# Unity Manual使用手册

通过Unity 你可以创建2维和3维游戏，应用程序和使用体验。此手册帮你学习掌握如何使用uninty以及其关联服务。你可以完整学习此手册或作为参考引用

此文档最新的功能对应Unity5.3版，关于从旧版本更新项目，你可以参考本手册的更新指南部分

## 更多的信息请参考

* [Unity Answers](http://answers.unity3d.com/) or [Unity Forums](http://forum.unity3d.com/); 问答和论坛
* [The Unity Knowledge Base](https://support.unity3d.com/); 开发团队提供的知识库
* [Tutorials](http://unity3d.com/learn/tutorials); 从头开始学习的教程
* [Unity Ads Knowledge Base](http://unityads.unity3d.com/help/index); 如何在项目中插入广告
* [Everyplay documentation](https://developers.everyplay.com/documentation); Everyplay 重播平台文档
* [Asset Store help](http://unity3d.com/asset-store/help); 商城的帮助文档

## 使用反馈请到

[issuetracker.unity3d.com](https://issuetracker.unity3d.com/).

# Manual Versions手册版本

随着我们对Unity的持续开发，不断会有新功能加入，现有功能的改善以及一些旧功能特性的移除。每次新版本的发布，本手册和脚本引用都会相应做出改变，一定要确保你所使用的首测版本与Unity版本相对应。

在线版本的最新手册查看地址： [**docs.unity3d.com**](http://docs.unity3d.com/)

除了在线手册之外，本文档也可以在开发机上安装，先于5.3版本的Unity软件中本文档是和Unity捆绑安装的。而在5.3及之后的版本中你可以通过 [Unity Download Assistant](http://docs.unity3d.com/Manual/InstallingUnity.html) 选择是否本地安装此文档。

对大多数用户来说，一般都会使用最新版本的Unity引擎和对应的最新参考文档。对一些需要使用旧版本Unity的用户来说，可能会遇到维护旧项目或是长期开发中被限制在特定坂本Unity的情况，如果你在使用旧版本的Unity，本地安装的文档会匹配你用的Unity坂本，如果你不想本地安装参考文档，你也可以在以下地址找到。

## Older versions of the Unity 5 documentation: Unity5以上版本

* Version 5.2: [docs.unity3d.com/520](http://docs.unity3d.com/520)
* Version 5.1: [docs.unity3d.com/510](http://docs.unity3d.com/510)
* Version 5.0: [docs.unity3d.com/500](http://docs.unity3d.com/500)

## Older versions of the Unity 4 documentation: Unity4以上版本

* Version 4.6: [docs.unity3d.com/460](http://docs.unity3d.com/460)
* Version 4.5: [docs.unity3d.com/450](http://docs.unity3d.com/450)
* Version 4.3: [docs.unity3d.com/430](http://docs.unity3d.com/430)
* Version 4.2: [docs.unity3d.com/420](http://docs.unity3d.com/420)
* Version 4.1: [docs.unity3d.com/410](http://docs.unity3d.com/410)
* Version 4.0: [docs.unity3d.com/400](http://docs.unity3d.com/400)

## Older versions of the Unity 3 documentation: Unity3以上版本

* Version 3.5.5: [docs.unity3d.com/355](http://docs.unity3d.com/355)
* Version 3.5.3: [docs.unity3d.com/353](http://docs.unity3d.com/353)
* Version 3.5.2: [docs.unity3d.com/352](http://docs.unity3d.com/352)
* Version 3.5.1: [docs.unity3d.com/351](http://docs.unity3d.com/351)

# New in 5.3版本的新功能

每个新版笨的Unity引擎都包含很多新的功能，既有功能的改进，bug的修复。

本篇介绍手册中新加入和更新的功能，完整列表可以在这里找到 [Unity 5.3 release notes](https://unity3d.com/unity/whats-new/unity-5.3).

## Documentation for new features in 5.3:

### [Multi-Scene Editing](http://docs.unity3d.com/Manual/MultiSceneEditing.html): 多场景编辑

同时打开和编辑多个场景，层次视图（hierarchy）中这个新功能让你可以管理多个场景，你就可以把工作分割成多个可管理的独立加载的主分支

[**New 2D Joints**](http://docs.unity3d.com/Manual/Joints2D.html)**,**[**Buoyancy Effector**](http://docs.unity3d.com/Manual/class-BuoyancyEffector2D.html)**&**[**Sprite Creator**](http://docs.unity3d.com/Manual/SpriteCreator.html): 二维关节、漂浮效果器、精灵创建

新的二维关节和能模拟液体流动的浮动效果器，精灵创建器，提升工作效率

[**OpenGL Core Support**](http://docs.unity3d.com/Manual/OpenGLCoreDetails.html): OpenGL Core的支持

OpenGL Core是OpenGL在DirectX11上对等的实现

并且在OpenGL平台上开放了很多和DirectX 11同等先进的渲染器功能

[**Euler Curve Animation Import**](http://docs.unity3d.com/Manual/AnimationEulerCurveImport.html): 欧拉曲线动画导入

导入时旋转曲线默认会被[四元数法](http://www.cprogramming.com/tutorial/3d/quaternions.html)重新采样，有时这并不符合需要并且会改变原有曲线。现在你可以选择保留旋转曲线为欧拉曲线。

[**LZ4 Asset Bundle Compression**](http://docs.unity3d.com/Manual/AssetBundleCompression.html):

一个新的对资源绑定的压缩选项，它可以部分解压绑定中需要的部分。这就避免了以前解压整个绑定时的时间浪费

A new compression option for asset bundles which allows partial decompression of only the required portion of the bundle. This avoids wait times which were previously incurred from having to decompress the entire bundle.

[**Sprite Flipping**](http://docs.unity3d.com/Manual/class-SpriteRenderer.html): 精灵翻转

选中复选框翻转精灵

[**Particle System Improvements**](http://docs.unity3d.com/Manual/PartSysRotOverLifeModule.html): 粒子化系统更新

所有的属性都可以通过脚本访问，以及 [3D Rotation](http://docs.unity3d.com/Manual/PartSysRotOverLifeModule.html) & [Emission from Skinned Mesh surfaces](http://docs.unity3d.com/Manual/PartSysShapeModule.html).

[**In-App Purchases**](http://docs.unity3d.com/Manual/UnityIAP.html)**App内购买**

让你的应用更容易的在多个商城中实现内购功能iOS App Store、 Mac App Store、 Google Play 、 Windows Store.

[**Multi-Display support**](http://docs.unity3d.com/Manual/MultiDisplay.html)**多显示支持**

同时支持最多8个摄像机输出8个不同的显示器，可用在模拟器或者户外展台上

[**Network Host Migration**](http://docs.unity3d.com/Manual/UNetHostMigration.html)**网络主机迁移**

此功能允许一客户端在主机连接丢失时接管连接并成为新的主机

[**Asynchronous Texture Upload**](http://docs.unity3d.com/Manual/AsyncTextureUpload.html)**异步材质上传**

可以分时上传**材质**数据到GPU，减少游戏主线程等待时间

[**Speedtree Improvements**](http://docs.unity3d.com/Manual/SpeedTree.html)

Multi-core batching improvements now give better performance for billboard trees.

[**JSON Serialization**](http://docs.unity3d.com/Manual/JSONSerialization.html)**Json序列化**

**更容易的相互转换Unity对象和Json**

# Working In Unity使用Unity工作

此部分展示您完整的Unity使用说明

## 本部分包含如下章节:

[Basics](http://docs.unity3d.com/Manual/UnityBasics.html) 基础：下载安装Unity，设置新项目，编辑器的快速入门

[Asset Workflow](http://docs.unity3d.com/Manual/AssetWorkflow.html) 资源工作流：如何从不同来源的文件导入图形、绘画和声音等资源。导入来自于其他开发者的包文件，或是Unity内置、商城中做好的资源

[The Main Windows](http://docs.unity3d.com/Manual/UsingTheEditor.html) 主窗口：深入了解Unity的主窗口，包括各种快捷键

[Creating Gameplay](http://docs.unity3d.com/Manual/CreatingGameplay.html) 创建游戏：介绍如何制作场景，游戏对象，组件，集成输入控制到你的项目中

[Editor Features](http://docs.unity3d.com/Manual/EditorFeatures.html) 编辑器功能

关于编辑器强大功能的介绍，帮助你定制工作流，集成外部工具，扩展编辑器

[Advanced Development](http://docs.unity3d.com/Manual/AdvancedDevelopment.html) 高级开发：

面对高级开发者，使用插件，资源绑定和高级开发技能

[Advanced Editor Topics](http://docs.unity3d.com/Manual/AdvancedEditor.html) 高级编辑器话题

完全控制编辑器，随心所欲的控制它，并且学习如何使用脚本定制资源通信和定制编辑器

[Licenses and Activation](http://docs.unity3d.com/Manual/LicensesAndActivation.html) 许可证

[Upgrade Guides](http://docs.unity3d.com/Manual/UpgradeGuides.html) 升级旧项目的向导

# 

**Basics基础**



此部分是你开始学习Unity的关键，这里会解释Unity用户界面，菜单项，使用资源，创建场景和构建发布。

当你学习完此部分时，你会了解到Unity的工作方式，如何有效的使用Unity，并且创建简单的游戏

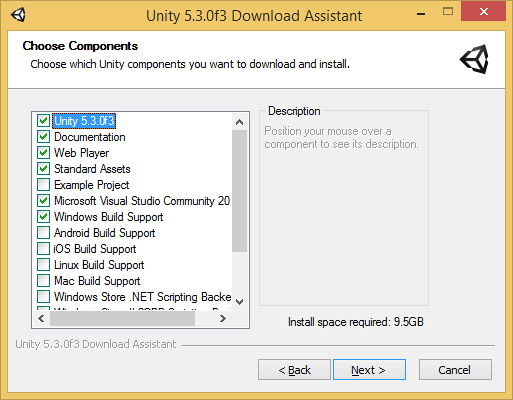
# Downloading and Installing Unity下载安装

从这里下载 [unity3d.com/download](http://unity3d.com/download).

在线安装器有详细的说明可供参考，如果你希望下载完整的安装包或者安装不同版本的Unity，你可以选择使用种子文件下载。

## Unity Download Assistant在线安装向导

从Unity5.0开始，Unity安装可以使用在线安装的方式，此向导是一个1Mb左右的可执行文件，你可以根据需要下载和安装。如果不清楚如何选择，可以选默认安装。



如果机器是PC则会安装VS2015

## Installing Unity without the Download Assistant离线安装

你可以离线安装Unity，安装过程会有更多步骤，如果公司需要批量自动化部署可以用命令行方式来安装。

### Installing Unity on Windows from command line命令行安装

下面的选项是在Windows下进行命令行安装所需的，注意区分大小写

#### Unity Editor install安装Unity编辑器

|  |  |
| --- | --- |
| /S | 静默安装 |
| /D=PATH | 静默安装时设置软件安装目录C:\Program Files (x86)\Unity\ (32-bit) or C:\Program Files\Unity\ (64-bit) |

**例如**

UnitySetup64.exe /S /D=E:\Development\Unity

静默安装Unity到根目录 E:\Development\Unity Unity编辑器的可执行文件会装在E:\Development\Unity\Editor\Unity.exe. “/D” 这个参数必须在最后并且不包括引号

#### Unity Editor uninstall卸载

静默卸载

命令行运行Uninstall.exe /S

注意，即使命令行很快执行完，程序文件也需要一会儿的时间才能移除。因为卸载器会被复制到系统临时目录下，为了确保本身也能被删除，一定要保证工作目录不在Unity安装目录下，否则将无法删除。

#### Web Player install网页播放器

静默安装（这个东西在Chrome下不能工作了）

UnityWebPlayer.exe /S

#### Standard Assets install标准资源集的安装

静默安装资源集，可选安装目录，是包含编辑器的目录，而不是包含Unity.exe的目录

UnityStandardAssetsSetup.exe /S /D=E:\Development\Unity

#### Example Project install示例项目

静默安装至C:\Users\Public\Documentation\Unity Projects\Standard Assets Example Project

UnityExampleProjectSetup.exe /S /D=E:\Development\Unity

### Installing Unity on OS X from command line苹果机的安装

默认

#### Unity Editor install

默认

#### Web Player install

默认

#### Standard Assets install

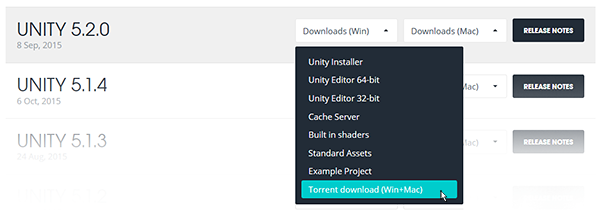
默认

#### Example Project install

默认

## Torrent Download离线安装包下载

如果用BT下载器下载，种子在这里[download archive page](http://unity3d.com/get-unity/download/archive)，目前仅有部分版本的提供离线下载，就是标注为“downloads”的



## Installing Several Versions at Once一次安装多个版本

在一台电脑上可以安装任意多版本的Unity程序，在苹果机上安装程序会创建一个Unity目录，并覆盖已有此名称的目录。在PC上安装目录名如下“Unity X.Y.Z[fp]W”f是主板本p是补丁版本

我们强烈建议你仔细选择文件夹名称，并且注意已经存在的指向旧文档的快捷方式，别名和链接都将不可用。此操作可能会是你的离线文档管理变得混乱。

Basics

# 2D or 3D Projects2维项目还是三维

Unity能创建2D和3D，如何选择？

当你在Unity中创建新项目时，你要选择2D、3D模式，你可能已经清楚自已要构建的东西，但是还有一些要注意的地方。选择2维和3维会影响Unity编辑器，比如图形是以贴图还是精灵方式导入。不用担心错误的选择，因为之后你可以在两之间任意切换，一下是一些参考

### Full 3D全3D

一些场景在商城能下载到

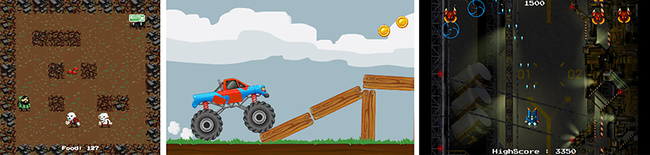
3D games usually make use of three-dimensional geometry, with materials and textures rendered on the surface of these objects to make them appear as solid environments, characters and objects that make up your game world. The camera can move in and around the scene freely, with light and shadows cast around the world in a realistic way. 3D games usually render the scene using perspective, so objects appear larger on screen as they get closer to the camera. For all games that fit this description, start in **3D** mode.

### Orthographic 3D

Some 3D games using an Orthographic view

Sometimes games use 3D geometry, but use an orthographic camera instead of perspective. This is a common technique used in games which give you a bird’s-eye view of the action, and is sometimes called “2.5D”. If you’re making a game like this, you should also use the editor in **3D** mode, because even though there is no perspective, you will still be working with 3D models and assets. You’ll need to switch your [camera](http://docs.unity3d.com/Manual/CamerasOverview.html) and [scene view](http://docs.unity3d.com/Manual/SceneViewNavigation.html) to **Orthographic** though. (scenes above from[*Synty Studios*](https://www.assetstore.unity3d.com/en/#!/publisher/5217)and[*BITGEM*](https://www.assetstore.unity3d.com/en/#!/publisher/1299))

### Full 2D

Some examples of typical 2D game types

Many 2D games use flat graphics, sometimes called sprites, which have no three-dimensional geometry at all. They are drawn to the screen as flat images, and the game’s camera has no perspective. For this type of game, you should start the editor in **2D** mode.

### 2D gameplay with 3D graphics

A side scrolling game with 2D gameplay, but 3d graphics

Some 2D games use 3D geometry for the environment and characters, but restrict the gameplay to two dimensions. For example, the camera may show a “side scrolling view” and the player can only move in two dimensions, but game still uses 3D models for the obstacles and a 3D perspective for the camera. For these games, the 3D effect may serve a stylistic rather than functional purpose. This type of game is also sometimes referred to as “2.5D”. Although the gameplay is 2D, you will mostly be manipulating 3D models to build the game so you should start the editor in **3D** mode.

### 2D gameplay and graphics, with a perspective camera

A 2D “cardboard theatre” style game, giving a parallax movement effect

This is another popular style of 2D game, using 2D graphics but with a perspective camera to get a parallax scrolling effect. This is a “cardboard theater” style scene, where all graphics are flat, but arranged at different distances from the camera. It’s most likely that **2D** mode will suit your development in this case. However, you will want to change your game [camera’s](http://docs.unity3d.com/Manual/CamerasOverview.html)projection mode to **Perspective** and the [scene view](http://docs.unity3d.com/Manual/SceneViewNavigation.html) mode to **3D**. (scene above from[*One Point Six Studio*](https://www.assetstore.unity3d.com/en/#!/publisher/8138))

### Other styles

You may have plans for a project that fits one of the above descriptions, or you may have something else entirely different or unique in mind! Whatever your plans are, hopefully the above will give you some idea which mode to start the editor in, and remember you can switch modes at any time later.

You can read about how to change the 2D/3D mode and find more detail about how the modes differ here: [2D and 3D Mode Settings](http://docs.unity3d.com/Manual/2DAnd3DModeSettings.html).

## Useful 2D Project Information

Whichever type of project you are working in, 2D or 3D, there are some useful pages to help you get started. There are also many specific pages for 2D features.

### Getting Started with Unity

* [Unity Basics](http://docs.unity3d.com/Manual/UnityBasics.html)
* [Creating Scenes](http://docs.unity3d.com/Manual/CreatingScenes.html)
* [Creating Gameplay](http://docs.unity3d.com/Manual/CreatingGameplay.html)

### 2D Overview

* [Gameplay in 2D](http://docs.unity3d.com/Manual/Overview2D.html)

### 2D Graphics

* [Sprites](http://docs.unity3d.com/Manual/Sprites.html)
* [Sprite Creator](http://docs.unity3d.com/Manual/SpriteCreator.html)
* [Sprite Editor](http://docs.unity3d.com/Manual/SpriteEditor.html)
* [Sprite Packer](http://docs.unity3d.com/Manual/SpritePacker.html)
* [Sprite Renderer](http://docs.unity3d.com/Manual/class-SpriteRenderer.html)

### 2D Physics

* [Physics](http://docs.unity3d.com/Manual/PhysicsSection.html) - An introduction to 2D and 3D physics in Unity.
* [Rididbody 2D](http://docs.unity3d.com/Manual/class-Rigidbody2D.html) - Placing a game object under control of the physics 2D engine.
* [Physics 2D Manager](http://docs.unity3d.com/Manual/class-Physics2DManager.html) - Control global settings for 2D physics.

**2D Joints**

* [Distance Joint](http://docs.unity3d.com/Manual/class-DistanceJoint2D.html)
* [Fixed Joint 2D](http://docs.unity3d.com/Manual/class-FixedJoint2D.html)
* [Friction Joint 2D](http://docs.unity3d.com/Manual/class-FrictionJoint2D.html)
* [Hinge Joint 2D](http://docs.unity3d.com/Manual/class-HingeJoint2D.html)
* [Relative Joint 2D](http://docs.unity3d.com/Manual/class-RelativeJoint2D.html)
* [Slider Joint 2D](http://docs.unity3d.com/Manual/class-SliderJoint2D.html)
* [Spring Joint 2D](http://docs.unity3d.com/Manual/class-SpringJoint2D.html)
* [Target Joint 2D](http://docs.unity3d.com/Manual/class-TargetJoint2D.html)
* [Wheel Joint 2D](http://docs.unity3d.com/Manual/class-WheelJoint2D.html)

**2D Colliders**

* [Box Collider 2D](http://docs.unity3d.com/Manual/class-BoxCollider2D.html)
* [Circle Collider2D](http://docs.unity3d.com/Manual/class-CircleCollider2D.html)
* [Edge Collider 2D](http://docs.unity3d.com/Manual/class-EdgeCollider2D.html)
* [Polygon Collider 2D](http://docs.unity3d.com/Manual/class-PolygonCollider2D.html)

**2D Physics Material**

* [Physics Material 2D](http://docs.unity3d.com/Manual/class-PhysicsMaterial2D.html)

**2D Effectors**

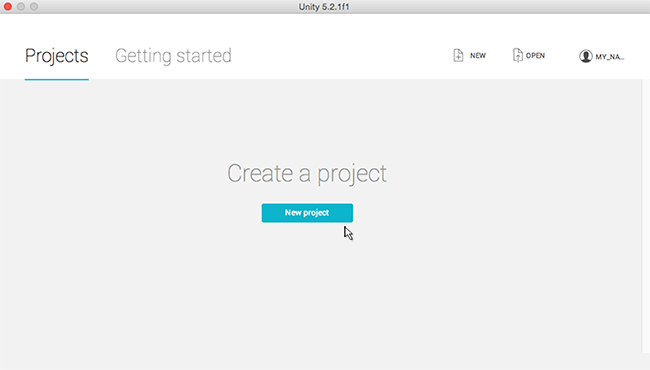
* [Area Effector 2D](http://docs.unity3d.com/Manual/class-AreaEffector2D.html)
* [Buoyancy Effector 2D](http://docs.unity3d.com/Manual/class-BuoyancyEffector2D.html)
* [Platform Effector 2D](http://docs.unity3d.com/Manual/class-PlatformEffector2D.html)
* [Point Effector 2D](http://docs.unity3d.com/Manual/class-PointEffector2D.html)
* [Surface Editor 2D](http://docs.unity3d.com/Manual/class-SurfaceEffector2D.html)

# Getting Started

## Starting Unity for the First Time

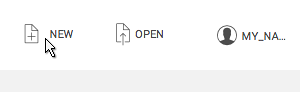
Whenever you launch the Unity editor, the Home Screen displays. If you have no existing Unity projects on your computer, or Unity doesn’t know where they are, it asks you to create a project.

To get started, you can click on New project which will take you to the Home Screen’s Create Project view. See the section on this in Creating a Project to find out more. Alternatively, if you already have a Unity project on your computer, you can open it from this screen. See Opening a Project to find out more.

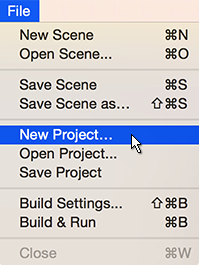
The Home Screen displays on launch, click on ‘New project’ to get started

## Creating a Project

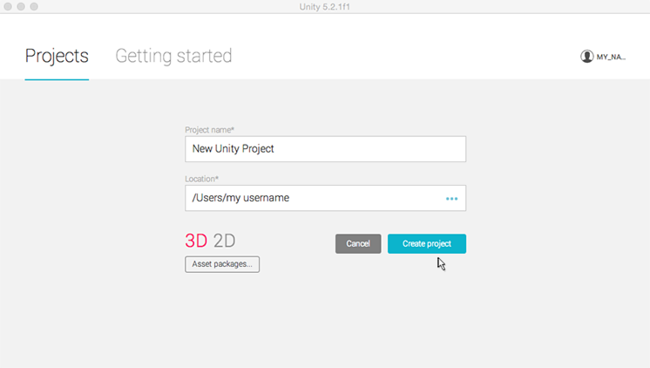
Whenever you start the Unity editor, the Home Screen displays. From here, you can select NEW in the top right corner, to switch to the Create Project view.

In the top right corner of the Home Screen, select ‘New’ to see the Create Project view

To bring up the Home Screen’s Create Project view when you are already in the Unity editor, select New Project… from theFile menu.

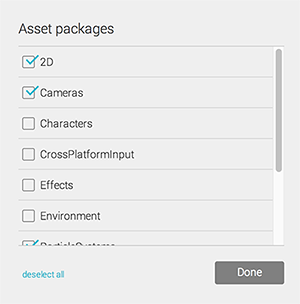
File>New Project - Displays the Home Screen’s Create Project view from within the Unity editor

From the Home Screen’s Create Project view, you can name, set options, and specify the location of your new project.

The Home Screen’s Create Project view

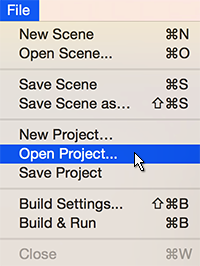
To create a new project:

1. The name defaults to New Unity Project but you can change it to whatever you want. Type the name you want to call your project into the Project name field.
2. The location defaults to your home folder on your computer but you can change it. **EITHER** (a) Type where you want to store your project on your computer into the Location field. **OR** (b) Click on the three blue dots **‘…’**. This brings up your computer’s Finder (Mac OS X) or File Explorer (Windows OS).
3. Then, in Finder or File Explorer, select the project folder that you want to store your new project in, and select “Choose”.
4. Select 3D or 2D for your project type. The default is 3D, coloured red to show it is selected. (The 2D option sets the Unity editor to display its 2D features, and the 3D option displays 3D features. If you aren’t sure which to choose, leave it as 3D; you can change this setting later.)
5. There is an option to select Asset packages… to include in your project. Asset packages are pre-made content such as images, styles, lighting effects, and in-game character controls, among many other useful game creating tools and content. The asset packages offered here are free, bundled with Unity, which you can use to get started on your project.**EITHER:** If you don’t want to import these bundled assets now, or aren’t sure, just ignore this option; you can add these assets and many others later via the Unity editor. **OR:** If you do want to import these bundled assets now, select Asset packages… to display the list of assets available, check the ones you want, and then click on Done.
6. Now select Create project and you’re all set!

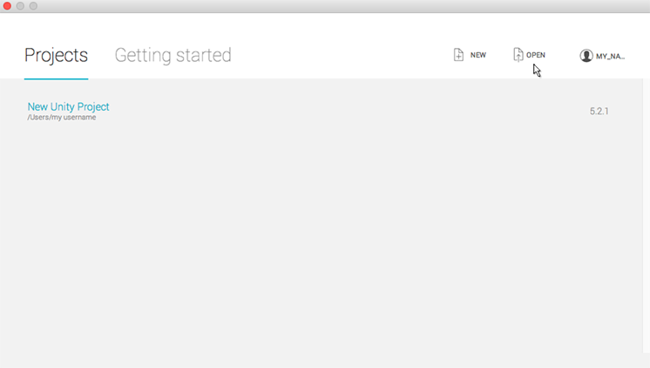
Asset packages options - You can choose to add them now or ignore this option and add them later

## Opening a Project

When you start the Unity editor, the Home Screen’s Open Project view displays. From here you can choose the project you want to open. To bring up the Home Screen’s Open Project view when you are already in the Unity editor, select Open Project from the File menu.

File>Open Project - Displays the Home Screen’s Open Project view from within the Unity editor

The Home Screen’s Open Project view lists all the projects the Unity editor knows about. (If the editor is newly installed and doesn’t know the location of your existing projects, it prompts you to create a new project. See Starting Unity for the First Time to find out more.)

The Home Screen’s Open Project view - Select ‘Open’ to locate and open an existing project which isn’t listed

Click on any of the projects listed to open them. If your project is not listed, you need to tell the editor where it is.

To locate and open an existing project which isn’t listed:

1. Select Open. This brings up your computer’s Finder (Mac OS X) or File Explorer (Windows OS).
2. In Finder or File Explorer, select the project folder that you want to open and select “Open”.

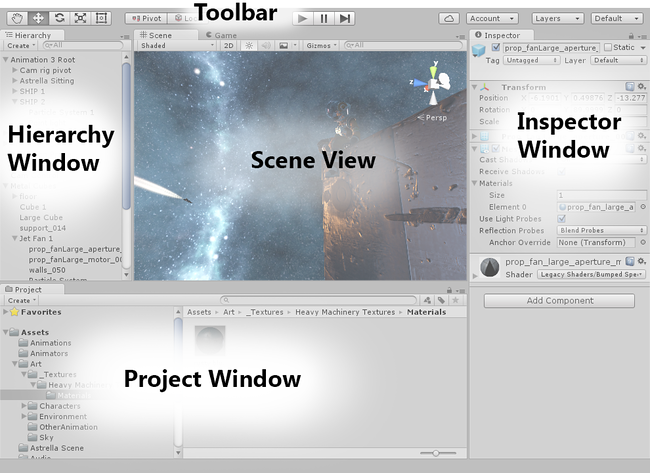
(**NOTE**: To open a Unity project, there is no specific Unity project file that you select. A Unity project is a collection of files, so you need to tell the Unity editor to open a folder, rather than a specific file.)

# Learning the Interface

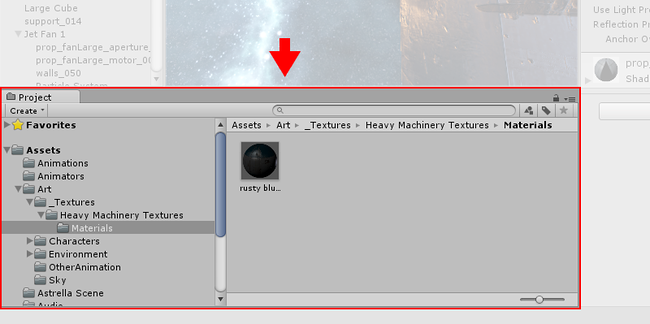
Take your time to look over the editor interface and familiarize yourself with it. The main editor window is made up of tabbed windows which can be rearranged, grouped, detatched and docked.

This means the look of the editor can be different from one project to the next, and one developer to the next, depending on personal preference and what type of work you are doing.

The default arrangement of windows gives you practical access to the the most common windows. If you’re not yet familiar with the different windows in Unity, you can identify the, by the name in the tab. The most common and useful windows are shown in their default positions, below:

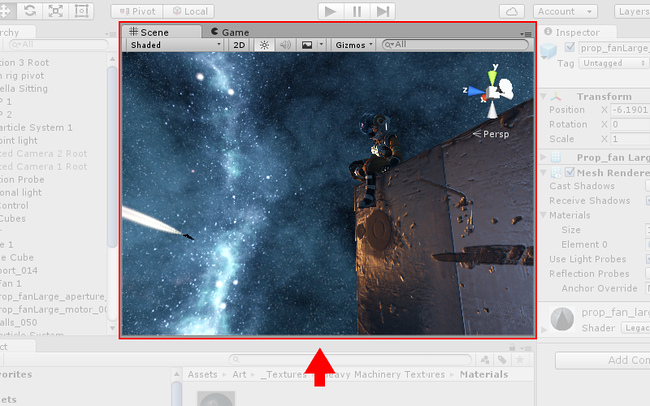


## The Project Window



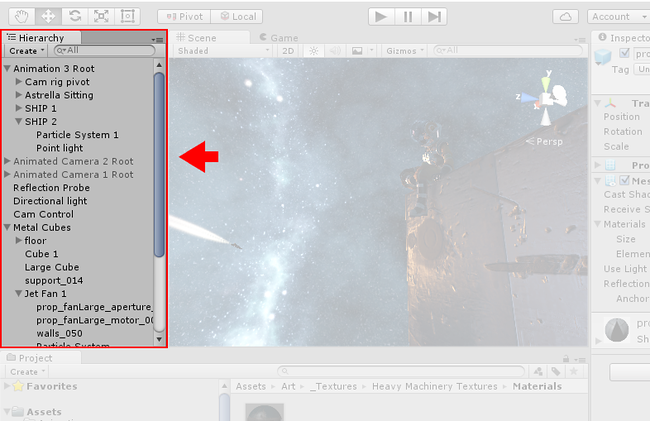
**The Project Window** displays your library of assets that are available to use in your project. When you import assets into your project, they appear here. Find out more about the [Project Window](http://docs.unity3d.com/540/Documentation/Manual/ProjectView.html).

## The Scene View



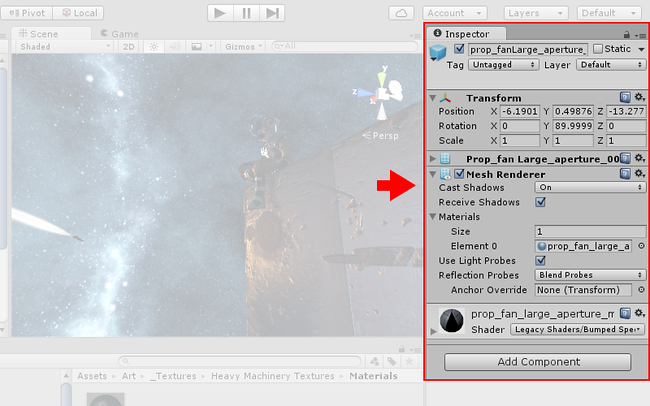
**The Scene View** allows you to visually navigate and edit your scene. The scene view can show a 3D or 2D perspective, depending on the type of project you are working on. Find out more about the [Scene View](http://docs.unity3d.com/540/Documentation/Manual/UsingTheSceneView.html) and the [Game View](http://docs.unity3d.com/540/Documentation/Manual/GameView.html).

## The Hierarchy Window



**The Hierarchy Window** is a hierarchical text representation of every object in the scene. Each item in the scene has an entry in the hierarchy, so the two windows are inherently linked. The hierarchy reveals the structure of how objects are attached to one another. Find out more about the [Hierarchy Window](http://docs.unity3d.com/540/Documentation/Manual/Hierarchy.html).

## The Inspector Window



**The Inspector Window** allows you to view and edit all the properties of the currently selected object. Because different types of objects have different sets of properties, the layout and contents of the inspector window will vary. Find out more about the [Inspector Window](http://docs.unity3d.com/540/Documentation/Manual/UsingTheInspector.html).

The Toolbar  


**The Toolbar** provides access to the most essential working features. On the left it contains the basic tools for manipulating the scene view and the objects within it. In the centre are the play, pause and step controls. The buttons to the right give you access to your Unity Cloud Services and your Unity Account, followed by a layer visibility menu, and finally the editor layout menu (which provides some alternate layouts for the editor windows, and allows you to save your own custom layouts).

The toolbar is not a window, and is the only part of the Unity interface that you can’t rearrange.

Find out more about the [Toolbar](http://docs.unity3d.com/540/Documentation/Manual/Toolbar.html).

# Asset Workflow

These steps will give you a general overview of the basic principles of working with assets in Unity.

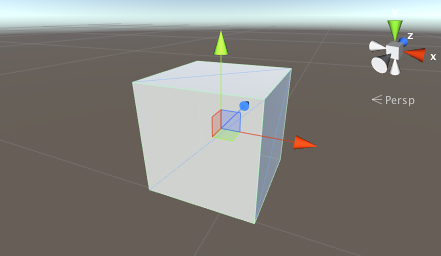
An asset is representation of any item that can be used in your game or project. An asset may come from a file created outside of Unity, such as a 3D model, an audio file, an image, or any of the other types of file that Unity supports. There are also some asset types that can be created within Unity, such as an Animator Controller, an Audio Mixer or a Render Texture.

# Some of the asset types that can be imported into UnitySome of the asset types that can be imported into Unity

# Primitive and Placeholder Objects

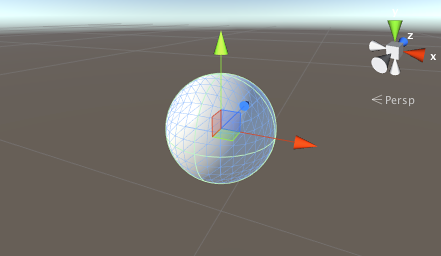
Unity can work with 3D models of any shape that can be created with modelling software. However, there are also a number of primitive object types that can be created directly within Unity, namely the Cube, Sphere, Capsule, Cylinder,Plane and Quad. These objects are often useful in their own right (a plane is commonly used as a flat ground surface, for example) but they also offer a quick way to create placeholders and prototypes for testing purposes. Any of the primitives can be added to the scene using the appropriate item on the GameObject > 3D Object menu.

## Cube



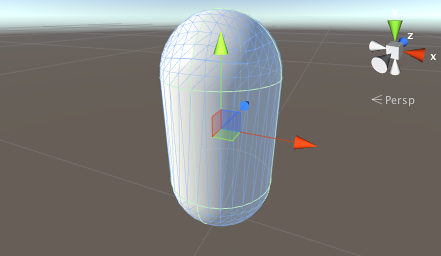
This is a simple cube with sides one unit long, textured so that the image is repeated on each of the six faces. As it stands, a cube isn’t really a very common object in most games but when scaled, it is very useful for walls, posts, boxes, steps and other similar items. It is also a handy placeholder object for programmers to use during development when a finished model is not yet available. For example, a car body can be crudely modelled using an elongated box of roughly the right dimensions. Although this is not suitable for the finished game, it is fine as a simple representative object for testing the car’s control code. Since a cube’s edges are one unit in length, you can check the proportions of a mesh imported into the scene by adding a cube close by and comparing the sizes.

## Sphere



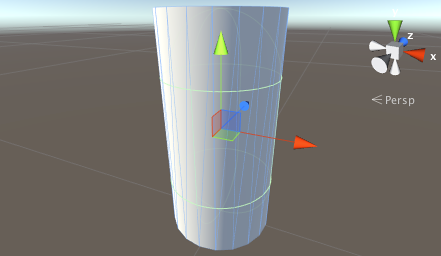
This is a sphere of unit diameter (ie, 0.5 unit radius), textured so that the entire image wraps around once with the top and bottom “pinched” at the poles. Spheres are obviously useful for representing balls, planets and projectiles but a semi-transparent sphere can also make a nice GUI device for representing the radius of an effect.

## Capsule



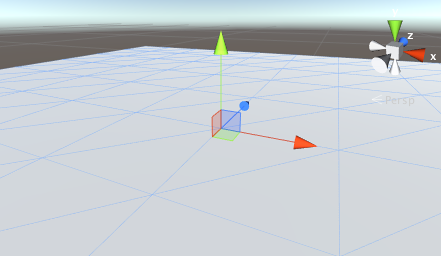
A capsule is a cylinder with hemispherical caps at the ends. The object is one unit in diameter and two units high (the body is one unit and the two caps are half a unit each). It is textured so that the image wraps around exactly once, pinched at each hemisphere’s apex. While there aren’t many real world objects with this shape, the capsule is a useful placeholder for prototyping. In particular, the physics of a rounded object are sometimes better than those of a box for certain tasks.

## Cylinder



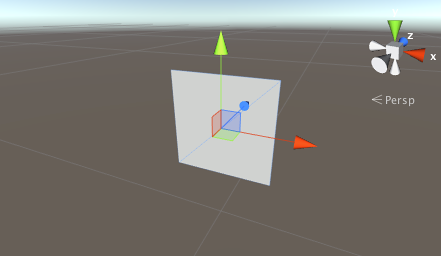
This is a simple cylinder which is two units high and one unit in diameter, textured so that the image wraps once around the tube shape of the body but also appears separately in the two flat, circular ends. Cylinders are very handy for creating posts, rods and wheels but you should note that the shape of the collider is actually a capsule (there is no primitive cylinder collider in Unity). You should create a mesh of the appropriate shape in a modelling program and attach a mesh collider if you need an accurate cylindrical collider for physics purposes.

## Plane



This is a flat square with edges ten units long oriented in the XZ plane of the local coordinate space. It is textured so that the whole image appears exactly once within the square. A plane is useful for most kinds of flat surface, such as floors and walls. A surface is also needed sometimes for showing images or movies in GUI and special effects. Although a plane can be used for things like this, the simpler quad primitive is often a more natural fit to the task.

## Quad

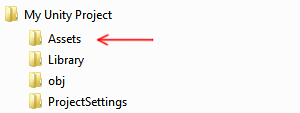


The quad primitive resembles the plane but its edges are only one unit long and the surface is oriented in the XY plane of the local coordinate space. Also, a quad is divided into just two triangles whereas the plane contains two hundred. A quad is useful in cases where a scene object must be used simply as a display screen for an image or movie. Simple GUI and information displays can be implemented with quads, as can particles, sprites and “impostor” images that substitute for solid objects viewed at a distance.

# Importing Assets

Assets created outside of Unity must be brought in to Unity by having the file either saved directly into the “Assets” folder of your project, or copied into that folder. For many common formats, you can save your source file directly into your project’s Assets folder and Unity will be able to read it. Unity will notice when you save new changes to the file and will re-import as necessary.

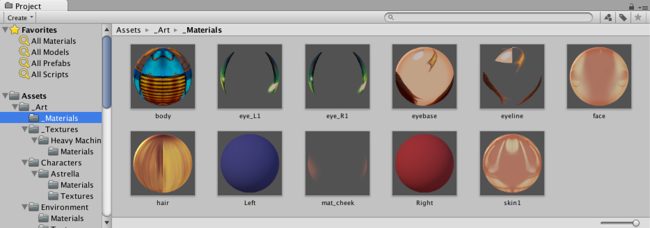
When you create a Unity Project, you are creating a folder - named after your project - which contains the following subfolders:

The basic file structure of a Unity Project

The **Assets** folder is where you should save or copy files that you want to use in your project.

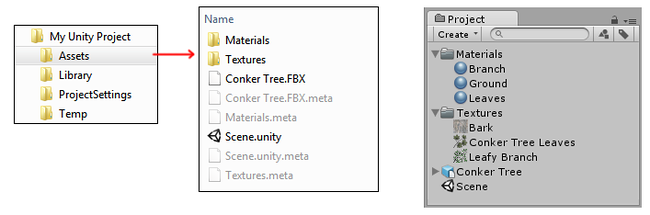
The contents of the **Project Window** in Unity shows the items in your Assets folder. So if you save or copy a file to your Assets folder, it will be imported and become visible in your Project Window.

Unity will automatically detect files as they are added to **Assets** folder, or if they are modified. When you put any asset into your Assets folder, you will see the asset appear in your Project View.

The Project Window shows assets that have been imported into your project

If you drag a file into Unity’s Project Window from your computer (eg, from the Finder on Mac, or from Explorer on Windows), it will be copied into your Assets folder, and will appear in the Project window.

The items you see in your Project window represent (in most cases) actual files on your computer, and if you delete them within Unity, you are deleting them from your computer too.

The relationship between the Assets Folder in your Unity Project on your computer, and the Project Window within Unity

The above image shows an example of a few files and folders inside the Assets folder of a Unity project. You can create as many folders as you like and use them to organise your Assets.

You’ll notice in the image above that there are **.meta** files listed in the file system, but not visible in Unity’s Project Window. Unity creates these .meta files for each asset and folder, but they are [hidden](https://en.wikipedia.org/wiki/Hidden_file_and_hidden_directory) by default, so you may not see them in your Explorer/Finder either.

They contain important information about how the asset is used in the project and they must stay with the asset file they relate to, so if you move or rename an asset file in Explorer/Finder, you must also move/rename the meta file to match.

The simplest way to safely move or rename your assets is to always do it from within Unity’s project folder. This way, Unity will automatically move or rename the corresponding meta file. If you like, you can read more about .meta files and what goes on [behind-the-scenes during the import process](http://docs.unity3d.com/540/Documentation/Manual/BehindtheScenes.html).

If you want to bring collections of assets into your project, you can use Asset Packages. See [Asset Packages](http://docs.unity3d.com/540/Documentation/Manual/AssetPackages.html) for more details.

## Some common types of Asset

### Image Files

Most common image file types are supported, such as BMP, TIF, TGA, JPG, and PSD. If you save your layered Photoshop (.psd) files into your Assets folder, they will be imported as flattened images. You can find out more about [importing images with alpha channels from photoshop](http://docs.unity3d.com/540/Documentation/Manual/HOWTO-alphamaps.html), or [importing your images as sprites](http://docs.unity3d.com/540/Documentation/Manual/SpriteEditor.html)

### 3D Model Files

If you save your 3D files from most common 3D software packages in their native format (eg, .max, .blend, .mb, .ma) into your Assets folder, they will be imported by calling back to your 3D package’s FBX export plugin (\*). Alternatively you can export as FBX from your 3D app into your Unity project. Read more about [importing 3D files from your 3D app](http://docs.unity3d.com/540/Documentation/Manual/HOWTO-importObject.html).

### Meshes & Animations

Whichever 3D package you are using, Unity will import the meshes and animations from each file. For a list of applications that are supported by Unity, please see [this page](http://docs.unity3d.com/540/Documentation/Manual/HOWTO-importObject.html).

Your mesh file does not need to have an animation to be imported. If you do use animations, you have your choice of importing all animations from a single file, or importing separate files, each with one animation. For more information about importing animations, please see [Importing animations](http://docs.unity3d.com/540/Documentation/Manual/AnimationsImport.html).

### Audio Files

If you save uncompressed audio files into your Assets folder, they will be imported according to the compression settings specified. Read more about [importing audio files](http://docs.unity3d.com/540/Documentation/Manual/AudioFiles.html).

### Other Asset Types

In all cases, your original source file is never modified by Unity, even though within Unity you can often choose between various ways to compress, modify, or otherwise process the asset. The import process reads your source file, and creates a game-ready representation of your asset internally, matching your chosen import settings. If you modify the import settings for an asset, or make a change to the source file in the Asset folder, will cause Unity to re-import the asset again to reflect your new changes.

**Note**: Importing native 3D formats requires the 3D application to be installed on the same computer as Unity. This is because Unity uses the 3D application’s FBX exporter plug-in to read the file. Alternatively, you can directly export as FBX from your application and save into the Projects folder.

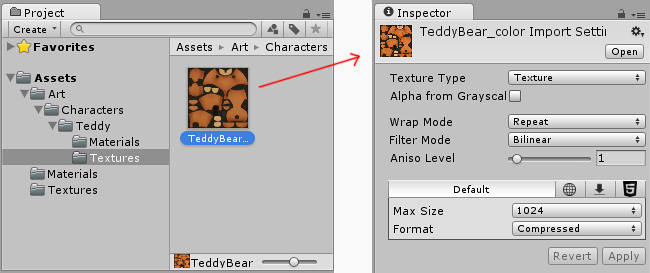
See also:

* [Asset Packages](http://docs.unity3d.com/540/Documentation/Manual/AssetPackages.html)
* [Importing Meshes](http://docs.unity3d.com/540/Documentation/Manual/class-FBXImporter.html)
* [3D formats](http://docs.unity3d.com/540/Documentation/Manual/3D-formats.html)
* [Animation Import](http://docs.unity3d.com/540/Documentation/Manual/AnimationsImport.html)
* [Materials and Shaders](http://docs.unity3d.com/540/Documentation/Manual/Materials.html)
* [Textures and Videos](http://docs.unity3d.com/540/Documentation/Manual/Textures.html)
* [Sprite Editor](http://docs.unity3d.com/540/Documentation/Manual/SpriteEditor.html)
* [Sprite Packer](http://docs.unity3d.com/540/Documentation/Manual/SpritePacker.html)
* [Procedural Materials](http://docs.unity3d.com/540/Documentation/Manual/ProceduralMaterials.html)
* [Audio Files](http://docs.unity3d.com/540/Documentation/Manual/AudioFiles.html)
* [Tracker Modules](http://docs.unity3d.com/540/Documentation/Manual/TrackerModules.html)

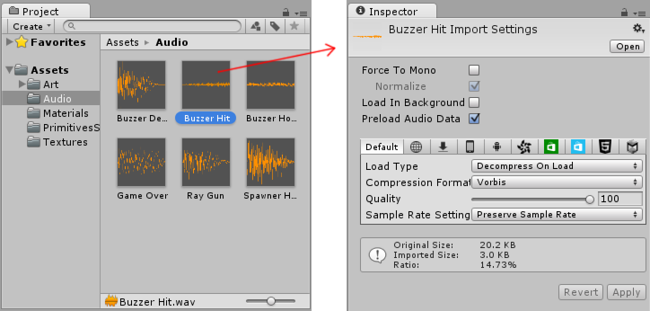
# Import Settings

Each type of asset that Unity supports has a set of Import Settings, which affect how the asset appears or behaves. To view an asset’s import settings, select the asset in the Project View. The import settings for this asset will appear in theInspector. The options that are displayed will vary depending on the type of asset that is selected.

For example, the import settings for an image will allow you to choose whether it’s imported as a texture, a 2D sprite, or a normal map. The import settings for an FBX file allow you to adjust the scale, generate normals or lightmap coordinates, and split & trim animation clips defined in the file.

Clicking on an image asset in the Project Window shows the import settings for that asset in the Inspector

For other asset types, the import settings will look different. The various settings you see will relate to the type of asset selected. Here’s an example of an Audio asset, with its related import settings shown in the inspector.

An Audio asset selected in the Project Window shows the Audio import settings for that asset in the Inspector

If you are developing a cross-platform project, you can override the “default” settings and assign different import settings on a per-platform basis.

# Importing from the Asset Store

The Unity Asset Store is home to a growing library of free and commercial assets created both by Unity Technologies and also members of the community. A wide variety of assets is available, covering everything from textures, models and animations to whole project examples, tutorials and Editor extensions. The assets are accessed from a simple interface built into the Unity Editor and are downloaded and imported directly into your project.

Unity users can become publishers on Asset Store, and sell content they have created. To find out more, see [Asset Store Publishing](http://docs.unity3d.com/540/Documentation/Manual/AssetStorePublishing.html).

# Asset Store Access and Navigation

You can open the Asset Store window by selecting Window->AssetStore from the main menu. On your first visit, you will be prompted to create a free user account which you will use to access the Store subsequently.

The Asset Store front page.

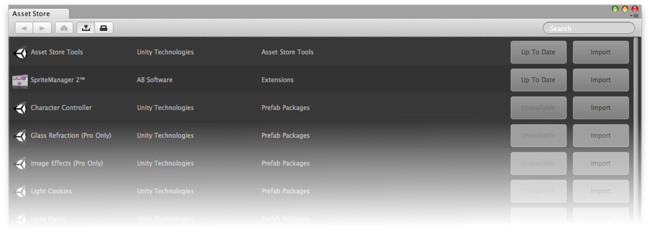
The Store provides a browser-like interface which allows you to navigate either by free text search or by browsing packages and categories. To the left of the main tool bar are the familiar browsing buttons for navigating through the history of viewed items:-

http://docs.unity3d.com/540/Documentation/uploads/Main/AssetStore-Navigation.png

To the right of these are buttons for viewing the Download Manager and for viewing the current contents of your shopping cart.

http://docs.unity3d.com/540/Documentation/uploads/Main/AssetStore-ToolButtons.png

The Download Manager allows you to view the packages you have already bought and also to find and install any updates. Additionally, the standard packages supplied with Unity can be viewed and added to your project with the same interface.

The Download Manager.

## Location of Downloaded Asset Files

You will rarely, if ever, need to access the files downloaded from the Asset Store directly. However, if you do need to, you can find them in

~/Library/Unity/Asset Store

…on the Mac and in

C:\Users\accountName\AppData\Roaming\Unity\Asset Store

…on Windows. These folders contain subfolders that correspond to particular Asset Store vendors - the actual asset files are contained in the appropriate subfolders.

# Asset Packages

Unity packages are a handy way of sharing and re-using Unity projects and collections of assets; Unity Standard Assetsand items on the Unity Asset Store are supplied in packages, for example.  
Packages are collections of files and data from Unity projects, or elements of projects, which are compressed and stored in one file, similar to Zip files. Like Zip files, a package maintains its original directory structure when it is unpacked, as well as meta-data about assets (such as import settings and links to other assets).

In Unity, the menu option Export Package compresses and stores the collection, while Import Package unpacks the collection into your currently open Unity project.

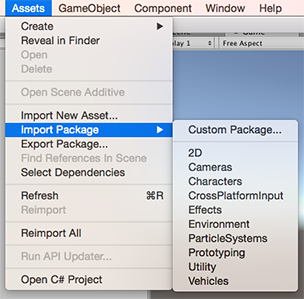
This page contains information on:

* Import Package: - Standard Asset Packages - Custom Packages
* Export Package
* Exporting Updated Packages

## Import Package

You can import Standard Asset Packages, which are asset collections pre-made and supplied with Unity, and Custom Packages, which are made by people using Unity.

Choose Assets > Import Package > to import both types of package.

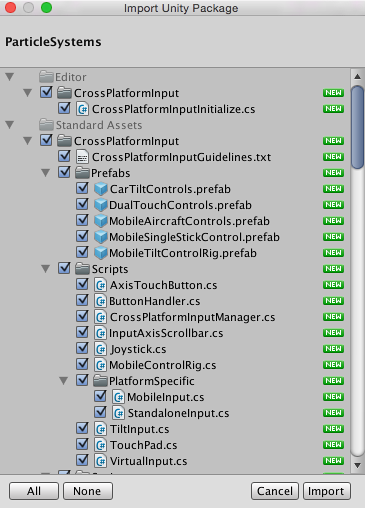
Fig 1: Asset>Import Package menu

### Standard Asset Packages

Unity ‘Standard Assets’ consist of several different packages: 2D, Cameras, Characters, CrossPlatformInput, Effects, Environment, ParticleSystems, Prototyping, Utility, Vehicles.

To import a new Standard Asset package:

1. Open the project you want to import assets into.
2. Choose Assets > Import Package > plus the name of the package you want to import, and the Import Unity Package dialog box displays, with all the items in the package pre-checked, ready to install. (See Fig 2: New install Import Unity Package Dialog Box.)
3. Select Import and Unity puts the contents of the package into a Standard Asset folder, which you can access from yourProject View.

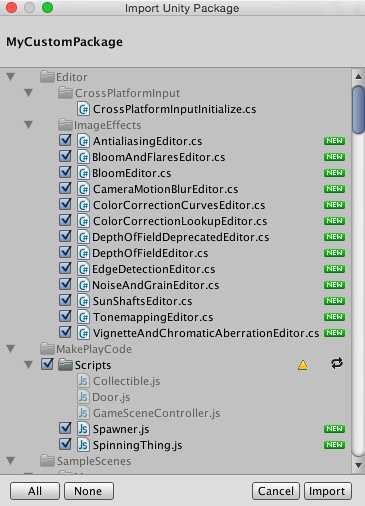
Fig 2: New install Import Unity Package dialog box

### Custom Packages

You can import custom packages which have been exported from your own projects or from projects made by other Unity users.

To import a new custom package:

1. Open the project you want to import assets into.
2. Choose Assets > Import Package > Custom Package… to bring up up File Explorer (Windows) or Finder (Mac).
3. Select the package you want from Explorer or Finder, and the Import Unity Package dialog box displays, with all the items in the package pre-checked, ready to install. (See Fig 4: New install Import Unity Package dialog box.)
4. Select Import and Unity puts the contents of the package into the Assets folder, which you can access from yourProject View.

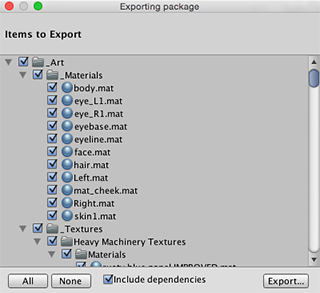
Fig 4: New install Import Unity Package dialog box

## Export Package

Use Export Package to create your own Custom Package.

1. Open the project you want to export assets from.
2. Choose Assets > Export Package… from the menu to bring up the Exporting Package dialog box. (See Fig 6: Exporting Package dialog box.)
3. In the dialog box, select the assets you want to include in the package by clicking on the boxes so they are checked.
4. Leave the include dependencies box checked to auto-select any assets used by the ones you have selected.
5. Click on Export to bring up File Explorer (Windows) or Finder (Mac) and choose where you want to store your package file. Name and save the package anywhere you like.

**HINT:** When exporting a package Unity can export all dependencies as well. So, for example, if you select a Scene and export a package with all dependencies, then all models, textures and other assets that appear in the scene will be exported as well. This can be a quick way of exporting a bunch of assets without manually locating them all.

Fig 6: Exporting Package dialog box

## Exporting Updated Packages

Sometimes you may want to change the contents of a package and create a newer, updated version of your asset package. To do this:

* Select the asset files you want in your package (select both the unchanged ones and the new ones).
* Export the files as described above in **Export Package**, above.

**NOTE:** You can re-name an updated package and Unity will recognise it as an update, so you can use incremental naming, for example: MyAssetPackageVer1, MyAssetPackageVer2.

**HINT:** It is not good practise to remove files from packages and then replace them with the same name: Unity will recognise them as different and possibly conflicting files and so display a warning symbol when they are imported. If you have removed a file and then decide to replace it, it is better to give it a different, but related name to the original.

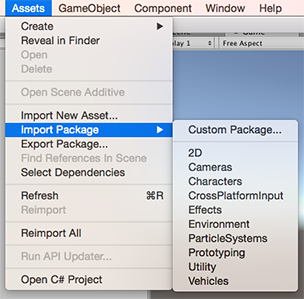
# Standard Assets

Unity ships with multiple Standard Assets. These are collections of assets that are widely used by most Unity customers. These are: 2D, Cameras, Characters, CrossPlatformInput, Effects, Environment, ParticleSystems, Prototyping, Utility, Vehicles.

Unity transfers Standard Assets into and out of projects using Unity packages.

## Importing Standard Assets

See [Asset Packages](http://docs.unity3d.com/540/Documentation/Manual/AssetPackages.html) for information about installing (importing) Standard Assets, as well as sharing Custom Packagesbetween projects and via the Asset Store.

Asset>Import Package menu

#### Standard Assets Do Not Upgrade Automatically

When you create a new project in Unity, you can choose to include Standard Assets collections in your project. Unity copies the assets you choose to include from the Unity install folder into your new project folder. This means that if you upgrade your Unity Editor to a newer version, the Standard Assets you have already imported into your project do not upgrade: You have to manually upgrade them.

**HINT:** A newer version of a Standard Asset might behave differently to your existing install (for performance or quality reasons, for example). A newer version might make your project look or behave differently and you may need to re-tweak its parameters. Check the package contents and Unity’s release notes before you decide to re-install.

Asset Packages

# 