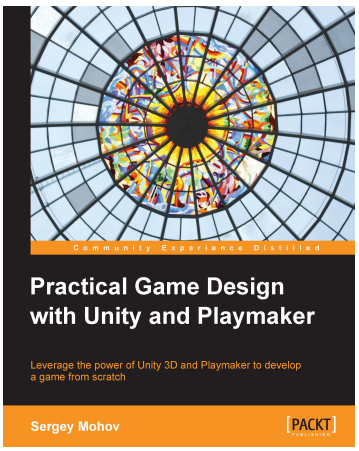
[PACKT] PUBLISHING

Practical Game Design with Unity and Playmaker

Sergey Mohov



Chapter No. 1

"Getting Started with Unity and Playmaker"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.1 "Getting Started with Unity and Playmaker"

A synopsis of the book's content

Information on where to buy this book

About the Author

Sergey Mohov is a game developer and designer with over three years of experience in working on games in Unity. His prominent projects include Dédale, Paradis Perdus, and Lune. The rest of Sergey's games can be found on his website at `http://sergeymohov.com` along with his personal blog.

I would like to thank everyone in the Unity community who helped me get from nothing to something. Special thanks to Mike Renwick who leaves no question unanswered. Many thanks to my friends and family for your continuing support. Thanks to Gwen, because everyone needs to get distracted every now and then.

I can't go without thanking the staff at Packt Publishing for suggesting that I write this book and for their guidance throughout the process.

Practical Game Design with Unity and Playmaker

Historically, game design has been a broad term. It is still often associated uniquely with creating the rules of a game, establishing a system, defining what the game will be in broad brushstrokes and then creating particular gameplay scenarios or levels. Sometimes, we go a little farther and endow game designers with the power to define how the game's universe and aesthetics will work. Broad as this definition might seem, it remains very limited in one area, and that is technology.

As the game industry moves toward its maturation, many small studios begin to emerge, often teams of one, two, or three individuals, attributing their success to the new generation of tools now available to game developers. While the larger game companies employing hundreds of developers have mostly been able to stand their ground, in this new landscape, the focus has already shifted toward polyvalence and self-sufficiency.

Game designers have always wanted more control over the games that they make, and, for the first time in history, there is no reason why they should not reach out and take it. Knowing how your game works on the inside means having more control and creative freedom. It also means more effective communication with other members of the team, provided there are any. It empowers you to go ahead and simply make the game on your own, be it a mere prototype, a game jam project, or a full commercial release.

Unity 3D is a game authoring tool that has changed the way we think about game development forever, making it much cheaper and more democratic. It allows anyone willing to invest their time into it to craft amazing game worlds and experiences for next to nothing. It has been one of the major forces driving this incredible change, with millions of new developers joining the game industry simply because they can now. The industry becomes more open every day, which enriches the pool of ideas that people bring to it from different backgrounds.

This book is an introduction to practical game design using Unity and Playmaker by Hutong Games. The latter lets anyone make a game without writing any programming code, while not giving up any of the power that Unity has to offer. Knowing the basics of programming is still valuable while working on the technical side of a game, but going deep into it is no longer an imperative, which means that creating a game is now easier than it has ever been before.

Even if you know how to code, learning how to use Playmaker in Unity may help you see the synergy between your game's mechanics and technology more clearly, making the development of your project more about finding the right design and less about working around the computer code.

There can be no doubt about the bright future of the practical aspect of game design, and this book offers you to become part of this future.

What This Book Covers

Chapter 1, *Getting Started with Unity and Playmaker*, gives us information about installing Unity, buying and installing Playmaker, creating and setting up the project, and making some adjustments to the interface layout.

Chapter 2, *Unity's and Playmaker's User Interface*, gives an introduction to the essential interface elements of Unity and Playmaker: menus, panels, and views.

Chapter 3, *Components and State Machines*, gives details about the project structure in Unity with and without Playmaker and introduces the component-based approach to game development.

Chapter 4, *Creating Your First Game*, gives information about applying and expanding your Unity and Playmaker skills learned in previous chapters to create real game mechanics.

Chapter 5, *Scripting and Custom Actions*, gives an introduction to Unity scripting in JavaScript and C#, making a custom Playmaker action.

Chapter 6, *Networking and Multiplayer*, gives details about adding multiplayer to your game using the plugin called Photon Cloud.

Chapter 7, *Working with External APIs*, gives information on working with external APIs, making your game available online on Kongregate, and integrating its online leaderboard.

1
Getting Started with Unity and Playmaker

In this chapter, we are going to go through the process of getting the project ready to work with this book. We will cover the following subjects:

- Downloading, installing, and setting up Unity
- Buying and installing Playmaker using the Asset Store
- Creating a new project and changing project settings

If you are familiar with the process of downloading and installing Unity and plugins for it, you can move to the section of this chapter called *Setting up your project*. It shows the layout of Unity and project settings that are going to be used for examples in this book.

Downloading and installing Unity

Unity is available for both Mac OS X and Windows PC, and the process of installation for these platforms is very similar. This book is going to use the Mac version of Unity, but I will provide you with alternative directions and hotkeys when they are different in the Windows version. Luckily, everything is, in fact, quite similar, and you should not have any problems switching between the two platforms if you choose to do so.

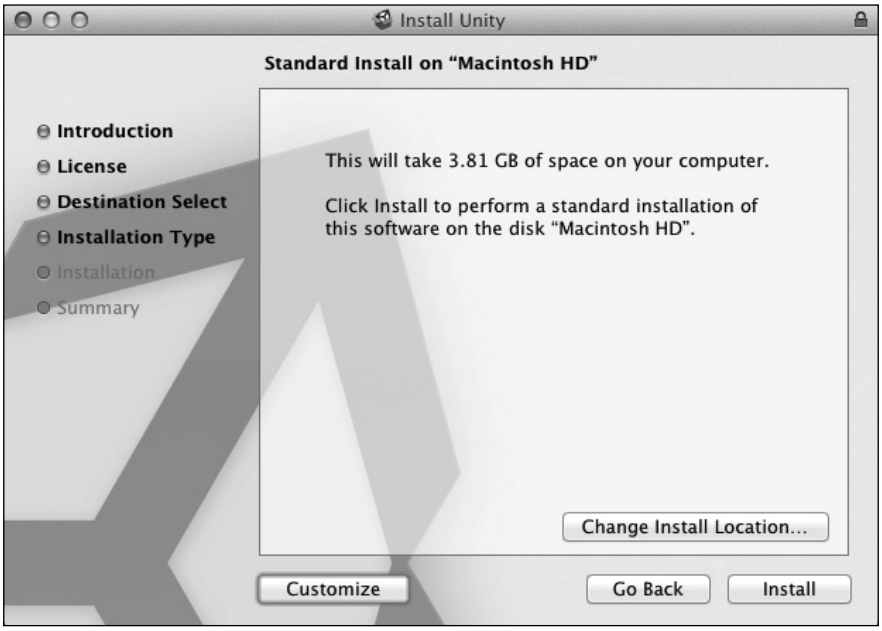
First things first, let us head over to `http://unity3d.com/unity/download/` and download the latest version of Unity available (as of writing this book the version is 4.3). Just click the blue **Download** button, and the file should start being downloaded to your hard drive. The file in question will be a DMG file for Mac users or an installer EXE for Windows users. Save it wherever you want on your hard drive.

If you are on a Mac, double-clicking the DMG file will reveal the PKG file, which is the installer. Double-clicking `Unity.pkg` will produce essentially the same result as double-clicking `Unity.exe` in Windows—it will prompt the installation wizard. You are not given any Unity-specific options except for the location where you want to install the engine.

You can install multiple versions of Unity on the same computer. To do this, you just have to change the name of the folder in which Unity is being installed and put the new installation into a different folder. You might want to do this if you want to have different versions installed at the same time, either because of license restrictions or because of some version-specific features that you want to access. Keep in mind that if you update a project to a newer version of Unity, you cannot go back to an older one.

The following screenshot shows the installation window of Unity. In this stage of the wizard you can specify the directory where Unity will be installed or change some installation parameters. In most cases you will want to install Unity with the default parameters. If that is the case, simply click on **Install** and skip the next step.

If you want to change the install location, click on **Change Install Location...**. The **Customize** button lets you include or exclude specific packages, such as the main Unity installation, example project, and Unity Web Player. Leave all of the boxes checked for this installation. We will talk about packages later in the book.

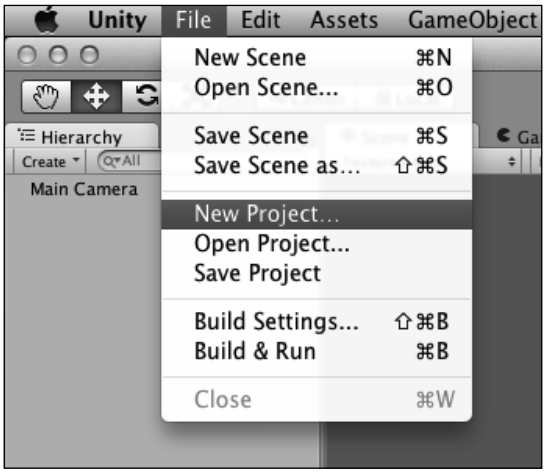


Clicking on **Install** will lead to installation of Unity on your hard drive. Once installation is done, you can go to the folder where you installed it (by default, `Applications/Unity` on Mac or `C:\\Program Files\Unity` in Windows) and launch it.

If you are launching Unity for the first time, a window will pop up asking you to select your type of license and enter your Unity account login and password. Do create an account if you do not have one and enter the details. Later you will need it for things like forum access and Unity answers—both very helpful when you are looking for solutions to Unity-related problems. In this book we are going to be using the free version of Unity, since none of the examples here will require any pro-license features. So select **Activate the free version of Unity** and click on **OK**.

The first time you open Unity, a default project will likely be loaded. These test projects included with the installation (and available through Unity's website) will be quite helpful later on when you want to find out how to do some specific thing (for example, mini-map or water physics), but do not know where to start.

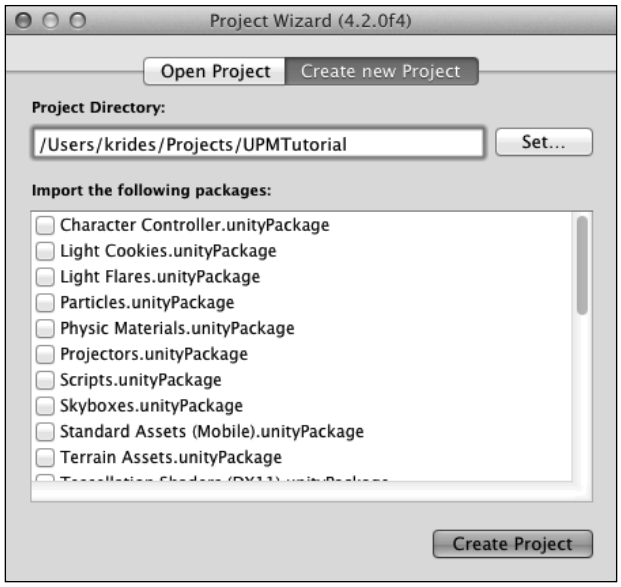
For now, though, we want a new empty project. Once Unity is loaded, go to the **File** menu in the main menu on the top-left corner of the screen and select **New Project…**.



This will prompt the **Project Wizard** window, in which you can choose where you want your future project to be located, as well as how it will be called. In Unity, a project is a folder, so the name that you give to your project folder will automatically be assigned to the project. For the examples in this book, you will not need any packages at this point, so leave all the checkboxes empty, unless you already have Playmaker downloaded, in which case check Playmaker. When you are ready, click on **Create Project**.

If you want to always start from the **Project Wizard** (which is highly recommended if you are planning to have multiple projects), go to **Unity** | **Preferences** and check **Always Show Project Wizard**, then close the **Preferences** window. For Windows users it is **File** | **Preferences** instead.

Since we are not loading any heavy asset packages, the project will load really quickly, and you should see the main Unity interface again, except this time everything will be empty.

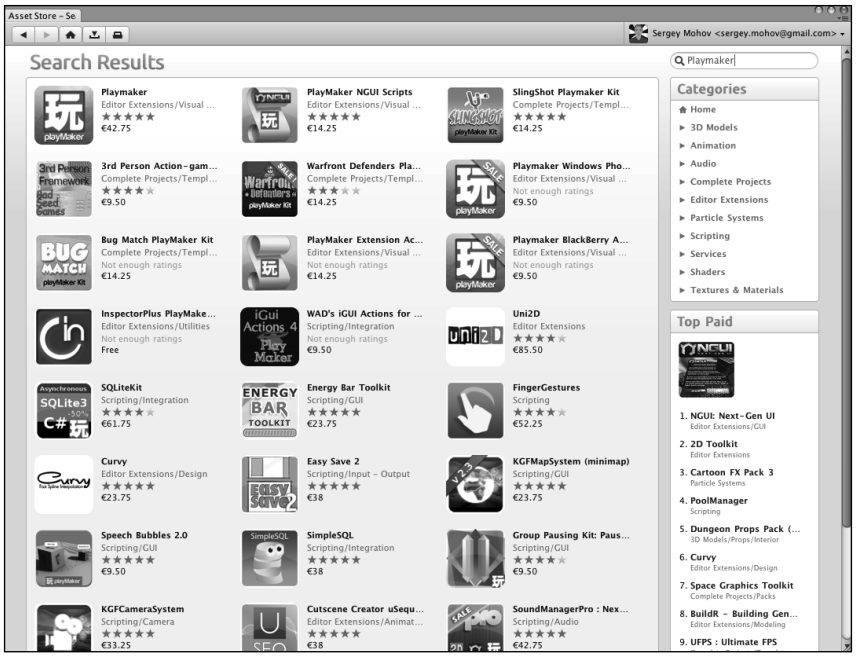


Buying and importing Playmaker

Before we change anything in the layout of our project, there is one missing component that we have to import, which is Playmaker by Hutong Games (`http://www.hutonggames.com/`)—the main tool (apart from Unity itself) that we are going to be using in most of the exercises in this book. To get Playmaker, we are going to use the Asset Store. The latter is an online market place valuable for any Unity developer, allowing us to save a lot of time by importing useful plugins and asset packs made by other users into our own projects. Some of the assets available through the Asset Store are free, but most of them, like Playmaker, are user-made and cost money. You too can make your tools and assets available through the Asset Store. Think of it as a developer-targeted analogue of the iTunes Store or Amazon.

Although you can access and search the Asset Store through a web page (`http://unity3d.com/asset-store/`), it is integrated directly into the engine in its most practical form. To access the Asset Store directly from Unity, go to the **Window** submenu in the main menu and select **Asset Store** from the list. This will open the **Asset Store** window in Unity (you need to be connected to the Internet to see the content of the store).

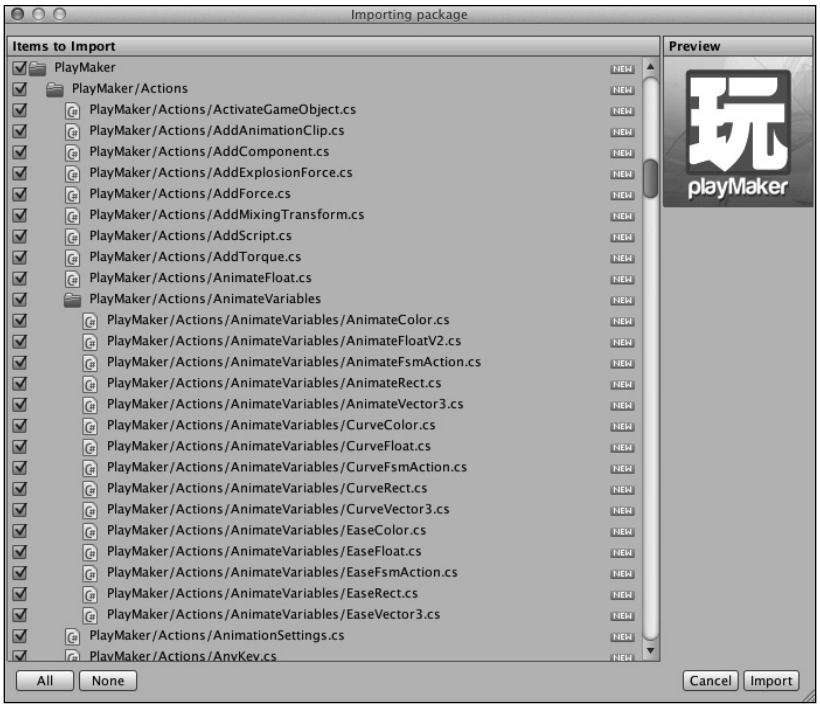
In the top-right corner of the **Asset Store** window, there is a search box. Type `Playmaker` in there and press *return* (*Enter*) on your keyboard. When the search results are displayed, click on the first one in the top-left corner of **Search Results**, simply called **Playmaker,** as shown in the following screenshot:



The **Asset Store** window will now display the description of Playmaker, user comments, some screenshots, its price, and the **Buy** button. Click on this button, enter your credit card or PayPal details, and complete the purchase as you would do with any other online store or service. After that Unity will offer to import the package. Click on the **Import** button and when Playmaker is downloaded, in the import window, leaving all the boxes checked, click on **Import** again. This will add the Playmaker files to your new Unity project.

Another way of installing Playmaker (or any other plugin for that matter) is going to **Assets** | **Import Package** | **Custom Package...**, provided that you already have the `unitypackage` file with it on your hard drive.

Playmaker also automatically imports some other popular Unity plugins it depends on: iTween for movement and animation and Photon for networking and multiplayer. We will talk about Photon later in *Chapter 6*, *Networking and Multiplayer*.

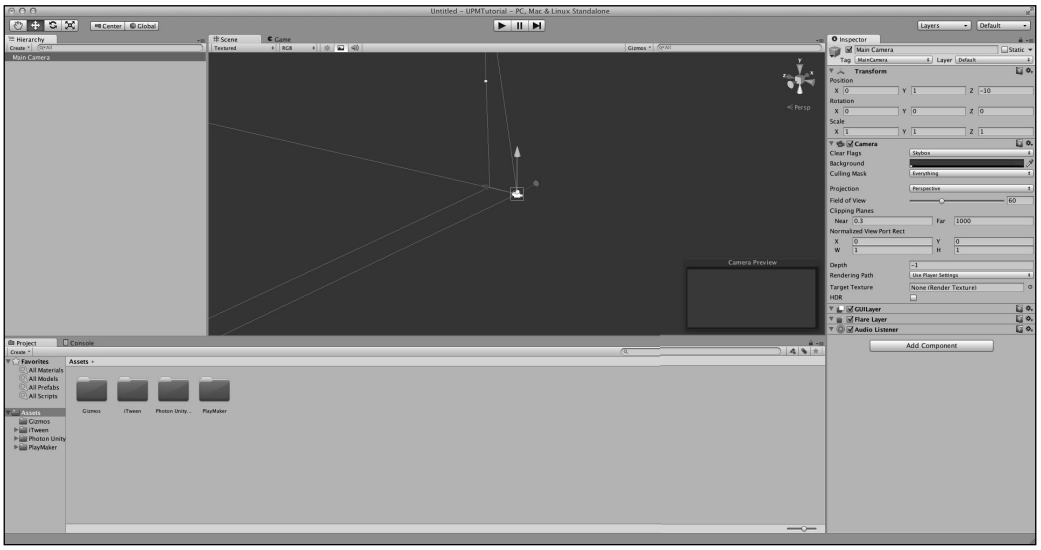


Setting up your project

There is an important thing that you have to know about Unity: you can customize it in many different ways and even expand its interface with your own custom windows and panels. We are not going to discuss this just yet, as the user interface is examined in more detail in *Chapter 2, Unity's and Playmaker's User Interface*, but you should know that there is nothing to be surprised about if the user interface changes when you install a new extension, such as Playmaker. You can also drag different tabs around and see which layout fits your screen size/configuration and project best. Now we are going to set up the interface in a way that will simplify the explanations in the next couple of chapters. Once you are comfortable with the interface, you can feel free to customize it however you want.

If you haven't dragged anything around, you should be seeing the default Unity layout. If you have, go to **Window | Layouts** in the main menu and click on **Default** in order to reset to the default window layout. You will see a tab called **Hierarchy** on the left, **Scene** and **Game** in the middle, **Inspector** on the right, and **Project** and **Console** on the bottom. We will talk more about these tabs and their functions in *Chapter 2, Unity's and Playmaker's User Interface*. Right now, in order to save screen space, we will drag some of them around and change their appearance.

The following screenshot shows the default interface layout in Unity.



Let us start with the Project tab that you can see on the bottom of the screen. If it is not highlighted, click on the word **Project**. Right now both **Project** and **Console** tabs are pinned to the bottom of the screen, so you can switch between them in order to select one or the other. Unfortunately, this layout is not ideal, as in most cases you will need direct access to both **Project** and **Console** tabs at the same time. Click and drag the **Project** tab to the right of the **Scene** and **Game** tabs, until it snaps into place next to the **Inspector** tab. It will become tall and narrow.

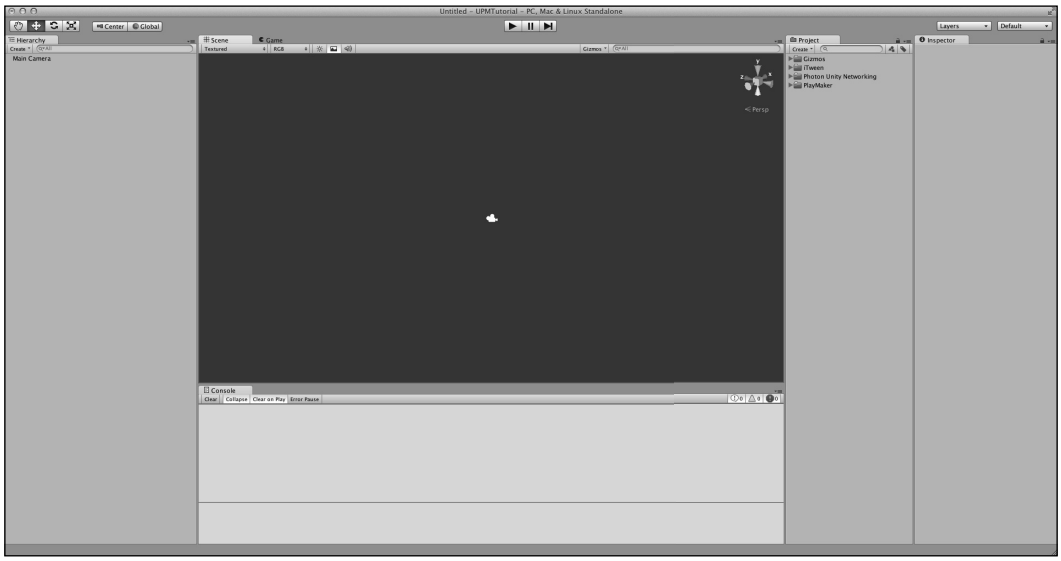
Unfortunately, now there's very little space in the Project panel with those large icons. Luckily, Unity allows us to solve this problem easily by clicking on the small options icon () in the top-right corner of the **Project** panel and selecting **One Column Layout** from the list.

Now your **Project** panel will need much less space, and you will be able to see all of your project's file structure without getting frustrated over icons and sub-panels getting in your way.

It is important to note here that some developers do prefer the **Two Column Layout** for its **Favorites** section and file manager-like subfolder representation; however, you will most likely want to have as much free work space as you can get, especially when working with plugins like Playmaker that add additional tabs that you need to have constant access to. You should consider the **Two Column Layout** only if you have a relatively big screen.

Now that the **Project** panel is set up and in place, let us free some more space for the **Hierarchy** panel by moving the **Console** panel under **Scene** and **Game**. To do this, click and drag the **Console** tab as you did with the **Project** tab, and move it to the bottom of the **Scene** and **Game** panels until it snaps into place directly under them.

The following screenshot shows what your editor window should look like if you followed all of the instructions in this chapter correctly.



Finally, your work space is all set! Now it is time to change the project settings. It is a good idea to do this before you start working in order to get this out of the way, so let us get to it. We will not go into much detail about why certain project settings are set in a certain way, and most of them are quite self-explanatory.

First, select **Edit** | **Project Settings** | **Player** from the main menu. A new menu should appear in the **Inspector** panel on the right. You can put your name or the name of your company in the **Company Name** field near the top of the menu, and **Product Name** (just below it) by default should be the same as the name of your project folder that you specified while creating the project. Changing the project name will not rename your project folder. In fact, everything you change in the Player settings only affects the output game or application. We are going to be building a Unity WebPlayer project for this book's examples, so click on the tab with a little planet icon ().

There are only two things here that we are interested in right now: screen resolution and WebPlayer template. Make sure that the **Default Screen Width** and **Height** are set to `960` and `600`, respectively, and the active template is **No Context Menu**. There are more options here, but we are going to leave them at their default values for now.

Next, we will change the build settings to match our target platform. Go to **File** | **Build Settings...** and select **WebPlayer** from the list of available platforms (it should be the topmost item on the list). Then click on the **Switch Platform** button (as shown in the following screenshot). This will allow you to test the game directly with the output resolution. Once Unity finishes reimporting files, you can close the **Build Settings** window.

Summary

In this chapter you learned how to download and install Unity, purchase and install plugins for it from the Asset Store, prepare your project for work, optimize your workspace, and select the output platform. In the next chapter, we will look at Unity's interface in more detail. In particular, we will examine various panels and views (for example, **Inspector**, **Hierarchy**, **Game**, **Scene**) that you have encountered in this chapter and start using them.

Where to buy this book

You can buy Practical Game Design with Unity and Playmaker from the Packt Publishing website: `http://www.packtpub.com/practical-game-design-with-unity-and-playmaker/book`.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

For More Information:
www.packtpub.com/practical-game-design-with-unity-and-playmaker/book