1
Getting Started

In this chapter, we will cover:

- Installing Unity
- Understanding and optimizing the UI
- Setting your preferences
- Saving assets created in Unity as Prefabs
- Discovering Unity's content
- Importing your own content
- Importing Unity Packages into your project
- Importing Custom Packages into your project
- Exporting Custom Packages from your project
- Adding Custom Packages to Unity's quick list
- Using the Project browser

Introduction

This chapter is tailored for those who are about to start using Unity or have just arrived to it. In this chapter, you will find some introductory steps into making this engine more comfortable and familiar.

Installing Unity

Unity is a very powerful and versatile game engine. It is available in both Indie (free) and Pro (paid) version. In case you haven't installed Unity yet, this recipe will show you how to do it.

Getting Ready...

You will need Internet access to follow this recipe.

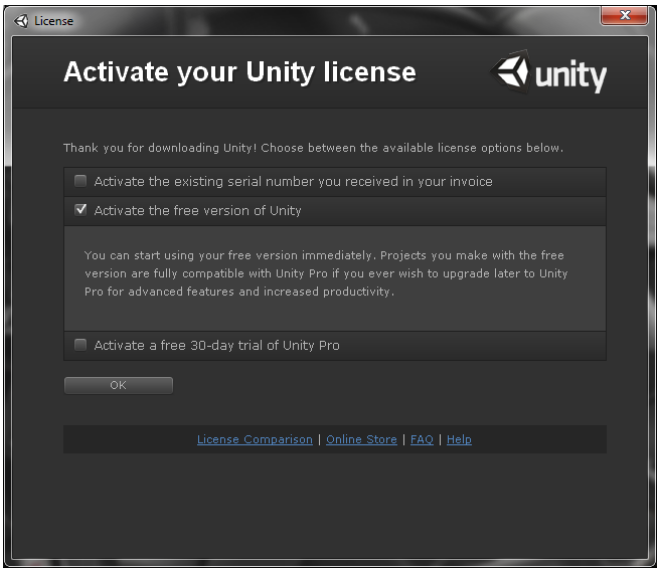
How to do it...

To Install Unity, please follow these steps:

1. Access Unity website at www.unity3d.com.
2. Locate and click the *Download* button, placed in the top right corner.

0423_01_01.png

3. Now, in the Download page, click the button to get the latest version of Unity. Wait for the download to complete.
4. Run the Installer. This is a very straightforward process that will install everything you need in a couple of minutes.
5. Once the software is installed, run Unity. That should take you to the Activation dialog, where you can choose between activating *Unity Pro* (provided you have a valid serial number), *Unity Free* or a 30-day trial of *Unity Pro*.



0423_01_02.png

6. Make your choice and click Ok. You should be prompted to log in or create an account.

7. Register (if necessary) and log in to activate your copy of Unity and start using it right away.

There's more...

You can expand Unity capabilities and reach new audiences by adding more platforms to your editor.

Acquiring IOS, Flash, Android and Flash licenses

IOS, Flash, Android and Flash exporters can be bought from the Unity Store at `https://store.unity3d.com`.

Setting your preferences

Setting the editor to your preferences might sound superfluous to some. However, it could accelerate your development process and make Unity even more comfortable to use. In this recipe, we will learn how to adjust some of those settings to your taste.

How to do it...

To adjust Unity's preferences, follow these steps:

1. Inside the Unity editor, access the menu *Edit > Preferences...*

2. As the Preferences windows shows up, notice that it is divided into three sections: *General*, *External Tools*, *Colors*, *Keys* and *Cache Server*.

3. Select the *General* tab. If you're work at multiple projects, you might want to leave the option *Always Show Project Wizard* checked.

4. Also, if you use OSX and are used to its native color picker, leave the option *OSX Color Picker* checked.

5. Now select the *External Tools* tab. In case you want to use a different Script Editor than Unity's built-in MonoDevelop, you can use the drop-down menu in *External Script Editor* to browse to your favorite application.

6. If *Image Application* is set as *Open by File Extension*, you might end up working with several image editors simultaneously. To avoid that, use the drop-down menu to browse to your favorite software.

7. Also, if you happen to develop to Android, make sure to browse to the SDK in *Android SDK Location*.

8. Let's move on to *Colors* tab. The default settings are fine, but feel free to change colors that make you most comfortable.

9. Now select the *Keys* tab. You might select any *Action* to change its shortcut. Again, the default settings are perfectly fine. Use the opportunity to learn them.

There's more...

As you probably noticed, Unity's *Preferences* window has more options than it was covered here. If you want a full explanation for each setting, please check its online documentation at
http://docs.unity3d.com/Documentation/Manual/Preferences.html

Changing the editor's player quality settings

If you want to experience a particular quality setting when running your game from the Editor, access the menu *Edit* > *Project Settings* > *Quality* and select it from the table in the *Inspector* view.

See also

Understanding and optimizing the UI

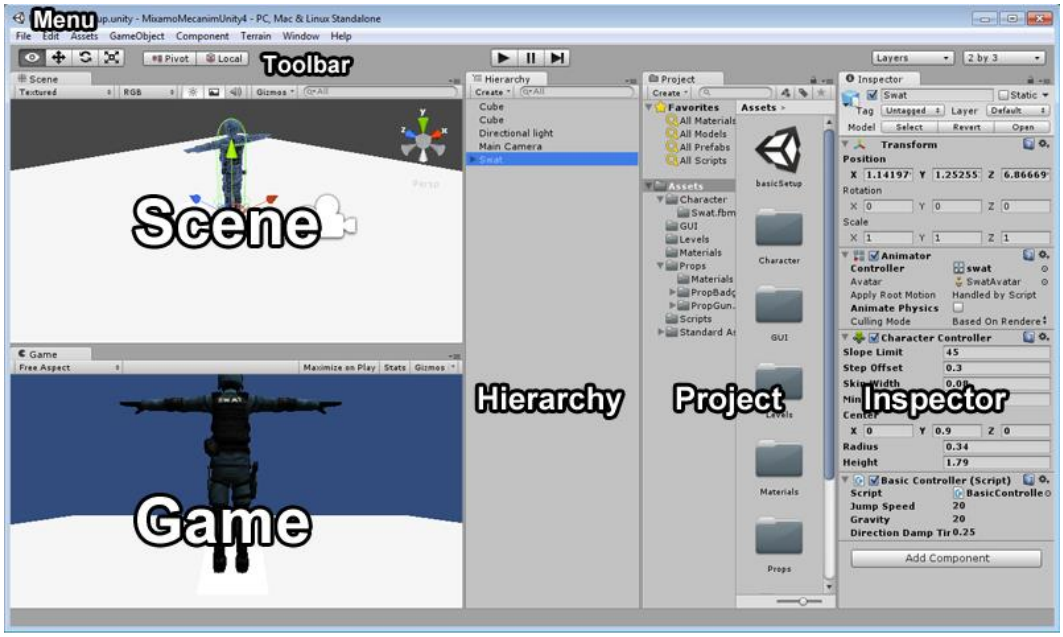
Understanding and optimizing the UI

Game engines, especially 3D-capable ones, can be a bit intimidating the first time you open them. Although Unity is particularly intuitive, user-friendly, and well documented, we have provided you this recipe to show you how to operate inside its UI.

How to do it...

Let's take a look at Unity's user interface:

1. Run Unity. Unless you have previously changed it, its layout should initiate in *Wide* mode. Access the menu *Window* > *Layouts* and choose another option such as *4 split* or *2 by 3*, and notice how the interface is organized into *Views*.



0423_01_03.png

- Scene: Those are viewports where to position, rotate, scale and select the game objects, and also navigate your level.
- Game: This is the place to play and test your game. It will reproduce the player's experience as accurately as possible.
- Hierarchy: Game objects (as diverse as characters, cameras, level geometry, lights, and even GUI Textures) placed in our scene will be listed here.
- Project: This is where you create, organize and access your game assets. From 3D models and 2D textures to C# Scripts and Prefabs, every reusable element will be listed here.
- Inspector: From the Inspector you can configure any game objects (selected from the Hierarchy view) or assets (selected from the Project view). That includes changing its Transform settings, configuring existing components and attaching new ones. Also, you can adjust other preferences for your game, once you have accessed them from the menu, in the Inspector view.
- Toolbar: Includes Transform tools (used for manipulating game objects and navigating the scene); Control tools (used for playing / pausing and stopping the level); and Drop-down tools (used for managing Layers and Layouts)

- Menu: Access to a diversity of commands covering asset import/export, preferences setting, game object creation, components, terrain, layout and documentation.

2. If you want to customize the layout any further, drag and drop the views to relocate and / or dock them.

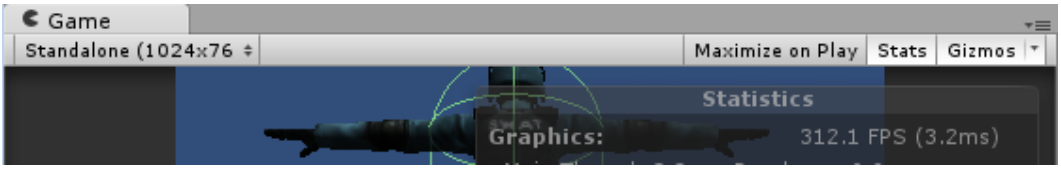
3. If you like your custom layout, save it through the menu *Window* > *Layouts* > *Save Layout*...

4. When testing your game, it might be a good idea to check the button *Maximize on Play*, in the *Game* view. Also, if you work with more than one display monitor, you could drag the *Game* view into the second display, leaving a display exclusively for the Editor.

5. You can also adjust the *Game* view resolution. It's a good idea to test your game running on its standard *Standalone Resolution* and every supported Aspect Ratio.

6. Also, in case you need to optimize the graphics performance of your game, activate the *Stats* button before testing it.

7. Finally, Activate *Gizmos* if you want them to be drawn at runtime, making it easier to spot rays, colliders, lights, cameras, etc in your scene.



0423_01_04.png

8. There is another view you should pay attention to: the *Console*. Access it through the menu *Window* > *Console*. This is a very important view when it comes to Debugging your game.

9. Another interesting view (for those with Unity Pro) is the *Profiler* (menu *Window* > *Profiler*), where you can check out detailed statistics of your game performance in real time.

There's more...

To get an extensive explanation on each UI feature, please check out Unity's documentation at:
http://docs.unity3d.com/Documentation/Manual/LearningtheInterface.html

See also

Setting your preferences

Searching assets with the Project browser

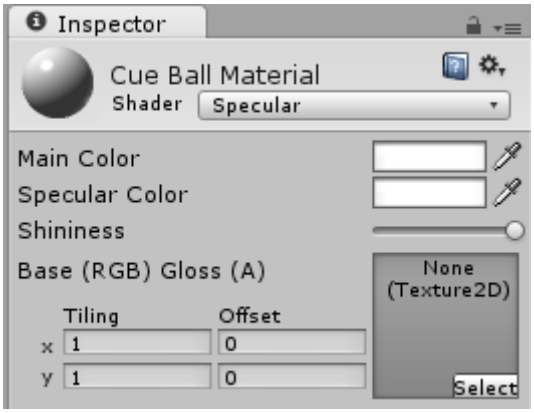
Saving assets created in Unity as Prefabs

Although Unity can be very limited as an asset creation tool (there are no internal 3d modelers or image editors, for instance), you can always create very basic game objects with it. In this recipe, we will create a game object from Unity resources and keep it in our Project as a *Prefab*.

How to do it...

To create a prefab, follow these steps:

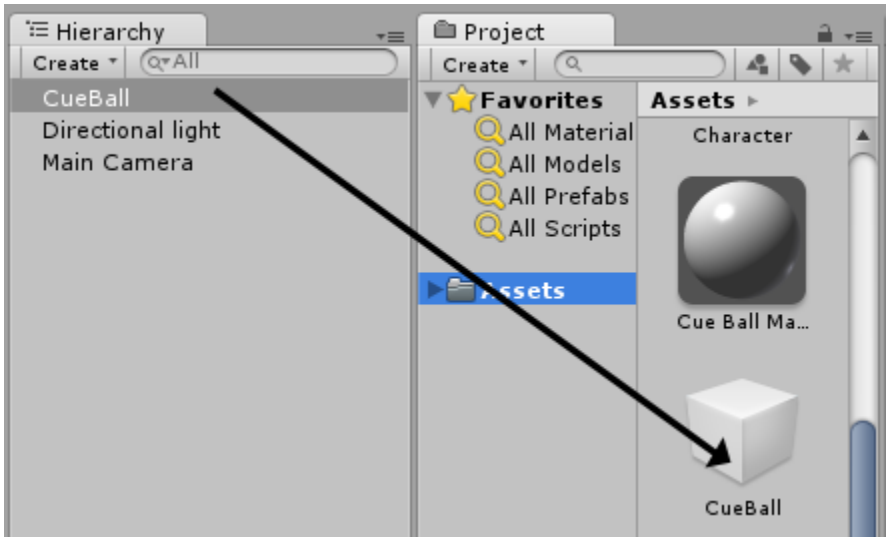
1. Inside the Unity editor, access the menu *GameObject* > *Create Other* > *Sphere*.
2. In the *Hierarchy* view, rename it as *Cue Ball*.
3. Now, in the *Project* view, click the *Create* button and choose the option *Material*. Then, rename the new material as *Cue Ball Material*.
4. In the Project view, select the *Cue Ball Material*. Then, in the *Inspector* view, change its *Shader* to Specular.
5. Also, change the *Specular* color to white and set its *Shininess* to the maximum.



0423_01_05.png

6. From the *Project* view, drag *Cue Ball Material* into the *Cue Ball* game object, in the *Hierarchy* view.
7. Select the *Cue Ball*. Then, access the menu *Component* > *Physics* > *Rigidbody*. That should attach a Rigidbody component to that game object.

8. Now that your game object is complete, click the *Create* button in the *Project* view and choose the option *Prefab*. Then, rename it as *Cue Ball Prefab*.
9. Drag the *Cue Ball* game object from the *Hierarchy* view into the Prefab in the *Project* view. Your game object is ready to be reused in this project.



0423_01_06.png

How it works...

In Unity, *Game Objects* can be stored as *Prefabs*. This is very useful in case you want to reuse a *Game Object* in several levels or instantiate it through scripting. Adobe Flash users can think of it as Unity equivalent of *MovieClips*.

There's more...

Adding external files

In this recipe, we haven't used any external asset. However, there's no reason you could not have imported a texture and used it as the *Cue Ball Material* base map, for instance.

Taking your Prefabs to another project

Also, if you plan of reusing your Prefab in other projects, you can do it by exporting it as a *Custom Package*.

Creating other kinds of Game Objects

As you probably noticed, Spheres are not the only entities you can create directly with Unity. Other primitives are available, as well as many other types of entities: lights, camera, GUI textures, etc. Access the menu *GameObject* > *Create Other* and experiment with the options.

See also

Exporting Custom Packages from your project.

Discovering Unity's content

As you enter Unity for the first time, you might think you'll have to build and code everything from scratch. However, Unity comes with several collections of content called "Packages", designed to save you time when implementing commonly required features.

How to do it...

Let's find out what's inside Unity's standard pakages:

1. Inside the Unity editor, access the *Assets* menu.
2. Expand the *Import Package* submenu.
3. You will see a list of available Packages from Unity. These are filled with ready to use content.

How it works...

Unity makes implementing commonly requested features easy by making them available as *Packages* ready to be imported and used in your project. These packages include First-Person and Third-Person Character Controllers; Image Effects (Pro Only); Terrain textures and assets; Skyboxes; Water, Tree Creator, and more.

There's more...

Sample Project

Unity also comes with a sample project ready to be dissected by you. It automatically opens the first time you start the software.

Download more resources

You can find and download even more resources, such as Packages, Projects, Tutorials and Assets, at Unity's Resources page at: `http://unity3d.com/support/resources/`.

See also

Importing Custom Packages into your project.

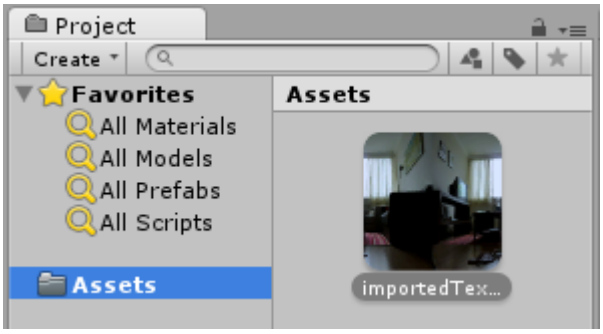
Importing your own content

After you have created a 3d model, an audio clip, a movie clip or a texture, you can import it into your project. In this recipe, we will learn how it can be done.

How to do it...

Follow the steps to import an asset:

1. Inside the Unity editor, access the *Assets* menu.
2. Select the option *Import New Asset…*.
3. Browse to your file and click *Import*.
4. Your file should be now listed in the *Project* browser.



0423_01_07.png

How it works...

Unity makes a copy of your file, converts it to an appropriate format (if necessary) and saves it into the Projects *Assets* folder.

There's more...

Organize it with Unity View

Unity updates its *Project* view whenever a new file is added to the *Assets* folder. You could, then, save or export your work directly into that folder. You could also paste or move multiple files into there. However, you should not re-organize or rename your

imported files via your OS file management system (Window's *Explorer* or Mac OS *Finder*), as this could damage important information kept by Unity about those files.

Exporting your assets

If you are not sure about how to export your work to Unity, or which file format you should use, please check out Unity's documentation at http://docs.unity3d.com/Documentation/Manual/AssetImportandCreation.html for a very comprehensive guide on the subject. Some other useful pages regarding the subject are:

- Importing Objects From 3D Studio Max - http://docs.unity3d.com/Documentation/Manual/HOWTO-ImportObjectMax.html
- Importing Objects From Maya - http://docs.unity3d.com/Documentation/Manual/HOWTO-ImportObjectMaya.html
- HOW TO – Export FBX - http://docs.unity3d.com/Documentation/Manual/HOWTO-exportFBX.html

See also

Importing Custom Packages into your project.

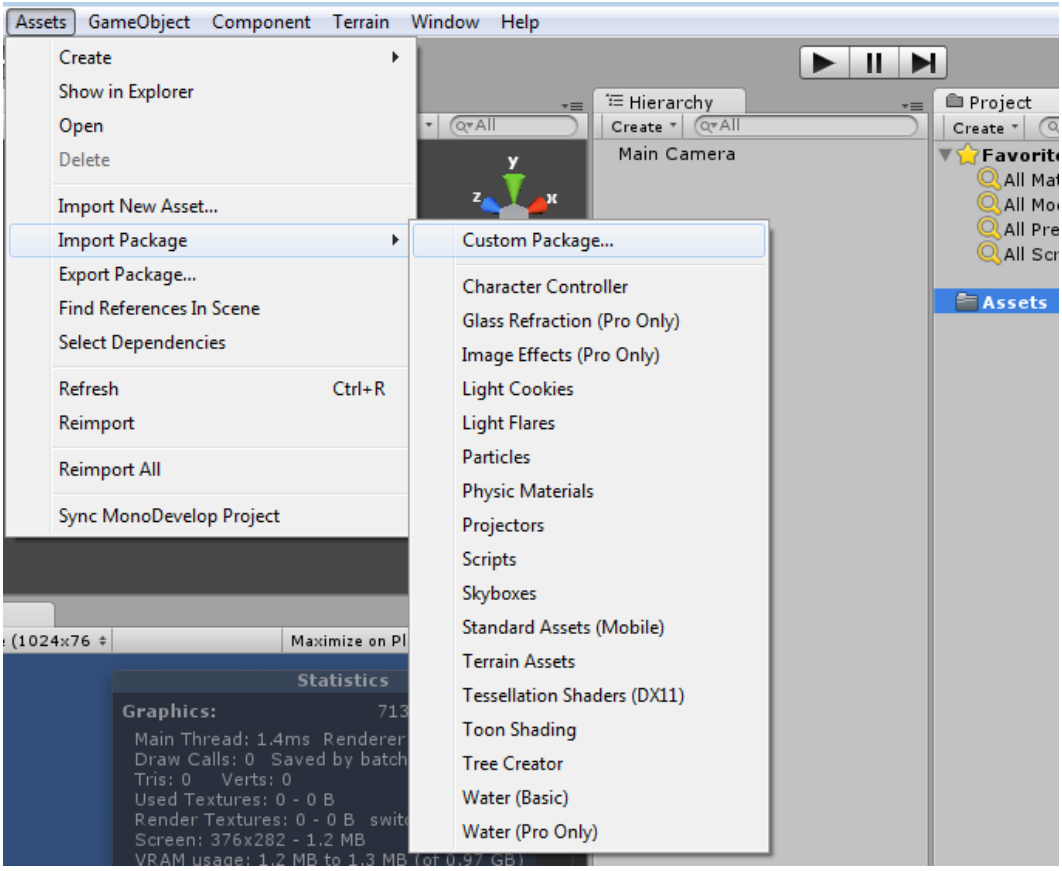
Importing Unity Packages into your project

The packages provided by Unity can save you a lot of development time. They usually contain resources (such as texture maps, materials, etc.) and fully implemented features ready to go into your project. When creating a new project Unity offers to install those packages into the Assets folder. However, if you've missed it at first, you can still import them into your project at any time.

How to do it...

To import a Unity Package, follow these steps:

1. Inside the Unity editor, access the *Assets* menu.
2. Enter the *Import Package* sub-menu and choose a package from the list.
3. Make sure every needed component is selected and click *Import*.
4. Package contents should be ready and listed in the Project window.



0423_01_08.png

How it works...

Unity installation files include a number of packages that can be imported into your project as ready-to-use resources. Inside those packages are all the assets needed to implement a specific feature or functionality. Once imported, new features can be accessed through the *Project* window (and dragged and dropped into your level) or through newly added menu items (the *Tree Creator* package, for instance, adds the option *Tree* into the *Create Other* sub-menu of the *GameObject* menu).

There's more...

Importing Unity Packages can also be done through the Project Wizard. When starting a new project, check the boxes of the packages you want to import.

See also

Importing Custom Packages into your project.
Exporting Custom Packages from your project.
Adding a Custom Package to Unity's quick list.

Importing Custom Packages into your project

Custom Unity Packages are available from a variety of sources, and they can be very helpful when developing a project.

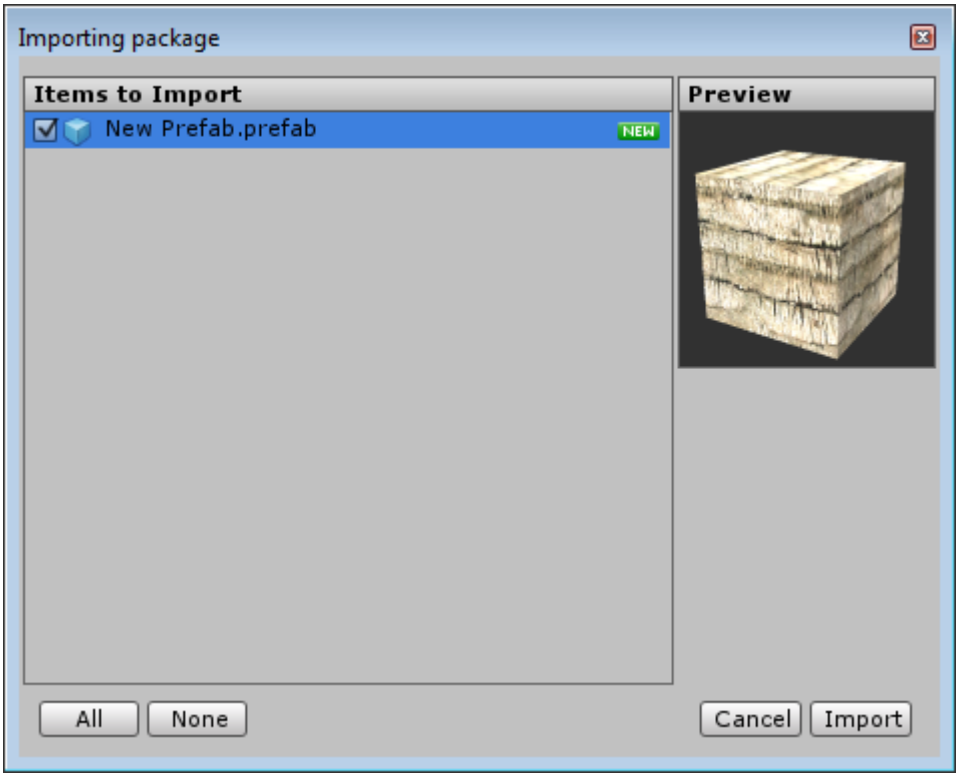
Getting ready

As its title implies, this recipe requires a custom package to be imported. If you need one for testing purposes, please use the one inside the folder named *0423_01_09-11*.

How to do it...

To import a custom package, follow these steps:

1. Inside the Unity editor, access the *Assets* menu.
2. Enter the *Import Package* sub-menu and choose the option *Custom Package...*
3. Browse to the package file you have saved on your disk and click *Open*.
4. Preview package contents in the top right *Preview* window, if you like.
5. Make sure every needed component is selected and click *Import*.
6. Package contents should be ready and listed in the *Project* view.



0423_01_09.png

How it works...

Custom packages are vastly used to distribute a number of assets inside a single compressed file. As they are made by third parties, the content inside those packages may vary, as they can include scripts, 3d models, texture maps, materials and any other file handled by Unity. Once imported, the package content is uncompressed into your project's Assets folder, and can be accessed through the *Project* window.

There's more...

3rd-party made content can also be found, downloaded and bought at the Unity Asset Store. For more information, access http://unity3d.com/unity/asset-store/.

See also

Importing Unity Packages into your project.
Exporting Custom Packages from your project.
Adding a Custom Package to Unity's quick list.

Exporting Custom Packages form your project

Creating Packages can be a very useful and practical way of storing your game objects and assets for future use and reference. If you want to save a feature, a group of assets, or even a prefab from the project you are currently working at, it's a good idea to export them as a package, so you can easily import them into your future projects.

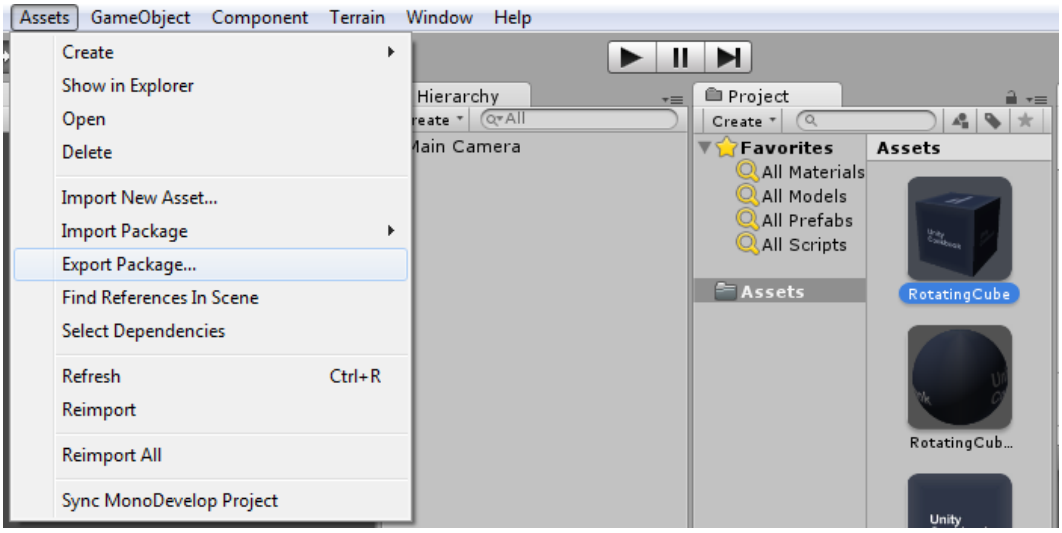
Getting ready

In order to export a package, you will need a project containing some assets. If you need one testing purposes, please use the one inside the downloadable content for this book, which can be found inside the folder 0423_01_10.

How to do it...

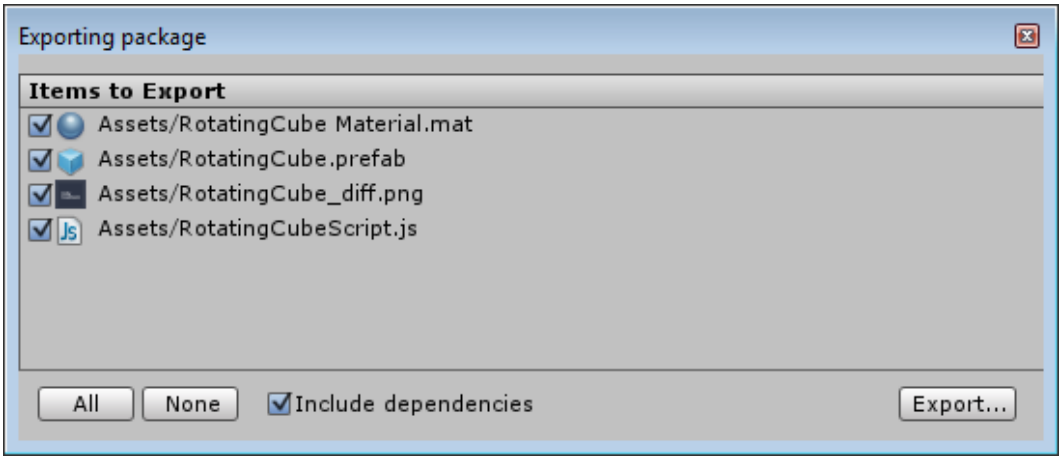
To export content as a custom package, follow these steps:

1. Select, in the *Project* view, the *RotatingCube* prefab.
2. Go to the *Assets* menu and choose the option *Select Depencies*. This will highlight, inside the *Project* tab, all assets that are linked to the *RotatingCube*.
3. Once again, select the *RotatingCube* prefab, only.
4. Go to the *Assets* menu and choose the option *Export Package...* A new window will pop up.



0423_01_10.png

5. Inside the *Exporting Package* window, make sure the *checkbox Include dependencies* is selected. It is important that checkboxes for all listed objects are also selected.
6. Click *Export* and save your package into your disk. You can give it any name you want (although a name similar to *RotatingCube* will make things easier later, when you want to use it).
7. Your Custom Package is ready to be imported.



0423_01_11.png

How it works...

By exporting a package, you have stored your selected objects and dependencies into a single compressed file. Importing them to your project will uncompress them into its Assets folder.

See also

Importing Unity Packages into your project.
Importing Custom Packages into your project.
Adding a Custom Package to Unity's quick list.

Adding Custom Packages to Unity's quick list

If you have one or more packages you'd like to include frequently in your projects, it might be a good idea to add them to Unity's package quick list.

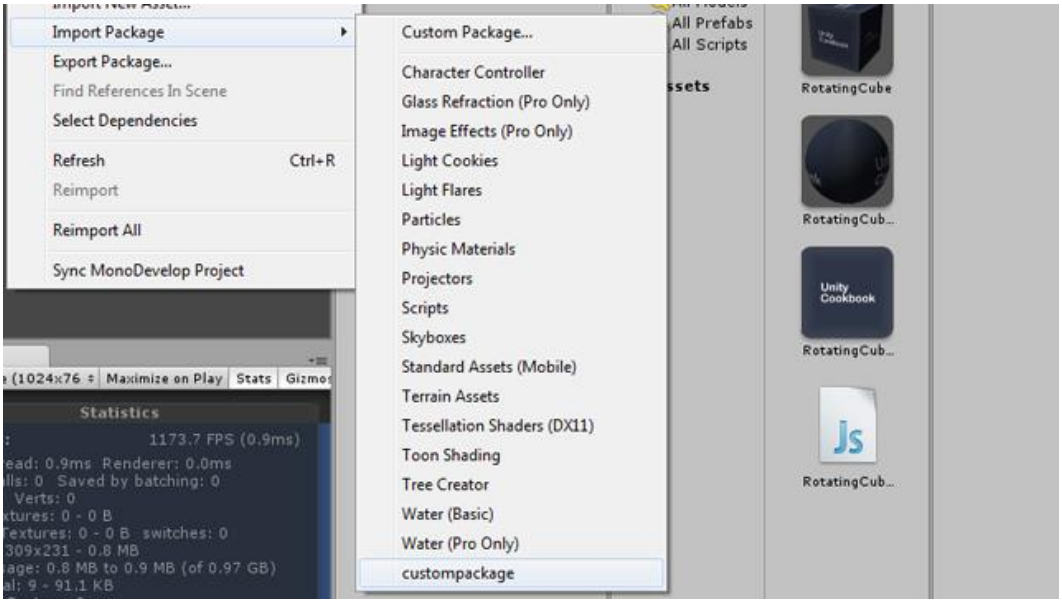
Getting ready

In order to complete this recipe, you will need a custom package (any package will do). If you need one for testing purposes, please use the one inside folder 0423_01_09-11.

How to do it...

To add a custom package to the quick list, follow these steps:

1. Using your file manager (Windows Explorer on Windows, Finder on Mac OS), browse to the package file and copy it by pressing Ctrl+C or Command+C).
2. Go to Unity's Editor folder. On Windows, that would typically be *C:/Program Files (x86)/Unity/Editor* or *C:\Program Files\Unity\Editor*. On Mac OS, it should be *Applications/Unity*.
3. Access the *Standard Packages* folder.
4. Paste the previously copied Package into this folder.
5. Restart Unity. It should now be listed on the *Assets* menu, under the *Import Package* sub-menu.



0423_01_12.png

How it works...

Unity's menu actually reads the *Standard Packages* folder content when starting up, instead of getting that information somewhere else. This is very practical, as it always reflects the actual content of that folder -- and also allows the user to quickly retrieve his favorite packages.

There's more...

Custom packages stored in the *Standard Packages* folder will also appear in the *Create New Project Wizard* window, making it simple to add them to new projects.

See also

Importing Unity Packages into your project.
Importing Custom Packages into your project.
Exporting Custom Packages from your project.

Using the Project browser

It doesn't matter how organized you keep your project folders: there will be times when you will need to search for one or more specific assets. To make thinks easier, Unity 4 includes the Project browser. In this recipe, we will learn how to save time by using it.

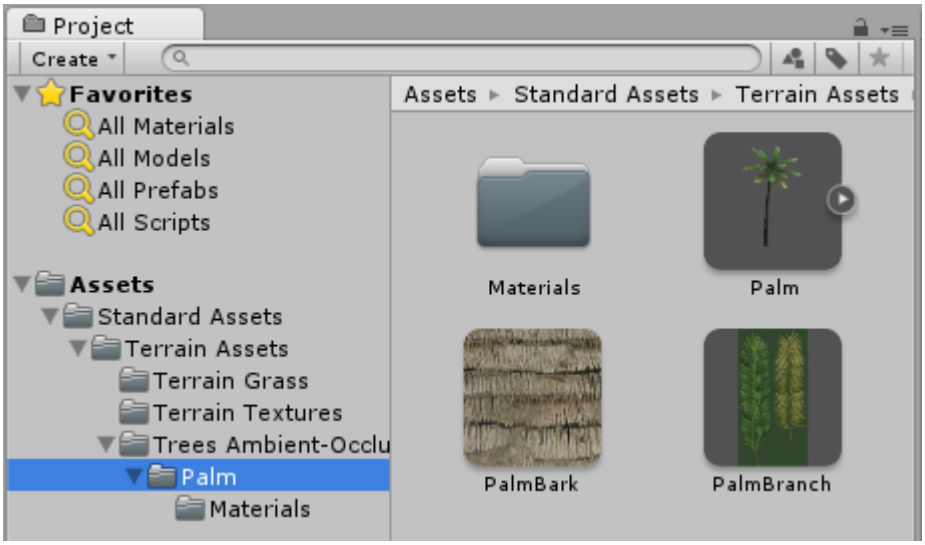
Getting ready

All we need to follow this recipe is a collection of assets. We will use Unity's *Terrain Assets* to populate our *Project* view.

How to do it…

Let's take a look at the Project browser:

1. Import the package Terrain Assets (Menu *Assets* > *Import Package* > *Terrain Assets*).
2. Browse through the Assets sub-folders to see the files that have been imported and how they are organized.



0423_01_13.png

3. Now let's search for all assets containing the word *palm* in their names. On the search field located above, type in *palm*. Observe how every file and folder is listed on the search results window.
4. Click the *Save Search* button (the one with the star icon) and name your search as *Palm*.

0423_01_14.png

5. We could filter our search results by tag or file type. Click the *Search by Type* button (the one with a circle, triangle and square) and select only the categories Models and Textures (you can select multiple options by holding the Ctrl key).
6. Click the *Save Search* button and save it as a new filter (name it Palm Models and Textures).
7. Browse through the other filters saved in *Favorites* – they are shortcuts to browse through specific file types (*All Materials*, *All Models*, etc.)

How it works...

Unity's new browsing system scans through the assets and lets you save and organize your search results.

See also

Understanding and optimizing the UI.