Getting Organized – VS Model

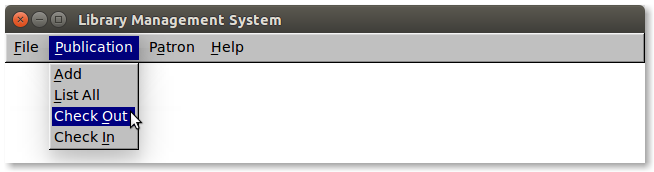
Sprint #3

CSE 1325 – Fall 2017 – Homework #6

Due Sunday, October 15 at 11:59 pm

**IMPORTANT: You may NOT work on any homework project in teams until explicitly instructed to do so. Team-optional projects begin AFTER the second exam.**

While Graphical User Interfaces (GUI) are better received by most computer users than Command Line Interfaces (CLI), users expect to have a “main window” with menus providing access to the programs functions rather than a stream of dialogs. In this third and final sprint, we will replace the fl\_input-based main menu with a main window complete with menu bar, tool bar, and display area listing our publications.



# Requirements

Port your solution (or the suggested solution provided with this assignment if you prefer), to a fuller GUI implementation. For this sprint, you [w](http://www.fltk.org/doc-1.3/group__group__comdlg.html)ill implement a main window with pull-down menu bar (full credit level), add a task bar (bonus level), and provide a scrolling list of publications below the task bar (in the extreme bonus level).

(**You will receive partial credit if you only implement a portion of the requirements.** You will be graded on the extent to which you cover the requirements and the quality of your code. **For FLTK programs only,** you are permitted – but not required – to put all of your code into a single source code file. Use Scrum to manage this third and final sprint. If you don't understand the *intent* of a requirement, feel free to ask – although reasonable assumptions without asking are also fine if you've used a library.)

# Suggested Approach

FLTK can be... temperamental. Therefore, I recommend the following approach.

1. First **build and verify the application from sprint #2 that you will be using for this assignment.** If using the instructor-supplied solution, a simple “make” should suffice. Your solution, if you prefer, may use “fltk-config --compile video\_store\_gui.cpp”. Identifying the correct command(s) to compile your sprint #2 baseline code is your responsibility – I cannot answer vague questions such as “my code doesn’t compile, why not?”, though I’m willing to advise given your exact source files, Makefiles, and compilation command(s) when used in our standard environment.
2. Add the necessary additional FLTK headers to the top of the single source file. For the full credit level, that’s probably FL/Fl\_Window.H and FL/Fl\_Menu\_Bar.H. Verify that the program still compiles and runs. It should behave identically to the pre-modification version.
3. Now, make the necessary changes to create a main window, perhaps using as a pattern the FL Paint 1325 program. Keep it simple – remember, baby steps, although you’ll have to move the entire main menu to the menu bar in one step. (Hint: **If you followed the Model – View – Controller pattern, then FL::run() replaces Controller entirely.**  The code in the “if” or “switch” clauses of Controller::execute\_cmd in sprint #2 now becomes (virtually unmodified) the code for the callbacks from your menu entries. If you switch to pointers for your Video Store and View instances, then remember to change e.g., video\_store.add\_publication to video\_stgore->add\_publication – but using pointers is NOT required.)  
     
   You may use the same dialogs as in sprint #2 to gather information and respond to the menu-initiated request. If you did the bonus or extreme bonus levels on earlier sprints, you can leverage that for additional points for this sprint – **as in real life, doing solid extra work early builds momentum quickly toward superior and valuable software.**  
     
   Build and interactively test your modification. The debugger is your friend. Don't proceed until it works well, and you understand *why* it works well.
4. Now, successively make any other necessary changes to meet the full credit, bonus, and extreme bonus levels and test carefull. These can generally be implemented in baby steps – a tool bar with one button is a giant baby step toward a full tool bar, and a simple publication table below the tool bar can be incrementally improved until it offers the capabilities you desire.

# Grading

* **Full Credit** – For the third sprint, update the Product Backlog tab of the Scrum spreadsheet with the new features (you can **manually merge** rows from the included Scrum\_P6 spreadsheet – do not use it directly!). Then, duplicate the Sprint\_02\_Backlog tab and rename it Sprint\_02\_Backlog, **leaving Sprint\_01\_Backlog and Sprint\_02\_Backlog unmodified** (this will be kept permanently as a record of the first and second sprint, respectively). Then update cells B1:B3 with the updated info. Delete all of the task rows completed in sprint #2, then add rows for the tasks needed to complete the new features for sprint #3. Use this tab to track your progress during sprint #3.  
    
  You will deliver your .h and .cpp class implementations, or a single file named abc1234\_video\_store\_gui.cpp, along with a Makefile that is competent to rebuild only files that have been modified, if applicable. Also include screenshots demonstrating each menu of the main window as images in PNG format, and your updated Scrum spreadsheet for the second sprint, including the Scrum results from your first and second sprint.
* **Bonus** – If you modify the main window to also include a tool bar immediately below the menu bar, with at least buttons to list all media, add a media, check out a media, check in media, list all patrons, add a patron, and display minimal help. Each button must include a (copyright-compatible) icon and tool tip that displays an short English hint (e.g., “Add media”) for each button. Additional tool bar functionality will be worth additional points.  
    
  You will deliver your updated class implementations or abc1234\_video\_store\_gui.cpp, a Makefile, screen shots of each tool tip in PNG format, and your updated Scrum spreadsheet for the third sprint showing the additional tasks.
* **Bonus 2** – If you add a scrollable display of media to the main window below the tool bar, using e.g., the Fl\_Table widget. The table should list the attributes of each media (title, leading actor, director, etc.) in columns, with one row per publication. You will receive even more points if the publications can be sorted by each column, and a ludicrous number of points if the tool bar includes a text field that can be used to filter the table to only include media matching the string or pattern entered.

You will deliver your updated class implementations or abc1234\_video\_store\_gui.cpp, a Makefile, a screenshot of the main window in PNG format **with enough publications created to make the scroll bars visible**, and your updated Scrum spreadsheet for the third sprint showing the additional tasks.

# Frequently Asked Questions

**Q. Where do I get icons to use for my tool bar?**

Oh, anywhere you like – *as long as you respect copyright law*.

* You may draw your own, if you have a vaguely artistic bent, or just don’t care. The graders will not be judging your artistic ability.
* You may go to a public domain collection, or to a creative commons library, as I did for the icon for the Homework #5 solution (from The Noun Project’s Sathish Selladurai collection). The Wiki Commons is a good place to find art licensed under the Creative Commons or the Gnu Free Documentation License (GFDL).
* You may use licensed clipart that came with a program you own, or a clip art collection you previously purchased.

**If the grader detects a violation of copyright law,** your grade could be reduced as low as a 0, and for more egregious offenses may result in your being reported for an honor code violation.

Integrity matters.