Building VSim

I am using a 64bit PC with Windows 7 Enterprise Installed. Fully updated as of 2/11/16

wxWidgets and OSG will not compile with VS2015

Must install VS2013 (if you leave VS2015 installed you must ensure you are running VS2013 throughout this process)

Otherwise, I Installed the latest versions of the following:

VS2013 (Free Edition)
Putty
Git
WinSCP
CMake
Python (this is only for Creator plugin development)

vSim has two major library dependencies - wxWidgets and Open Scene Graph.

I installed both into the root of the C: drive (do the same)

I found this post to build wxWidgets useful (notes follow)

http://liuxingguang.blogspot.com/2014/02/using-wxwidgets-30-in-microsoft-visual.html

Current version of wxWidgets is 3.0.2, which works
Loaded Visual Studio solution file wx_vc12
Build is not F7 it's Ctrl+Shift+B or the menu option BUILD|Build Solution (which is what I did / rebuild solution is probably better)
Build Debug and Release as shown but Win32 and x64 as well
This will build wxWidgets with Unicode support (apparently the default) - which matters later

I found this post to build OSG useful (notes follow)

https://xinyustudio.wordpress.com/2014/02/25/building-openscenegraph-with-visual-studio-2013-2/

Current version of OSG is 3.4.0, which works

I created two build directories under the top level OSG directory. CMake uses an ‘out of source’ build process so you need someplace for the code and build to occur.

build-x86 and build-x64

OSG builds a little differently because it uses CMAKE. Basically we need to redo the CMAKE for each Visual Studio version (x86/x64) we want to make available.

Run CMake per instructions in the BLOG notes
I selected the Visual Studio 12 - 32 or 64 bit
You can check on the OSG_EXAMPLES
(slows build down enormously - not necessary but handy to learn OSG
maybe only do for one build since it will use a lot of disk space and time)
If you do you need to included wxWidgets in the plugins on a subsequent Configure run
Picked the 32/64bit wxWidgets directory

Repeated Configure until the red stuff was gone then clicked on Generate to create VS Solution files.

Opened the Solution file in the build directory in Visual Studio 2013 (make sure you load it into 2013 if you have more than one version installed!)

Selected BUILD|Rebuild Solution for both Debug and Release for the platform (x86/x64) you are building.

Waited... Go do something else

The default (for me) in VS was the Debug build - you can rebuild against the Release by selecting it and rebuilding. Do not just Build, you must either Rebuild or Clean and Build.

Once I finished this I close VS and reopened with the 64 bit solution file and repeated the process. This is the same process but you guys can skip this if you like for now.

This takes a long time.

Building vSim

git clone VSim into C:\src or wherever you are building things (I used C:\src)

$ git clone git@gitlab.idre.ucla.edu:neh/vsim

I performed some amount of cleanup and compiler/linker adjustments to new library versions and location. You should get these changes automatically from the repository.

Both x86 (32bit) Debug and Release build cleanly
Turned off post build batch file (postbuild.bat) in Release build - not sure what this is

I also committed back a bunch of configuration changes for Visual Studio 2013
at this point the following configurations build cleanly (executing I have not got to yet)

32bit (Win32) Release
32bit (Win32) Debug
64bit (x64) Release
64bit (x64) Debug

Eventually, we will move to the 64bit build only. For now we can stick with the 32bit since that is what was being developed before. It -should- not be a super big deal to transition.

I also had to manually install the DroidSans font which you can find in one of the vSim folders.

I was able to run all versions of Vsim - I did notice the fonts are rendering in the wrong location so that will have to be debugged.

Send me email if you have questions and I will try to help.

-Scott