Unity 3D

Everything About Unity

Reference:

http://docs.unity3d.com/Documentation/Manual/index.html

Contents:

- Introduction
- Installing
- Unity Basics
- Scripting
- Assets
- Scenes
- Game Objects
- Prefabs
- Build

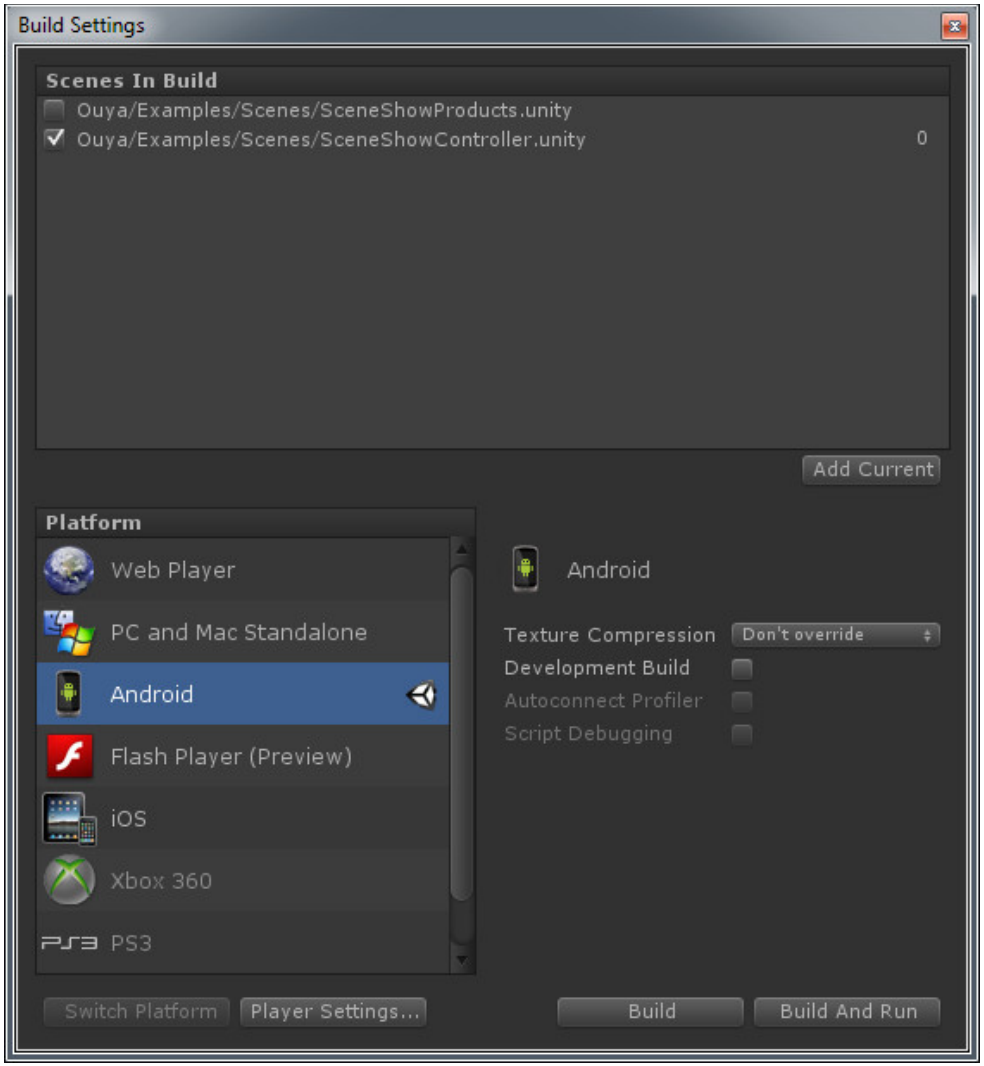
Graphics of Unity:

A screen shot of the game created using Unity 3D Engine for PC.

As you can see in the screen shot that the graphics are realistic. See how the shadow of tree is accordingly to the sun light and the rest of graphics are also giving it a realistic view. You also play these type of games now a days in your PC's. Ever wonder how they are created? Well now you can do it using Unity Engine.

Platform:

The extension of a unity file is .unity. You can make games on unity for different platforms that include Android and iOS.



- This screen shot is showing the platforms on which you can export your .unity file.
- So now you see the unity is a huge tool if you are interested in game development.
- However, in these slides we will only cover Unity for PC games.
- Unity is a freeware software. But it is now freeware if you want to use it for commercial purpose. (Selling your games in market)

What is Unity?

Unity is a multi-platform, integrated IDE for scripting games, and working with 3D virtual worlds Including:

- Game engine
  3D objects / lighting / physics / animation / scripting

- Accompanying script editor
  MonoDevelop (win/mac) << RECOMMENDED TO USE
  Can also use Visual Studio (Windows)

- 3D terrain editor
- 3D object animation manager
- GUI system
- Executable exporter many platforms:
  Native application / web player / iPhone / Android / Wii

Installing Unity 3D

System Requirements

Windows: XP SP2 or later;
Mac OS X: Intel CPU & "Snow Leopard" 10.6 or later.
Note that Unity was not tested on server versions of Windows and OS X.
Graphics card with DirectX 9 level (shader model 2.0) capabilities. Any card made since 2004 should work.
Using Occlusion Culling requires GPU (Graphics Processing Unit) with Occlusion Query support (some Intel GPUs do not support that).

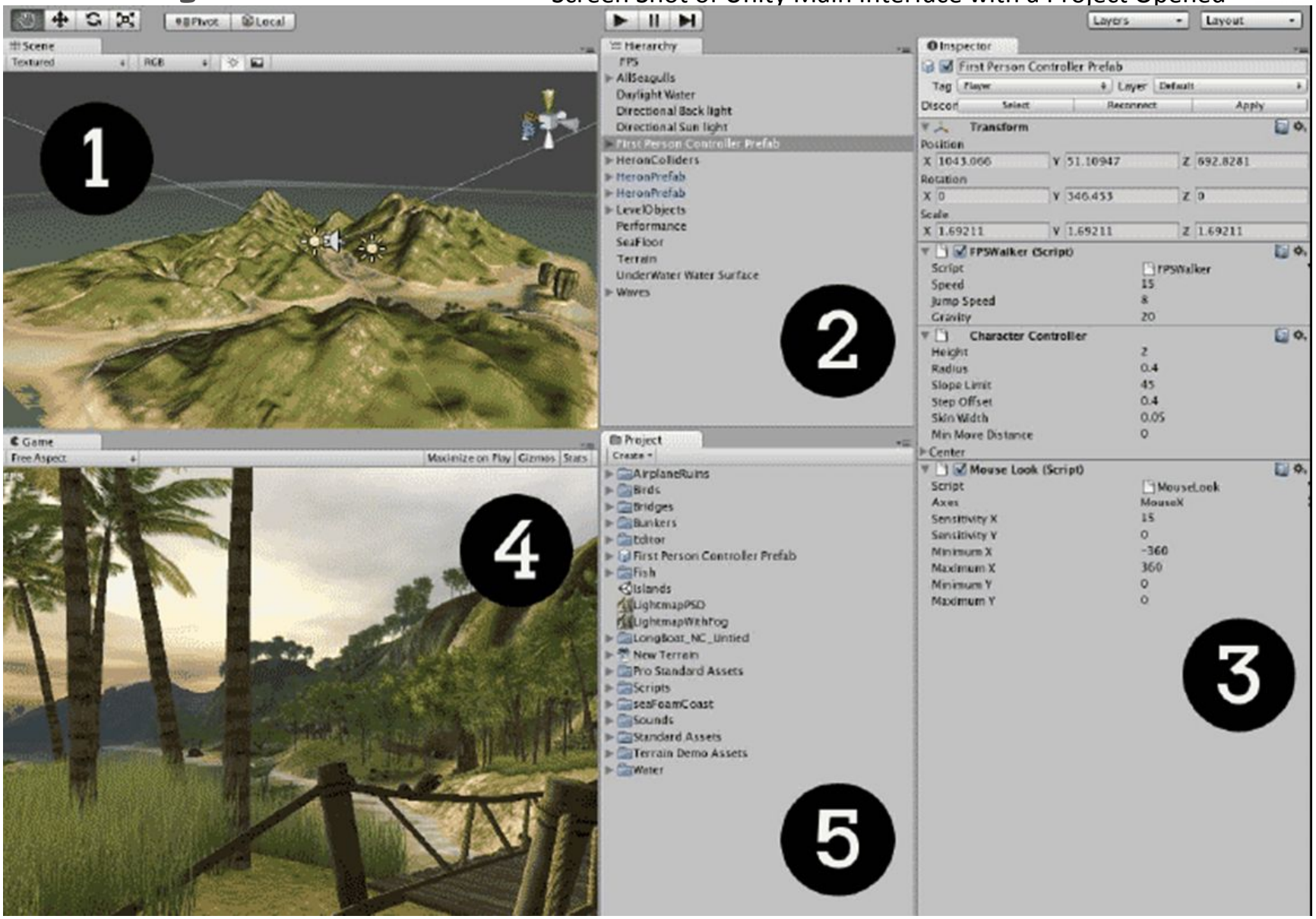
Installation

Installing Unity 3D is simple.

- Download the latest Unity distribution from http://unity3d.com/unity/download/
- Run the installer with default settings.
- Run Unity for the first time. It may open the demo project, or it will ask you to create a project. Open the demo or create a new project without changing any settings.
- Follow the instructions to register online, fill out your email address and just use the free/indie license.
- Go to File > Open Project
- When your project will be loaded, it will open in unity. More on next slide

Unity Basics

Screen Shot of Unity Main Interface with a Project Opened



Unity Basics

1 – Scene

Scene is same as Hierarchy(2) = same, just diff. views

Editable (design-time) 3D game objects in the current scene

2 – Hierarchy

Text list of game objects and sub-objects in the current scene

3 – Inspector

Properties for currently selected

4 – Game

Preview how game will look when executing

5 – Project

Contents of Project ‘assets’ folder (i.e. files in that folder)
– library of scripts, digital media files, and scenes

Unity Basics

This is the top most bar below Menu Bar in Unity Main Interface Window.

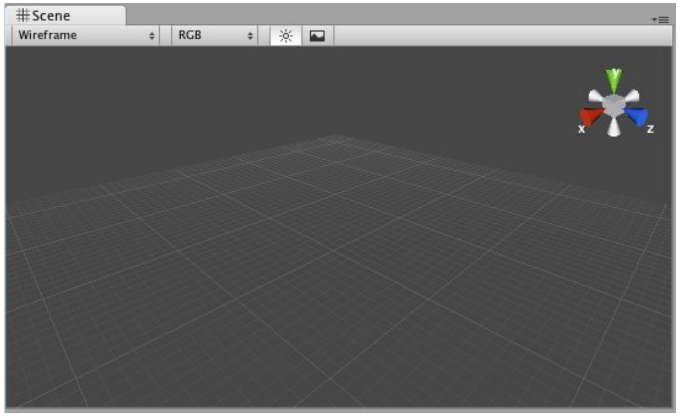
Click-drag to drag the camera around
Hold **Alt** and click-drag to orbit the camera around the current pivot point.
Hold **Control** and click-drag to zoom the camera.

These three buttons help you test your game and control playback. As you've seen, the **Play** button starts and stops your game. The **Pause** button works as expected—it pauses your game so that you can make changes to it on the fly. The third button is a **Step-Through** control; use it to advance frame-by-frame through your game so that you can more tightly control what's going on.

Above the **Inspector** panel, you'll see the **Layers** and **Layout** dropdowns. **Game Objects** can be grouped into layers, much like in Photoshop or Flash. Unity stores a few commonly used layouts in the **Layout** dropdown. You can also save and load your own custom layouts.

The Scene window

The **Scene** window is where you can position your Game Objects and move things around. This window has various controls to change its level of detail. Use these controls to toggle lighting on and off, and to display the window contents with textures, wireframes, or a combination of both. You can use the colorful gizmo in the top-right corner to constrain the window to the X, Y, and Z axes to view the top and sides of your **scene**. Click on the white box in the middle to return to perspective view. This is what the **Scene** window looks like when you start a new project or create a new **Scene**.



You can think of scenes as levels or stages in your game.
A unity “scene” is essentially a “level” or “screen”

Typical game

Welcome / main menu screen
Buttons: play game / see high scores / read instructions / change input settings

Level 1 / Level complete / Level 2 etc…
Game Over / Enter details for new High Score …

All the above would be separate “scenes” in unity
Some scenes may be entirely based around the Unity GUI scripts / components – i.e. be text / buttons on screen

The Game Window

The **Game** window shows you what your players will see. When you click on the **Play** button to test your game (as you just did with the Island Demo), the results of your efforts play out in this window. Toggle the **Maximize on Play** button to test your game in full-screen mode.

Hierarchy Panel

The **Hierarchy** panel lists all of the **Game Objects** in your **Scene**. **Game Objects** - cameras, lights, models, and prefabs—are the things that make up your game.
They can be "tangible" things like the birds and the bridge in the Island Demo. They can also include intangible things, which only you as the game developer get to see and play with, such as the cameras, the lights, and colliders, which are special invisible shapes that tell the game engine when two **Game Objects** are touching.
So, **Game Objects** can include touchy-feely "physical" objects like birds and bridges, as well as behind-the-scenes intangible things like lights, cameras, and actions (scripts).

Click on a **Game Object** in the **Hierarchy** panel, and then hover your mouse over the **Scene** window. Press the *F* key on your keyboard, and the **Scene** window will automatically pan and zoom directly to that object. Alternatively, you can go to **Edit | Frame Selected**, which can be more reliable than using the keyboard shortcut. (I like to think of the *F* as standing for *Focus* to help me remember what this shortcut does).

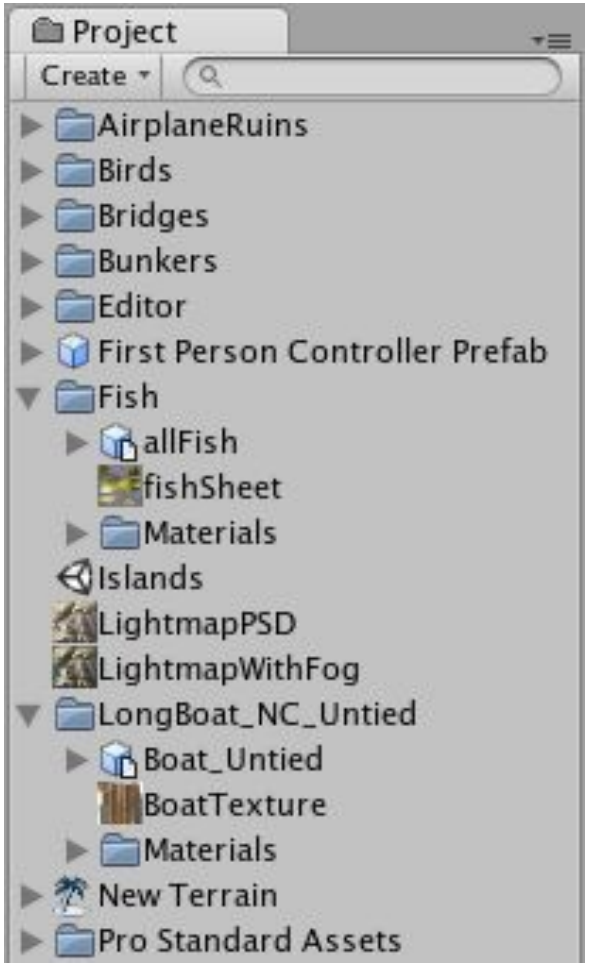
Project Panel

It lists all of the elements that you'll use to create **Game Objects** in your project.

The **Project** panel contains many of the elements, or ingredients, that go into making our **Game Objects**.

The **Project** panel displays the contents of a special folder called **Assets**. Unity automatically creates the *Assets* folder for you when you create a new project. If you drag a compatible file, like a 3D model, a sound effect, or an image into the **Project** panel, Unity copies it to the *Assets* folder behind the scenes, and displays it in the **Project** panel.

Don't mess with the Assets folder!
Unity stores metadata about the folder, and by moving stuff around or deleting things through your operating system, you may break your project. If you need to make changes, make them right inside Unity in the **Project** panel.

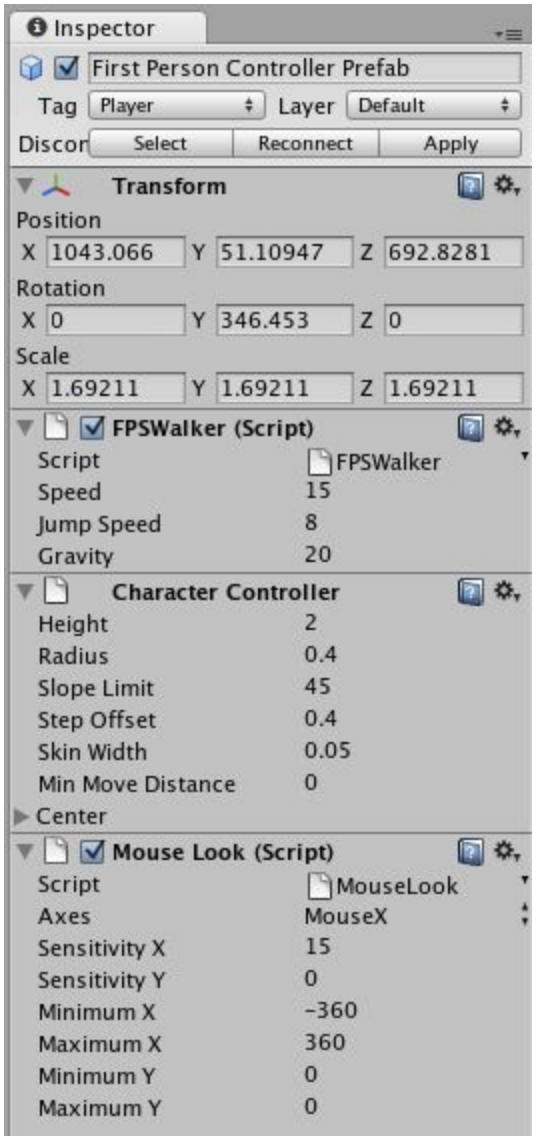


The Inspector

The **Inspector** is a context-sensitive panel, which means that it changes depending on what you select elsewhere in Unity. This is where you can adjust the position, rotation, and scale of **Game Objects** listed in the **Hierarchy** panel.

The **Inspector** can also display controls to configure components that add functionality to Game Objects.

Between the three main panels in Unity (**Hierarchy**, **Project**, and **Inspector**), the **Inspector** is where you'll likely spend most of your time because that's where you'll be tweaking and fiddling with every aspect of the elements that comprise your game projects.



Scripting

- Unity implements a MONO compiler

- Scripts can be written in
  - **JavaScript**
    - Note – most introductory tutorials are written in Javascript – for those learning programming its fine
  - **C#**
    - Very similar to Java, Unity can be integrated with the Microsoft Visual Studio editor, to get full benefits of code completion, source version control etc. Serious developers work in C#

- In Unity, select from the menu Assets->Sync VisualStudio Project
- Find the newly created .sln file in your Unity project (one folder up from your Assets folder)
- Open that file with Visual Studio Express.
- In Unity, go to Edit->Preferences, and make sure that Visual Studio is selected as your preferred external editor.
- Doubleclick a C# file in your project. Visual Studio should automatically open that file for you.
- You can edit the file, save, and switch back to Unity.
- You can now edit all your script files, and switch back to Unity to use them.

Game Objects

- Everything in a scene is either a Game Object or a component inside a Game Object
- Every Game Object has at least 1 COMPONENT
  - Its TRANSFORM – an object's position, scale, rotation
  - Other components depend on object type (audio, mesh, material, script etc.)
- Game objects can be in a HIERARHCY – so an object can be a sub-object of another object
  - E.g. an "arm" object can be a sub-object of a "body" object etc.

Prefabs

Since object-oriented (although this is partially hidden when scripting in JavaScript) instances can be INSTANTIATED at run time.

Unity uses the term PREFAB for a pre-fabricated object template (i.e. a class combining 3D objects and scripts)

At DESIGN TIME (in editor) a prefab can be dragged from Project window into the Scene window and added the scene's hierarchy of game objects. The object can then be edited (i.e. customised from the prefab default settings) if desired

At RUN TIME a script can cause a new object instance to be created (instantiated) at a given location / with a given transform set of properties

Asset

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

- **Asset Store**

Unity's 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.

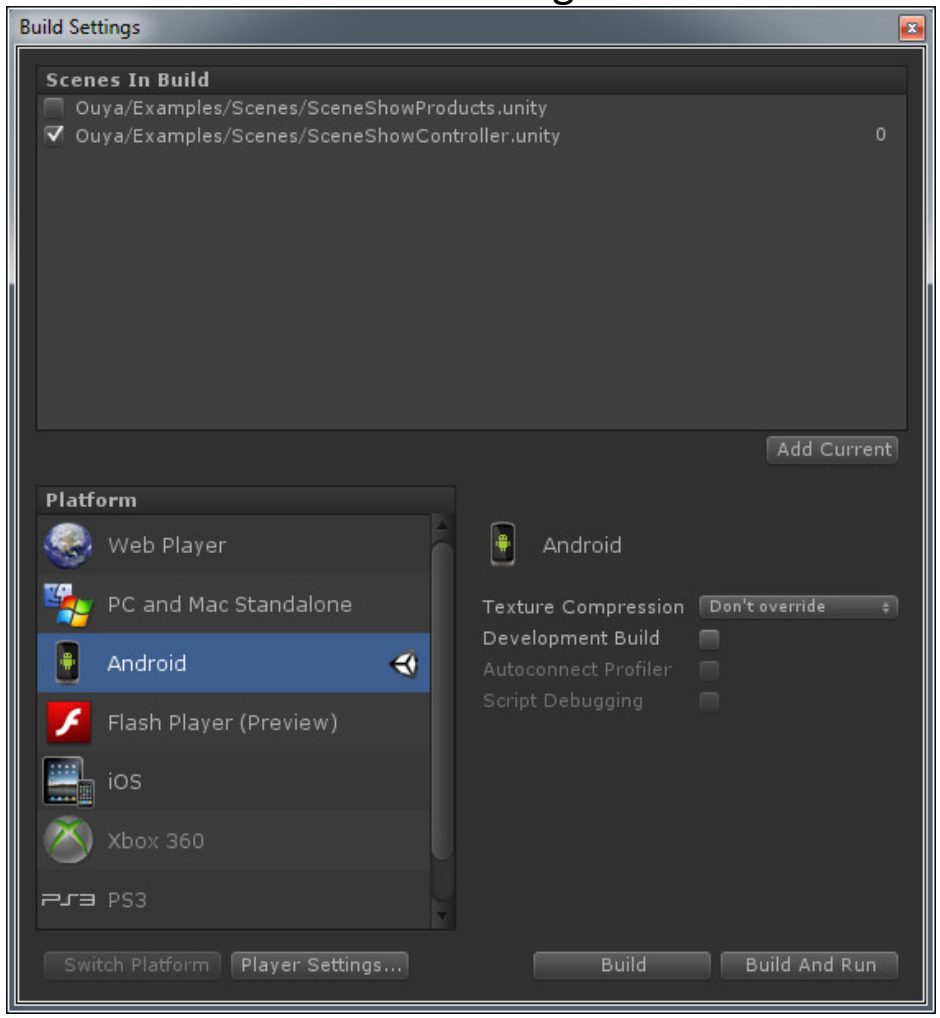
You can open the Asset Store window by selecting Window->AssetStore from the main menu. On your first visit, you will be prompted to create a free user account which you will use to access the Store subsequently.

- **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 subfolder

Build

From the menu bar select File -> Build Settings



Select Scenes to Build

Select Platform

Click Build And Run