Introduction to the Unity Game Engine
Part 1 : The UI
By Christian Barrett

Game development is a tricky thing. While there's quite a lot of people who want to be involved in amateur game development, relatively few of them can manage to create a 3D game on their own. It could be the high barrier of entry; traditionally, creating a 3D game either resulted in a large amount of coding to do the simplest of tasks or a lowering of standards to fit with the engines targeted at beginners. It could also be a resources thing; game development traditionally takes a lot of time and money. So what is the game development enthusiast to do?

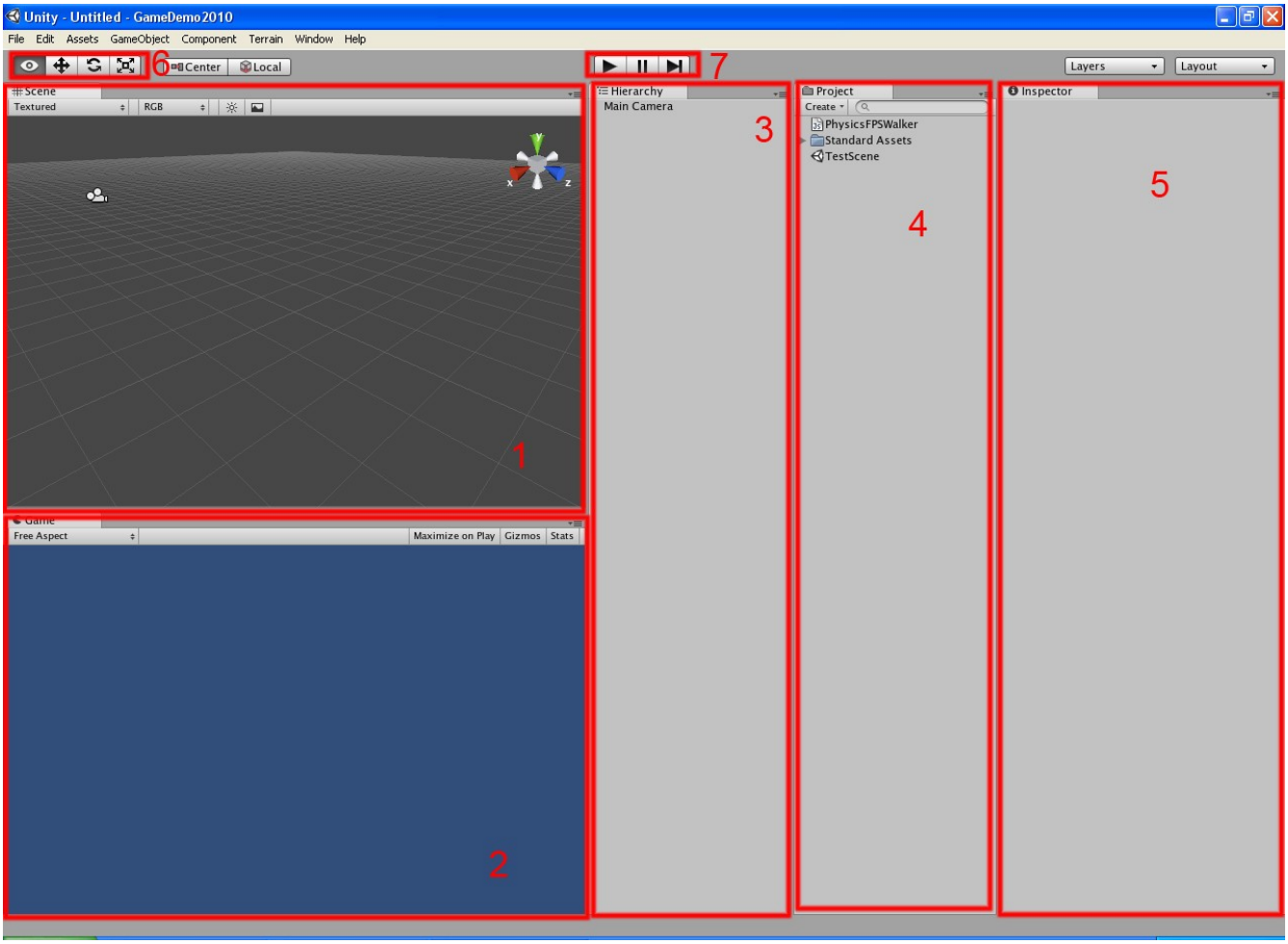
The solution is in a game engine targeted at independent developers and allows for rapid testing of ideas. Luckily, there is such an engine on the market: the very capable Unity 3D. Unity provides a strong combined graphics, audio, physics, and input engine that encourages an implement-test-tweak model of game development. It's easy to pick up, works well with most 3D modeling packages (including the most popular modeling package, Blender, which is completely free), runs quickly, and can deploy to Windows, Mac, the Web, and a number of hand-held devices.

Another huge advantage Unity3D has is in price. For no cost at all, anyone can download a copy of Unity with a license that allows commercial distribution of any products created with it. While there are some features only available in the professional versions and licensing (such as real-time shadows and deploying to platforms other than desktops and the web), the free version of Unity3D is more than enough to introduce somebody to game development.

A quick note:
The goal of this workshop series is to teach people how to quickly implement a game design to be tested and tweaked. Due to the limited time we have, we will not be covering the creation of 3D or 2D art assets. There are many stock models that can be found online - some links will be provided on the workshop website.

In the next section, we'll cover the basic user interface of Unity3D.

The Unity3D User Interface



Pictured above is the main Unity game engine interface with an empty scene. *Note: if yours does not look like this, go to Window->Layouts->2 by 3*. I've numbered the initial parts of note.

Number one (1) is the Unity Scene view. This is where you will place any visual assets in your Unity environment. It will update in real-time when you are previewing the game. Note the manipulator on the top right; this allows you to switch between a number of standard views. We are currently in the perspective view. Although this doesn't matter too much, it allows us to view our scene with a vanishing point, which is the standard way Unity games will display.

Number two (2) is the Game view. When you're not actively running the game, it will show a render of how the game will look, ignoring graphical effects that need to be

computed at run-time, from the point of view of the main camera. When you're previewing the game, you'll be playing through this window. Since our scene is currently empty, all this window is showing is the background color.

Number three (3) is the Hierarchy. This lists all the objects in the currently loaded scene, and any children they may have. **Children** are objects that can be thought of as subordinate to the parent object; wherever the top object moves, they'll follow, keeping the current offset they have to this object. This is an important concept for Unity beginners to understand; we'll cover it more later and in the workshops.

Number four (4) is the Project/Assets view. This is a list of all custom assets for our game, including graphical assets, sound, scripts (more on these later), prefabs (pre-assembled game objects), and much more. Our current game is only using the Standard Assets (which come with Unity, and provide templates to quickly get going in a virtual environment), our current scene, and a custom character control script.

Number five (5) is the Inspector. Since we currently don't have any objects selected in the Hierarchy or the Project/Assets view, it's completely blank. The inspector allows us to look at and tweak individual settings of various game objects and assets, as well as adjust some global settings.

Number six (6) are the graphical icons for moving the scene and it's contents. The hand allows us to pan around the scene; when combined with other scene camera controls, Unity becomes very easy to navigate (we'll cover these later). The icon on it's right, which looks like four arrows, allows you to move a selected object around. We call this transforming the object. The next icon allows for rotation of the object, and the final one allows for uniform scaling of the object.

Number seven (7) is our playback bar. This allows us to play, pause, and stop running our game in the Unity editor. This is the quickest and easiest way to test and tweak your game.

This concludes our lightning introduction to Unity's user interface. We'll cover navigating Unity and creating a simple static scene in the next pdf.