Ogre到底有什么好？

以前了解实在太浅，不好说什么，现在有些体会了，

觉得得赞美他：

1. 开源，活跃，MIT授权；
2. 底层抽象，跨平台，跨API（DX，OpenGL）；
3. 插件化体系设计，相关Addon丰富（PageGeometory,PCZSM,HydraX,Caelum,SkyX,Hikari…）；
4. SceneMgr，场景图系统，及很是受用的RenderQueue；
5. 十余年世界各地高手的积累，稳定完善的功能,数学，内存，字符串；
6. 优雅的代码设计，设计模式，STL，第三方库集成；
7. 很好的ResourceManager
8. 完善的Material系统，及RtShaderSystem；
9. 完善的异常机制，及Log系统（做过大型的项目之后对这点感觉更是明显）；
10. 很好的动画系统，独特的Compositor，Paging,Terrain（不断在完善）
11. 丰富的教材，且有许多成功的典范，而且在愈来越多；

想了许多，但静下来一写，发现其实也就这样，慢慢再去探究吧，Ogre作为一个图形引擎，本身要完善的也有很多，缺点还有什么？只能说我现在功力不够，还不足以去批评他，但是如果以单纯架构的复杂，设计模式等为由去诟病，敝人实在不敢苟同。

但要作为一个真正的游戏引擎，要做的还有更多更多，当然Ogre社区上还有很多上层的架构，如：

多线程（Boost，POCO），

音频（OpenAL,FMOD）

物理（Bullet，NxOgre，Havok），

编辑器（ETM，GOOF，Ogitor）,

编辑器环境（MFC, C++/CLI, C#, wxWidget, QT）

UI（MyGUI，CeGUI，Hikari，Awesomium），

网络（RakNet），

AI,

脚本系统（Lua，Python，AngelScript）

游戏引擎（Yake，OGE）

借鉴一位前辈们的思路，一个完整的游戏引擎应该这样：

**一层：**

**数学库**

**I/O支持**

**资源管理**

**日志系统**

**第二层：**

**渲染器**

**（Ogre完成了这前面几项）**

**物理系统**

**声音层**

**网络层**

**第三层：**

**场景管理系统**

**特效系统**

**动画管理系统**

**GUI系统**

**第四层：**

**AI，脚本驱动系统**

**开发工具：场景编辑器、材质编辑器、动画编辑器、GUI编辑器、逻辑编辑器、特效编辑器等辅助管理工具。**

比如基于Ogre的一个需要商业授权的Neoxis，试了下发现还是不错的，努力，要早日完成属于自己的一个~~

