, <http://www.n3evin.com>.

I would like to thank my parents for giving me this opportunity and being supportive of my interest in programming. Moreover, I am thankful to the rest of my family and friends for encouraging me to pursue my passion.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	v
Chapter 1: Getting Started on an FPS	1
Prerequisites	2
Project creation	4
Getting started with the Asset Store	6
Installing UFPS	9
Installing Prototype	12
File organization	15
Customizing Unity's layout	16
Summary	17
Chapter 2: Building Custom Weapons	19
Prerequisites	20
Setting up a testbed	20
Getting models/sounds for weapons	22
Building our weapon – the mesh	24
Creating a UnitBank	29
Creating the weapon	30
Customizing our weapon's properties	34
Summary	39
Chapter 3: Prototyping Levels with Prototype	41
Prerequisites	42
Level design 101 – planning	42
Creating the architectural overview	44
3D modeling software	44
Constructing geometry with brushes	44
Modular tilesets	45
Creating geometry	46
Building a doorway	60

Duplicating rooms / creating a hallway	68
Preventing falls - collision	73
Adding stairways	76
Coloring your world	84
Summary	86
Chapter 4: Creating Exterior Environments	87
Prerequisites	87
Introduction to Terrain	88
Height maps	88
Hand sculpting	90
Creating the Terrain	91
Adding color to our Terrain – textures	97
Adding water	106
Adding trees	108
Adding details – grass	112
Building the atmosphere – Skyboxes and Fog	114
Summary	120
Chapter 5: Building Encounters	121
Prerequisites	121
Adding a simple turret enemy	122
Integrating an AI system – RAIN	129
Integrating an AI system – Shooter AI	153
Spawning enemies with the help of a trigger	164
Spawning multiple enemies at once	179
Cleaning up dead AI	184
Placing healthpacks/ammo	186
Summary	188
Chapter 6: Breathing Life into Levels	189
Prerequisites	189
Building an explosive barrel	190
Using triggers for doors	198
Creating an elevator	208
Summary	214
Chapter 7: Adding Polish with ProBuilder	215
Prerequisites	215
Upgrading from Prototype to ProBuilder	216
Creating material	223
Working with ProBuilder – placing materials	227
Meshing your levels	249
Summary	253

Chapter 8: Creating a Custom GUI	255
Prerequisites	255
Creating a main menu: part 1 – adding text	256
Creating a main menu: part 2 – adding buttons	261
Creating a main menu: part 3 – button functionality	265
Replacing the default UFPS HUD	270
Summary	278
Chapter 9: Finalizing Our Project	279
Prerequisites	279
Building the game in Unity	279
Building an installer for Windows	283
Building an installer for Windows	287
Summary	296
Index	297

Preface

Unity, available in free and pro versions, is one of the most popular third-party game engines available. It is a cross-platform game engine, making it easy to write your game once and then port it to PCs, consoles, and even the Web, making it a great choice for both indie and AAA developers.

Building an FPS Game with Unity takes readers on an exploration of how to use Unity to create a 3D first-person shooter (FPS) title. Over the course of the book, you will learn how to work with Unity's own tools while also leveraging the powerful UFPS framework by VisionPunk and Prototype/ProBuilder 2.0 by ProCore3D. In addition, readers will learn how to create AI characters using both RAIN and Shooter AI.

After setting up the computer, you will start by learning how to create custom weapons, prototype levels, create exterior and interior environments, and breathe life into your levels. You will then polish the levels. Finally, you will create a custom GUI and menus for your title to create a complete package.

What this book covers

Chapter 1, Getting Started on an FPS, will give readers a brief overview, from a beginner's point of view, of the exciting world of Unity development for the creation of a first-person shooter (FPS) title. We will start off by creating a project and then set up our project by installing UFPS and Prototype. We'll also see how we can customize Unity's layout and organize our files effectively.

Chapter 2, Building Custom Weapons, will talk about one of the most important things in the FPS game: the weapons. In this chapter, we will create a testbed to work on and learn how to build a weapon in UFPS from a model while also learning about UFPS features such as UnitBanks and various weapon properties to give your weapons a feel "just right" for you.

Chapter 3, Prototyping Levels with Prototype, will help us take on the role of a level designer, who has been tasked to create a level prototype to prove that our gameplay is solid. We will make use of the free tool Prototype to help in this endeavor. In addition, you will also learn some beginning level designing.

Chapter 4, Creating Exterior Environments, explores the various ways to add more organic-feeling areas to your levels, making use of Unity's different terrain tools while also adding water, trees, grass, custom skyboxes, and fog to create a complete environment that can be used in your project.

Chapter 5, Building Encounters, will show how to create various types of encounters that players may experience to create effective gameplay scenarios starting with a simple turret, then creating a melee enemy using the free RAIN AI Engine as well as a ranged enemy using Shooter AI. You will then learn how to place enemies as well as spawn them into our scene. Lastly, you will learn how to place ammo and healthpacks in the level to guide players through the level.

Chapter 6, Breathing Life into Levels, will explore some of the ways we can breathe life into our levels with moving objects and more things that the player can interact with, such as exploding barrels, moving doors, and elevators.

Chapter 7, Adding Polish with ProBuilder, will take all of the pieces we created in the previous chapters and put them together to create a finished level. You will learn how to upgrade Prototype levels to using ProBuilder, gaining the ability to add materials to wall faces and create custom UVs. You will also learn how to create materials and how to mesh your levels to create a complete environment.

Chapter 8, Creating a Custom GUI, will demonstrate how to use Unity's new GUI system to create a custom GUI to replace the one given to us by UFPS to help our project stand out.

Chapter 9, Finalizing Our Project, will focus on exporting our game from Unity and then creating an Installer so that we can give it to all of our friends, family, and prospective customers.

What you need for this book

Throughout this book, we will be working within the Unity 3D game engine that you can download from <http://unity3d.com/unity/download/>. The projects were created using version 5.0.1, but the project should work with minimal changes.

For simplicity's sake, we will assume that you are working on a Windows-powered computer. Though Unity allows you to code in either C# or UnityScript, for this book, we will be using C#.

We will also be using assets that are available from the Asset Store in Unity, most notably UFPS which costs money (\$95 normally, but it also has sales where it can be gotten for cheaper). Aside from UFPS, all of the other topics will cover how to use a free asset to accomplish things while also discussing an easier to use asset which does cost. Readers will use ProCore3D's free Prototype tool for the creation of levels while also learning how you may upgrade your project to using ProBuilder in *Chapter 9, Finalizing Our Project*, for advanced functionality, but it also costs money. For enemies, you will also learn how to use the free RAIN toolkit for melee enemies and how to use Shooter AI for ranged enemies, which costs money.

Who this book is for

This book is for those who want to create an FPS game in Unity and gain knowledge on how to customize it to be their very own. If you are familiar with the basics of Unity, you will have an easier time, but it should make it possible for someone with no prior experience to learn Unity at an accelerated pace.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "We can include other contexts through the use of the `include` directive."

A block of code is set as follows:

```
void OnTriggerEnter(Collider other)
{
    //If the player touches the trigger, and if it hasn't
    //been triggered before
    if(other.tag == "Player" && hasTriggered == false)
    {
        // Spawn a new enemy using the properties from the
        // spawnPoint object
        GameObject newEnemy = Instantiate(enemy,
                                           spawnPoint.position,
                                           spawnPoint.rotation)
                                   as GameObject;
        // We only want this to happen once.
        hasTriggered = true;
    }
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:


```
void OnTriggerEnter(Collider other)
{
    //If the player touches the trigger, and if it hasn't
    //been triggered before
    if(other.tag == "Player" && hasTriggered == false)
    {
        // Spawn a new enemy using the properties from the
        // spawnPoint object
        GameObject newEnemy = Instantiate(enemy,
                                           spawnPoint.position,
                                           spawnPoint.rotation)
                                as GameObject;

        //Tell enemy to go to the player's position
        newEnemy.GetComponent<NavMeshAgent>().SetDestination(other.
        transform.position);

        // We only want this to happen once.
        hasTriggered = true;
    }
}
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Once completed, select **Create project**."

 Warnings or important notes appear in a box like this.

 Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail feedback@packtpub.com, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files from your account at <http://www.packtpub.com> for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from https://www.packtpub.com/sites/default/files/downloads/Building_FPS_Games_with_Unity.pdf.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at questions@packtpub.com, and we will do our best to address the problem.

1

Getting Started on an FPS

Welcome to *Building an FPS Game with Unity*! This chapter is dedicated to offer a brief overview, from a beginner's point of view, of the exciting world of Unity development for the creation of a **First Person Shooter (FPS)** title, leveraging the powerful **Ultimate First Person Shooter (UFPS)** framework by VisionPunk and Prototype/ProBuilder 2.0 by ProCore3D. But, before we get started, we first need to get all of the resources we'll need and set up our project for success.

Over the course of this book, we will be creating a 3D FPS game similar to the popular games in the market such as *Call of Duty: Black Ops III* and *Halo 5: Guardians*.

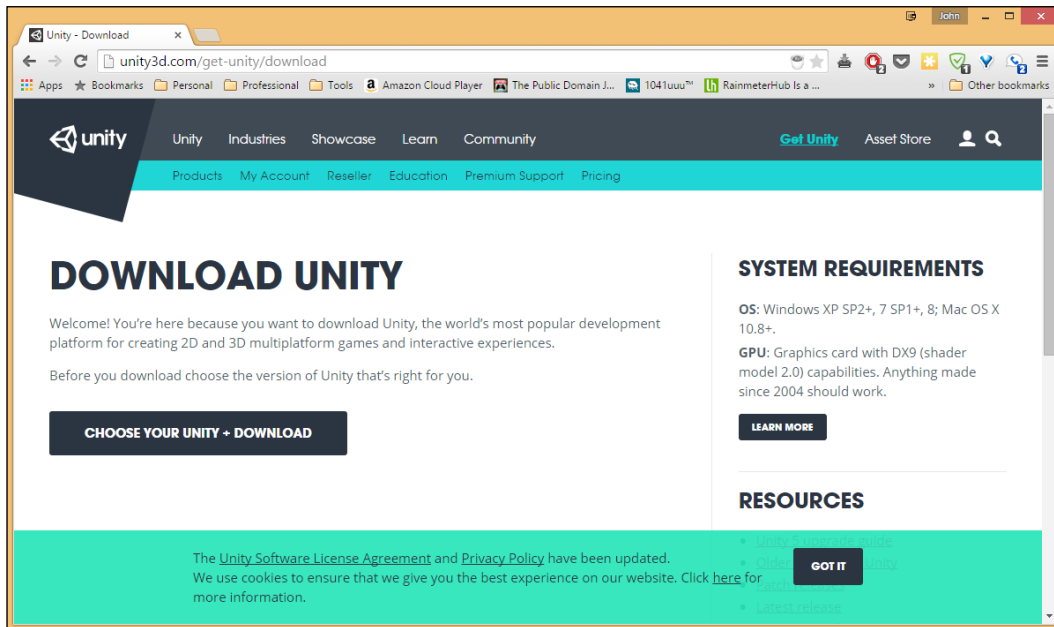
We will learn how we can create custom weapons of our own as well as how we can create interior and exterior environments. After creating our environments, we will populate them with different combat encounters for players to fight as well as include intractable objects such as exploding barrels. We'll then customize our user interface using Unity's new GUI system before we package our game and create an installer to get the game out into the world!

This project will be split into a number of tasks. It will be a simple step-by-step process from the beginning to the end. Here is the outline of our tasks:

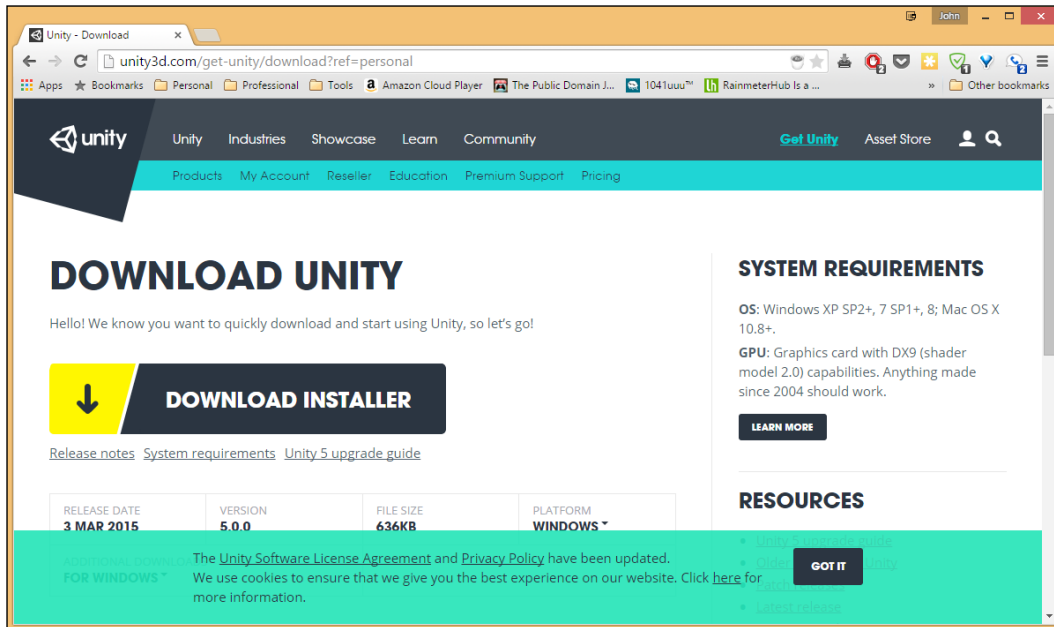
- Project creation
- Getting started with Unity's Asset Store
- Installing UFPS
- Installing Prototype
- File organization
- Customizing Unity's layout

Prerequisites

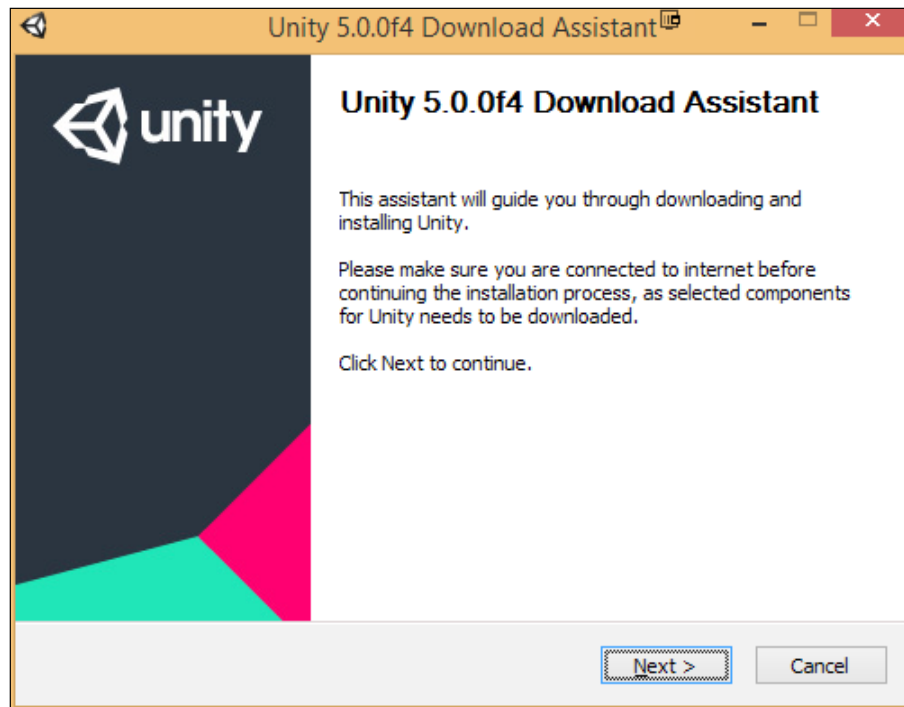
Before we start, we will need to get the latest Unity version. You can always download it from <http://unity3d.com/unity/download/>. At the time of writing this book, the page looks like this:



Once you get to this page, click on the **CHOOSE YOUR UNITY + DOWNLOAD** button (this page onward, I will be using the **PERSONAL EDITION** version). Then, click on the **DOWNLOAD INSTALLER** option.



The download assistant should be installed from your default downloads directory. Once it is installed, double-click on the file to open it.



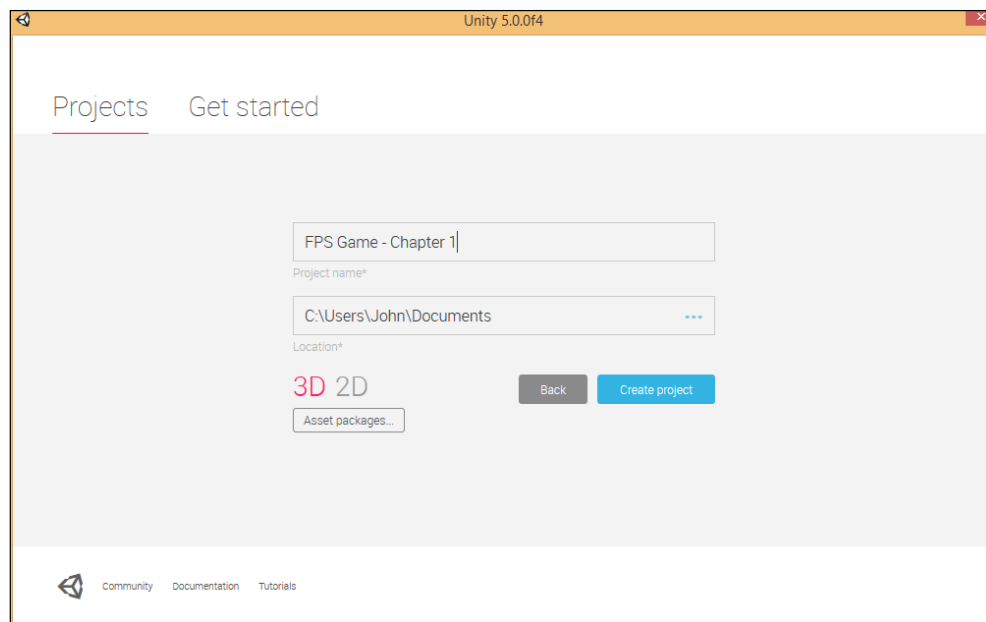
Once there, go through the installation using the default properties. At the time of writing, we are using **Unity 5.0.0f4**, but most things should work with minimal changes but be sure to check out the book's website to check for any errata.

Since it needs to download Unity, this process may take some time.

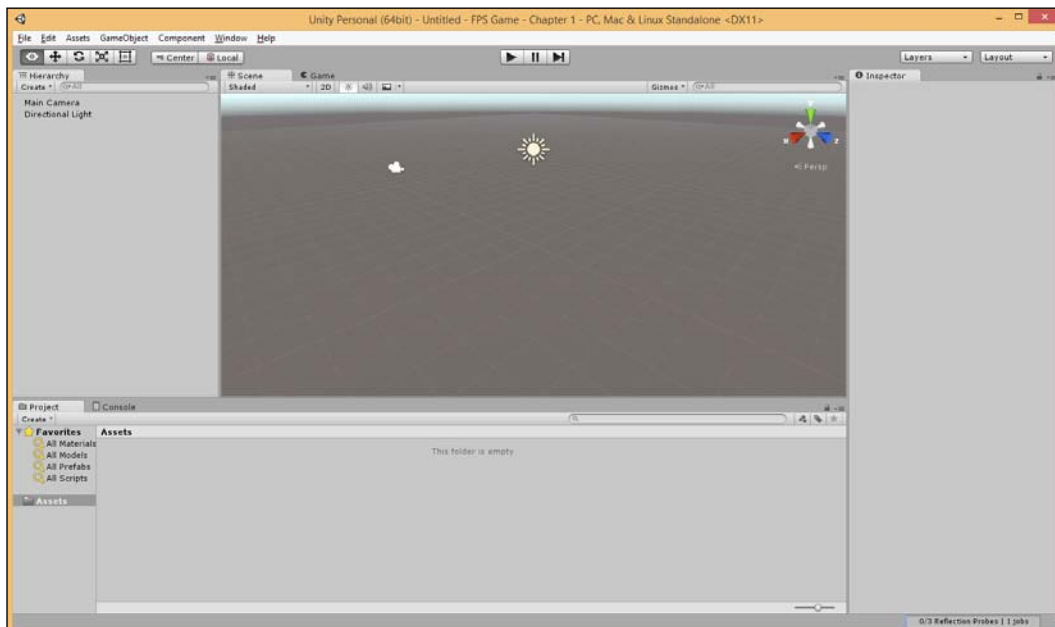
Project creation

At this point, I assume that you have freshly installed Unity and have started it up. Follow these steps to create a new project in Unity:

1. With Unity started, go to **New Project**. Enter a **Project Name** value such as `FPS Game - Chapter 1` as that's what we are making and the chapter we're making it for, or whatever you want to call your project. Select **Location** of your choice somewhere on your hard drive and ensure that you have your game set to 3D. Once completed, select **Create project**. At this point, we do not need to import any packages as we'll be doing it manually.



2. Here on, if you see the **Welcome to Unity** pop up, feel free to close it as we won't be using it. At this point, you will be brought to the general Unity layout, which should look as follows:





I'm assuming you have some familiarity with Unity before you read this book. If you want more information on the interface, please visit <http://docs.unity3d.com/Documentation/Manual/LearningtheInterface.html>.

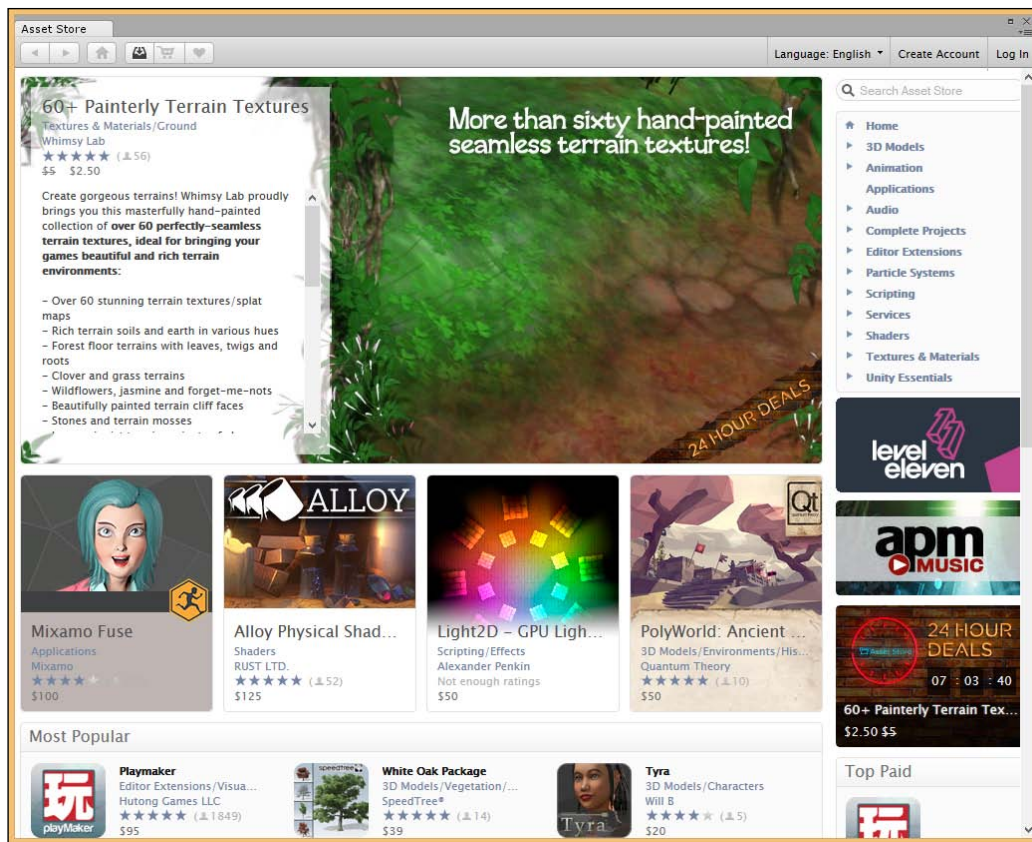
Getting started with the Asset Store

Since Unity 3, the Asset Store has been similar to Apple's/Android's app stores, except, instead of apps, you can buy prebuilt assets that can be imported directly into your project. We will be using this in our project; but, before we do so, we will need to have an account, that can be created using the following steps:

1. To open the **Asset Store** via Unity, we can go to **Window | Asset Store** from the top toolbar.

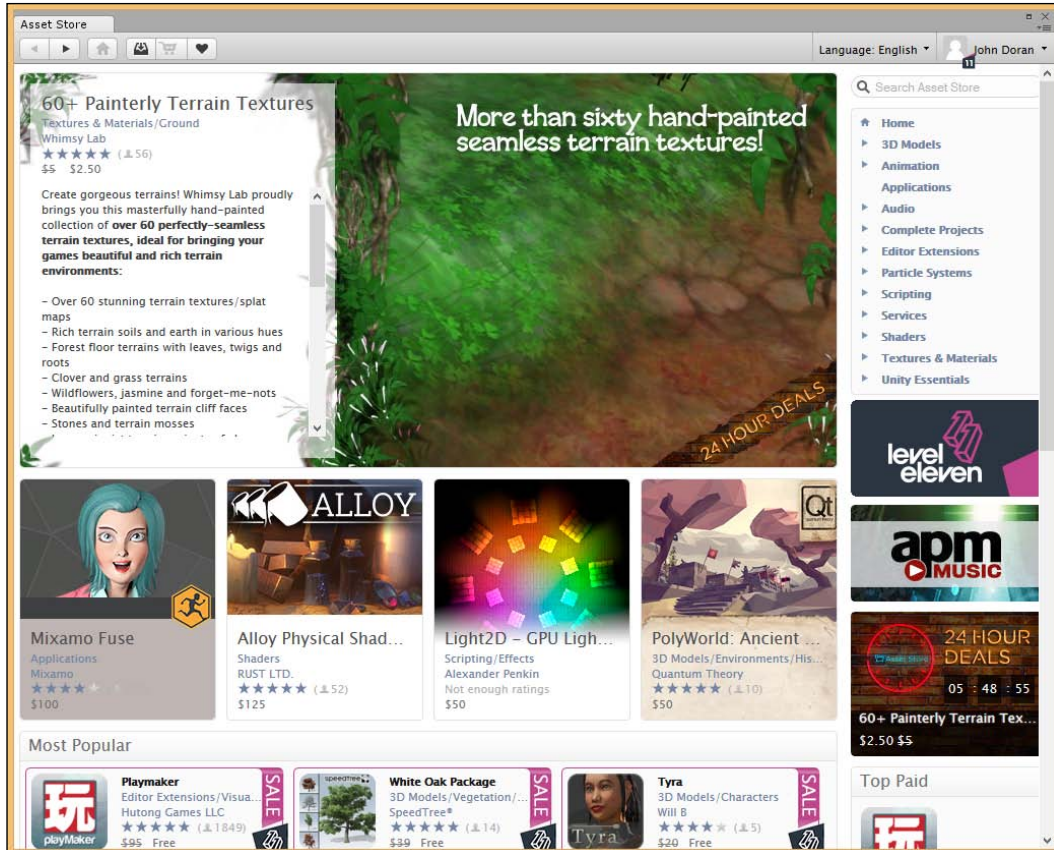


You may also open the panel by pressing *Ctrl + 9* (*command + 9* on Mac). If nothing happens, make sure Unity is focused on window by clicking on the program and trying again. If you prefer working outside Unity, you may also go to the **Asset Store** website at <https://www.assetstore.unity3d.com/>. However, you'll need to download assets from Unity via the **Download Manager**, which you can learn about from the link given later in this chapter.



2. Next, go to the top-right corner of the panel and click on the **Create Account** button. Fill out your information and click on **Create account** at the bottom of the screen.
3. You should receive an activation e-mail shortly after you submit the form. Open it and click on the activation link provided to verify your account.

4. Once this is done, go to **Log In** from the top-right corner of the panel and enter the information you put in when you created the account. If all goes well, the top-right corner of the window will change to reflect that you are logged in.



With this, we can download and purchase assets from the Asset Store!



Do not be concerned if you do not see **11** by your account, it just shows that I have a Unity Pro account.

For more information on the Asset Store and how to navigate using it, visit <http://docs.unity3d.com/Manual/AccessNavigation.html>.

Installing UFPS

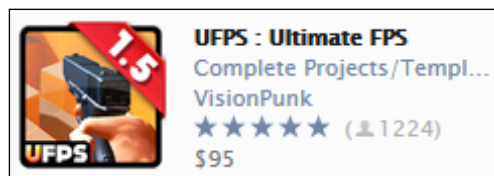
Creating a game completely from scratch takes a considerable amount of time. Creating a first-person shooter relies on having a lot of knowledge as well as problem solving with things such as physics, mathematics, graphics, and programming. Now, books of each of these subjects could have been written based on it; but we are assuming that you want results in the fastest amount of time possible.

Rather than taking the tens of thousands of hours it would take to create it from scratch, we will leverage the very popular Unity add-on UFPS (Ultimate FPS) from VisionPunk. It will give us a solid foundation to create our own project with hundreds of parameters, which we can customize to create a project exactly the way we want it to be.

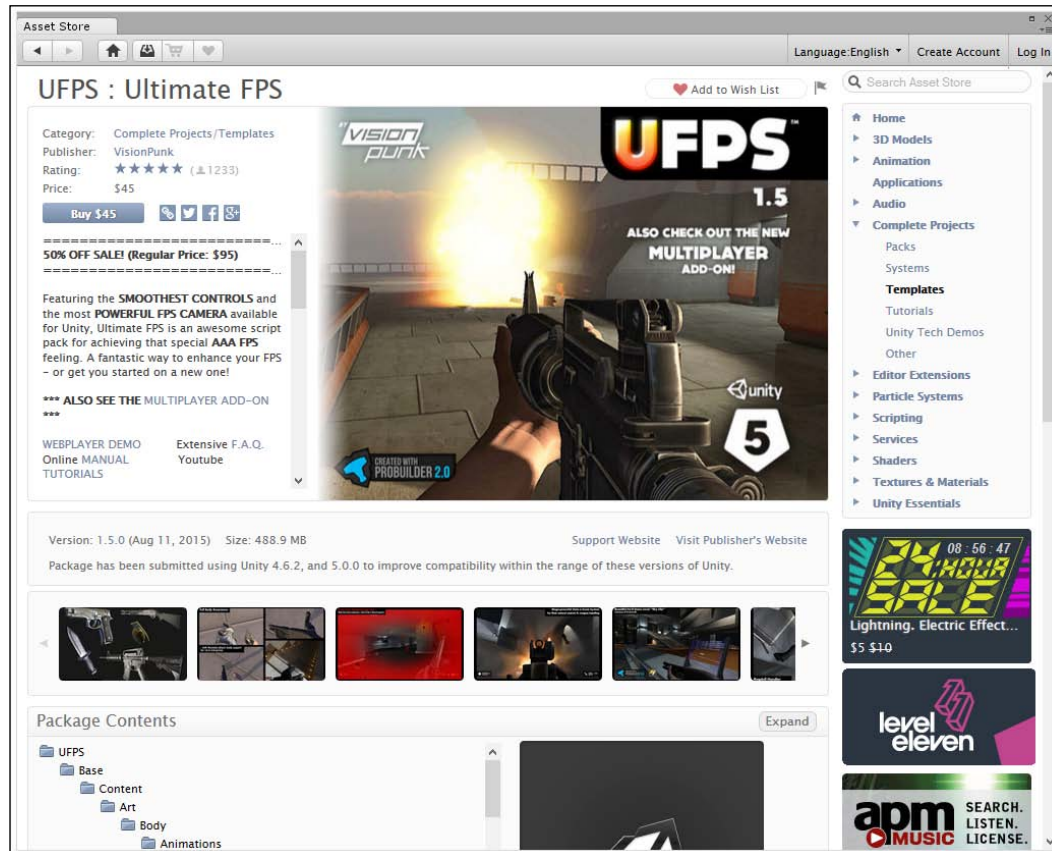
It's important to note that UFPS does cost money. But considering that a solid game programmer generally makes \$50 or more an hour, the amount of time saved really makes it a worthwhile investment.

Now that we are logged into the Asset Store, let's purchase and install UFPS:

1. Now, from the search bar directly below the login information, search for UFPS. You should see an icon that looks similar to the following:



- Click on the first option, which will bring you to the following page:



- Now, click on the **Buy** button and purchase the asset. Once it is purchased, the screen will change the **Buy** button to a **Download** button. You can click on it to download the asset and bring it into your project.