Mike Geig

In Full Color

Fourth Edition

Sams Teach Yourself

Unity® Game Development

in 24 Hours

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Mike Geig

Sams **Teach Yourself**

Unity Game Development in 24 Hours

SAMS

Sams Teach Yourself Unity Game Development in 24 Hours

ISBN-13: 978-0-13-744508-0
ISBN-10: 0-13-744508-3

Library of Congress Control Number: 2021943446

ScoutAutomatedPrintCode

Trademarks

Warning and Disclaimer

Special Sales

Editor-in-Chief
Mark Taub

Signing Editor
Malobika Chakraborty

Development Editor
Chris Zahn

Managing Editor
Sandra Schroeder

Senior Project Editor
Tonya Simpson

Copy Editor
Kitty Wilson

Indexer
Timothy Wright

Proofreader
Abigail Manheim

Technical Editor
Rocio Chongtay

Editorial Assistant
Cindy Teeters

Cover Designer
Chuti Prasertsith

Compositor
codeMantra

Contents at a Glance

Table of Contents

Preface

The Unity game engine is an incredibly powerful and popular choice for professional and amateur game developers alike. This book has been written to get readers up to speed and working in Unity as fast as possible (in about 24 hours, to be exact) while covering fundamental principles of game development. Unlike other books that cover only specific topics or spend the entire time teaching a single game, this book covers a large array of topics while still managing to contain four games. Talk about a bargain! By the time you are done reading this book, you won't have just theoretical knowledge of the Unity game engine. You will have a portfolio of games to go with it.

You can find all the code assets used in this book on GitHub at http://fixbyproximity.com/Downloads/UnityBook.html.

About the Author

Mike Geig works on the production team at Unity Technologies, where he helps democratize game development by developing and delivering high-impact learning resources. Mike has experience as an indie game developer, a university educator, and an author. A gamer at heart, Mike works to make the development of interactive entertainment fun and accessible for all skill sets. Hi, Mom!

Acknowledgments

A big “thank you” goes out to everyone who helped me write this book.

This edition comes at a very strange time in my life. Thank you to everyone who helps make it the good kind of strange.

Link and Luke: You continue to be the tornado that carries me to Oz…while also destroying most of Kansas.

Thanks to my parents. As I am now a parent myself, I recognize how hard it was for you not to strangle or stab me. Thanks for not strangling or stabbing me.

Thanks to Angelina Jolie. Due to your role in the spectacular movie *Hackers* (1995), I decided to learn how to use a computer. You underestimate the impact you had on 10-year-olds at the time. You’re elite!

To the inventor of beef jerky: History may have forgotten your name, but definitely not your product. I love that stuff. Thanks!

Thank you to Rocio Chongtay for being an awesome technical editor for this book. Also, thanks for reaching out and asking me when I am going to update this book. “Now.”

Thank you to Chris Zahn for helping with all of the formatting and structural edits. If the readers find any problems in this book, I will tell them to blame you!

Thank you, Laura, for convincing me to write this book. Also thank you for buying me lunch at GDC. I feel that lunch, the best of all three meals, specifically enabled me to finish this.

Finally, a “thank you” is in order for Unity Technologies. If you never made the Unity game engine, this book would be very weird and confusing.

Dedication

To all the people who read the dedications page. Nothing can stop you now.

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: community@informit.com

Reader Services

HOUR 1
Introduction to Unity

What You'll Learn in This Hour:

- How to install Unity
- How to create a new project or open an existing project
- How to use the Unity editor
- How to navigate inside the Unity Scene view

This hour focuses on getting you ready to rock and roll in the Unity environment. It starts by looking at the different Unity licenses and installing the one you choose. This hour you'll also learn how to create new projects as well as open existing ones. You'll open the powerful Unity editor and examine its various components. Finally, you'll learn to navigate a scene by using mouse controls and keyboard commands. This lesson is meant to be hands-on, so download Unity while reading and follow along.

Installing Unity

Before you can begin using Unity, you first need to download and install it. Software installation is a pretty simple and straightforward process these days, and Unity is no exception. Before you can install anything, though, you need to look at the three available Unity license options: Unity Personal, Unity Plus, and Unity Pro. Unity Personal is free and has everything you need to complete all the examples and projects in this book. In fact, Unity Personal contains everything you need to make games commercially, up to an annual revenue of $100,000! If you're lucky enough to start earning more than this, or if you want to access the advanced features of Unity Plus or Unity Pro (mainly aimed at teams), then you can always upgrade in the future.

NOTE

Unity Hub

From the Unity website you need to download and install the Unity Hub. The Hub is a launcher that acts as a centralized…well…hub for all your Unity editor installations and projects. (The Unity editor is the software that enables you to make games.) If you're coming to this book after having used a previous version of Unity without the Hub, don't worry. The concepts behind Unity installations and projects are exactly the same as they've been in the past. As the saying goes, "new packaging, same great taste!"

NOTE

Unity Updates

The images and instructions in this book are accurate as of this writing. That being said, the Unity Hub and editor are living and constantly evolving pieces of software. As such, new versions may come out, and the images or text you see here might not match what you see onscreen. Although the fine specifics of the software may change, the general steps and concepts outlined in this book will still be accurate and relevant!

Downloading and Installing Unity Hub

As mentioned earlier in this lesson, when you want to program games using Unity, the Unity Hub is your starting point. When you are ready to begin downloading and installing Unity Hub, follow these steps:

1. Go to the Unity Store website, at https://store.unity.com, and choose your license type.
2. If you chose the Unity Personal edition, you have the option to download the Unity Hub directly or to go through a guided installation experience aimed at assisting brand-new users. Either way will get you where you need to be.
3. Run the installer and follow the prompts as you would with any other piece of software.
4. Open up the Unity Hub application (see Figure 1.1). You may be prompted to sign in or create a new account. Doing so takes only a moment, and you will need an account later, so go ahead and do it now.

In addition to being the place where you manage all projects and installations, the Unity Hub is also a place where you can find community news and learning resources to assist with your development journey.

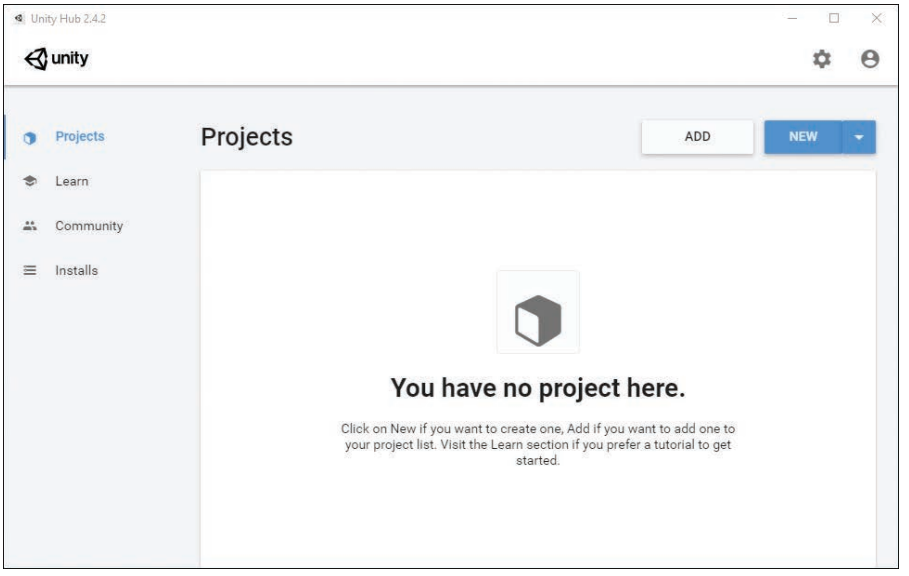

FIGURE 1.1
The Unity Hub.

NOTE

Internet Links

All Internet URLs in this book are current as of the time this book was published. Web locations do change sometimes, though. If the material you are looking for is no longer provided at the links listed, a good Internet search should turn up what you are looking for.

Installing the Unity Editor

Now that the hub is installed, it is time to install the Unity editor. Using the Hub, you can have as many versions of Unity installed as you'd like (and that your hard drive allows). To install the Unity editor, follow these steps:

1. In the Hub, click **Installs** and then click the **ADD** button.
2. Select the 2020 LTS release (see Figure 1.2). (The note "Why 2020 LTS?" explains why you want this version.) Click **Next**.

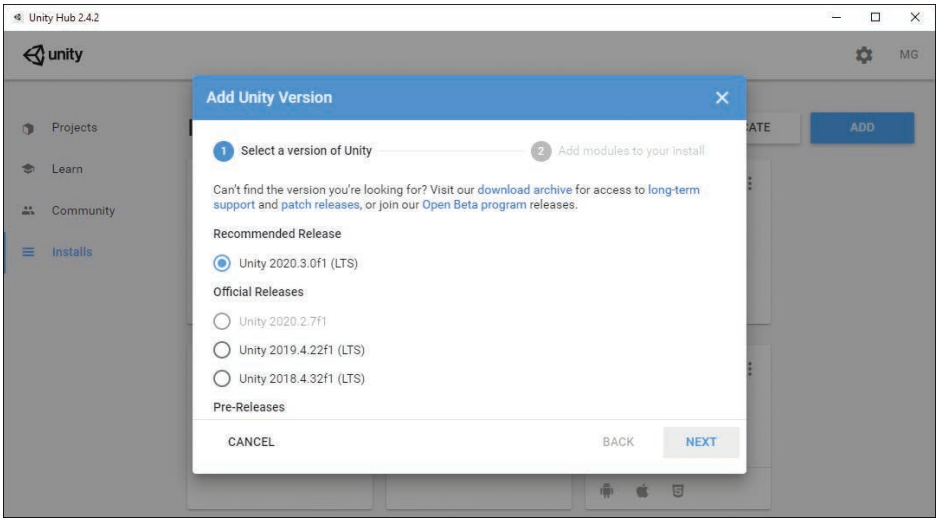

FIGURE 1.2
Selecting the 2020 LTS release of Unity.

3. In the next window, notice that you can select any add-ons or build platforms you'd like to support (see Figure 1.3). You can add platforms now, but you don't have to select any of them in order to continue with this book. You can also come back here later and add new items. So for now, just click **Done**.

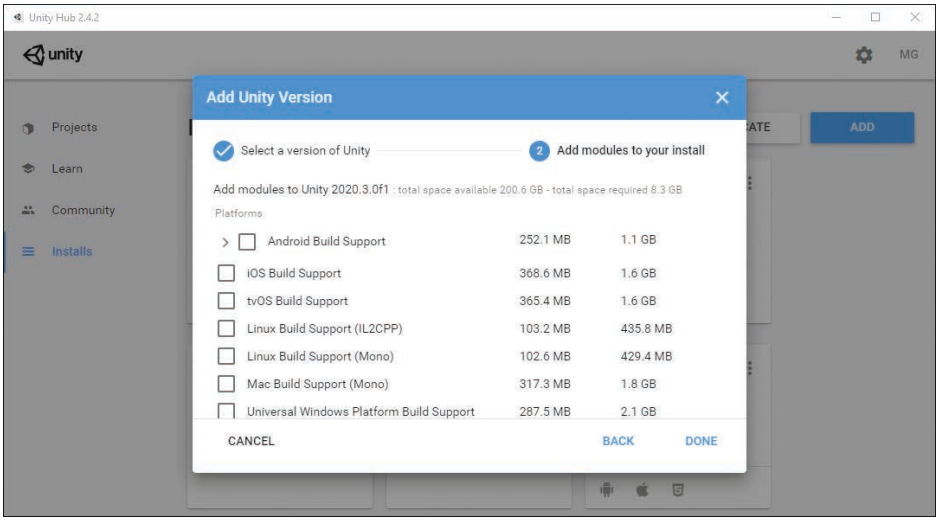

FIGURE 1.3
Additional installation modules.

4. Now you just need to wait for the installation to complete. When it is done, you will be good to go!

NOTE

Why 2020 LTS?

You might be wondering why you installed a 2020 version of the software and what LTS means. With Unity releases, the LTS (Long Term Support) version is the version that combines all previous features, is fully stable, and will be supported for at least the next two years. The LTS version of the engine usually doesn't come out until the beginning of the next year, so Unity 2020 LTS came out at the beginning of 2021 and was the newest version when this book was published.

NOTE

Supported Operating Systems and Hardware

To use Unity, you must be using a Windows PC or a Mac computer. Although there is a version of the editor that runs on Linux machines, Linux is not a directly supported OS. Your computer must also meet the minimum requirements outlined here (taken from the Unity website at the time this book was published):

- Windows 7 SP1+, Windows 8, or Windows 10, 64-bit versions only; OS X 10.12.6+. Note that Unity was not tested on server versions of Windows and OS X.
- Graphics card with DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.
- A CPU that supports the SSE2 instruction set (most modern CPUs).

Note that these are *minimum* requirements.

Getting to Know the Unity Editor

Now that you have Unity installed, you can begin exploring the Unity editor. The Unity editor is the visual component that enables you to build games in a "what you see is what you get" fashion. Because most interaction you have is actually with the editor, many people refer to it as simply *Unity*. This section examines the various elements of the Unity editor and how they fit together to make games.

The Projects Section

Not only is the Unity Hub where you manage your editor installations, it is where you create and select projects. At this point, the Projects section of the Hub is probably blank for you (see Figure 1.4).

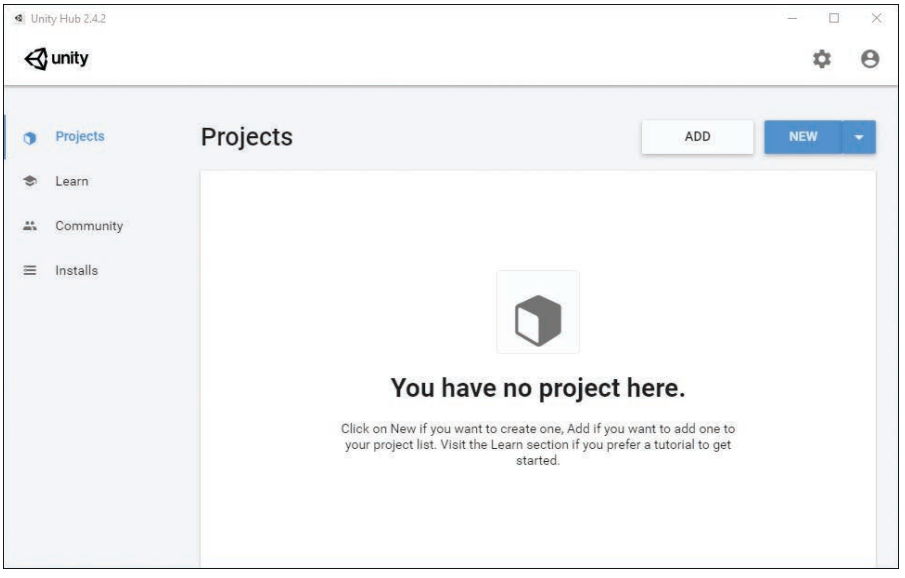

FIGURE 1.4
The empty Projects section of the Hub.

In order to create a new project, all you have to do is click **New**. If you have multiple editor versions installed and you'd like to pick the version, you can do so by clicking the downward arrow next to the New button. Finally, if you want to open a project you've already created (and that isn't in the project list already), you can do so by clicking **ADD**. In the following exercise, you will see exactly how to create a new project.

TRY IT YOURSELF

Creating Your First Project

You are ready to create a project. Pay special attention to where you save the project so that you can easily find it later. Figure 1.5 shows the dialog box you use to create a project. Follow these steps:

1. Open the Unity Hub and click **NEW**. The New Project dialog pops up.
2. Select a location for your project. I recommend that you create a folder called Unity Projects to keep all your book projects together. If you are unsure where to put your project, you can leave the default location.
3. Name your project **Hour 1 TIY**. Unity creates a folder with the same name as the project, in the location specified in this dialog.

4. Leave **3D** selected for now. You'll learn about some of the other options in the future.
5. Click **CREATE**.

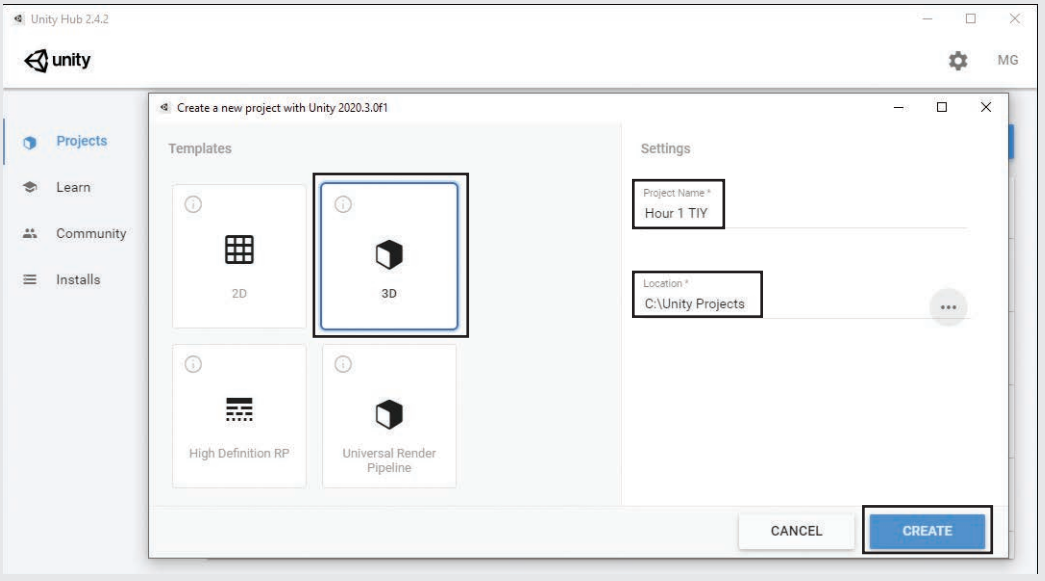

FIGURE 1.5
The settings for your first project.

NOTE

2D, 3D, Rendering Pipelines?

You might be wondering what the other options on the New Project dialog are for. The 2D and 3D buttons allow you to select which type of game you are planning on making. Don't worry about picking the wrong one or being unsure. These options just control editor settings and can be changed at any time. The High Definition RP and Universal Render Pipeline options are a part of Unity's Scriptable Rendering Pipeline, which gives developers a lot of control over exactly how their projects render but requires knowledge that is beyond the scope of this lesson.

The Unity Interface

So far, you have installed Unity and looked at the New Project dialog. Now it is time to dig in and start playing around. When you open a new Unity project for the first time, you see a collection of gray windows (called *views*), and everything is rather empty (see Figure 1.6). Never fear; you will quickly get this place hopping. The following sections look at each of the unique views, one by one. First, though, let's talk about the layout.

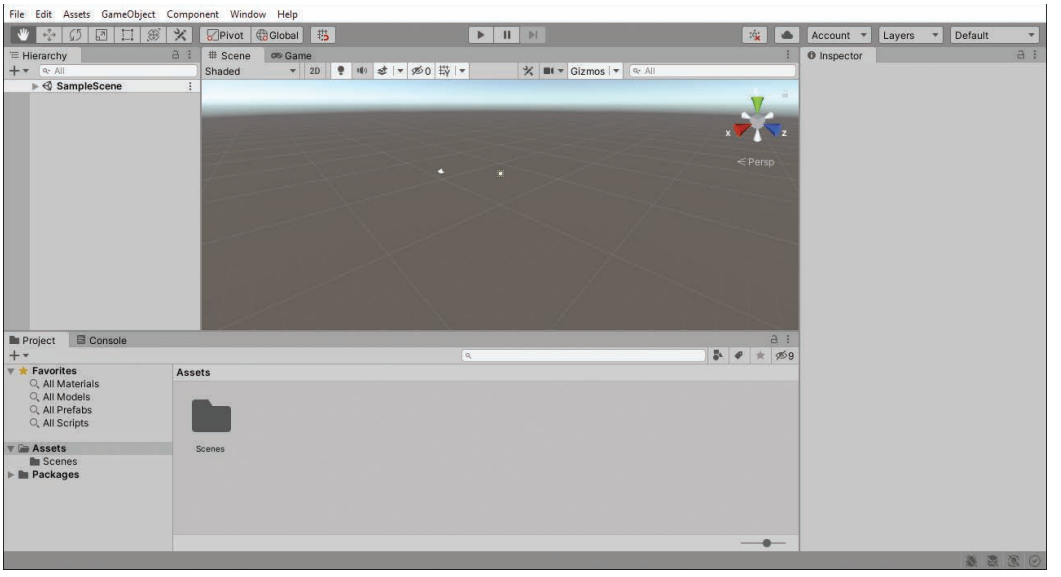

FIGURE 1.6
The Unity interface.

For starters, Unity allows you to determine exactly how you want to work. Any of the views can be moved, docked, duplicated, or changed. For instance, if you click the word **Hierarchy** (on the left) to select the Hierarchy view and drag it over to the Inspector (on the right), you can tab the two views together. You can also place your cursor on any line between views and resize the windows. In fact, why don't you take a moment to play around and move things so that they are to your liking? If you end up with a layout that you don't much care for, you can quickly and easily switch back to the built-in default view by going to **Window** > **Layouts** > **Default Layout**. While you are playing around, go ahead and try out a few of the other layouts. (I'm a fan of the Wide layout.) If you create a custom layout you like, you can save it by going to **Window** > **Layouts** > **Save Layout**. (I used a custom layout called Pearson for the writing of this book.) After you've saved a custom layout, if you accidentally change the layout, you can always get it back. It is worth noting that you can also control the layouts through the Layouts drop-down, located in the upper right of the Unity editor; it is the drop-down that says Default in Figure 1.6.

NOTE

Finding the Right Layout

No two people are alike, and likewise, no two ideal layouts are alike. A good layout will help you work on your projects and make things much easier for you. Be sure to take the time to fiddle around with the layouts to find the one that works best for you. You will be working with Unity a lot, and it pays to set up your environment in a way that is comfortable for you.

Duplicating a view is a fairly straightforward process as well. You can simply right-click any view *tab* (such as Inspector in Figure 1.7) and hover the mouse cursor over **Add Tab**, and a list of views pops up for you to choose from (see Figure 1.7). You may wonder why you would want to duplicate a view. Say that in a view-moving frenzy, you accidentally close a view. Re-adding the tab will give it back to you. Also, consider the capability to create multiple Scene views. Each Scene view could align with a specific element or axis within your project. If you want to see this in action, check out the 4 Split built-in layout by going to **Window > Layouts > 4 Split**. (If you have already created a layout that you like, be sure to save it before you check out 4 Split.)

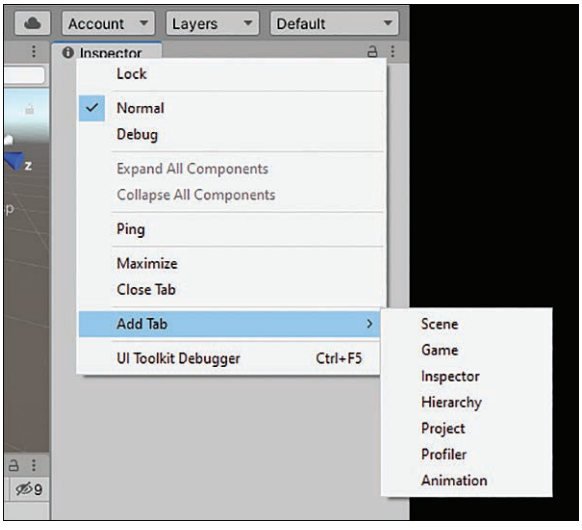

FIGURE 1.7
Adding a new tab.

Now, without further ado, let's look at the specific views in the Unity editor.

The Project View

Everything that has been created for a project (files, scripts, textures, models, and so on) can be found in the Project view (see Figure 1.8). This is the window that shows all the assets and organization of a project. When you create a new project, you see a folder section called Assets. If you go to the folder on your hard drive where you save the project, you also find an Assets folder. This is because Unity mirrors the Project view with the folders on the hard drive. If you create a file or folder in Unity, the corresponding file or folder appears in the explorer (and vice versa). You can move items in the Project view simply by dragging and dropping. Unity enables you to place items inside folders or reorganize your project on the fly.

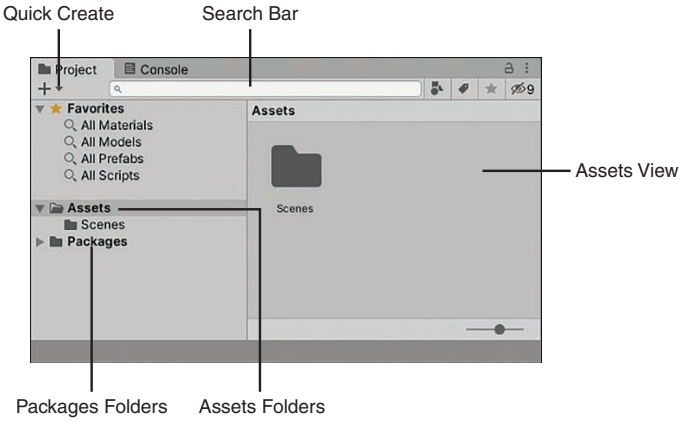

FIGURE 1.8
The Project view.

NOTE

Assets and Objects

An *asset* is any item that exists as a file in your Assets folder. All textures, meshes, sound files, scripts, and so on are considered assets. In contrast, a *game object* is an object that is part of a scene or a level. You can create assets from game objects, and you can create game objects from assets.

NOTE

Packages

When looking in the folder for a project, you can see the Assets folder just as it appears in the editor. You may also notice, however, that there isn't a Packages folder, even though one appears in the Project view in the editor. This is a special section where you can view the packages that have been added to a project. You can think of a package as a type of add-on or module that can change the functionality of a project. Unity has made many of them available, and you can even make your own. You will learn more about packages later, as they become important.

CAUTION

Moving Assets

Unity maintains links between the various assets associated with projects. As a result, moving or deleting items outside Unity could potentially cause problems. As a general rule, it is a good idea to do all your asset management inside Unity.

When you click a folder in the Project view, the contents of the folder are displayed under the Assets section on the right. As you can see in Figure 1.8, the Assets section contains a single

folder, named Scenes. If you open the Scenes folder, you see the contents of the folder—a single scene—listed on the right. To create assets, you simply click the **Create** drop-down (that is, the + drop-down). This menu enables you to add all manner of assets and folders to a project.

TIP

Project Organization

Organization is extremely important in project management. As your projects get bigger, the number of assets will grow, and eventually finding anything will be a chore. You can prevent a lot of frustration by employing some simple organization rules:

- Every asset type (scenes, scripts, textures, and so on) should get its own folder.
- Every asset should be in a folder.
- If you use a folder inside another folder, make sure the structure makes sense. Folders should become more specific and should not be vague or generalized.

Following these few simple rules will really make a difference.

The Favorites buttons enable you to quickly select all assets of a certain type. This makes it possible to get an "at a glance" view of your assets. When you click one of the Favorites buttons (All Models, for instance) or perform a search with the built-in search bar, you can narrow down the results between assets and packages (or both; see Figure 1.9). With a little practice, finding exactly what you need will become a breeze!

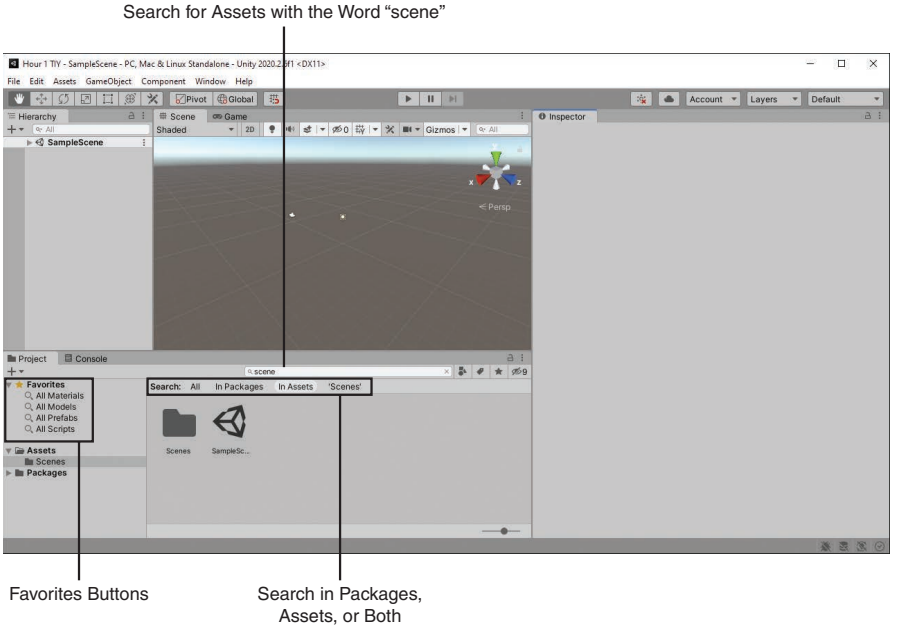

FIGURE 1.9
Searching the Project view.

The Hierarchy View

In many ways, the Hierarchy view (see Figure 1.10) is a lot like the Project view. The difference is that the Hierarchy view shows all the items in the current scene instead of in the entire project. When you first create a project with Unity, you get the default scene, which has just two items in it: the Main Camera and Directional Light game objects. As you add items to a scene, they appear in the Hierarchy view. Just as with the Project view, you can use the **Create** menu to quickly add items to your scene, search using the built-in search bar, and click and drag items to organize and nest them.

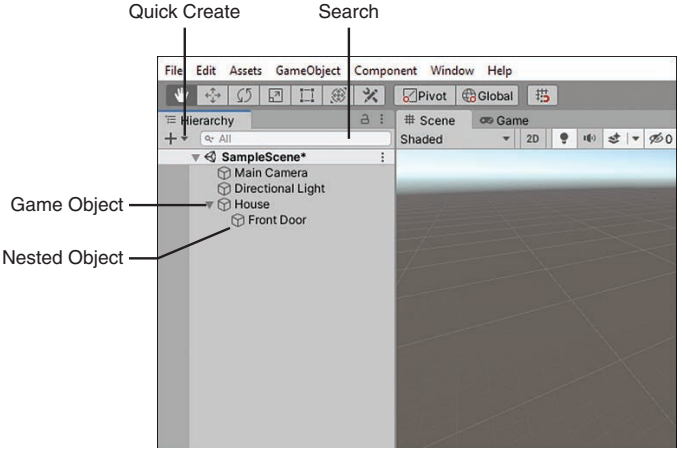

FIGURE 1.10
The Hierarchy view.

TIP

Nesting

Nesting is a way of establishing a relationship between two or more items. In the Hierarchy view, clicking and dragging an item onto another item nests the dragged item under the other one. This creates what is commonly known as a *parent/child relationship*. In this case, the object on top is the parent, and any objects below it are children. You can tell when an object is nested because it becomes indented. As you will see later, nesting objects in the Hierarchy view can affect how they behave.

TIP

Scenes

Scene is the term Unity uses to describe what you might already know as a level or map. As you develop a Unity project, each collection of objects and behaviors should be its own scene. For

example, if you were building a game with a snow level and a jungle level, those would be separate scenes. You will see the words *scene* and *level* used interchangeably as you look for answers on the Internet.

The Inspector View

The Inspector view enables you to see all the properties of a currently selected item. Simply click any asset or object in the Project or Hierarchy view, and the Inspector view is automatically populated with information.

In Figure 1.11, you can see the Inspector view after the Main Camera object is selected from the Hierarchy view.

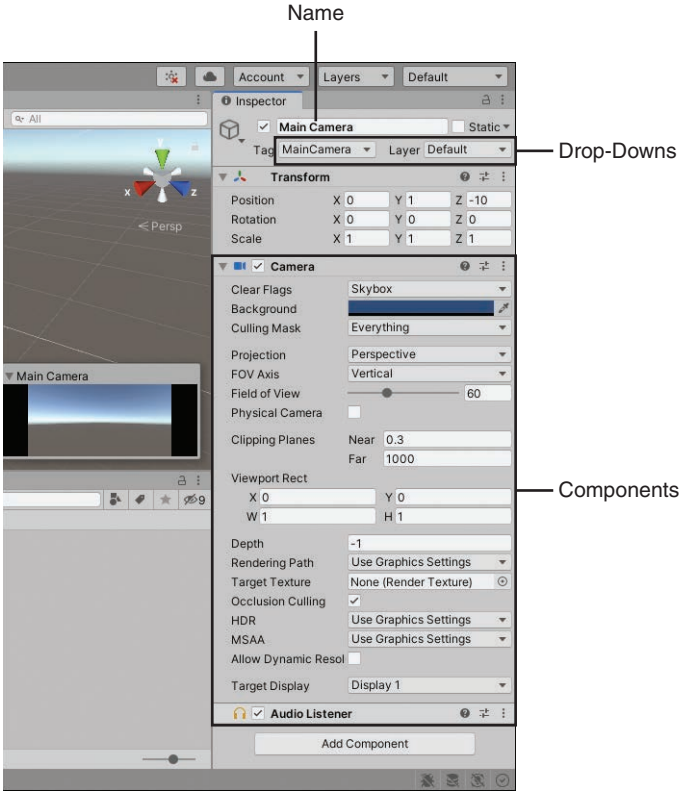

FIGURE 1.11
The Inspector view.

Let's break down some of the functionality available in the Inspector view:

- Unchecking the box next to an object's name disables it and ensures that it does not appear in the scene. Objects are enabled (that is, their boxes are checked) by default.

- Drop-down lists (such as the Layer and Tag lists, which are discussed later) are used to select from a set of predefined options.
- Text boxes, drop-downs, and sliders can have their values changed, and the changes are automatically and immediately reflected in the scene—even if the game is running!
- Each game object acts like a container for different components (such as Camera and Audio Listener in Figure 1.11). You can disable these components by unchecking them or remove them by right-clicking and selecting **Remove Component**.
- You can add components by clicking the **Add Component** button.

CAUTION

Changing Properties While Running a Scene

The capability to change the properties of an object and see those changes reflected immediately in a running scene is very powerful. It enables you to tweak things like movement speed, jumping height, and collision power on the fly, without stopping and starting the game. Keep in mind, though, that any changes you make to the properties of an object while a scene is running are reverted when the scene finishes. If you make a change and like the result, be sure to remember what it was so you can set it again when the scene is stopped.

The Scene View

The Scene view is the most important view you work with in Unity because it enables you to see your game visually as it is being built (see Figure 1.12). Using the mouse controls and a few hotkeys, you can move around inside a scene and place objects where you want them. This gives you an immense level of control.

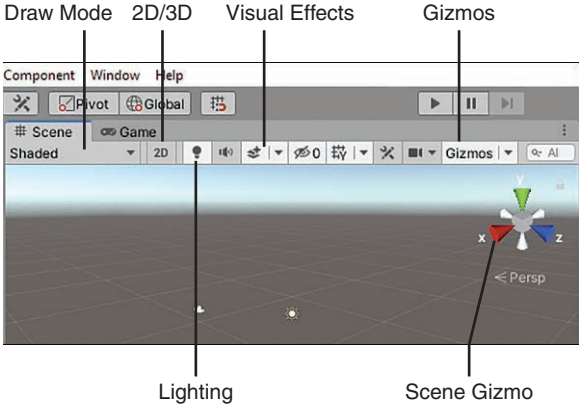

FIGURE 1.12
The Scene view.

In a little bit, you will learn about moving around within a scene, but for now, let's focus on the controls that are part of the Scene view:

- **Draw mode:** This controls how the scene is drawn. By default, it is set to Shaded, which means objects will be drawn with their textures in full color.
- **2D/3D view:** This control changes from a 3D view to a 2D view. Note that in 2D view, the scene gizmo (described later this hour) is not visible.
- **Scene lighting:** This control determines whether objects in the Scene view are lit by default ambient lighting or only by lights that actually exist within the scene. The default is to include the built-in ambient lighting.
- **Audition mode:** This control sets whether an audio source in the Scene view functions.
- **Visual effects:** This control determines whether items like skyboxes, fog, and other effects appear in the Scene view.
- **Hidden objects:** This control toggles the visibility of hidden objects in the Scene view.
- **Scene grid:** This control allows you to configure the Scene grid.
- **Tools:** This control toggles the Component Editor tool panel.
- **Scene camera:** This control allows you to configure the Scene camera (the camera used to view the scene while editing).
- **Gizmo selector:** This control enables you to choose which *gizmos*—that is, indicators that help with visual debugging or aid in setup—appear in the Scene view. This control also determines whether the placement grid is visible.
- **Scene gizmo:** This control shows which direction you are currently facing and aligns the Scene view with an axis.

NOTE

The Scene Gizmo

The scene gizmo gives you a lot of power over the Scene view. As you can see in Figure 1.12, the control has X, Y, and Z indicators that align with the three axes. This makes it easy to tell exactly which way you are looking in the scene. You will learn more about axes and 3D space in Hour 2, "Game Objects." The scene gizmo also gives you active control over the scene alignment. If you click one of the gizmo's axes, the Scene view immediately snaps to that axis and gets set to a direction such as top or left. Clicking the box in the center of the gizmo toggles between Iso and Persp modes.

Iso, which stands for *isometric*, is the 3D view with no perspective applied. *Persp,* which stands for *perspective*, is the 3D view with perspective applied. Try out these settings for yourself and see how they affect the Scene view. You'll notice that the icon changes from parallel lines for isometric to diverging lines like crow's feet for perspective.

The Game View

The last view we need to go over is the Game view. Essentially, the Game view allows you to "play" the game inside the editor by giving you a full simulation of the current scene. All elements of a game function in the Game view just as they would if the project were fully built. Figure 1.13 shows what the Game view looks like. Note that although the Play, Pause, and Step buttons are not technically part of the Game view, they control the Game view and therefore are included in this figure.

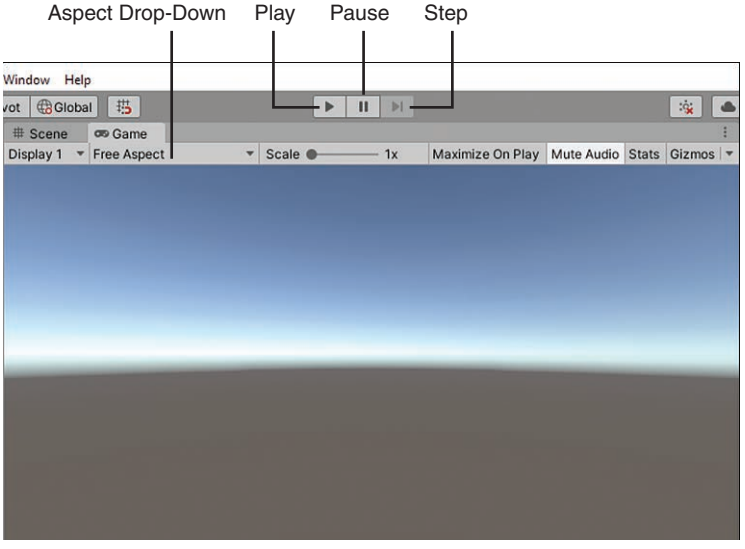

FIGURE 1.13
The Game view.

TIP

Missing Game View

If you find that the Game view is hidden behind the Scene view or that the Game view tab is missing entirely, don't worry. As soon as you click the **Play** button, a Game view tab appears in the editor and begins displaying the game.

The Game view comes with some controls that assist with testing games:

- **Play:** The Play button enables you to play the current scene. All controls, animations, sounds, and effects are present and working. Once a game is running, it should behave very similarly to how it would behave if it were actually being run in a standalone player (such as on your PC or mobile device). To stop a game from running, click the **Play** button again.

- **Pause:** The Pause button pauses the execution of the currently running Game view. When a game is paused, it maintains its state and stays exactly where it was when the Pause button was clicked. Clicking the Pause button again causes the game to continue running.
- **Step:** The Step button works while the Game view is paused and causes a single frame of the game to execute. This effectively allows you to "step" through the game slowly and debug any issues that exist. Clicking the Step button while the game is running causes the game to pause.
- **Aspect drop-down:** From this drop-down menu, you can choose the aspect ratio you want for the Game view window while running. The default is Free Aspect, but you can change it to match the aspect ratio of the target platform you are developing for.
- **Maximize on Play:** This button determines whether the Game view takes up the entirety of the editor when run. By default, this option is turned off, and a running game is only the size of the Game view tab.
- **Mute Audio:** This button turns off the sounds when playing the game. This is handy when the person sitting next to you is getting tired of hearing your repeated playtesting!
- **Stats:** This button determines whether rendering statistics are displayed on the screen while the game is running. These statistics can be useful for measuring the efficiency of a scene. The stats are turned off by default.
- **Gizmos:** This is both a button and a drop-down menu. The button determines whether gizmos are displayed while the game is running. Game view gizmos are not displayed by default. The drop-down menu (the small arrow) on this button determines which gizmos appear if gizmos are turned on.

NOTE

Running, Paused, and Off

It can be difficult at first to determine what is meant by the terms *running*, *paused*, and *off*. When a game is not executing in the Game view, the game is said to be *off*. When a game is off, the game controls do not work, and the game cannot be played. When the Play button is pressed and the game begins executing, the game is said to be *running*. *Playing*, *executing*, and *running* all mean the same thing. If a game is running and the Pause button is pressed, the game stops running but maintains its state. At this point, the game is paused. The difference between a paused game and an off game is that a paused game resumes execution at the point at which it was paused, whereas an off game begins executing at the beginning.

Honorable Mention: The Toolbar

Although not a view, the toolbar is an essential part of the Unity editor. Figure 1.14 shows the toolbar components:

- **Transform tools:** These buttons enable you to manipulate game objects and are covered in greater detail in later hours. Pay special attention to the button that resembles a hand. This is the Hand tool, and it is described later this hour.
- **Transform gizmo toggles:** These toggles allow you to manipulate how gizmos appear in the Scene view. You can leave them alone for now.
- **Game view controls:** These buttons control the Game view.
- **Account and Services controls:** These buttons allow you to manage the Unity account you are using as well as the services you are using in your project.
- **Layers drop-down:** This menu determines which object layers appear in the Scene view. By default, everything appears in the Scene view. Leave this alone for now. Layers are covered in Hour 5, "Lights and Cameras."
- **Layout drop-down:** This menu allows you to quickly change the layout of the editor.

FIGURE 1.14
The toolbar.

Navigating the Unity Scene View

The Scene view gives you a lot of control over the construction of a game. The ability to place and modify items visually is very powerful. None of this is very useful, though, if you cannot move around inside the scene. This section covers a couple different ways to change your position and navigate the Scene view.

TIP

Zoom

Regardless of what method you are using for navigation, scrolling the mouse wheel always zooms the view within a scene. By default, the scene zooms in and out of the center of the Scene view. If you press **Alt** while scrolling, however, you zoom in and out of wherever the mouse is currently pointing. Go ahead and give it a try!

The Hand Tool

The Hand tool (hotkey: **Q**) provides a simple way to move about the Scene view with the mouse (see Figure 1.15). This tool is especially useful if you are using a mouse with only a single button (because other methods require a two-button mouse). Table 1.1 briefly explains each of the Hand tool controls. (Don't worry about the other buttons next to the Hand tool yet. They are covered a little bit later.)

FIGURE 1.15
The Hand tool.

TABLE 1.1 The Hand Tool Controls

Action	Effect
Click-drag	Drags the camera around the scene
Hold **Alt** and **click-drag**	Orbits the camera around the current pivot point
Hold **Alt** (**Command** on Mac) and **right-click-drag**	Zooms the camera

You can find information on all the Unity hotkeys at https://docs.unity3d.com/Manual/SceneViewNavigation.html.

CAUTION

Different Cameras

When working in Unity, you deal with two types of cameras. The first type is the standard game object camera. You can see that you already have one in your scene (by default). The second type is more of an imaginary camera than a camera in the traditional sense. It determines what you can see in the Scene view. When this hour's instructions mention the camera, they are talking about the second type. You do not actually manipulate the game object camera.

Flythrough Mode

Flythrough mode enables you to move about a scene using a traditional first-person control scheme. This mode feels like home for anyone who plays first-person games (such as first-person shooters). If you don't play those games, this mode might take a little getting used to. Once you become familiar with it, though, it will be second nature.

Holding down the right mouse button while your mouse cursor is over the Scene view puts you into Flythrough mode. All the actions laid out in Table 1.2 require that the right mouse button be held down.

TABLE 1.2 Flythrough Mode Controls

Action	Effect
Move the mouse	Causes the camera to pivot, which gives the feeling of "looking around" within the scene.
Press the **WASD** keys	The WASD keys move you about the scene. Each key corresponds with a direction: forward, left, back, and right, respectively.
Press the **QE** keys	The QE keys move you up and down, respectively, within the scene.
Hold **Shift** while pressing the **WASD** or **QE** keys	Has the same effect as using the WASD or QE keys but much faster. Consider Shift to be your "sprint" button.

TIP

Snap Controls

You have many ways to attain precise control over scene navigation. Sometimes, you just want to quickly get around a scene, though. For times like these, it is good to use what I call *snap controls*. If you want to quickly navigate to, and zoom in on, a game object in a scene, you can do so by highlighting the object in the Hierarchy view and pressing **F** (which stands for *Frame Select*). The scene then "snaps" to that game object. You can also achieve the same effect by double-clicking any object in the Hierarchy view. Another snap control is one you have seen already: The scene gizmo allows you to quickly snap the camera to any axis. This way, you can see an object from any angle without having to manually move the scene camera around. Be sure to practice using the snap controls, and navigating through your scene quickly will become a snap!

TIP

Additional Learning

When using the Unity Hub, your eyes may have magically been draw to the Learn section (see Figure 1.16). You can click the **Learn** button to see Unity's new learning resources. These resources are a fantastic augmentation and are well worth your time if you'd like even more practice getting to know the basics of the Unity engine. (I am a little biased: I helped make them.)

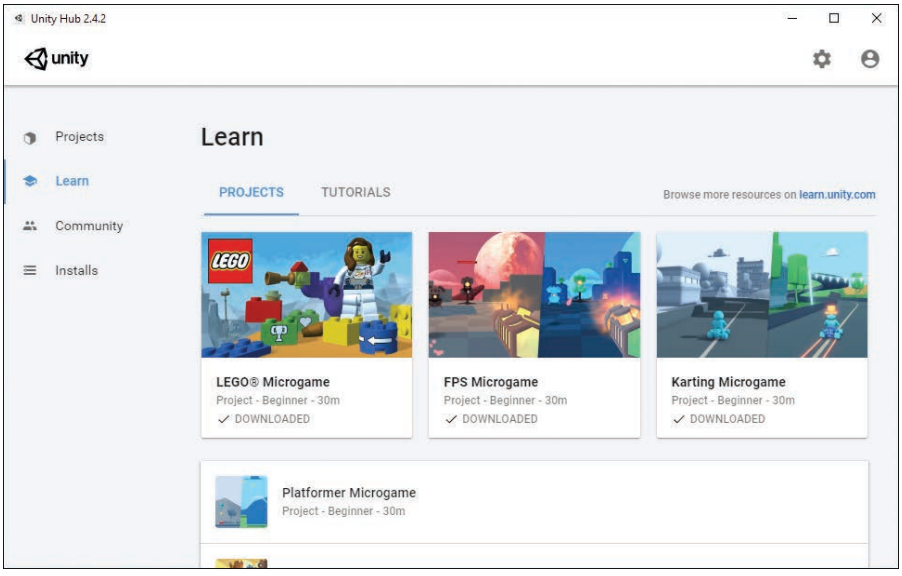

FIGURE 1.16
The Learn section of the Unity Hub.

Summary

In this hour, you took your first look at the Unity game engine. You started by downloading and installing Unity. From there, you learned how to open and create projects. Then you learned about all the different views that make up the Unity editor. You also learned how to navigate around the Scene view.

Q&A

Q. **Are assets and game objects the same?**

A. Not exactly. The big difference is that an asset has a corresponding file or group of files on the hard drive, whereas a game object does not. An asset may or may not contain a game object.

Q. **There are a lot of different controls and options. Do I need to memorize them all right away?**

A. Not at all. Most controls and options are already set to default states that cover most situations. As your knowledge of Unity grows, you will continue to learn more about the different controls available to you. This lesson is just meant to show you what's there and to give you a bit of familiarity.

Workshop

Take some time to work through the questions here to ensure that you have a firm grasp of the material.

Quiz

1. True or false: You must purchase Unity Pro to make commercial games.
2. Which view enables you to manipulate objects in a scene visually?
3. True or false: You should always move your asset files around within Unity instead of using the operating system's file explorer.
4. True or false: You manage Unity projects and installations from within the editor.
5. What mode do you enter in the Scene view when you hold down the right mouse button?

Answers

1. False. You can make games with Unity Personal or Unity Plus.
2. Scene view
3. True. This helps Unity keep track of the assets.
4. False. You manage projects and editor installations through the Unity Hub.
5. Flythrough mode

Exercise

Take a moment to practice the concepts presented in this hour. It is important to have a strong foundational understanding of the Unity editor because everything you will learn from here on out will utilize it in some way. To complete this exercise, do the following:

1. Create a new scene by going to **File > New Scene** or by pressing **Ctrl+N** (**Command+N** on a Mac).
2. Create a folder in the Project view by right-clicking **Assets** and selecting **Create > Folder**. Name the folder **Scenes**.
3. Save your scene by going to **File > Save Scene** or by pressing **Ctrl+S** (**Command+S** on a Mac). Be sure to save the scene in the Scenes folder you created and give it a descriptive name.

4. Add a cube to your scene. You can do this in one of three ways:
   - Click the **GameObject** menu at the top of the editor and select **3D Object > Cube**.
   - Click **Create > 3D Object > Cube** in the Hierarchy view.
   - Right-click in the Hierarchy view and select **3D Object > Cube**.
5. Select the newly added cube in the Hierarchy view and experiment with its properties in the Inspector view.
6. Practice navigating around the Scene view by using Flythrough mode, the Hand tool, and snap controls. Use the cube as a point of reference as you navigate.

Index

Symbols & Numerics

A

B

C

D

E

F

G

H

I

J-K-L

M

N

O

P

Q-R

S

T

U

V

W-X-Y-Z