

Project 0

Due: 5:00 PM on Wednesday, March 18th, 2015

Complete [documentation](#) of the supplied shell code.

Instructions

The purpose of this assignment is to familiarize you with the source code provided for the class, and to set up assignment hand-ins.

You have the option of using Visual C++ or DevC++ under Windows, or g++ under Linux or Mac OSX. The code is designed to compile and run on the Law 113 computers, but you also have the option of using your personal computer. If you choose to use your personal computer, you must always compile your assignments on a Law 113 computer before handing in your code. If you do not do this, I may be unable to grade your work.

The CSC321shell uses FLTK to generate the user interface and provide event handling. You will not have to mess with this too much but if you want more info you can visit fltk.org

The file [csc321files.zip](#) contains the source code for all the labs in this course, as well as a pre-compiled demo of all the labs (built for Windows and Linux). The demo is a very good way for you to check your work, so use it often.

To get started, download [csc321files.zip](#) and unzip it in your own directory.

Instructions for Windows Computers

Depending on which version of Visual C++ (labs have been tested extensively on VS 2010, other versions should work but are not guaranteed to) you are using, you need to open a different project file.

- Visual Studio 2008 or 2010: csc321shell/csc321shell.sln or csc321shell/csc321shell.vcproj
- Visual Studio .NET: csc321shell/csc321shell.sln

Compile by hitting ctrl-shift-b (F7 under VC6.0). By default, compiled object files will go into d:/temp. If your computer does not have a d: drive, or if you want to change the compile location for some other reason, follow these instructions:

- In Visual Studio, right-click on "csc321shell" in the Solution Explorer pane and select Properties
- Click Configuration Properties, then General
- Change Configuration to Debug
- Change the Output Directory and the Intermediate Directory. Make sure you do not change these directories to any location inside of csc321shell -- it is best to keep the intermediate and object files separate from the source code. Any other location is fine.
- Change Configuration to Release and change the Output Directory and Intermediate Directory again

- Click OK.

If you receive a "Device is not ready" error, you have not changed the location correctly.

You may get a warning message: LINK : warning LNK4098: defaultlib 'LIBCMT' conflicts with use of other libs;... This is because the fltk libraries are compiled optimized while the rest of the project isn't. Don't worry about this.

After you successfully compile, the executable will be placed in the `csc321shell` directory. It will have filename `csc321shell.exe` or `csc321shellDB.exe` (if you built in Debug mode). It is fine to build in Debug mode, but in later labs your code may run more slowly if you have built this way.

Run the compiled program by hitting F5. Or if you prefer, double-click the executable file that has appeared in your `csc321shell` directory.

Once you have this running perform the tasks outlined below under Tasks.

Before handing in, close Visual Studio (there are issues with un-saved files and visual studio "hanging on" to some files). Please delete the `csc321shell.ilc`, `.ncb`, and `.pdb` files as well.

Tasks

Add your first and last name to the program in the "name" box (replacing the name "demo"). Use Fluid to do this. Fluid is an interactive gui builder; there are Linux and Window builds included in the `csc321files.zip` file. Open up `ShellInterface.fl` file using Fluid. The name initialization takes place in the constructor method for `ShellInterface`. Tell Fluid to write out the `.cpp` and `.h` files and save the `.fl` file. DO NOT edit the `ShellInterface.cpp` file directly.

After you have finished your work in Fluid, recompile your program in Visual Studio (Windows) or by typing `make` (linux, mac). Then run the executable to check that your name appears in the upper-right corner of the window.

Create a README file for this assignment. Put in your name, email address and the programming environment you used.

You will submit your README file, your source code, and your compiled executable. See the Handin Instructions below for details.

Handin Instructions

1. Look at the grade sheet

Whenever you submit a project in this course, look at the grade sheet linked from the course home page. Use the grade sheet to learn how the project is graded and to check that you have not forgotten any of the required parts of the project.

2. Zip up your files

When you are ready to hand in your work, follow these steps:

- Save your README file in the `csc321shell` directory.
- Compile your code on a computer in Law 113.
- Run and test the compiled executable, which is in the `csc321shell` directory, with filename `csc321shell.exe` or `csc321shellDB.exe` (Windows) or `csc321shell` (Linux). Use a Law 113 computer for this test -- your code must run on the Law 113 computers.
- Clean up your `csc321shell` directory and its subdirectories by deleting object files and any large (>1MB), unnecessary files. DELETE files with extensions like `.obj`, `.o`, `.ilk`, `.ncb`, `.pdb`. DO NOT DELETE files with extensions `.cpp`, `.h`, `.mk`, `.fl`, `.exe`, `.sln`, or `.vcproj`. These lists may not be comprehensive -- the point is to erase the junk and keep the important files.
- Double-check that your `csc321shell` directory does not contain any large files (>1MB) other than your compiled executable.
- Use WinZIP or the built-in Windows zip feature (Windows), or tar and gzip (Linux) to create a zip file of the `csc321shell` directory, including all subdirectories. A good filename would be `csc321shell-assign0` (with appropriate extension).

Before you submit your zip file, check the following:

- Make sure your zip file contains all of the files needed to build your executable. This includes:
 - `.h`, `.cpp`, and `.fl` files
 - all subdirectories (`brush`, `camera`, `interactive`, `intersection`, `rendering`, `sceneview`, `shapes`, `vecmath`) and their contents
 - the build files (for Windows, `csc321shell.sln` and `csc321shell.vcproj` or equivalent; for Linux, `Makefile` and `linux.mk`).

The best way to test this is to unzip and recompile the code yourself.

- Make sure your zip file contains your README
- Make sure your zip file contains your compiled executable: `csc321shell.exe` or `csc321shellDB.exe` (Windows) or `csc321shell` (Linux). Verify that this executable will run on the Law 113 computers (Windows or Linux).

When grading your work, I will always read your source code and perform two tests: (1) Run the executable you have submitted on a Law 113 computer to check your work, and (2) Compile your code from scratch using the files you have submitted. If your executable does not run in Law 113, or if your code does not compile in Law 113, you will lose points on the assignment, even if your source code is correct. Before you submit, try these tests yourself!

3. Submit

To submit, upload your zip file to Moodle [here](#). Please use a different filename for each assignment (e.g., `csc321shell_project0.zip`). You are only allowed to submit one file, but you may delete and upload a new version until the deadline.