SDL2 / Codeblocks Installation
BJ Blair
01/25/17

- First, you need to install the compiler: in this case, MinGW
- If you have Code::Blocks running and working normally on your computer already (eg you can program regular stuff but not SDL2 stuff)then you probably don't have to do this

- Go to the mingw website at https://sourceforge.net/projects/mingw/files/
- Click the 'Download MinGW-get setup.exe' at the towards the top of the page

- Run the exe, just click continue until you get to the next window
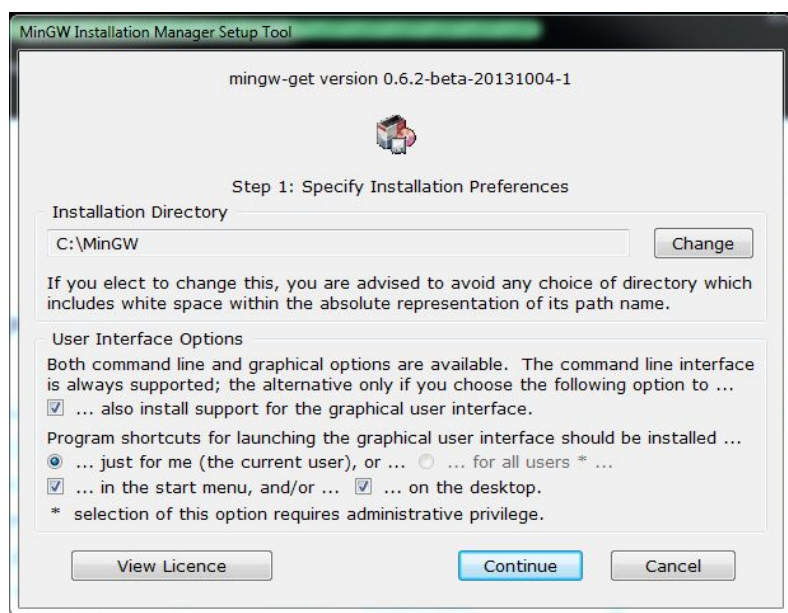
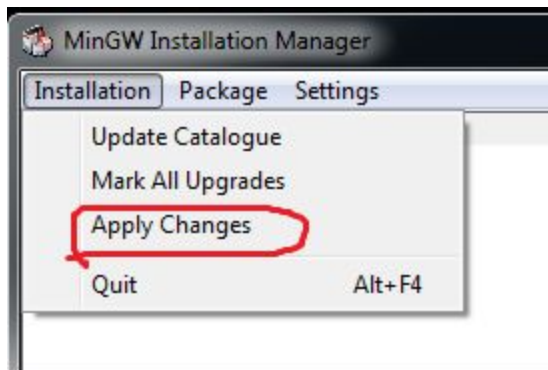- Right click the checkboxes for 'mingw32-base', 'mingw32-gcc-g++', and 'msys-base' and select 'mark for installation'
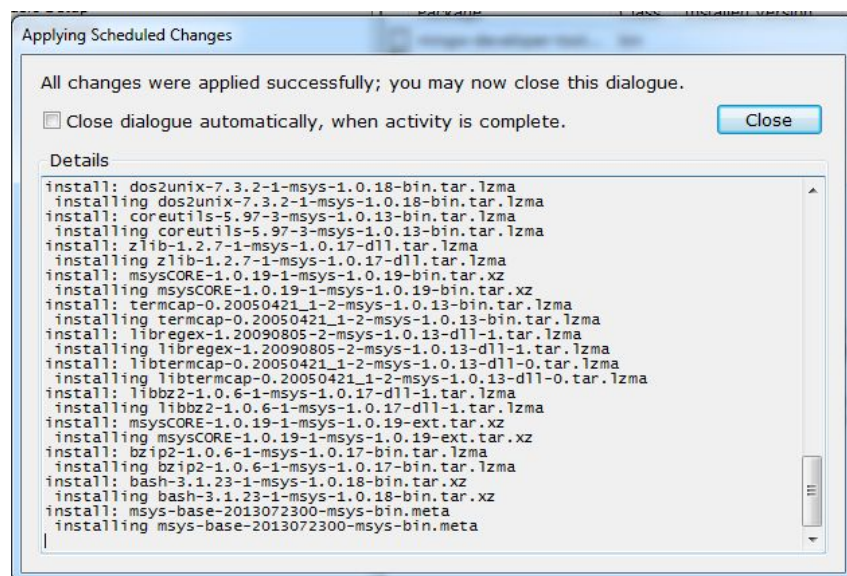
Package	Class	Installed Version	Repository Version	Description
mingw-developer-tool...	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
	bin		2013072200	A Basic MinGW Installation
			5.3.0-3	The GNU Ada Compiler
			5.3.0-3	The GNU FORTRAN Compiler
			5.3.0-3	The GNU C++ Compiler
			5.3.0-3	The GNU Objective-C Compiler
			2013072300	A Basic MSYS Installation (meta)

Unmark
Mark for Installation
Mark for Reinstallation
Mark for Upgrade
Mark for Removal

Package	Class	Installed Version	Repository Version	Description
mingw-developer-tool...	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
mingw32-base	bin		2013072200	A Basic MinGW Installation
mingw32-gcc-ada	bin		5.3.0-3	The GNU Ada Compiler
mingw32-gcc-fortran	bin		5.3.0-3	The GNU FORTRAN Compiler
mingw32-gcc-g++	bin		5.3.0-3	The GNU C++ Compiler
mingw32-gcc-objc	bin		5.3.0-3	The GNU Objective-C Compiler
msys-base	bin		2013072300	A Basic MSYS Installation (meta)

- On the top left corner, click 'Installation' then 'Apply changes'

MinGW Installation Manager

Installation Package Settings

Update Catalogue

Mark All Upgrades

Apply Changes

Quit   Alt+F4

- It should then take a while to download everything and when it's done you should get a window saying so

Applying Scheduled Changes

All changes were applied successfully; you may now close this dialogue.

☐ Close dialogue automatically, when activity is complete.   Close

Details

```
install: dos2unix-7.3.2-1-msys-1.0.18-bin.tar.lzma
 installing dos2unix-7.3.2-1-msys-1.0.18-bin.tar.lzma
install: coreutils-5.97-3-msys-1.0.13-bin.tar.lzma
 installing coreutils-5.97-3-msys-1.0.13-bin.tar.lzma
install: zlib-1.2.7-1-msys-1.0.17-dll.tar.lzma
 installing zlib-1.2.7-1-msys-1.0.17-dll.tar.lzma
install: msysCORE-1.0.19-1-msys-1.0.19-bin.tar.xz
 installing msysCORE-1.0.19-1-msys-1.0.19-bin.tar.xz
install: termcap-0.20050421_1-2-msys-1.0.13-bin.tar.lzma
 installing termcap-0.20050421_1-2-msys-1.0.13-bin.tar.lzma
install: libregex-1.20090805-2-msys-1.0.13-dll-1.tar.lzma
 installing libregex-1.20090805-2-msys-1.0.13-dll-1.tar.lzma
install: libtermcap-0.20050421_1-2-msys-1.0.13-dll-0.tar.lzma
 installing libtermcap-0.20050421_1-2-msys-1.0.13-dll-0.tar.lzma
install: libbz2-1.0.6-1-msys-1.0.17-dll-1.tar.lzma
 installing libbz2-1.0.6-1-msys-1.0.17-dll-1.tar.lzma
install: msysCORE-1.0.19-1-msys-1.0.19-ext.tar.xz
 installing msysCORE-1.0.19-1-msys-1.0.19-ext.tar.xz
install: bzip2-1.0.6-1-msys-1.0.17-bin.tar.lzma
 installing bzip2-1.0.6-1-msys-1.0.17-bin.tar.lzma
install: bash-3.1.23-1-msys-1.0.18-bin.tar.xz
 installing bash-3.1.23-1-msys-1.0.18-bin.tar.xz
install: msys-base-2013072300-msys-bin.meta
 installing msys-base-2013072300-msys-bin.meta
```
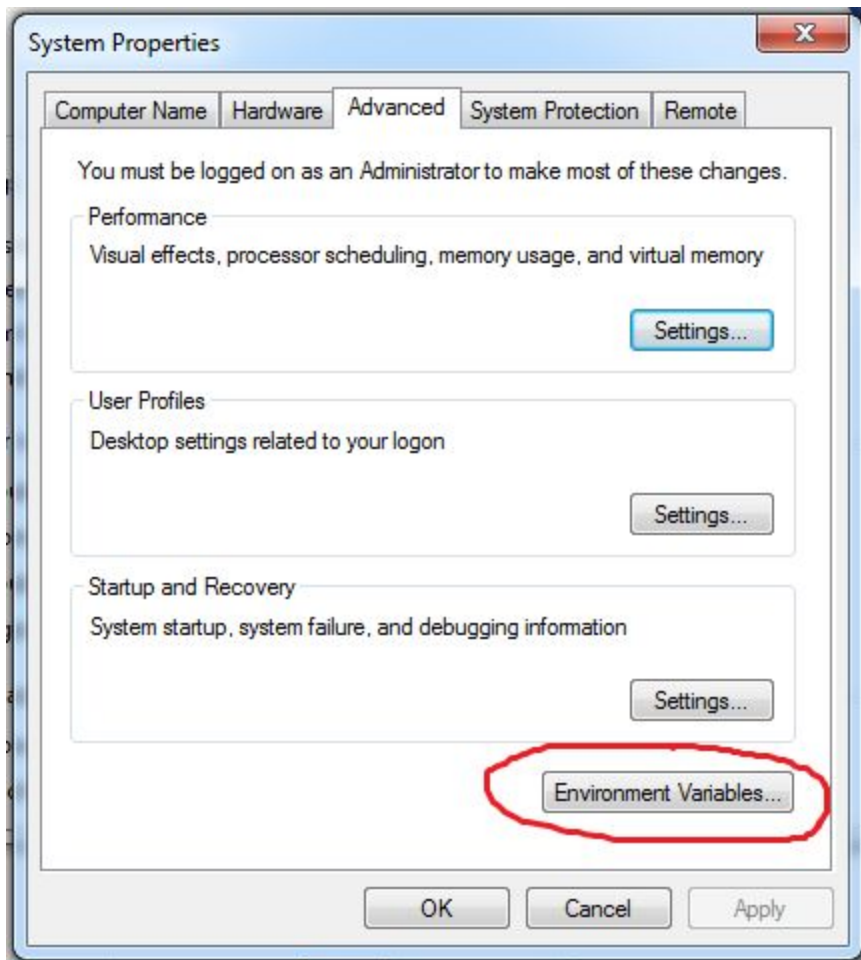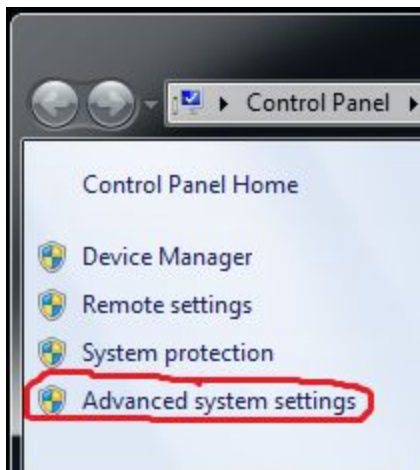
- Next you should add the compiler to your computer's path so you can compile stuff by command line if necessary. Open an explorer/file manager and right click on 'computer' or 'this pc' if in Windows 8 or 10 (the computer icon), then click 'properties'
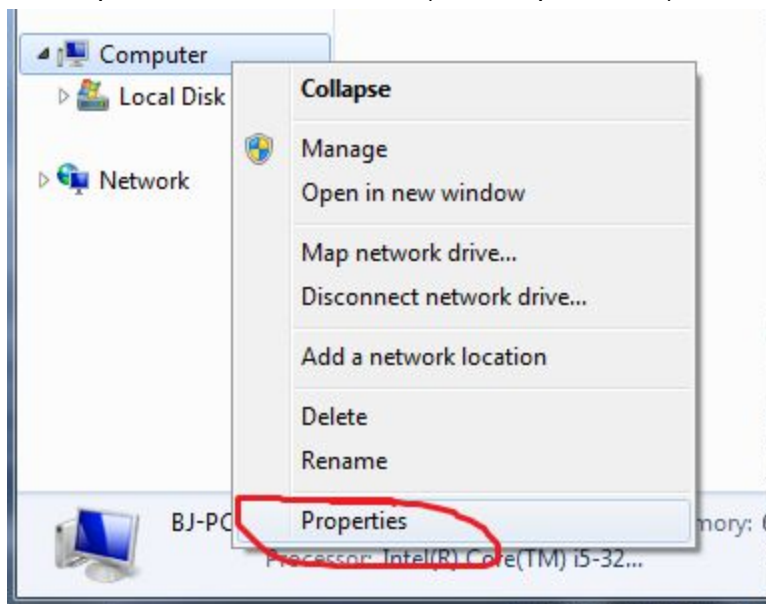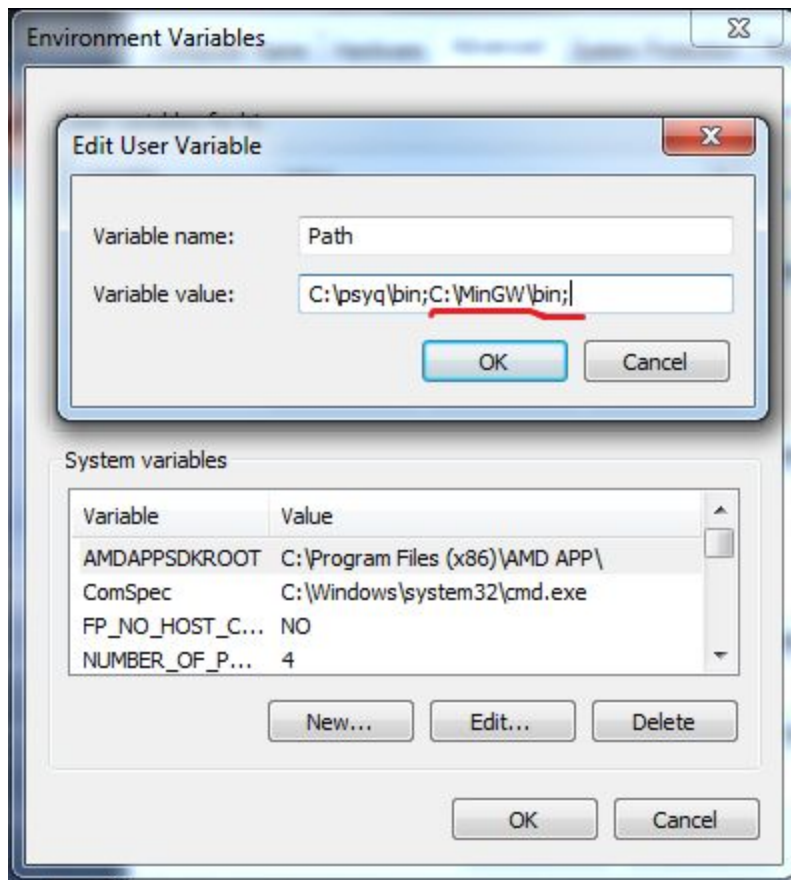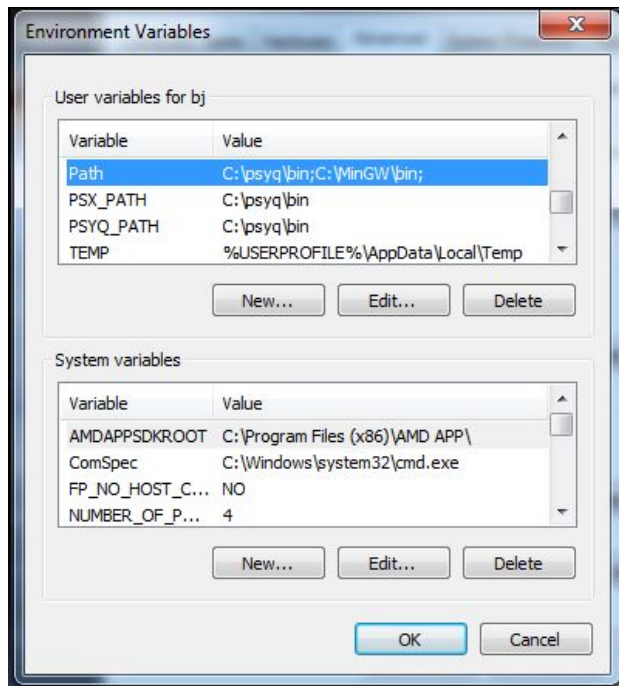

Or
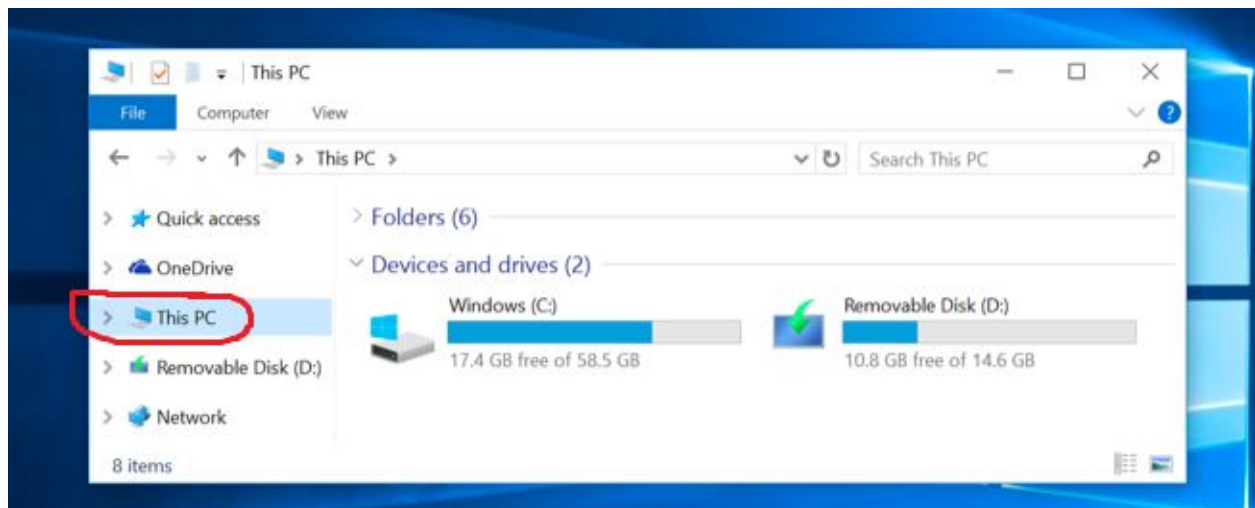
- Then go to 'Advanced system settings' -> 'Environment variables'

- under 'user variables', select 'Path' (or create new if it doesn't exist), and enter 'C:\MinGW\bin' (assuming you left MinGW to install to its defaults).

- If there are any other entries in path, separate them with a semicolon (I believe you don't have to worry about this in 10 as each entry has its own row or whatever)
- then click OK and OK for both windows to exit

- Now that the compiler is installed, we can install Code::Blocks
- This part is self explanatory, just install it as normal from https://sourceforge.net/projects/codeblocks/files/Binaries/16.01/Windows/codeblocks-16.01-setup.exe/download
- Click next for everything during the install
- Once you run codeblocks, you should get a screen saying GNU GCC compiler detected; select it and set it as the default
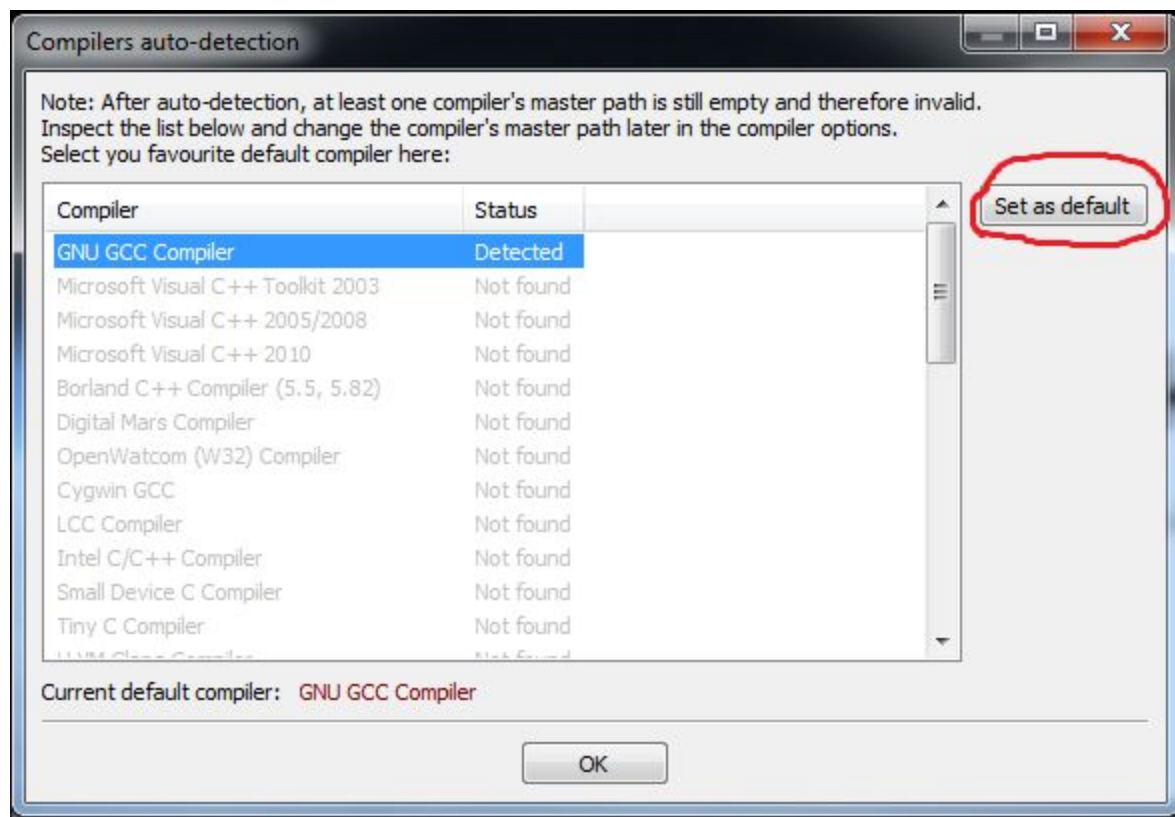
- I would create a new 'console application' project at this point just to make sure Code::Blocks is set up properly at this point (File->New Project->Console Application, write a helloworld c++ file, run it)
- Once that's good we can set up an SDL2 project in Code::Blocks

- First download the SDL2 Library from https://www.libsdl.org/download-2.0.php
- Select the MinGW 32bit/64bit library under 'Development Libraries:'

SDL version 2.0.5 (stable)

Source Code:

SDL2-2.0.5.zip - GPG signed
SDL2-2.0.5.tar.gz - GPG signed

Runtime Binaries:

Windows:
SDL2-2.0.5-win32-x86.zip (32-bit Windows)
SDL2-2.0.5-win32-x64.zip (64-bit Windows)

Mac OS X:
SDL2-2.0.5.dmg (Intel 10.5+)

Linux:
Please contact your distribution maintainer for updates.

Development Libraries:

Windows:
SDL2-devel-2.0.5-VC.zip (Visual C++ 32/64-bit)
SDL2-devel-2.0.5-mingw.tar.gz (MinGW 32/64-bit)

Mac OS X:
SDL2-2.0.5.dmg (Intel 10.5+)

Main

About
Bugs
Licensing
Credits
Feedback

Documentation

Wiki
Forums
Mailing Lists

Download

SDL 2.0
SDL 1.2
SDL Mercurial
Bindings

- I saved the file in a new 'external_libs' folder in C:\MinGW. Just make sure to save it somewhere **without spaces in the path**. This can mess up command line compiling
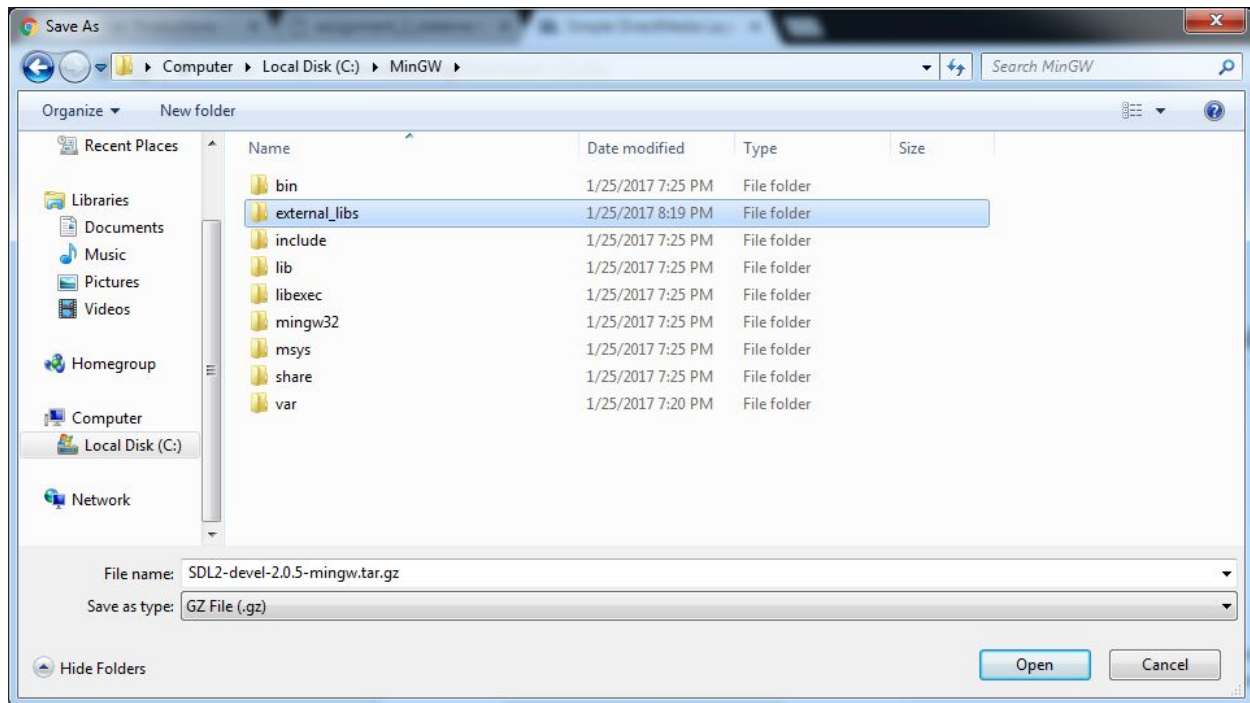
- Extract the file, using something like 7zip. You'll have to extract it twice - the first time you'll end up with a .tar file then the second time you actually get the folder.
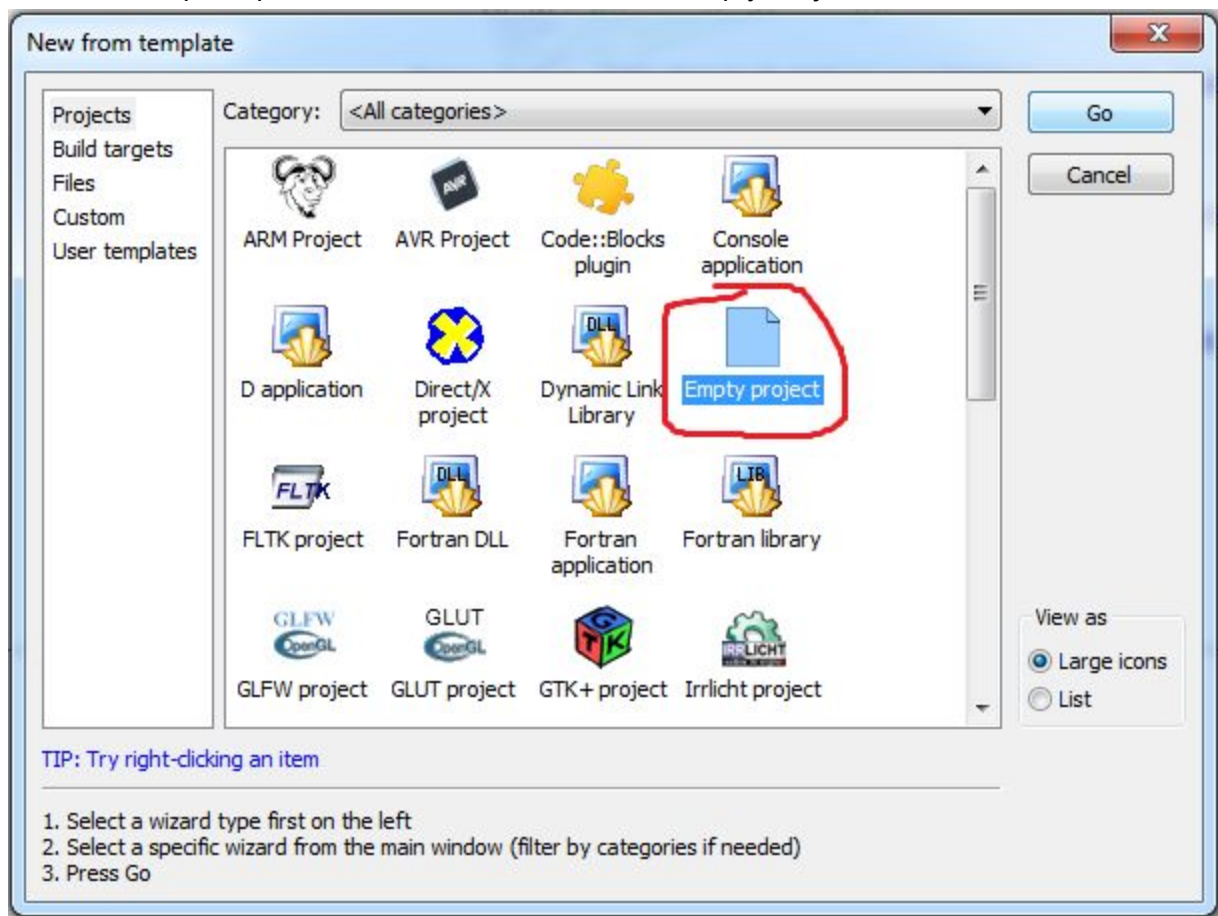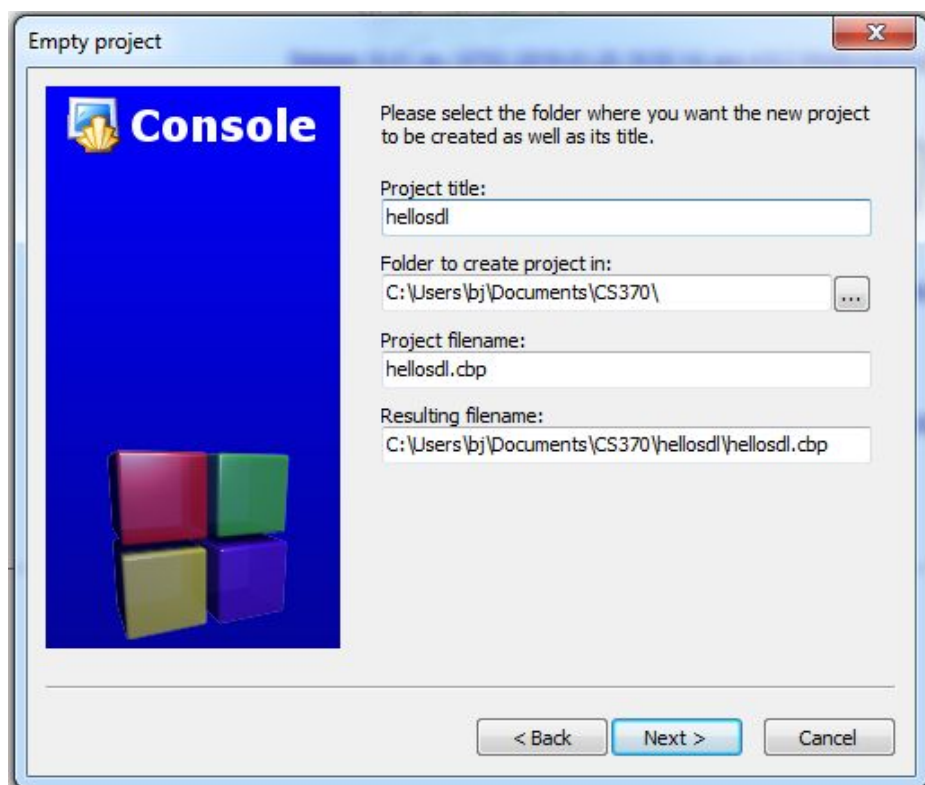

- Move the final actual extracted folder (SDL2-2.0.5) to wherever you want to keep the library. I just moved it up to the same directory where I downloaded the file; so its path ended up being 'C:\MinGW\external_libs\SDL2-2.0.5'. Inside should look like this:

- Now open up CodeBlocks and create a new Empty Project:

- Name the project whatever, doesn't matter, and it can be located anywhere also.

- Once you've done that, go to 'Project->Build Options'
- Under the 'Search Directories' tab, for the 'compiler' click 'add', and navigate to the '....include\SDL2' directory where the SDL2 library is. Ex. in my case it's C:\MinGW\external_libs\SDL2-2.0.5\i686-w64-mingw32\include\SDL2
- Note - we'll be using the 'i686-w64-mingw32' folder for everything because we want to build 32bit applications, since mingw is 32bit
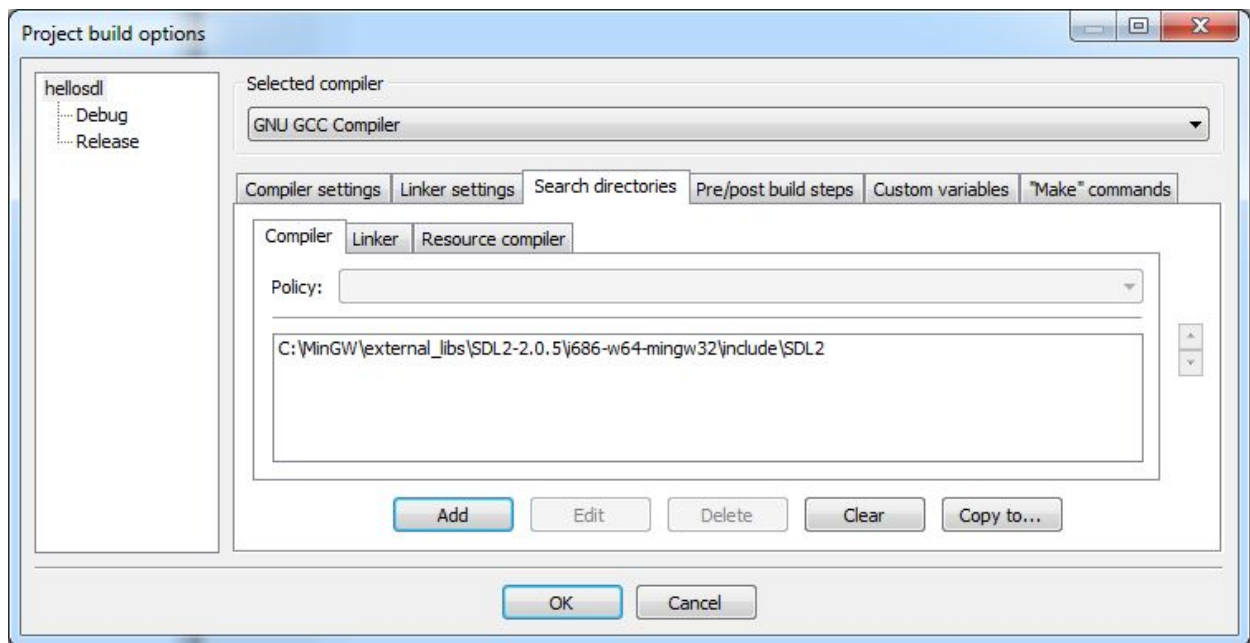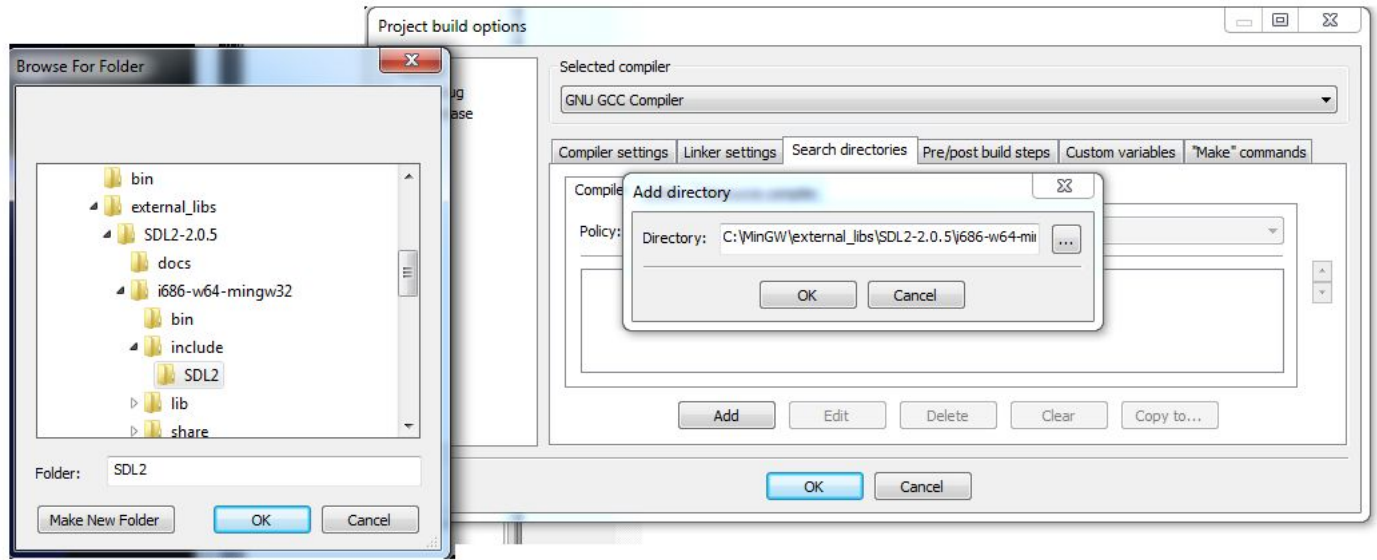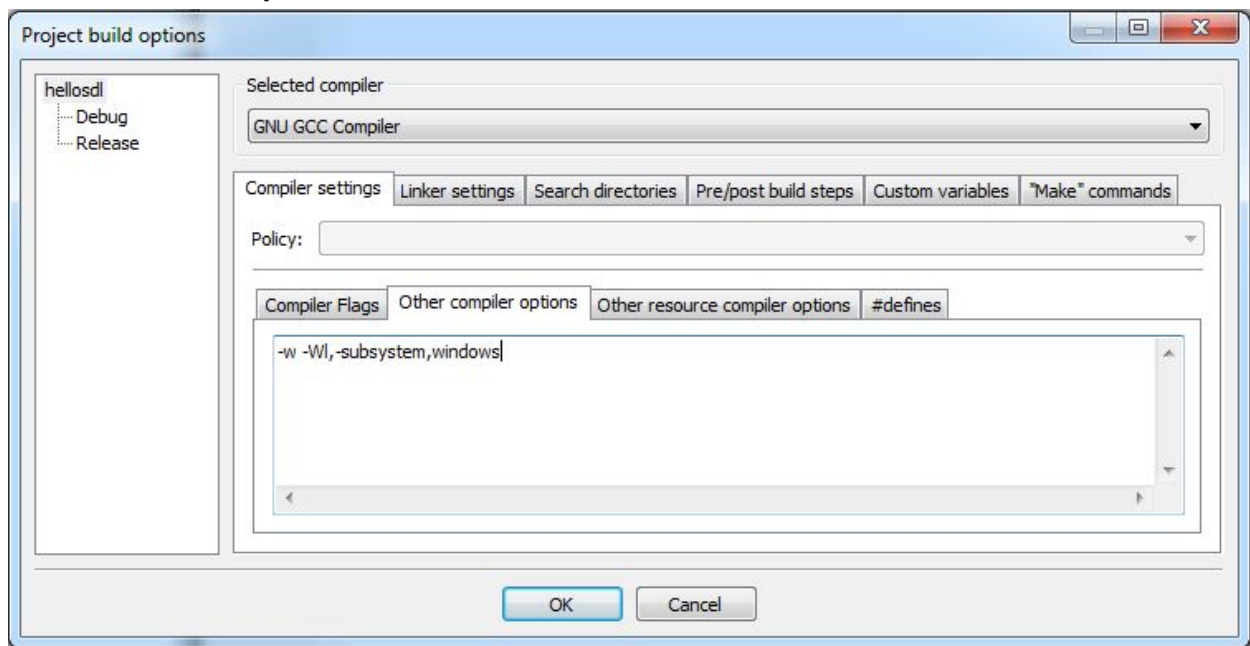
- Under the 'Linker tab' (still under 'Search Directories'), locate the 'lib' folder in SDL2
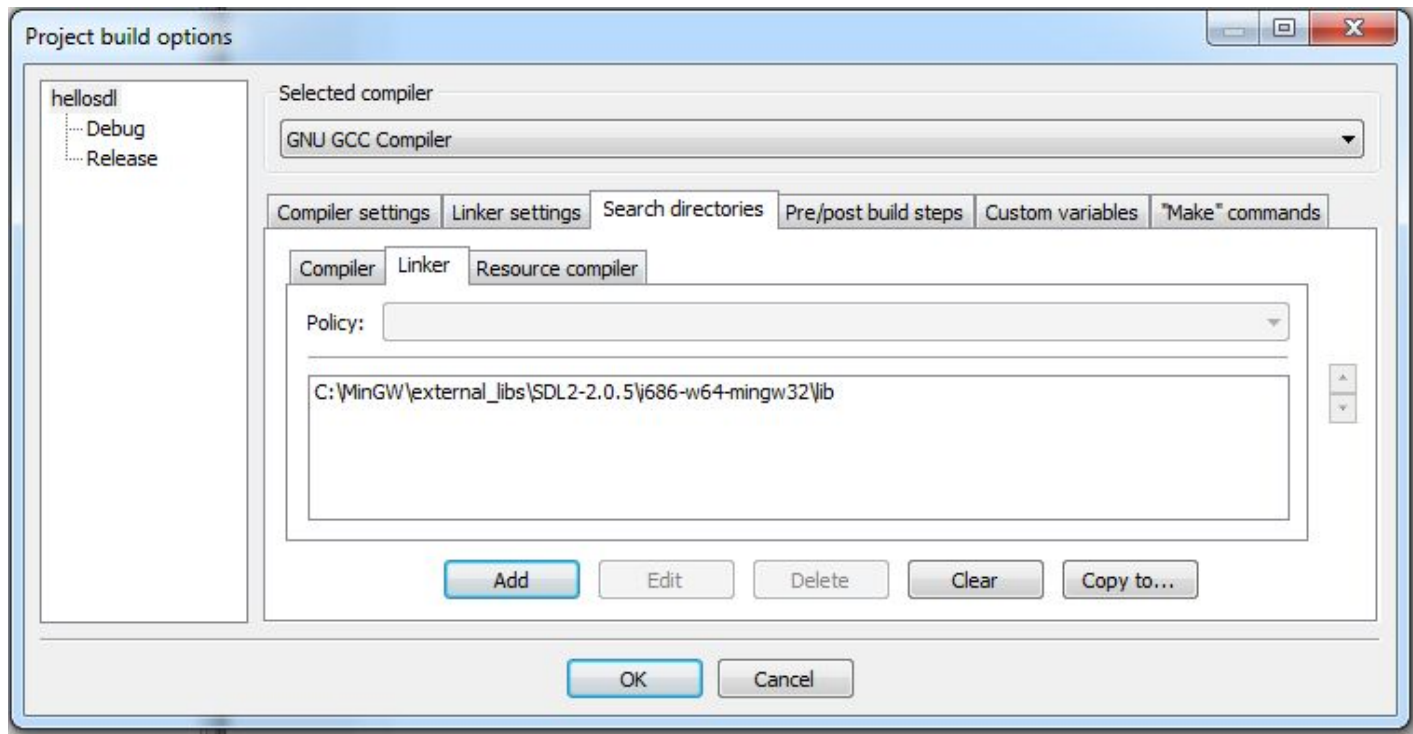

- Now move from the 'Search Directories' tab to the 'Linker Settings' tab, and add
  -lmingw32 -lSDL2main -lSDL2
 To the 'Other linker options' box
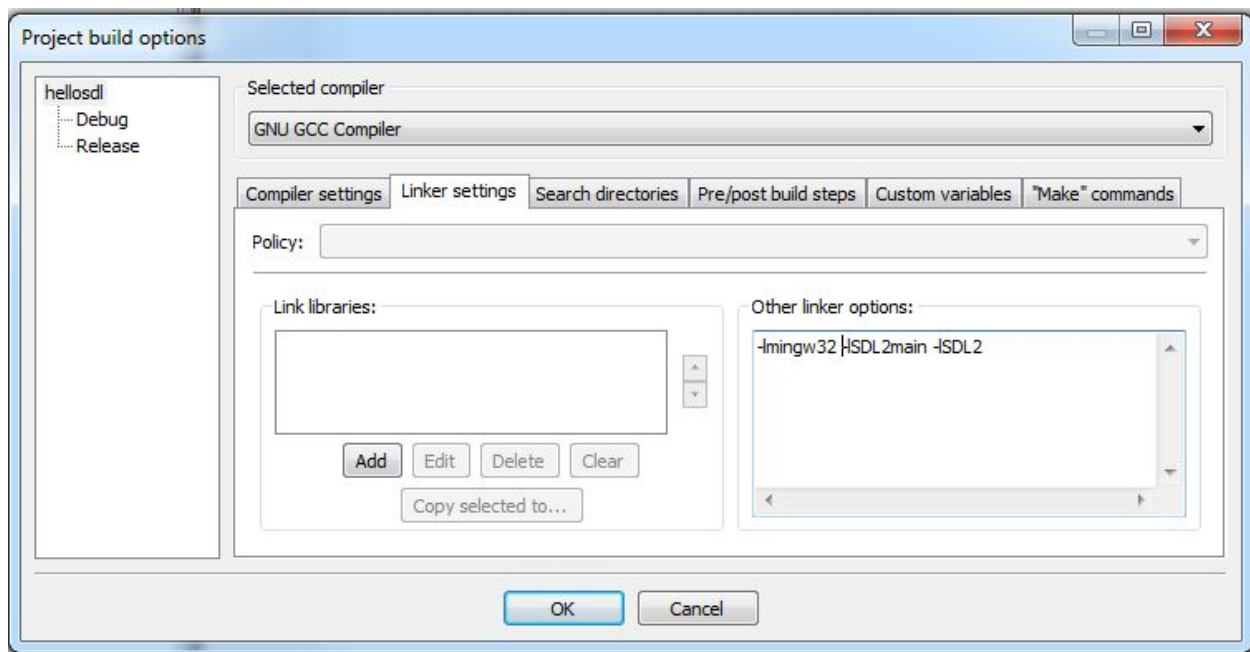
- Then move to the 'compiler settings' tab, and under this select the 'other compiler options' tab, and add
 -w -Wl,-subsystem,windows

- Finally, click OK, then click 'File->Save Project.'
- Almost done: go to the 'bin' folder of the SDL2 library;
  (eg C:\MinGW\external_libs\SDL2-2.0.5\i686-w64-mingw32\bin), and copy SDL2.dll and sdl2-config to the 'bin' folder of your codeblocks project directory
  (ex C:\Users\bj\Documents\CS370\hellosdl\bin\Debug)
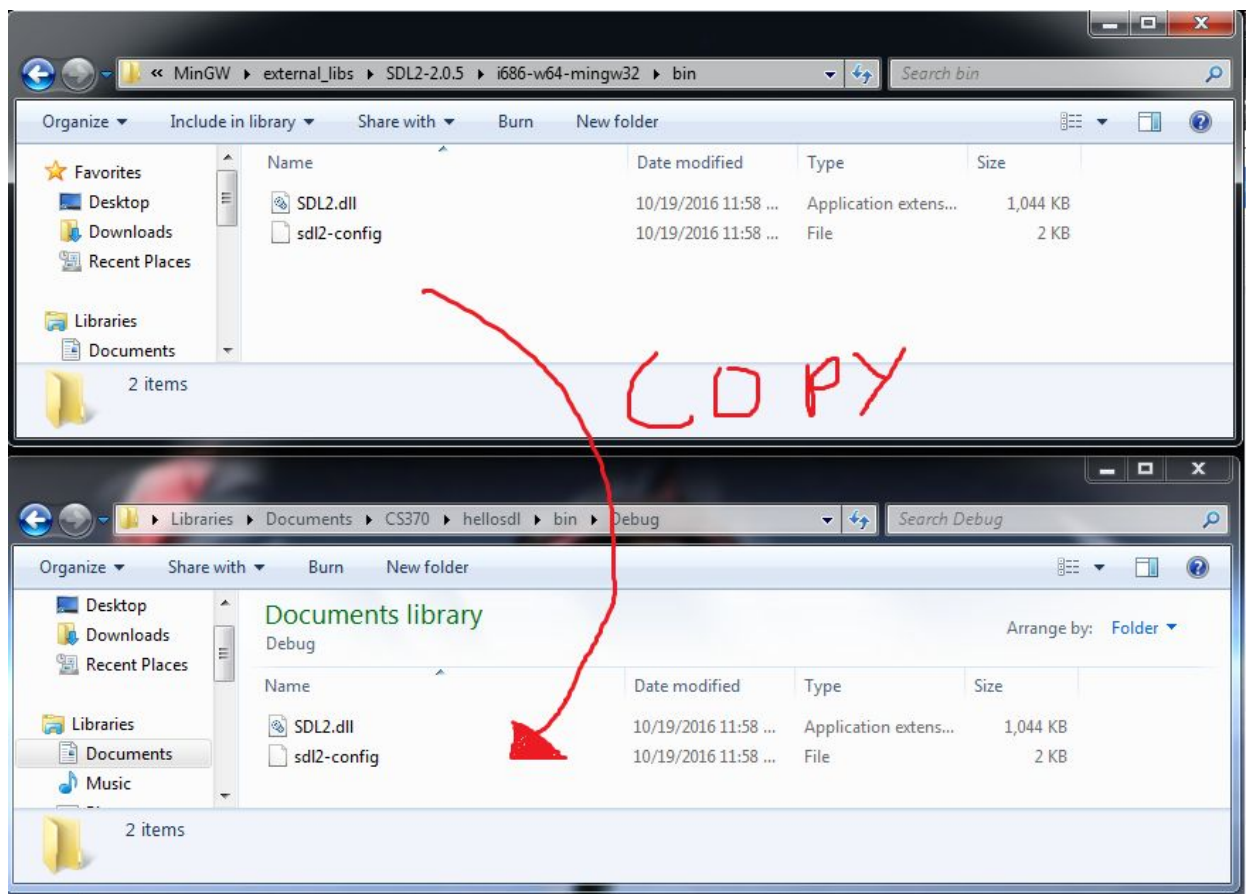
- Now for actual code
- In codeblocks, add a new source file to your project (File->New->File, C/C++ source, check both 'debug' and 'release' to add it to the current project)
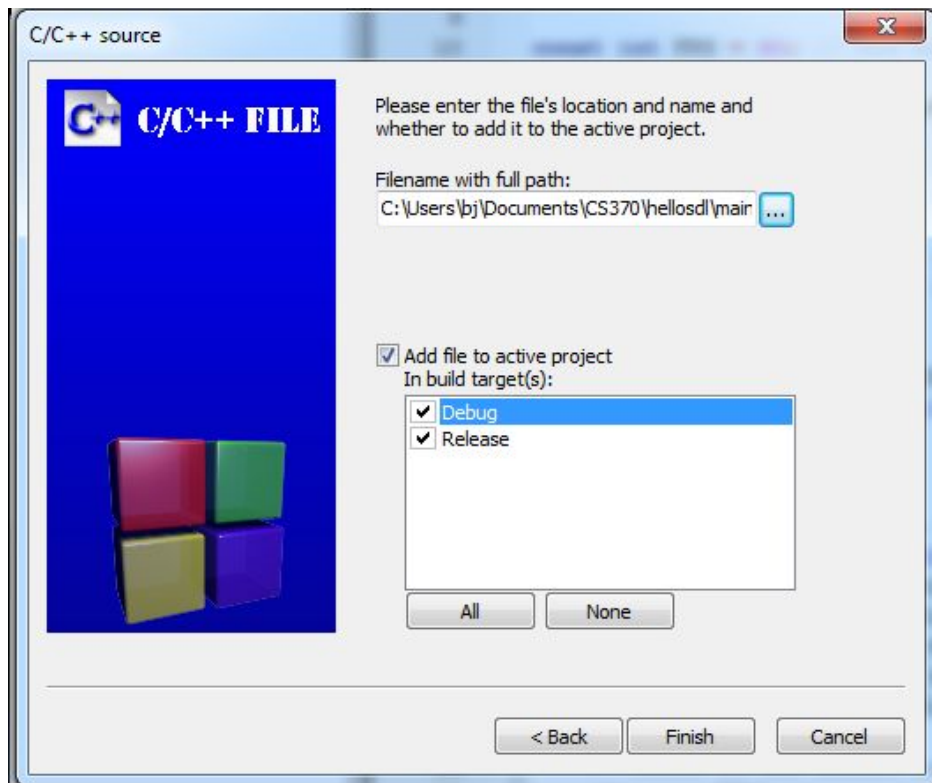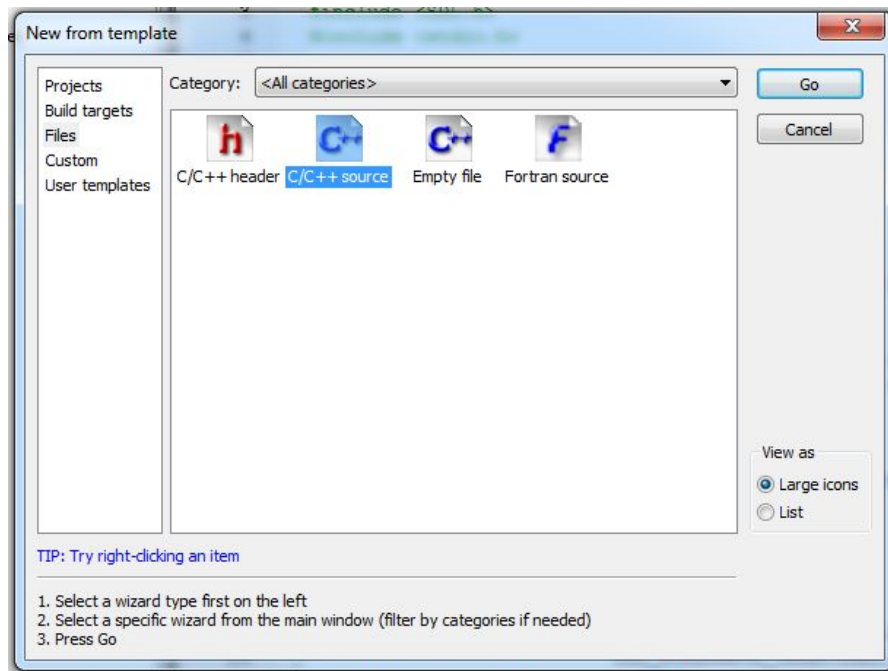
- Now you can actually program using SDL!
- You can test this by copying the contents of a hello world kinda thing I wrote:
- Copy the code from https://github.com/williamblair/CS370/blob/master/helloworld/main.c
- Paste this into your newly created file. Save it, then press F9 to compile and run. Hopefully, you get a blue screen that doesn't do anything. That's a success!
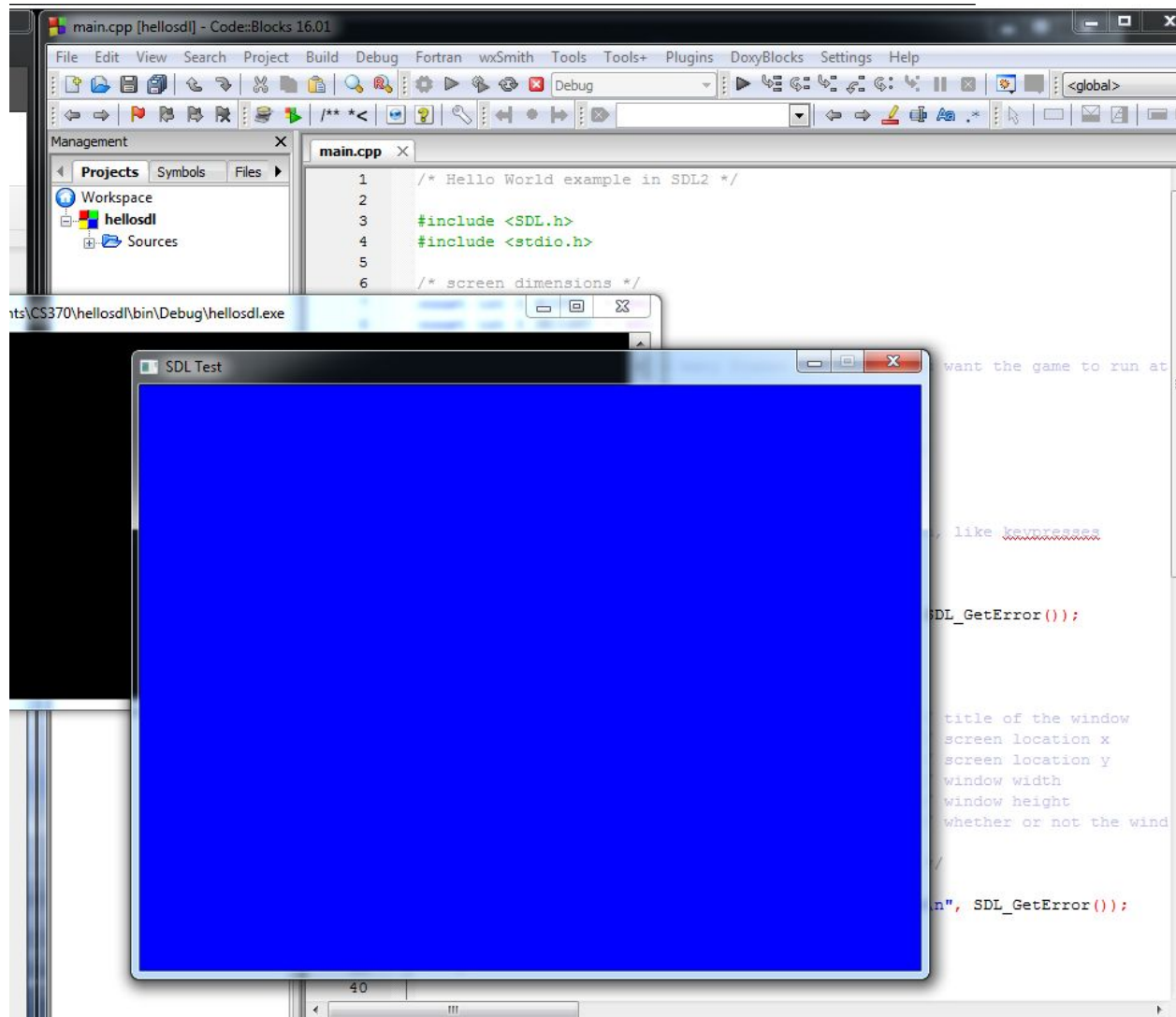
- Now I would start following the SDL2 tutorials from LazyFoo here: http://lazyfoo.net/tutorials/SDL/