该部分为osg编译记录，可跳过：

1. 按照<https://github.com/Microsoft/vcpkg>安装vcpkg
2. 进入vcpkg，打开管理员权限cmd，运行

vcpkg install osg:x64-windows sqlite3:x64-windows protobuf:x64-windows poco:x64-windows

1. 下载osgQt：git clone https://github.com/openscenegraph/osgQt.git
2. 编译osgQt：创建build文件夹，进入，打开管理员权限cmd：

cmake .. -G "Visual Studio 15 2017 Win64" -DCMAKE\_BUILD\_TYPE=Release -DWIN32\_USE\_MP=ON -DCMAKE\_TOOLCHAIN\_FILE=${vcpkg路径}\scripts\buildsystems\vcpkg.cmake -DCMAKE\_PREFIX\_PATH=${Qt路径}\${Qt版本}\msvc2017\_64\lib\cmake

1. 构建安装osgQt：

cmake --build . --target INSTALL --config Release

1. 下载osgearth：git clone <https://github.com/gwaldron/osgearth.git>
2. 创建build，进入，打开管理员权限cmd：

cmake .. -G "Visual Studio 15 2017 Win64" -DCMAKE\_BUILD\_TYPE=Release -DWIN32\_USE\_MP=ON -DCMAKE\_TOOLCHAIN\_FILE=${vcpkg路径}\scripts\buildsystems\vcpkg.cmake -DCMAKE\_PREFIX\_PATH=${Qt路径}\${Qt版本}\msvc2017\_64\lib\cmake

Extra：

使用代理加速加载：

set HTTP\_PROXY=socks5://127.0.0.1:1080/

set HTTPS\_PROXY=socks5://127.0.0.1:1080/

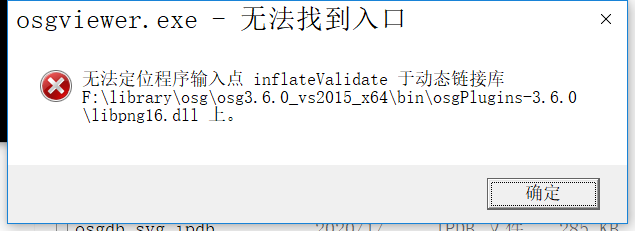
对非github网址容易出现SSL握手失败的错误，自己手动下载放在${vcpkg路径}\download即可。

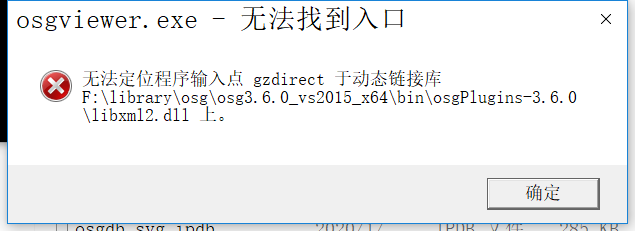
Osg添加新插件（我们编译好的osg库中已经有了部分插件dll，见{OSG\_DIR}\bin\osgPlugins-3.6.0，但要使用其他的插件，如svg插件则需要自己编译添加）。

1. 寻找要添加的插件源码，osg官网提供有许多，参考url：<https://github.com/openscenegraph/OpenSceneGraph/tree/master/src/osgPlugins>（也可以用svn签出部分）。
2. 源码中有Cmake工程，不过据说不能自动识别osg的部分东西。所以还是自己建立一个工程，vs、qt都可（由于依赖库的问题，后面会提到，所以这里建议使用vs工程），把源码拷过去。
3. 查看源码中有无依赖库，比如svg源码中就使用到librsvg的库（而这个库本身又依赖许多其他库！），这里推荐使用上文（编译osg过程）中提到的vcpkg。如下载librsvg库，只要一个命令vcpkg install librsvg:x64-windows即可，期间会把22个依赖包全部下好，同时一个命令vcpkg integrate install即可在VS中使用库，非常方便。Vcpkg教程参考：<https://blog.csdn.net/cjmqas/article/details/79282847>
4. 依赖库下好，如在VS项目，此时把项目属性中输出改为dll，名字要设置为osgdb\_xx.dll（如osgdb\_svg,dll），编译源码即可得到dll插件。（由于vcpkg主要提供dll，所以这样得到的插件应该也会有一大堆dll，一起拷到环境配置中的osgPlugins文件夹即可）。无论在全局环境变量path还是{OSG\_DIR}\bin\osgPlugins-3.6.0中，osg会根据名称自动去寻找插件。
5. 在程序或者使用osgViewer去读写文件即可，如：osg::Image image = osgDB::readImageFile(“xxx.svg”);

可能遇到的坑：

1. 关于怎么在VS中开发osg项目，可参考本人的github项目<https://github.com/kkkiitty/learnOSG>中的OSG\_VS，有一个VS上的osg框架（其实就是添加好include和lib，用到哪些库链接上就行）。
2. 一般插件都分为debug版和release版，debug版带d尾缀，注意编译插件源码时选择的配置。这里有一点未解决的问题是我们项目似乎暂时只支出debug版本，所以插件暂时只编译debug版本即可。所以链接库时记得链接的库都要带d尾缀（当然带d不带d一起链接也行）。如果编译的是debug版，链接库只链接了release版是不能用的！同时osgViewer这个命令本身就是release版本的，所以不能跑debug版本插件（换成osgViewerd即可）。编译release版可能会出现：





如果没碰到什么问题，但最简单的插件就是找（写）源码，编译且输出dll名字要符合格式，丢到path中，剩下的osg都会搞定了。

如果碰到问题，那就祝欧吧~