



TECHNOLOGY IN ACTION™

Unity



for Absolute Beginners

*GET ON THE FAST TRACK TO BECOMING A
GAME DEVELOPER WHILE LEARNING UNITY
IN A FUN AND INTERACTIVE ENVIRONMENT*



Sue Blackman

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.



Apress®

Contents at a Glance

About the Author	xiii
About the Contributor	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
■ Chapter 1: The Unity Editor.....	1
■ Chapter 2: Unity Basics	39
■ Chapter 3: Scene Navigation and Physics	103
■ Chapter 4: Importing Static Assets	153
■ Chapter 5: Introduction to Scripting with C#.....	197
■ Chapter 6: Mecanim and Animation	243
■ Chapter 7: Populating the Game Environment.....	291
■ Chapter 8: Weaponry and Special Effects	339
■ Chapter 9: Incorporating Unity 2D.....	389
■ Chapter 10: Menus and Levels	447

■ Chapter 11: Bonus Features	525
■ Appendix A: Rigging with Mixamo	565
Index.....	575

Introduction

The Unity community is very large and very helpful. There are videos, code, and 3D assets available on any number of topics, and the Unity help documents have recently undergone a major update. But if you are new to any or all of the components of game development (scripting, 2D or 3D art asset creation and manipulation, or game design in general), you have probably discovered how overwhelming it is to make sense of it all.

One of the biggest challenges is to learn how to bring it all together. Unlike short topic videos, this book takes you through the process of creating a game that seems very simple at first glance, yet becomes more sophisticated as you work your way through the design and creation process.

One of the great advantages books have over videos is the luxury to spend time on explanations. In game development, there are always several different ways to solve design and technical problems, but there is rarely a “best” solution. And unlike a large studio, where budget and time constraints dictate a highly detailed game document, you will discover that the game design and creation process must be flexible in order to be able to finish a game or even a working prototype.

About the Unity Game Engine

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 royalty free, allowing people to make and sell games with the very low overhead essential to the casual games market.

With its huge user community, Unity removes the elitist separation between programmers, artists, and game designers that is typical of high-priced game engines. It makes it possible for anyone to take the first step to bringing their ideas to life. In this book, you will get to wear many hats as you create your first Unity game, discovering where your interests lie as well as gaining an understanding of what is required should you reach the point where collaboration becomes appealing.

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 need 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 needs 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 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 with games that are largely controlled by physics, logic-driven cause and effect is what differentiates games from linear, plot-driven passive media. Even the most "artist friendly" game engines need scripting to move beyond simple environmental walkthroughs. If you have no previous scripting experience, you will find that this book's aim is to familiarize you with scripting a few lines at a time, while providing visual feedback as often as possible. If you find you enjoy the scripting part of the project, feel free to delve deeper into programming with a more conventional approach.

What About Math?

One of the most common things heard in the game industry by artists and programmers alike is, "If I'd known math was going to be so useful, I would have paid more attention in class." Although it helps to have a good background in math, there are plenty of resources for both code and mathematical functions to help you solve a particular problem. And, as always, there are plenty of people on the Unity Forum and Answers who may be willing to help as long as you can show that you have spent a reasonable amount of time trying to solve it yourself.

Assumptions and Prerequisites

This book assumes that you are new to scripting, 3D and game design, and/or the Unity engine.

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. More typically, you will find yourself building 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. It uses programming best practices when possible, but 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 generally 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 own language. He 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. What Is the General Structure of this Book?

Tip Follow this format.

Code looks like this

Platform

This book was written using Unity 4.5 in a Windows 8 environment. Differences for shortcut keys and operating system file storage with Unity on a Mac are noted throughout the book.

The Unity Editor

The most exciting part of learning a new application is getting it installed and firing it up for the first time. Following that, the most frustrating part can be the process of becoming familiar with its editing environment. Unity is no exception, especially if you are already familiar with DCC (digital content creation) applications. In this chapter, you will be introduced to the Unity editor and many of its key concepts. As this will serve as a light overview for the rest of the book, don't stress over trying to remember it all.

Installing Unity

The first order of business is to download and install Unity if you have not already done so.

Unity has two license types: free and Pro. When you download and install Unity for the first time, you will have the option of using a free 30-day trial of the Pro version. It can be selected as a third option right in the license dialog upon installing Unity or upon returning the currently active license (an option that allows you to move Pro licenses to different machines). With a few exceptions, most of the Pro features are aimed at the optimization of your projects and are beyond the scope of this book. If you are considering the purchase of Unity Pro, you should be aware that, with the exception of Windows Mobile, it will require a separate license for each of the mobile target platforms you wish to author for.

Unity User Account

Currently, after you download and install Unity, if you don't already have one, you will be required to create a Unity user account during the installation. Your user account will provide access to the downloads, Unity Forum, Unity Answers, and the Unity Asset Store. The forum provides a place for discussion of all things Unity. Answers is where you can go to get quick solutions to your queries, and the Unity Asset Store is where you can find assets of all kinds (3D models, animated characters, textures, scripts, complete game environments, etc.) to help you with your project. Once Unity is installed, you will find a direct link to each of these resources through the editor.

To get started, go to www.unity3d.com/unity/download.

With the account created, you can download Unity for Windows or Mac. The download site will automatically offer the version that matches the platform you are currently on.

Installing

This book was written using Unity 4.5. Because Unity regularly makes changes that can affect your projects, you may wish to use the 4.5 version even if the current version is newer. If you prefer, you *can* install multiple versions of Unity on the same machine, providing you name the folders accordingly (e.g., Unity 4.5, Unity 4.6). You may only run one instance of Unity at a time, regardless of the version. If you choose to have multiple versions, you will be required to start Unity from the desired version rather than from the desktop icon or the project itself, as the previously run version will open by default.

If that sounds like too much trouble, you can always check the book's thread on the Unity Teaching forum, <http://forum.unity3d.com/forums/23-Teaching>, where both errata and version-change solutions will be posted and updated as necessary. A search for the book's title will help you find its thread.

1. Install Unity, following the prompts, including the sample project.

On a Windows machine, it will be installed in the Program Files x86 folder. Upon the release of Unity 5.0, when it will be fully 64 bit, it should install itself in the regular Program Files folder. The sample project will be installed in Documents ► Public Documents ► Unity Projects. On a Mac, you will find it in /Users/Shared/Unity.

At this point, you should now see the Unity icon on your desktop (Figure 1-1).



Figure 1-1. The Unity application icon on the desktop

2. Click the icon to open Unity.

If you downloaded the sample project, Unity will open with it already loaded (Figure 1-2).

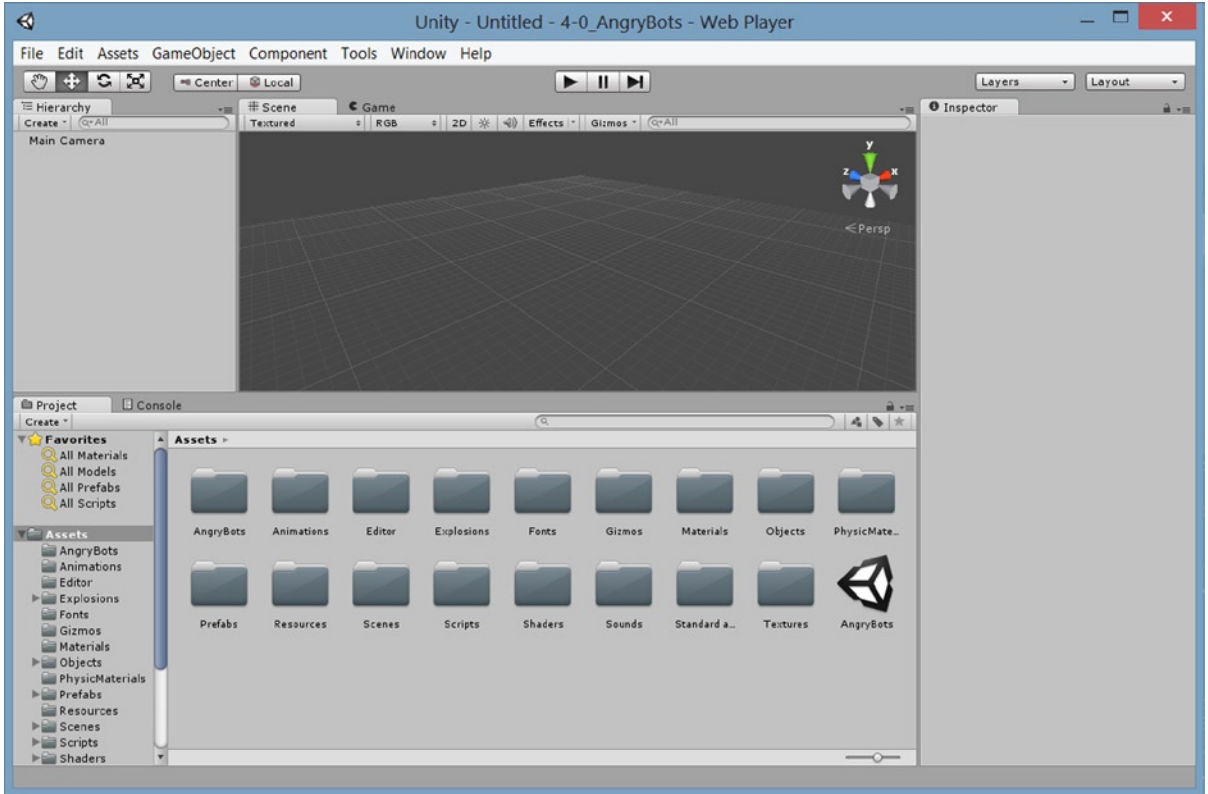


Figure 1-2. The Unity sample project loaded

If the Scene window is blank, you can double-click the Unity icon in the Assets folder to load the AngryBots scene. The demo scene Unity ships with changes occasionally, so yours may be different.

Tip When opening Unity from the desktop icon, it will open the previously opened project, provided it can be found.

If you did not choose to install the sample project, you will see the Project Wizard (Figure 1-3).

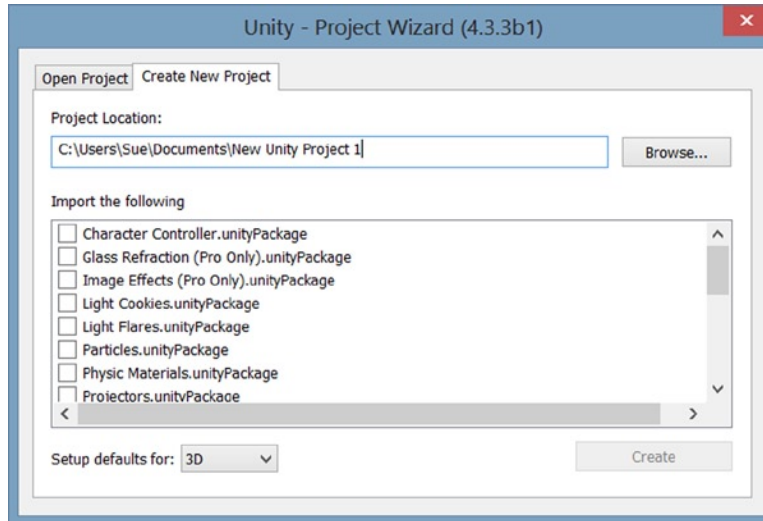


Figure 1-3. The Unity Project Wizard

From the Project Wizard, you can create a new project or browse to open an existing project. Initially, exploring the editor with a finished project will make more sense. You can obtain the sample project from the Unity Asset Store directly from a new project.

1. In the Project Wizard, select the Create New Project tab.
2. Set the Project Location manually using the Browse button, or accept the default name and location.

Tip When you create a new project, you will be creating a folder to house the project. The folder *is* the project. You can only create a new project in an empty folder.

3. With the project name and location specified, click the Create button.

The new project and, as yet nameless, *scene* opens in the Unity editor.

Originally, if you pressed the Alt key down immediately after clicking the Unity icon on the desktop, it would force the program to start with the Project Wizard. Because that required excellent timing and/or a good dose of luck, Unity finally added the option to the Preferences section. While it may seem like a nice time-saver to automatically open your last file, it has its drawbacks. If the project has become corrupted, Unity will not be able to open it (giving you access to the file menu where you can open a different file). If you are using multiple versions of Unity, it will recompile data if the last project was opened in a different version. Depending on the version, the changes made could be permanent. To avoid complications, the best practice is to set the preference to always start with the Project Wizard.

4. From the Edit menu, select Preferences.
5. Under General, turn on Always Show Project Wizard (Figure 1-4).

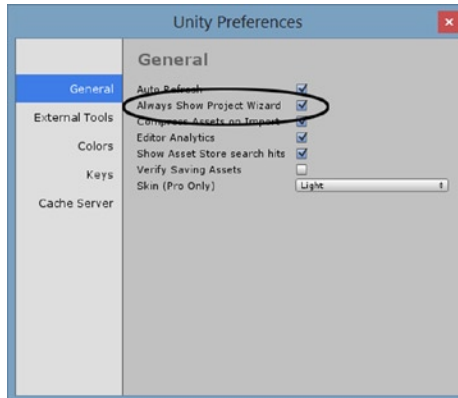


Figure 1-4. Preferences set to Always Show Project Wizard

General Layout

With the project loaded, you should see Unity sporting its default layout. If you are using the free version, the UI should appear in light gray (Figure 1-5). If you have purchased, or are evaluating, Unity Pro, the background will reflect the dark theme (Figure 1-6). For this book, the screenshots will utilize the light theme for better contrast. If you have Pro and prefer the lighter version, you can change the Skin in the General section of the Preferences.

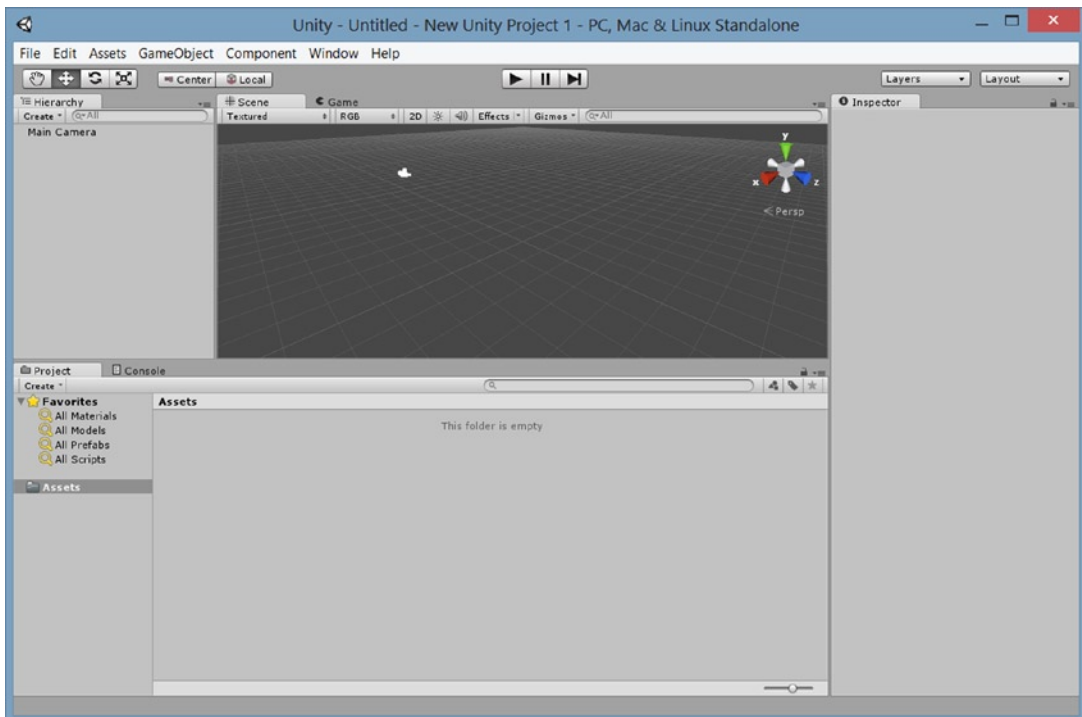


Figure 1-5. The Unity UI, light theme

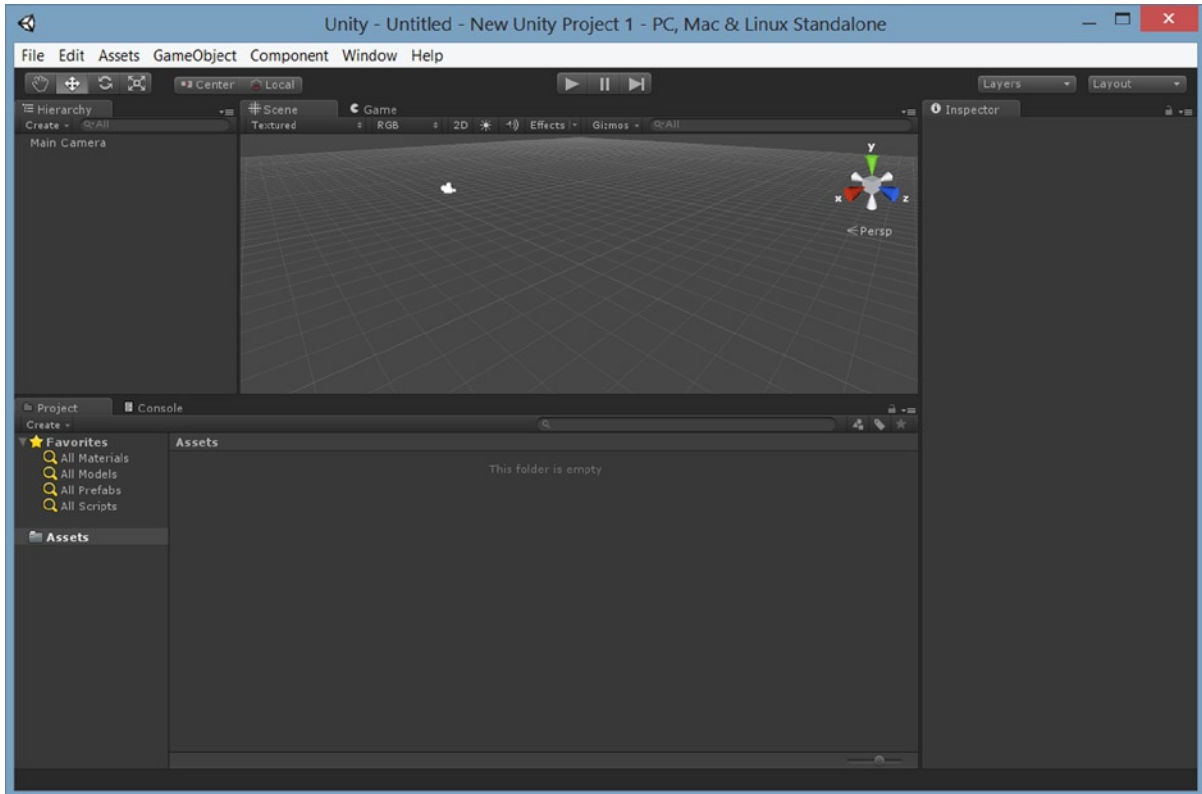


Figure 1-6. The Unity UI dark theme

Unity's UI consists of four main “views” and several important features. The Scene view, Game view, Hierarchy view, and Inspector view (typically referred to as the *Inspector*) are generally accessible in any of the layout options, as are the playback controls, coordinate system options, and object/viewport navigation buttons (Figure 1-7). If the term “views” seems a bit odd, it helps to know that depending on which view is *active*, your input (or the events it triggers) will generate different results. The first click in a different view will set the focus to it.

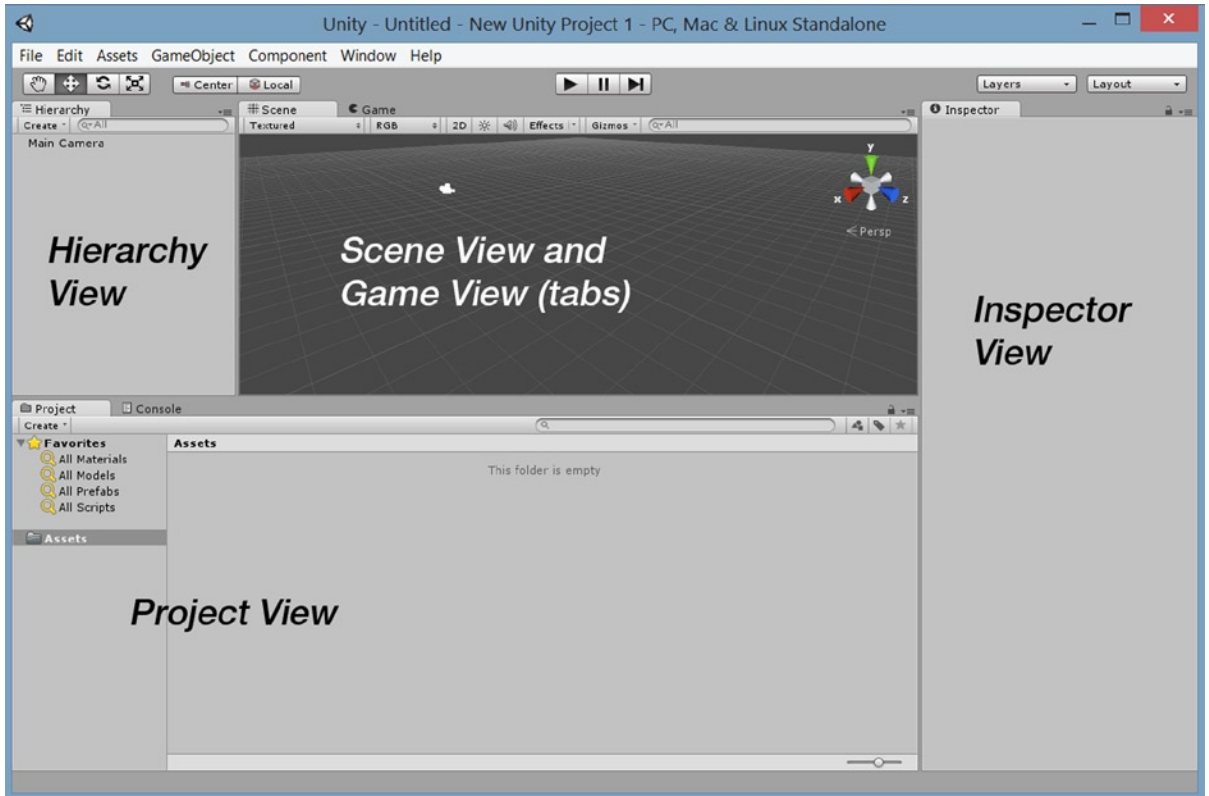


Figure 1-7. The Unity GUI and its main features

With the default layout, the main feature is the Scene view. This is where you will be able to arrange your 3D assets for each scene or level in your project.

In the tab next to the Scene view, you will find the Game view. In the default layout, when you click the Play button, the Game tab is automatically opened. The game view is where you will interact with your scene at runtime to test your game play.

To the left of the Scene/Game viewports, you will find the Hierarchy view. This is where you will find the objects that are currently in the loaded scene or level.

Below the Scene/Game view is the Project view. This contains the resources or assets available to the current project. This folder corresponds directly to the project's Assets folder on your operating system. Deleting files from the Project view will send them to the trash on your computer. If you author from an external hard drive, you should be able to locate the deleted files in the drive's own trash folder. The main point here is that there is no "undo" for these deleted files.

To the far right, you will find the Inspector. This is where you will have access to the parameters, options, and other particulars of selected assets, from the Hierarchy or Project views, as well as general project-related settings.

At the top left, you will find the navigation controls that will allow you to move and arrange objects in your scene and re-orient the Scene view itself for easier access to the various objects.

To the right of the navigation tool controls are the coordinate system options. Here you can specify Global or Local coordinates as well as using the object's designated Pivot Point or the Center of its bounding box as the rotation point.

In the top center, you will find the Playback buttons. The Play arrow will put you in Play mode, where you can test your game's functionality and presentation in a Windows or Mac environment. The Background color of the UI will also change, alerting you to the fact that you are in Play mode. The Play button becomes a Stop button once you are in Play mode. You also have the option to Pause the game, using the middle button, or to step through one frame at a time, using the right button.

At the top right, you will see two drop-down menus. The Layers button will allow you to set various Layers to be active or inactive during Play mode, or define new layers. The Layout drop-down menu provides a quick way to switch between several popular layout configurations.

Menus

The menu bar consists of seven menus, a few of which may already be familiar to you from other applications.

The File menu (Figure 1-8) is where you will save and load both scenes and projects. It is also where you will build, or compile, your game before sharing it with others.

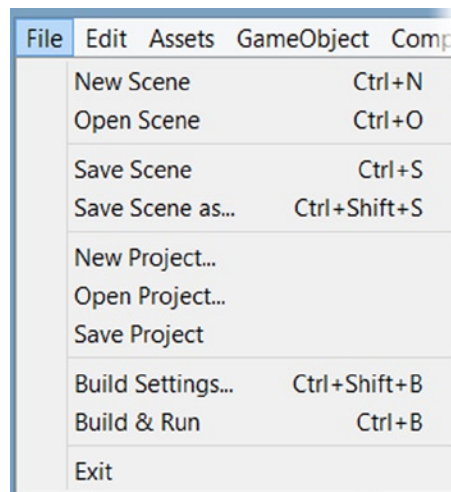


Figure 1-8. The File menu

In the Edit menu (Figure 1-9), you will find the usual editing options: Undo, Redo, Cut, Copy, Paste, Duplicate, and Delete. The next couple of sections deal with functionality generally accessed through keyboard shortcuts or UI buttons, but that are useful for identifying the shortcuts. The Play options are fairly self-explanatory. You will investigate Find later in the chapter. The Edit menu is also where you can find access to Project and Render settings. These are settings that are not associated with any particular scene object. Preferences is where you can go to customize the Unity editor for matters other than layout. At the bottom, you can gain access to the snap options to help with the arrangement of your scene assets.

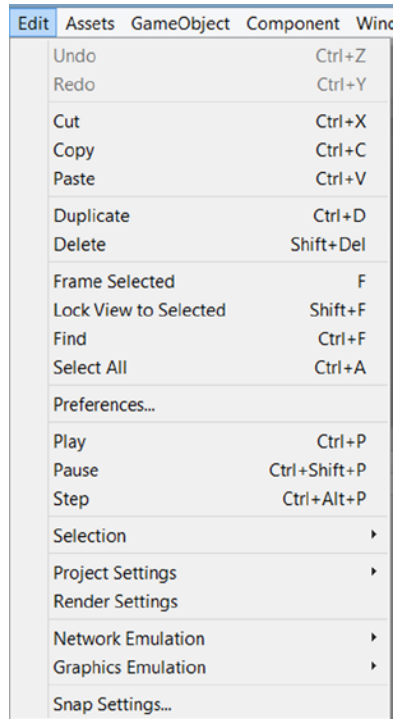


Figure 1-9. The Edit menu

In the Assets menu (Figure 1-10), you will see the various options for creating, importing, and exporting assets. This extremely useful menu can also be accessed in a couple of different places in the editor, as you will see throughout the book. Topping the list is the Create submenu. This is where you will create most of your Unity-specific assets, such as scripts, materials, and a variety of other useful things. Along with the menus for importing assets such as textures and 3D models, you will find a mainstay of Unity game development: the means of importing and exporting Unity “packages.” Packages are the vehicle for transferring all things Unity with their relationships and functionality intact.

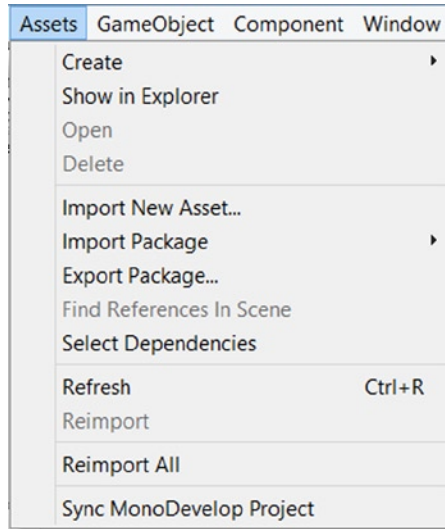


Figure 1-10. The Assets menu

In the GameObject menu (Figure 1-11), you will be able to create several types of pre-set objects, from the most basic of Unity objects, an “Empty GameObject,” to primitives, lights, cameras, and a nice variety of 2D and 3D objects. Also of note in this menu are the bottom three commands. They are the means for positioning objects relative to one another, including Cameras and their views. In Unity, anything that is put in your scene or level is called a *gameObject* (lower case g). More than just Unity’s name for an object, *gameObject* specifically refers to an object that inherits from the *GameObject* class (upper case G), that is the code that defines the top level object and its basic behavior.

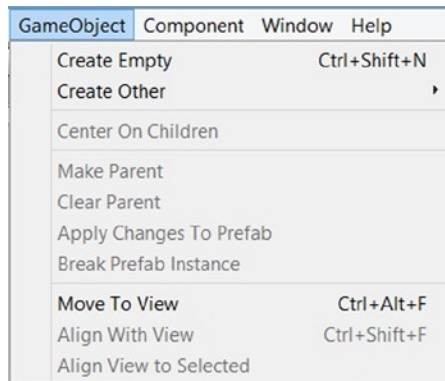


Figure 1-11. The GameObject menu

The Component menu (Figure 1-12), is where you can add components to define or refine your *gameObject*’s functionality. Any of the “Create Other” objects from the GameObject menu *could* be built from scratch by adding the appropriate components to an empty *gameObject*.

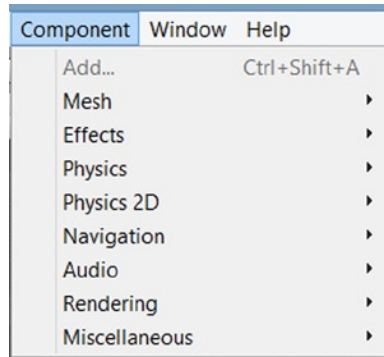


Figure 1-12. The Component menu

The Window menu (Figure 1-13) is where you can go to open or change focus to Unity’s standard and specialty views or editors. The shortcut keys are listed if they exist. Note the Asset Store item. This will take you directly through Unity’s Asset Store, where you can import assets directly into your game.

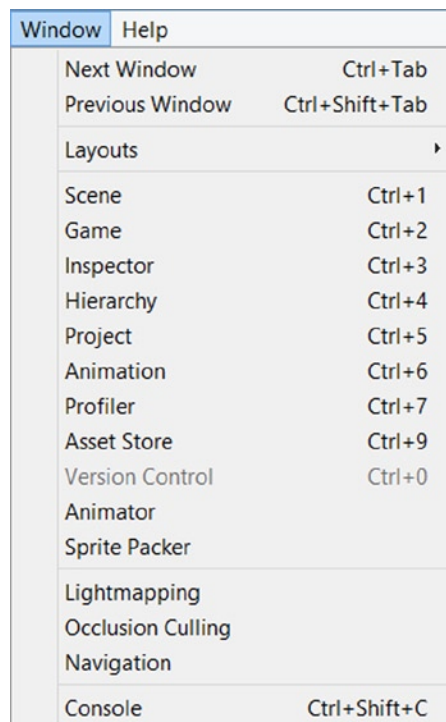


Figure 1-13. The Window menu

The Help menu (Figure 1-14), as expected, will provide you with the version number and license information of your Unity installation. In this menu, you can find access to the three main pieces of Unity documentation: the Unity manual, where you can find information on the workings and concepts behind much of the Unity workflow and features; the Reference Manual, where you can get specific information on Unity components; and the Scripting Manual, where you can find Unity-specific API classes, examples, and scripting tips. Additionally, the menu supplies links to the Unity Forum and Answers (when you just need a quick solution) and a means of reporting a bug should the need arise.

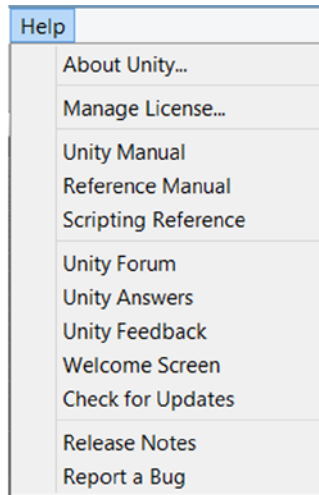


Figure 1-14. The Help menu

Getting Started

Each of Unity’s “views” comes with its own options and settings. Most are along the top of the viewport or panel, but there are a few that are located at the bottom. Once again, this is just a quick overview to familiarize you with some of their key features.

If you didn’t choose to download and install the sample scene, AngryBots, with your Unity install, you will need to fetch it from the Unity Asset Store if you wish to follow along with the next section’s exercises. It is a free download and can be found in the Complete Projects category, Unity Technologies section.

1. From the Windows menu, select Asset Store.
2. Select the Complete Projects category.
3. Select the Tutorials subsection.
4. Select Angry Bots.
5. Click the Download button’s drop-down arrow (on its right side).
6. Select Download Only.

7. Close the Asset Store window when the project has finished downloading.
8. In the Unity editor, from the files menu, select Open Project.

The Project Wizard opens (Figure 1-15).

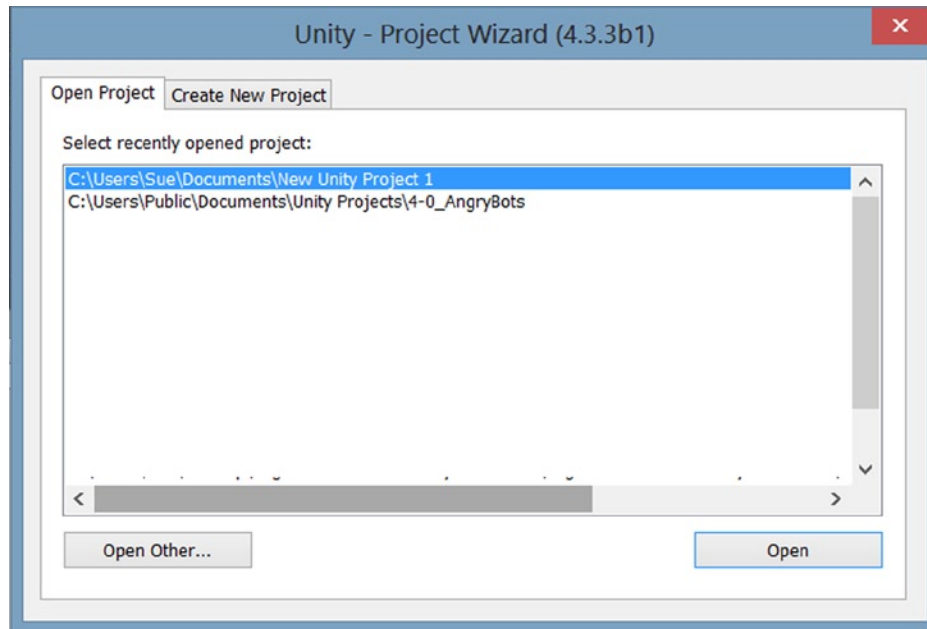


Figure 1-15. The Project Wizard

9. Select Open Other...
10. Navigate to where the project was installed, and select the parent folder, 4-0_AngryBots.

When Unity opens a project that you were previously working on, it will open to the scene you were editing or viewing. If this is the first time the project has been opened, it will open to a new, unnamed scene. Since all new scenes are created with a default camera, it won't be entirely empty, but it also won't do you much good as you check out the various views.

One of the first things you will learn to do in Unity is to keep your projects organized. While it's not mandatory to do so, you will find that most developers create a folder called *Scenes* to store the game's scenes or levels in. You will find that AngryBots does indeed have a *Scenes* folder, but the main scene was left on its own in the root folder. Unity helps you to locate various types of resources through the use of various icons. The icon used for scenes is, appropriately, the Unity production.

1. In the Project view, select the Assets folder in the left column and locate the AngryBots scene in the right column (Figure 1-16).

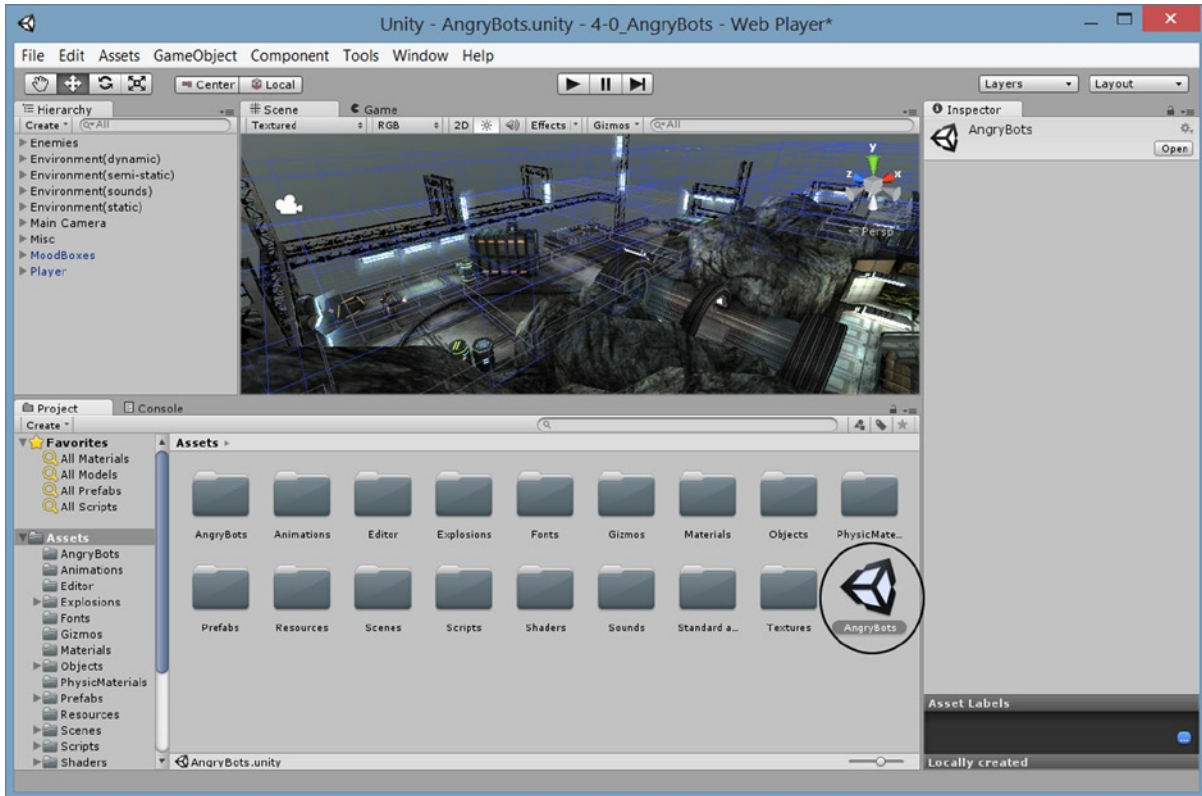


Figure 1-16. The AngryBots scene in the Project view

2. Double-click on the AngryBots scene name and/or icon to load the scene.

The contents of the scene are shown in the Hierarchy view. The Scene view reflects the last observation point from the last scene seen in the Unity editor, regardless of what project it was in.

Navigating the Scene View

The first thing you will probably want to do is explore the loaded scene in the Scene view. Because viewport navigation varies quite a bit in 3D applications, this is a good place to start your Unity experience. Let's start by focusing in on an object in the Hierarchy view.

1. In the Hierarchy view, double-click on the Main Camera to focus the Scene view to it.

Tip At the time of this writing, the double-click method to focus or find an object in the Scene view may no longer work on all platforms. An alternate means of zooming in on the selected object is to move the cursor into the viewport and press the F key. Both methods will “Frame” the selected object in the viewport.

Your exact view may vary, but from any particular location, it will be looking toward the Main Camera object (Figure 1-17). You will learn how to adjust the icon size and type in the Scene a little bit later.

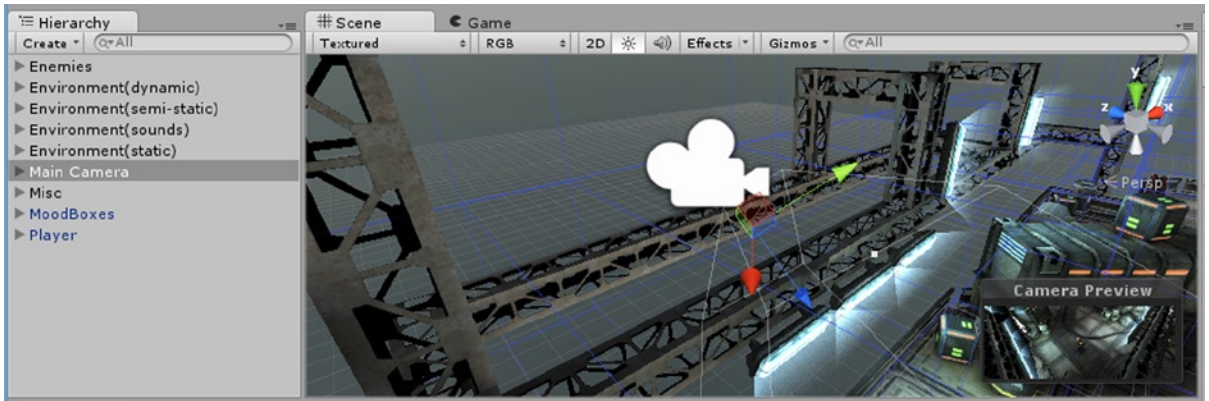


Figure 1-17. The Main Camera “found” in the Scene view

When a camera object is selected, you will also get an inset showing what it sees.

2. To orbit around the Scene view’s current focus, Main Camera, hold the Alt key down on the keyboard, press the left mouse button, and move the mouse around.

The scene view is orbited around the current center.

3. Release the Alt key, press the middle mouse button or roller, and move the mouse.

The view is panned (moved up/down/left/right).

4. Holding the Alt key down once again, press the right mouse button and move the mouse.

This time, you are zooming the scene in and out. You can also zoom by rolling the middle mouse roller.

5. Release the Alt key, and continue to hold the right mouse button down as you move the mouse.

With no Alt key, the view is rotated from the viewpoint’s origin.

You can also perform fly-through navigation through the scene.

6. Hold the W key down on the keyboard and then press the right mouse button. Move the mouse to perform direction changes.

The speed increases the longer the mouse button is held down, so you may want to try a few starts and stops. Try experiments with the A, S, and D keys along with the right mouse button.

Understanding the Scene Gizmo

In the Scene view, you have several options for working in the viewport as you create and refine your scene. The two main options are to view the scene in perspective mode or an isometric mode.

1. Double-click the Main Camera once again in the Hierarchy view to get back to some scene objects.
2. Using the right mouse button (or any of the alternative methods), rotate the view and observe the Scene Gizmo (Figure 1-18).



Figure 1-18. The Scene Gizmo in the upper right corner of the Scene view

The Scene Gizmo indicates the cardinal direction of the global or world directions. Currently you are seeing the view in a perspective view, Persp, but you can use the Scene Gizmo to change to an iso view, Iso. An 'iso' or isometric view is a view often used in drafting to create a 3D view without the vanishing point perspective of the human eye. Unlike perspective, where objects farther back in the scene *appear* smaller, in Iso, their size relative to each other is always correct. When coupled with the cardinal directions (front, back top, bottom, left and right), Iso will show a flat orthographic view that can make positioning objects much more accurate. When rotated off of the side or top/bottom views, the view becomes isometric. You can toggle the label to return to a perspective view.

3. Select the Y or top view arrow on the gizmo.

The scene is seen from the top (Figure 1-19), but it's still the perspective view, as indicated by the wedge-shaped icon next to the Top label, just below the gizmo.

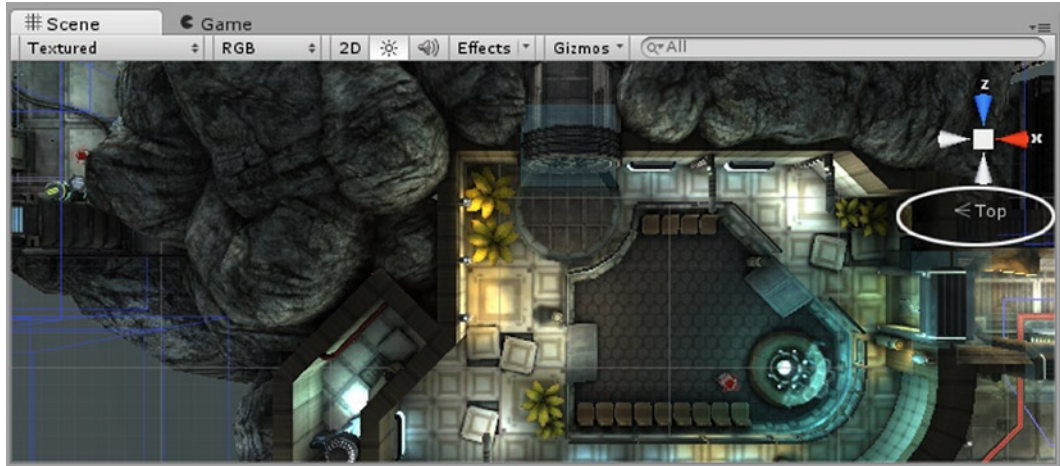


Figure 1-19. The perspective Top view

4. Click on the Top label.

Now the view is a flat orthographic view, as indicated by the three parallel bars of the new label icon (Figure 1-20).

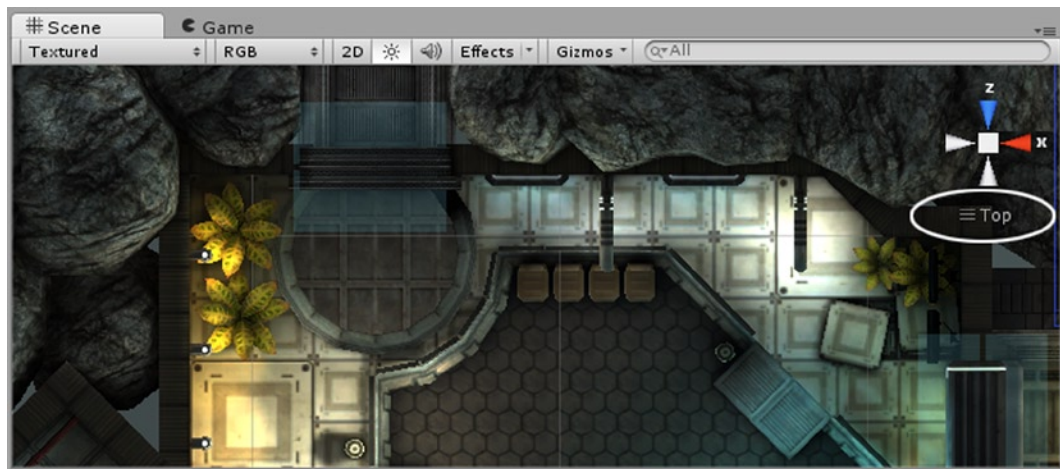


Figure 1-20. The ortho Top view

5. Try clicking the other Scene Gizmo arrows to see the other views.

These views will allow you to position objects with respect to each other with far more accuracy than a perspective view.

6. Use the right mouse button or its equivalent to rotate away from the current view.

This time the view becomes an isometric view (Figure 1-21). If you have an engineering background, you will recognize this as a typical means of depicting objects in 3D without having to calculate true (human-type) perspective. It is also used in many world-building games, such as Blizzard's Warcraft 3 or StarCraft.

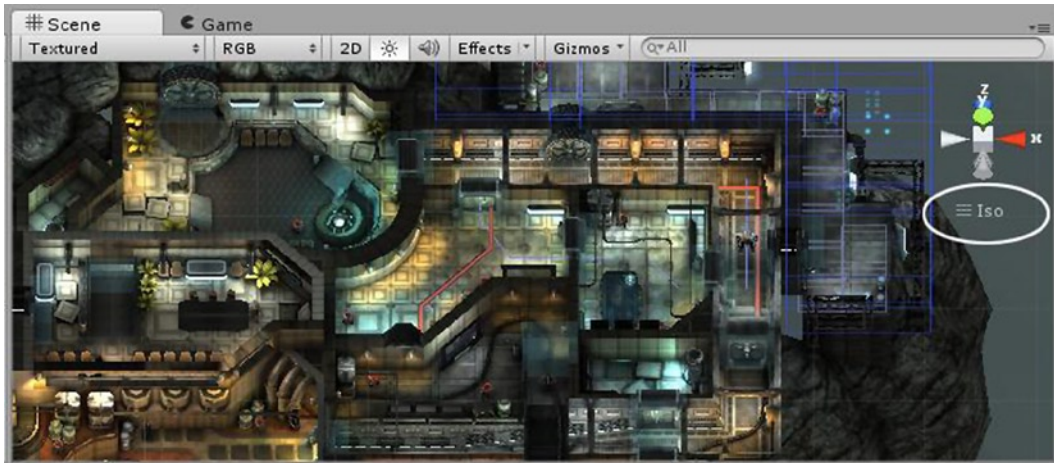


Figure 1-21. The Iso, or isometric view

7. Return to a perspective view by clicking the iso label.

The viewport and label are returned to a perspective view (Figure 1-22).

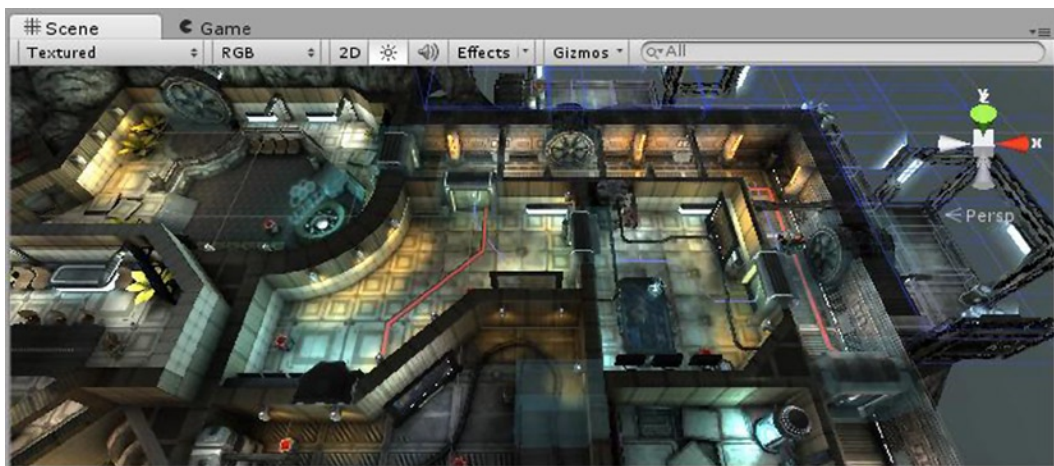


Figure 1-22. The Scene

Exploring the Views

Now that you are becoming more adept at manipulating the viewport, you can check out several of the tools and features available to you with each of the main views.

Hierarchy View

The Hierarchy view is essentially a list tree of the objects currently in the loaded or active scene. The key word here is “currently” because during run-time, objects can be “instantiated” (created on the fly) or destroyed (deleted from the scene). This is an important concept in Unity, and it is one of the reasons that most changes made during run-time are not permanent. On exiting Play mode, the Hierarchy view will be returned to its former state.

During your experimenting with the viewport, you may have accidentally clicked on an object in the Scene view. When selected that way, the hierarchy in the Hierarchy view will be expanded to show the picked object. As you may have noticed, some of the objects are children of children of the top-level objects in the scene. With this in mind, you can imagine how difficult it can be to locate particular objects in either the Scene or Hierarchy view. As the book’s project comes together, you will be importing and using several prefabs (Unity’s term for reference objects) and will want a quick means of locating them in the current scene.

Fortunately, Unity has a nice search function for that very purpose. At the top of the Hierarchy view, to the right of the Create drop-down arrow, you will see a small magnifying glass icon and its drop-down arrow. The drop-down menu will let you filter for All, Name, or Type (Figure 1-23).

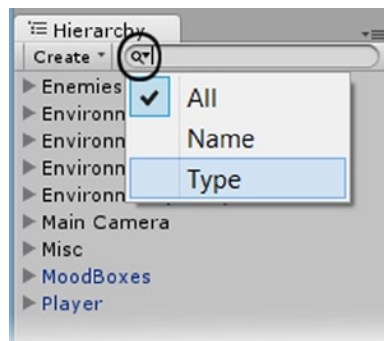


Figure 1-23. The Search feature’s filters

1. Press the drop-down arrow next to the search icon, and select Type.
2. In the text field to the right, type **camera**.

Before you can finish typing, Unity lists the cameras in the scene. In this case, there is only one, Main Camera. If you select it in the Hierarchy view, it is also selected in the Scene view, along with its view in the Camera Preview window. Note how everything else in the scene has been grayed out.

3. Return to a normal Scene view by clicking the x to the right of the word you typed in.

The Scene view is returned to normal, but the selected object remains highlighted in the Hierarchy view.

4. Next, change the filter back to All and type **confused**.
5. Select the only object that comes up, ConfusedEnemyMech.
6. Double-click to see the view focus in on it.
7. Click the x to return the view to normal.
8. This time, type in something more generic, **pipe**.

Because there are several objects in the scene with the word *pipe* in their name, the list of possible objects is quite long. In the Scene view, they are all “un-grayed,” helping you to find the one you want fairly quickly (Figure 1-24).



Figure 1-24. The options for a scene object with “pipe” in its name

9. Click the x to return the Scene view to normal.

Scene View

The Angry Bots scene provides you with plenty of variety to explore the Scene view’s tools and features. Because there are lots of options, you will just take a look at some of the more commonly used features.

1. Deselect any objects you’ve got selected by clicking in a blank area of either the Hierarchy or Scene view.
2. Zoom back far enough to see about half of the scene.
3. On the far left, just under the Scene tab, click on Textured to see the other viewport display options (Figure 1-25).

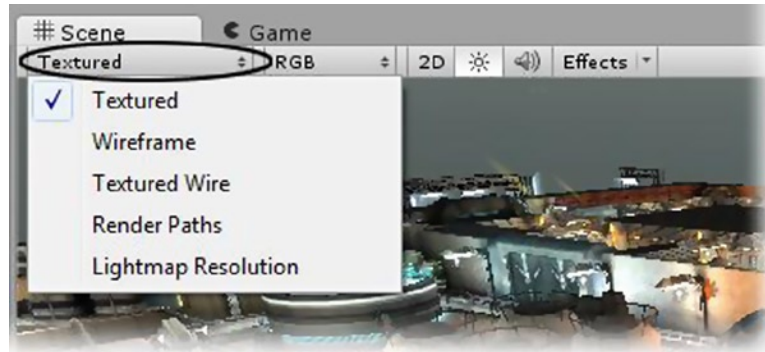


Figure 1-25. The viewport display options

Of the five, the first three should be easy to understand, especially if you have had any background in 3D modeling or have ever watched a “making of” video of any of the pioneers of the CG movie industry’s 3D animated films.

4. Try selecting Wireframe and then Textured Wire.

You may have noticed that meshes automatically show as textured wire when selected. The difference here is that the wire is black, not light blue, allowing you to continue to locate the selected item in the Scene view. In Wireframe, you can see how 3D objects came to be called “meshes,” as they resemble wire meshes.

The next option to the right shows RGB (Figure 1-26). This set of display options has to do mostly with texture maps. RGB, the default, shows the red, green, blue part of an RGBA texture.

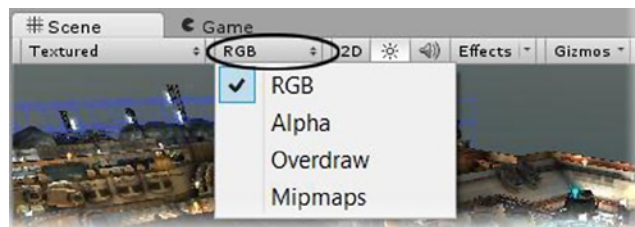


Figure 1-26. The scene view options

5. Select the Alpha option.

If you are familiar with the alpha channel of a 32-bit texture or image, you may be thinking that Alpha is about transparency, with 0 as fully transparent and 255 as fully opaque. In Unity, however, the alpha channel can do duty as anything from a glossiness channel to self-illumination to height maps. Each of Unity’s shaders is designed to make use of alpha channels of its textures to maximize effect and minimize texture usage. With the Alpha option selected for the viewport texture display, you will be able to see the alpha channel but not what it has been used for (Figure 1-27).



Figure 1-27. Textures displaying alpha channels

6. Select the Overdraw option.

Overdraw, the third option, shows how many objects must be drawn on top of each other from the current vantage point. The denser or brighter the display (Figure 1-28), the more objects there are that must at least be sorted, if not actually drawn. In mobile games, regardless of device and operating system, this type of sorting is very costly and should be kept to a minimum.

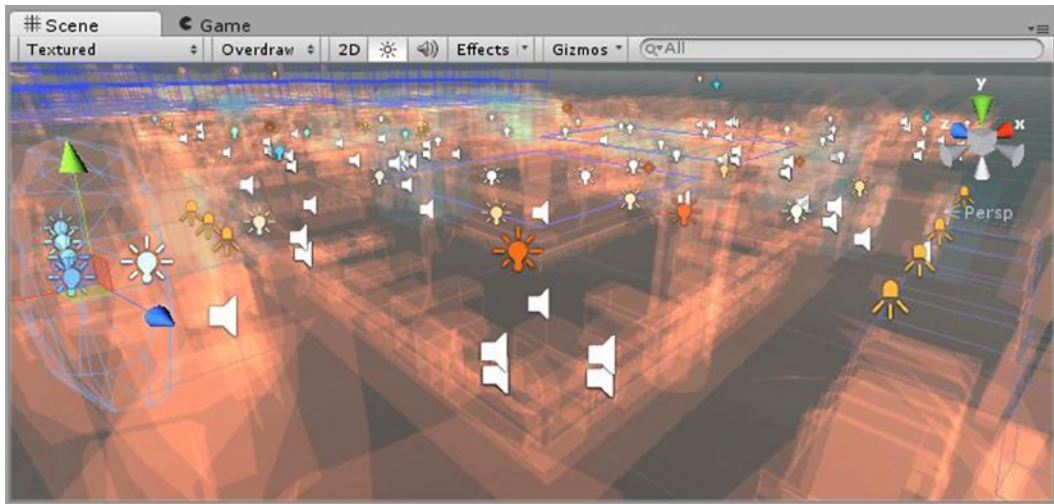


Figure 1-28. Possible overdraw problem areas showing as bright areas in the Scene view

7. Select the Mipmaps option.

The final option shows the MIP map distance, with no color being the full original-sized texture. As the tint goes from blue to clear to gold through to red, the smaller, blurrier maps are substituted, preventing the sparkly artifacting effect seen as the camera pulls back away from the object. Blue uses a blurred version of the map when the camera is close enough to see pixilation in the texture. You will learn more about MIP mapping when you work with imported images later on.

1. Zoom in close enough to the compound floor by using the mouse roller or by holding the Alt key, pressing the right mouse button, and moving the mouse to see the floor tinted blue at close range.
2. Zoom out slowly, and watch as the tint goes to no tint then to red (Figure 1-29).

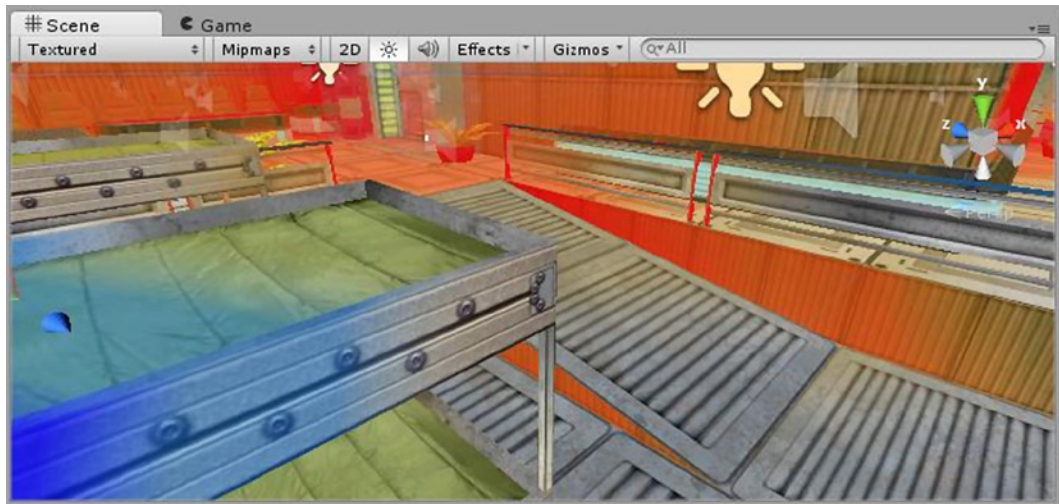


Figure 1-29. Color tinting indicating which version of MIP map is shown according to the distance

An absence of tint is the native image. Blue is a blurry version of it you can use so that you won't see each and every pixel up close. Red is a blurred version of it you can use so that you won't see the sparkly artifacting common to low-resolution videos. To see the MIP maps generated from the imported textures, you can select a texture and view it in the Inspector.

3. In the Project view, select the Textures folder from the Assets folder.
4. Locate and select the AngryBots texture in the second column.
5. At the bottom of the Inspector, in the Preview window, move the slider from the far right, slowly over to the far left.

The images are shown in increasingly blurry versions. They appear pixilated as Unity zooms in closer and closer and the versions are smaller and smaller.

6. Set the Scene view drop-down menu back to its default RGB.