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.

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.

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 all the different elements of the Unity editor and how they fit together to make games.

The Project Dialog

When opening Unity for the first time, you see the Project dialog (see Figure 1.3). You use this dialog to open recent projects, browse for projects that have already been created, or start new projects.

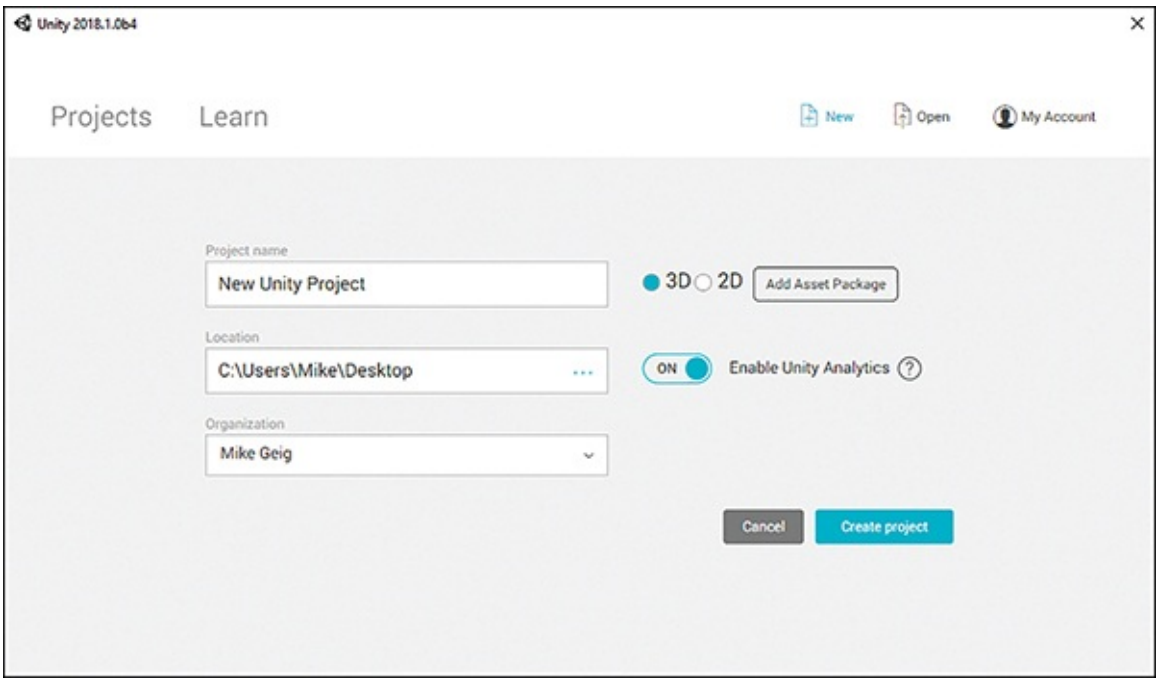

FIGURE 1.3
The Project dialog (Windows version shown; the Mac version is similar).

If you have created a project in Unity already, whenever you open the editor, you are taken directly to that project. To get back to the Project dialog, in Unity, select **File > New Project** to get to the Create New Project dialog or select **File > Open Project** to get to the Open Project dialog.

TIP

Opening the Project Dialog

You can choose whether you want Unity to open to the Project dialog or the last opened project by going to **Edit > Preferences (Unity > Preferences on a Mac)** and changing the value of the **Load Previous Project on Startup** check box.

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 find it easily later if necessary. Figure 1.4 shows the dialog box you use to create the project. Follow these steps:

1. Open Unity and locate the New Project dialog. (If Unity opens to a project, click **File > New Project.**)
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 and ignore the *Add Asset Package* and *Enable Unity Analytics* options for now.
5. Click **Create Project**.

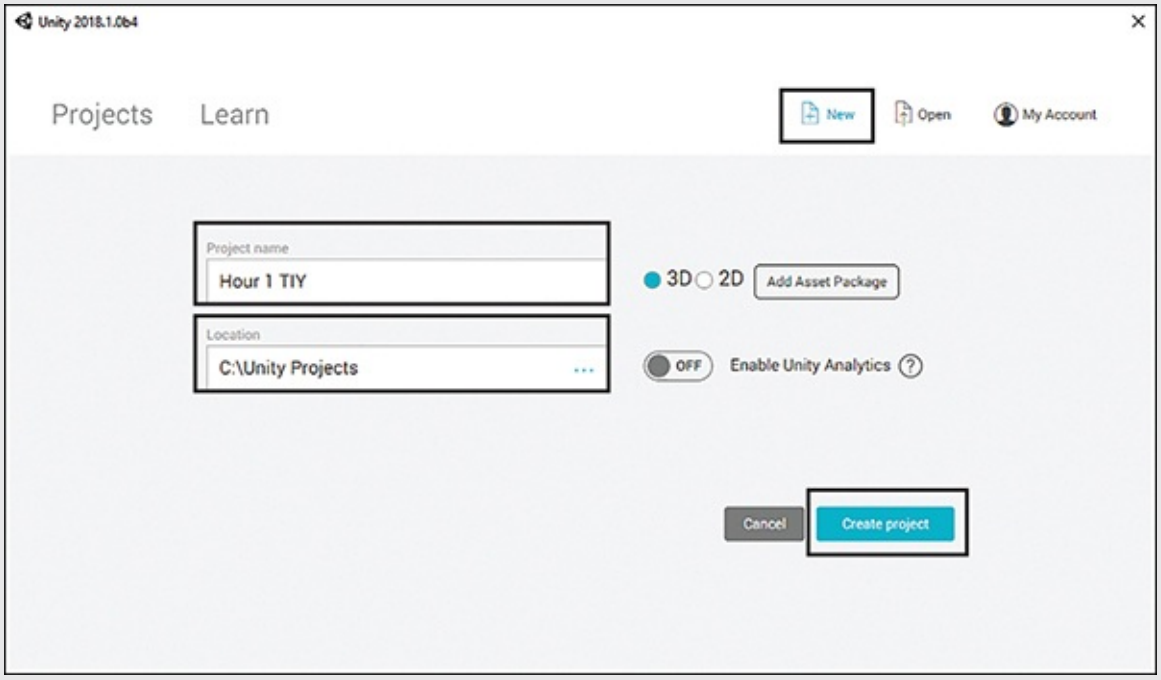

FIGURE 1.4
The settings for your first project.

NOTE

2D, 3D, Packages, Analytics?

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 just control editor settings and can be changed at any time. The *Add Asset Package* button allows you to add files that you commonly use in your projects into a new project automatically. Finally, the *Enable Unity Analytics* option is a very powerful tool that taps into Unity’s Analytics service. Using it allows you to generate data about your game audience; while that is totally awesome for game developers, it isn’t necessary for you at this time.

The Unity Interface

So far, you have installed Unity and looked at the 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 1.5). Never fear; you will quickly get this place hopping. The following section look at each of the unique views, one by one. First, though, let’s talk about the layout as a whole.

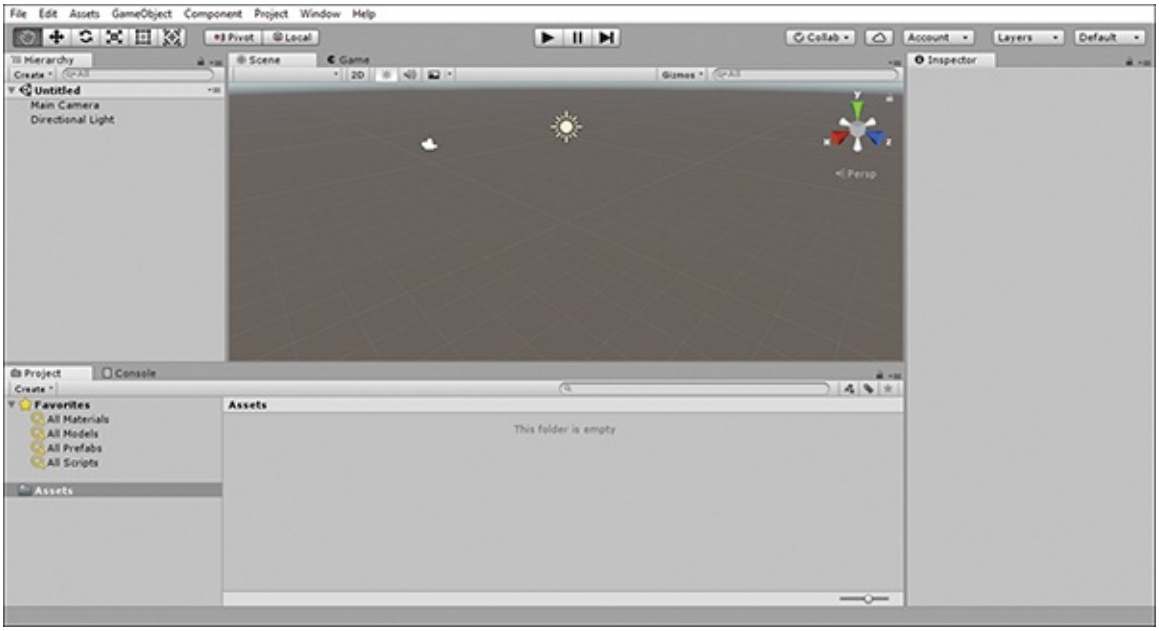

FIGURE 1.5
The Unity interface.

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.

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 a lot with Unity. 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 1.6), hover the mouse cursor over **Add Tab**, and a list of views pops up for you to choose from (see Figure 1.6). You may wonder why you would want to duplicate a view. Say that in your view-moving frenzy, you accidentally closed the 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 created a layout that you like, be sure to save it before you check out 4 Split.)

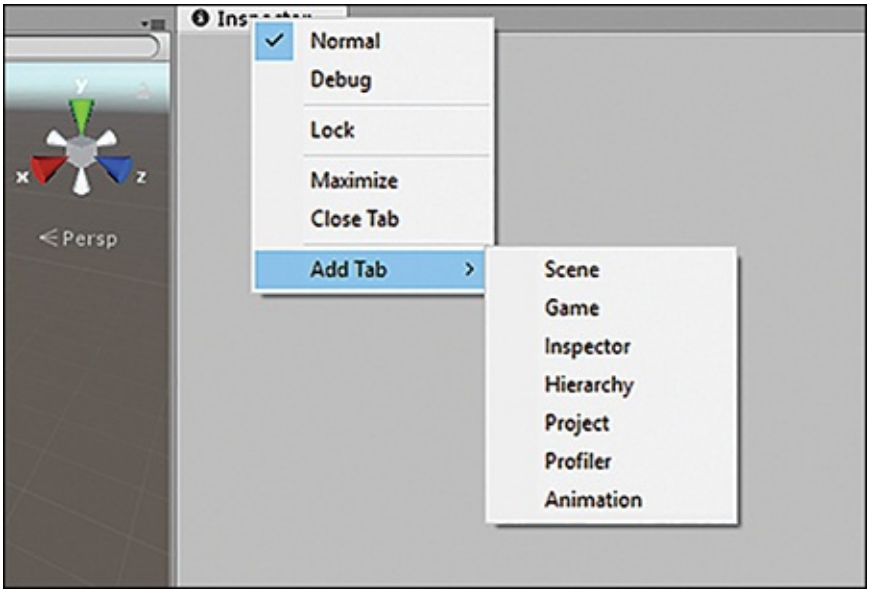

FIGURE 1.6
Adding a new tab.

Now, without further ado, let's look at the specific views themselves.

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 1.7). This is the window that shows all the assets and organization of a project. When you create a new project, you see a single folder item 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. This enables you to place items inside folders or reorganize your project on-the-fly.

FIGURE 1.7
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.

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 1.7, the Assets folder is currently empty; therefore, nothing appears on the right. If you would like to create assets, you can do so easily by clicking the **Create** drop-down menu. 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 help 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 that 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 for you to get an "at a glance" view of your assets quickly. 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 Asset Store. If you click **Asset Store**, you can browse the assets that fit your search criteria from the Unity Asset Store (see Figure 1.8). You can further narrow your results by free and paid assets. This is a fantastic feature because it enables you to grab assets that you need for your project without ever leaving the Unity interface.

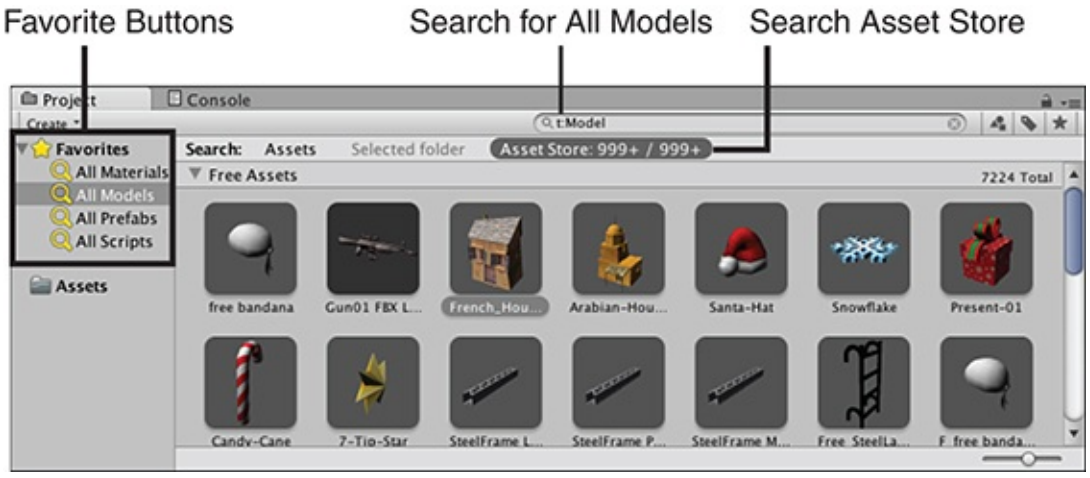

FIGURE 1.8
Searching the Unity Asset Store.

The Hierarchy View

In many ways, the Hierarchy view (see Figure 1.9) is a lot like the Project view. The difference is that the Hierarchy view shows all the items in the current scene instead of the entire project. When you first create a project with Unity, you get the default scene, which has just two items in it: Main Camera and Directional Light game objects. As you add items to your scene, they appear in the Hierarchy view. Just like 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.

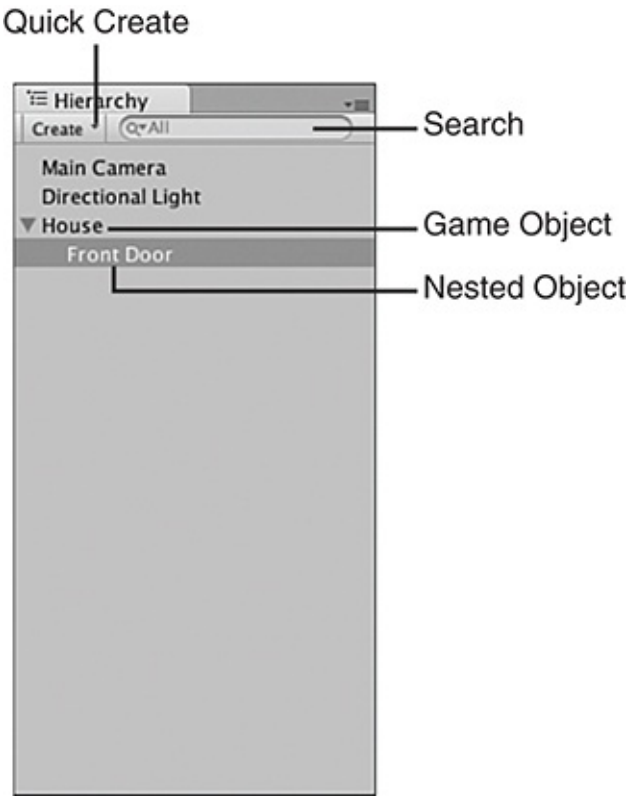

FIGURE 1.9
The Hierarchy view.

TIP

Nesting

Nesting is the term for 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. Therefore, 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.

TIP

Scene Organization

The first thing you should do when working with a new Unity project is create a Scenes folder under Assets in the Project view. This way, all your scenes (or levels) will be stored in the same place. Be sure to give your scenes descriptive names. Scene1 may sound like a great name now, but when you have 30 scenes, such unclear naming can get confusing.

The Inspector View

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

In Figure 1.10, you can see the Inspector view after the Main Camera object is selected from the Hierarchy view.

Let’s break down some of this functionality:

- If you click the check box next to the object’s name, it becomes disabled and does not appear in the project.

- Drop-down lists (such as the Layer and Tag lists; more on those 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 Transform, Camera, and Audio Listener in Figure 1.10). You can disable these components by unchecking them or remove them by right-clicking and selecting **Remove Component**.
- Components can be added by clicking the **Add Component** button.

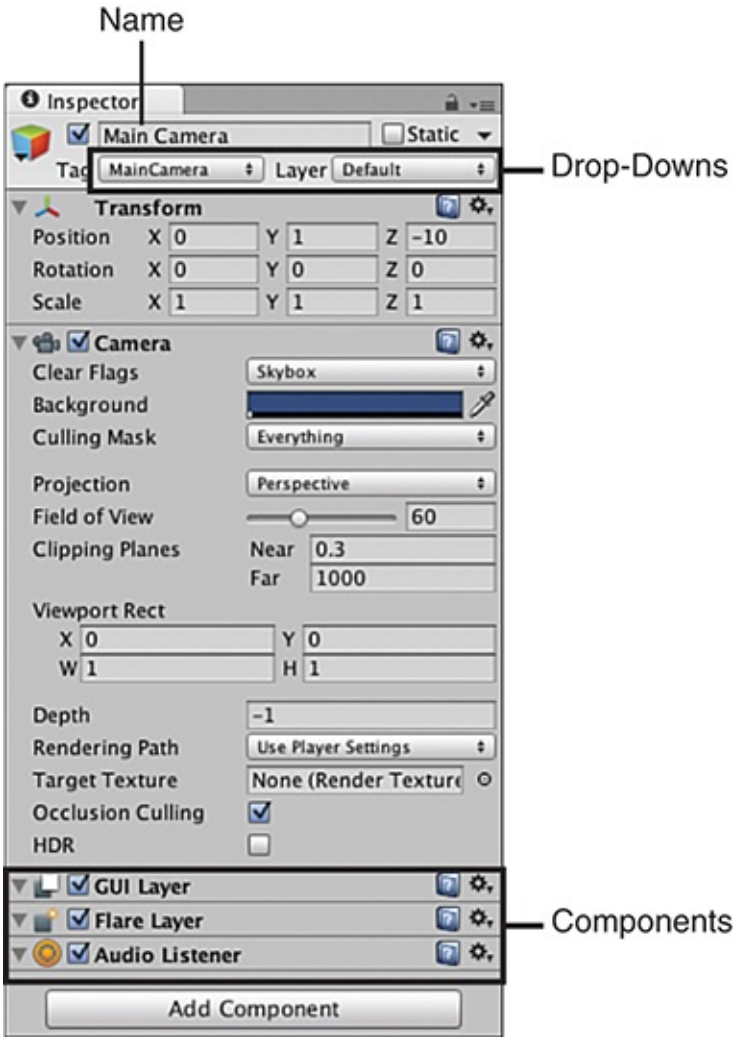

FIGURE 1.10
The Inspector view.

CAUTION

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, collision power, and so on, all on-the-fly, without stopping and starting the game. Be wary, though, as 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 because it enables you to see your game visually as it is being built (see Figure 1.11). Using the mouse controls and a few hotkeys, you can move around inside your scene and place objects where you want them. This gives you an immense level of control.

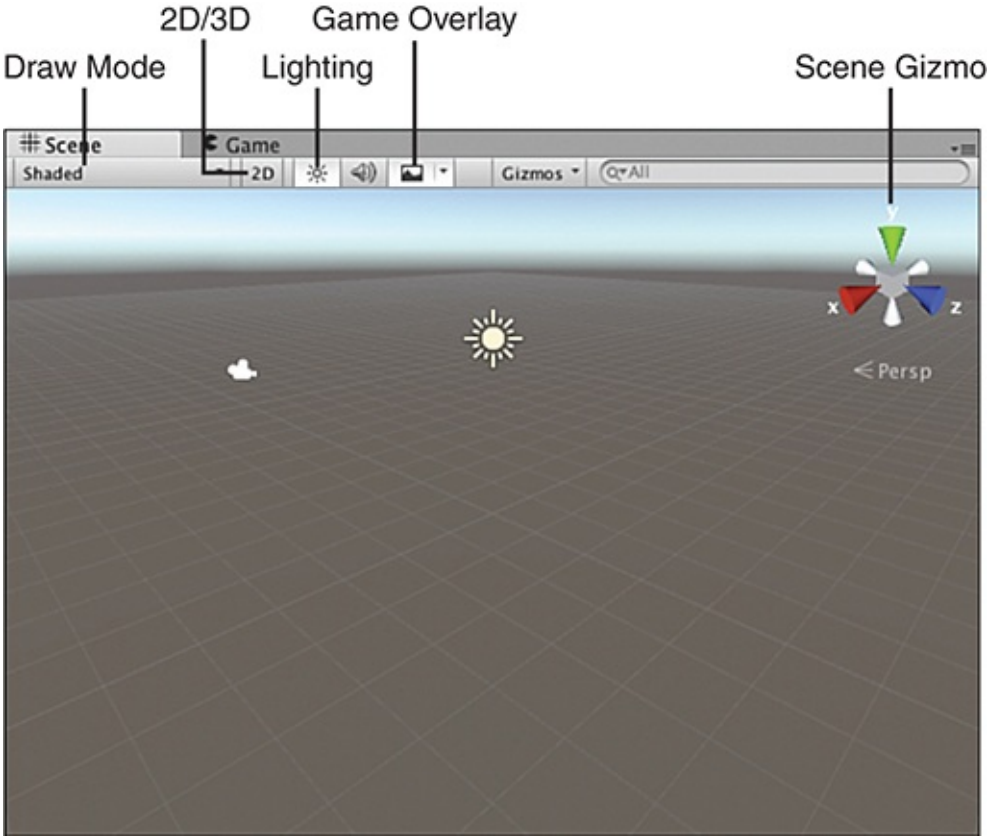

FIGURE 1.11
The Scene view.

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) does not show.
- **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.
- **Game overlay:** This control determines whether items like skyboxes, fog, and other effects appear in the Scene view.
- **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, 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 it out for yourself and see how it affects 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 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 1.12 shows what a Game view looks like. Note that although the Play, Pause, and Step buttons are not technically a part of the Game view, they control the Game view and therefore are included in the image.

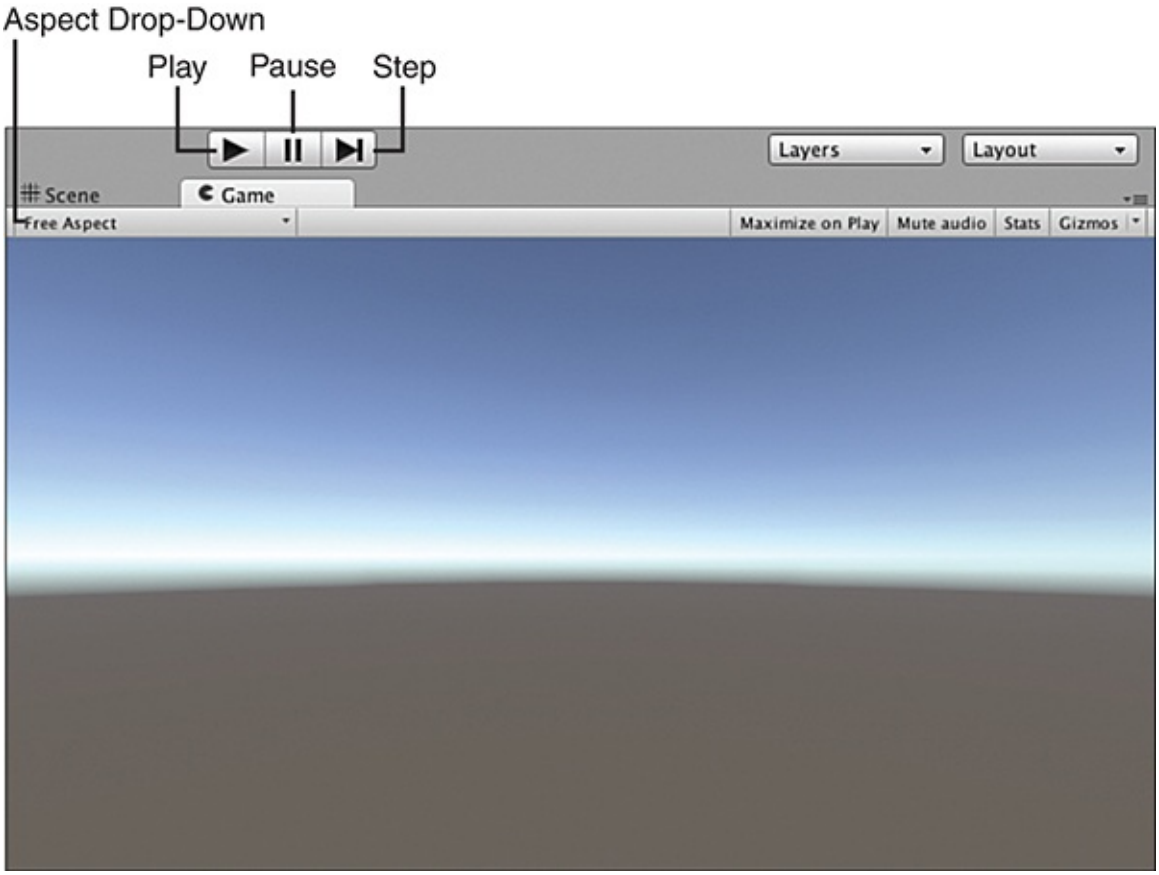

FIGURE 1.12
The Game view.

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 closely 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 the game from running, click the Play button again.
- **Pause:** The Pause button pauses the execution of the currently running Game view. The game maintains its state and continues exactly where it was when paused. Clicking the Pause button again causes the game to continue running.
- **Step:** The Step button works while the Game view is paused and causes the game to execute a single frame of the game. This effectively allows you to “step” through the game slowly and debug any issues you might have. 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 this 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 is 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 play-testing!
- **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 the 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 the game is running and the Pause button is pressed, the game stops running but still 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 1.13 shows the toolbar components:

- **Transform tools:** These buttons enable you to manipulate game objects and are covered in greater detail later in this book. 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. 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 1.13
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 hold **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 1.14). 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 1.14
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 **Ctrl** (**Command** on Mac) and **right-click-drag**	Zooms the camera

You can find all the Unity hotkeys at http://docs.unity3d.com/Manual/UnityHotkeys.html.

CAUTION

Different Cameras

When working in Unity, you deal with two types of cameras. The first 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 is what determines what you can see in the Scene view. When this hour's instructions mention the camera, it is talking about the second type. You do not actually manipulate the game object camera.

Flythrough Mode

Flythrough mode enables you to move about the 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.

	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 the 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** (short 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 New Project or Open Project dialog, your eyes may have magically been draw to the Learn button present there (see Figure 1.15). Clicking that button displays 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.)

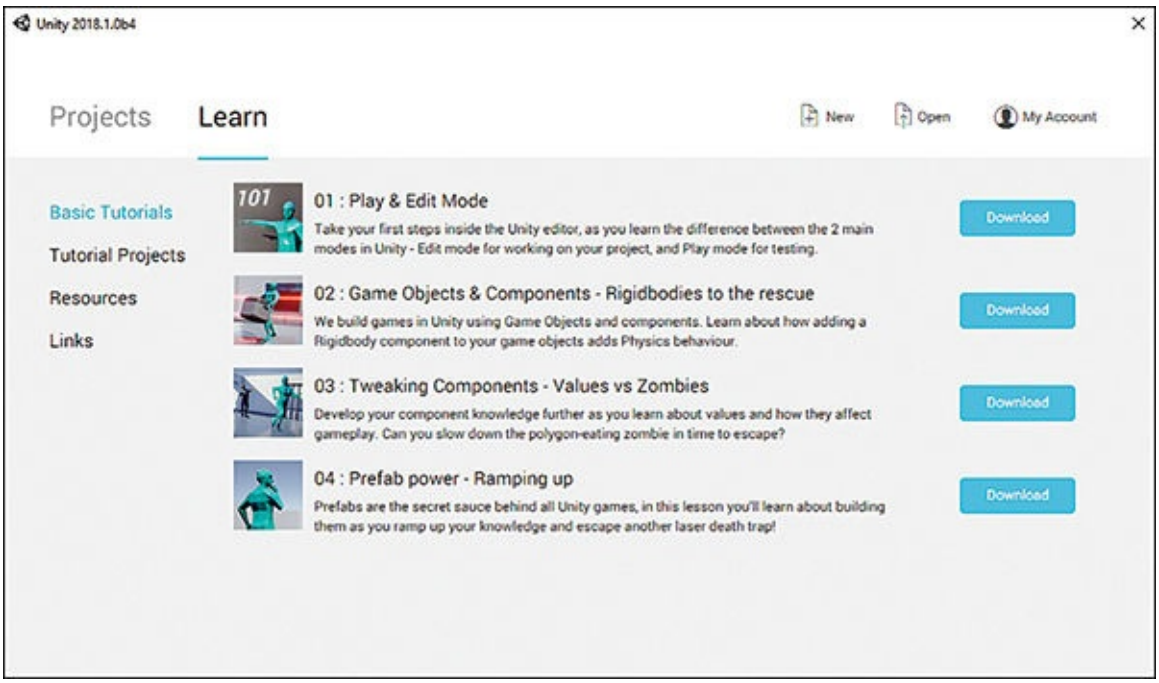

FIGURE 1.15
The Learn section of the Project dialog.

Summary

In this hour, you took your first look at the Unity game engine. You started off 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 and not use the operating system's file explorer.
4. True or False: When creating a new project, you should include every asset that you think is awesome.
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. This takes up space and slows down your project.
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 to help navigate.