Ogre3D: A Complete Tool for Easy 3D Graphics

by Diwant Vaidya
mangokiwi.com

Ogre 3D is an open source (LGPL) graphics engine, which for you gamers means it is what to use to draw your masterpiece to the screen for free. While I am slightly biased toward it, there are solid reasons for its increasing popularity with the developer folk (outside of the fact that it is FREE to use for any purpose). First, the engine was "design-led rather than feature-led," which means that the code is very well organized around Object Oriented principles. Second, the engine is being actively developed with several new features and bug fixes being added to each release by the Ogre dev team (it is 4 years old so the code is mature enough to use in production). Finally, and perhaps the deciding factor for many, the Ogre3D engine has a massive user community which comes in handy not only for support, but for add-on development to the core engine and tool development to interface other software with the engine (i.e. physics engine wrappers, game engine frameworks, etc.).

What to use it for?

As mentioned earlier, Ogre3D is a graphics engine that is used to display hardware-accelerated 3D graphics. It abstracts interaction with the graphics language so the same Ogre code that will run OpenGL also will run DirectX. While gamers immediately see its use as the graphics half of a game engine, Ogre was not designed to be used only for games. When I told Steve Streeting (AKA "Sinbad"), who is the lead developer on Ogre, that I was writing an article for the ASP on Ogre, he requested two key points to be stressed. The first was that "...we're a graphics engine for all sorts of software, not just games..." The second was that "...we take a lot of care in making a quality product...," which I will get to later. So Ogre is not just for games but for other graphical applications as well. Having clarified this point, I would like to reiterate how amazingly kickass Ogre is as a graphics engine for a game!

Why is it so great?

When you first start learning the engine, the most obvious reason you would begin to like it is because of the code. Ogre is written in C++, which itself is a fast but well structured language (kind of a 'best of both worlds' deal). However, the real magic lies in the organization of the code. Ogre is built around the principles of Object Orientation, which means that the code is Polymorphic (same function acts different for different classes), Abstracted (can ignore details of implementation when calling a routine), and Encapsulated (data is private within a class) (Note: Wikipedia has a good article on Object Orientation for further reading. Maybe an article for ASPects is forthcoming if there is enough interest?).

With OO design patterns such as Singleton in play, access to the functions of the engine become very clean, and the split between the Ogre engine and the game code is very clear. As Steve had mentioned, the team has definitely put a lot of attention on good code, good design.

A little detail, please!

I will try to explain how the code hooks in with the rest of the game, so if you are aversive to detail or don't care for it at the moment skip ahead to the next section. First of all, Ogre initializes a 'Root' object, which learns which graphics language should be used (OpenGL or DirectX), loads plugins needed for rendering, and connects to the graphics devices (your graphics card and monitor). Then we ask Root to create a Scene Manager (Root-> getSingletonPtr()-> getSceneManager()). What you see in Ogre is contained in a Scene. A SceneManager decides what to cull, what to light, etc. (so different scene managers can have different algorithms for culling. Can you guess which algorithm the BSP Scene Manager uses?). Once the SceneManager is created, and suppose we store a pointer to it in mSceneMgr, we add our game objects in scene nodes which 'pin' our game objects to a position [, rotation, and scale] in the Scene. That code looks like: mSceneMgr-> createSceneNode("flower").

To this scene node, you can attach a 3D mesh, a 2D billboard, a sound, a child

Game Developers' Corner

Sashiburidana! [Welcome back!] First, a bit of news: our column shall have a name, and this name shall be..."Game Developers' Corner"! Yes, after four fierce rounds of voting, we have finally found our column a title. Has anyone noticed how the acronym for this name interestingly becomes GDC?

Moving on, last month's Marketing article answered many core questions asked by new and seasoned shareware game developers alike! If you have an itch to write an article or read about a certain something, please let me know at admin@mangokiwi.com. This month I present an article about Ogre3D, a tool that can be used for making some great games. The article is written by Yours Truly, and presents the engine from the standpoint of one who has learned the tool, but is still relatively new to its full set of features. Enjoy!

scene node, etc. Moving this SceneNode then moves whatever is attached to it as well. This is very dirtily how the engine works, and there is a lot more to the engine that what is presented here, so if you are interested in learning more, head over to www.ogre3d.org and start by reading the Wiki which boasts several tutorials and a lot of sample code.

What about when the code gets old?

Yes, the code is written very well and the design can make a CS PhD student tear up, but the code is going to get obsolete eventually, isn't it? To this my answer is no, and justly so. The Ogre dev team has been developing the engine for a little over 4 years now, constantly adding new features and updating older ones. Goals for the next release are made public, as is the ownership and progress on those goals. The community offers code updates and fixes, when these are found necessary. Add-ons and tools seem to follow as rigorous a development plan as the core Ogre team. So no, my friend, I don't think Ogre is going to become obsolete anytime soon!

Who will help me if I get stuck?

One of the greatest assets of Ogre is its user community. It is a large and growing crowd which is very friendly and extremely helpful. For many tools, the only documentation comes from the developers, and too often only the comments included in the code. Ogre is different. Ogre is supported by a community-maintained Wiki that offers tutorials, advice, and sample code on most of what an Ogre noob would need to know, a forum where questions are patiently answered (yea, I know. A developer forum with so little flaming of noobs is a hard find!), and an internet relay chat channel where users discuss solutions to each others' problems. All of this is backed by a detailed online manual for Ogre, and a lot of sample code that comes packaged with the engine. Of course, if nothing else, then I am available to answer questions as best I can on the engine!

How do you know so much about it?

I have used Ogre to make my first 3D game: Temple Of Blocs (templeofblocs.mangokiwi.com). Part of the reason that the game took so long was that I was learning new tools (including Ogre) while developing the game. Part of the reason the game did not take as long as I was expecting was that I was able to learn Ogre pretty quickly thanks to its good design, good code, and great community. Making the game has given me insight on how to do several cool things, a couple of which I have added to the Ogre Wiki to help it grow. When looking at Temple Of Blocs, keep in mind that the game does not even come close to pushing the engine's potential. I wanted to keep the game simple, but the engine itself is capable of doing a lot of cool stuff.

Nice! Where can I see the engine showing off its features?

A good place to see what the engine can do is the "Featured Projects" section on the Ogre website (www.ogre3d.org). It is a gallery of screenshots from some cool stuff done using Ogre.

How do I start using Ogre?

First, review your Object Orientation and C++ skills. Then download the appropriate SDK from the Ogre website (appropriate means the correct one for the development environment you work in–VC 7, VC 8, gcc, etc.). After the download is complete, head to the Wiki and start with the article 'Installing an SDK'. Once you have the Ogre SDK installed, you can go through the tutorials on the Wiki, play around with the given examples, or read through the relevant 'how-to-do-this' posts on the forum. And if ever you get stuck, I'll be happy to help you out. Just send me an email. Good luck!

Discuss this article on the newsgroup a.s.games under the topic "GDC Tool Review Ogre3D"

Diwant Vaidya is CEO of mangokiwi.com (www.mangokiwi.com), and a full-time game developer. At the time of writing this article, he has recently released his first game, Temple Of Blocs (templeofblocs.mangokiwi.com), which uses the aforementioned Ogre graphics engine. He can be contacted at diwant_vaidya@hotmail.com for further information regarding the article, his company, Temple Of Blocs, or the weather.