How to add a new dependency in Physika?

Fei Zhu

05/05/2014

Step 1: Is it really necessary?
We're very cautious about adding a new dependency in Physika. So please ask yourself the following questions before you add one:

- Is the functionality you need already in Physika? We avoid adding dependencies that have redundant functionality. If the functionality you need is already covered by either Physika itself or any existing dependency, don't add a new one.
- Is the functionality you need easy to implement with short code? If so, implement it yourself.
- Are you confident that the library to be imported will be used frequently by all(most) Physika developers? Generally we don't recommend to import 3rd code(probably huge) merely for the need of a few.

Step 2: Add a new dependency.
The path of Physika dependency is "Physika_Src/Physika_Dependency/". The dependency in Physika may be provided in **3** forms:

1. **Only headers.** This is when the 3rd party code is written in C++ template. In this case, place the code in corresponding dependency path and update the "pure_copy" variable in scons script of Physika. The headers will be copied to target include directory while building Physika.
2. **Source code.** In this case, you need to write a scons script for the library, rename it as "SConscript" and place it at root directory of this library. Take the "LodePNG" library as an example, the script is put in "Physika_Src/Physika_Dependency/LodePNG/". In this script you need to compile the library into libs and copy the headers and libs to target build directories of Physika. The build script will be called by build script of Physika. I provide a template of this script at root directory of dependency.
3. **Headers and precompiled libraries.** In this case, you need to put all headers in a subdirectory of the library with the name "include" and put all libs in a subdirectory called "lib". The libs are categorized according to the OS name and architecture. In "lib" directory, there are 3 directories named as "Apple","Linux", and "Windows". Each of the 3 directories has 2 subdirectories called "X86" and "X64". The libs for different platforms are put in corresponding directories respectively. We support both "msvc" and "g++" on Windows, hence we require that libs incompatible between the 2 compilers are put in 2 deeper subdirectories named "msvc" and "g++" while the compatible ones are placed in root directory with architecture name. For example, the lib files compatible between "msvc" and "g++" for 64 bit Windows are put in "LibraryName/lib/Windows/X64/". The compiler specific ones are placed in "LibraryName/lib/Windows/X64/name", where name is "msvc" or "g++". The build script of Physika will be able to find lib files for the target platform and copy them to public directories. *Please provide lib files for all supported platforms while adding a new dependency!*

Step 3: Inform other developers.
It's a big decision anyway, they deserve to be noted.