用到的Pyside基本模块介绍：

1、QtCore模块

用于核心的非 GUI 功能函数,用于以下方面:日期、文件和目录、数据结构、数据流、URL、MIME、线程和进程。

2、QtGui模块

用于绘图组件以及与绘图相关的类,比如按钮、窗口、状态栏、工具栏、滑块、位图、颜色、字体等。

3、QtXml模块

用于处理 XML 文件的类,该模块提供了 SAX 和 DOM API 两种 XML 文件处理方式的实现。

4、QtOpenGL 模块

用于渲染使用 OpenGL 库创建的 3D 或 2D 图形。并且它支持 Qt GUI 库和 OpenGL 库的无缝结合。

<https://doc.qt.io/qtforpython/PySide2/QtOpenGL/QGLWidget.html>

3月29日

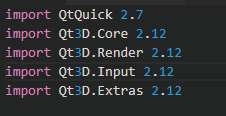
<https://doc.qt.io/qt-5/qt3d-qml.html#qt-3d-extras-module>



3月30日 参加两场考试，所以没更新。

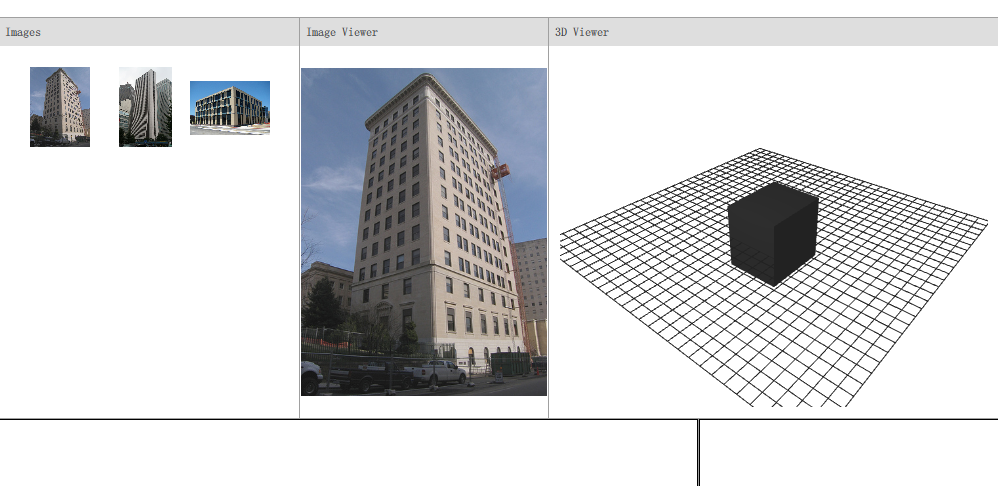
3月31日

import部分：

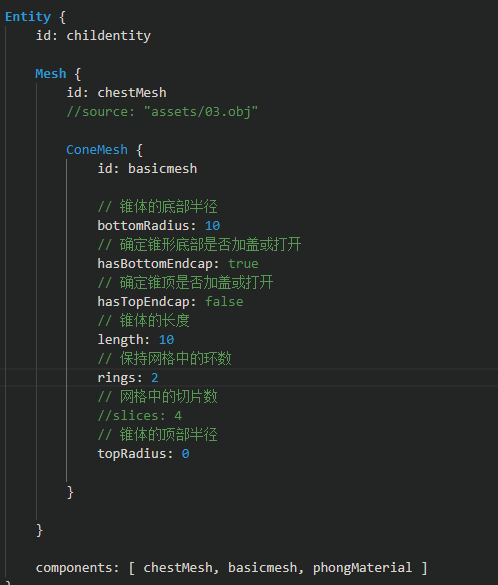


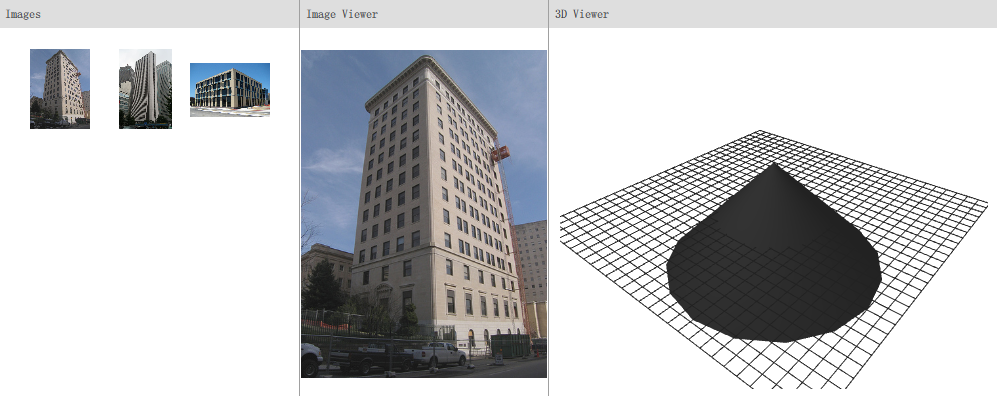
1. CuboidMesh



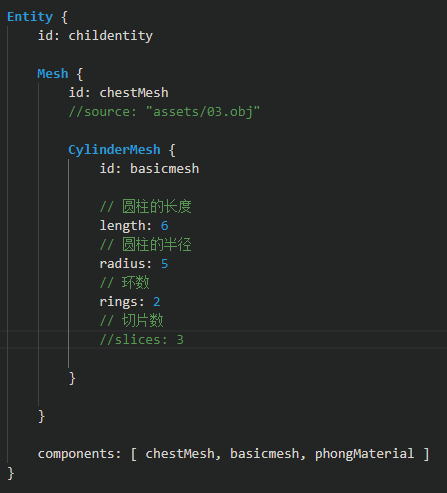


1. ConeMesh





1. CylinderMesh



注意：slics=3时,柱体为三棱柱，slice=4时，为四棱柱即长方体（如下图1），未设置slices时，柱体为圆柱体（图2）。slices控制底边多边形的边数（取值>=3）。椎体也是如此。

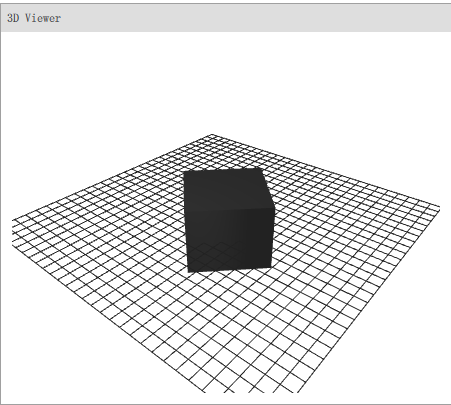


图1

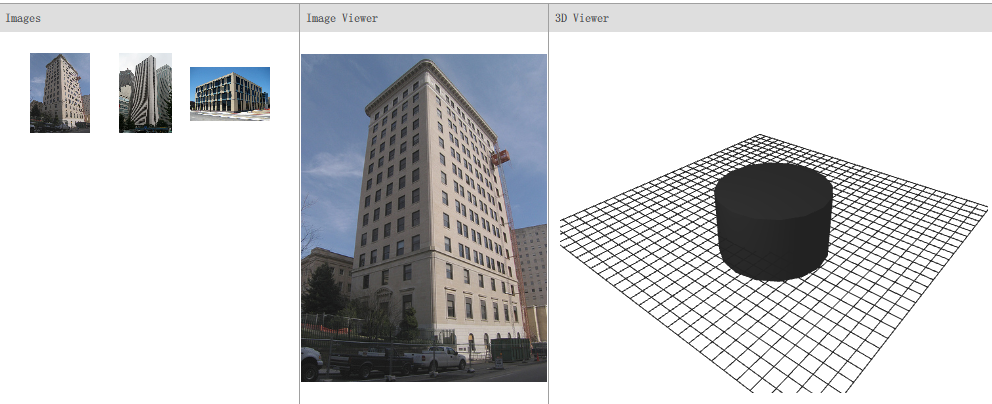


图2



PyOpengl环境搭建：

<https://blog.csdn.net/BigBoySunshine/article/details/80218245>

4月1日 QtXML——读取XML文件

1 创建一个QDomDocument类对象，代表整个XML文档

2 使用QFile打开要读取得xml文档，使用QDomDocument类的setContent()函数来设置整个文档的内容，它会将XML解析成一个DOM树,并保存在内存中。

QFile file(filename);

QDomDocument doc;

doc.setContent(&file, true);

3 获取根节点元素,QDomDocument类也是QDomNode的子类，使用firstChild()函数可以获得它的第一个子节点。使用documentElement()可以获得他的根节点，这是访问XML文档的入口。

QDomElement root = doc.documentElement();

QDomNode child\_node = root.firstChild();

4 后面就可以通过QDomNode来遍历整个文档。

QDomNode的函数：

firstChild()：获得第一个子节点

lastChild()：获得最后一个节点

childNodes()：获取该节点的所有孩子节点的一个列表

nextSibling()：获取下一个兄弟节点

previousSibling()：获取前一个兄弟节点

// 对xml文件的每个grammar节点进行分析

while (!child\_node.isNull()) {

if (child\_node.toElement().tagName() == "grammar") {

Grammar grammar;

parseGrammar(child\_node.toElement(), grammar);

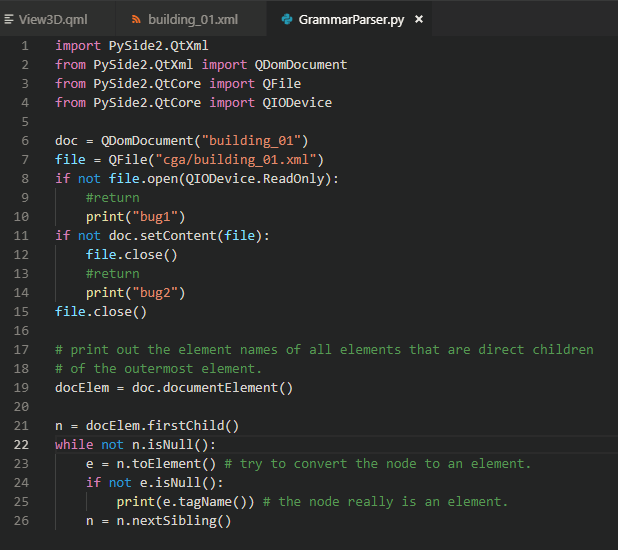
grammars.push\_back(grammar);

}

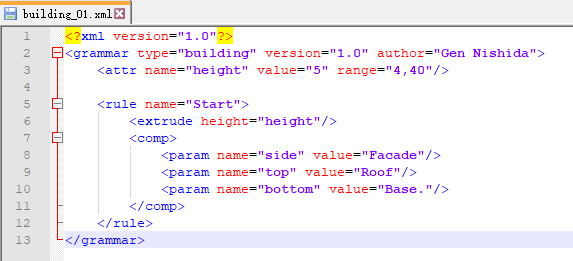
child\_node = child\_node.nextSibling();

}

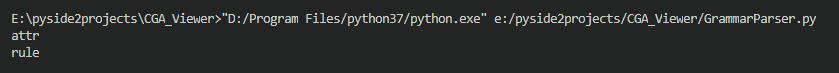
新建GrammarParser.py，第一个示例：



输入：



输出：



后续的代码还没调试好。。。

4月2日 敲代码中，rule部分C++转python，代码提交到GitHub的CGA\_Viewer项目中。

4月3日 上午科目一考试，下午和晚上继续码。。。依旧是rule中对xml进行语法分析部分，这段敲完之后会进行调试。