Members and Work Done:

Casey Brooks - File I/O

Brannan Bowlin - GUI

Matthew Bowersox - Connecting File I/O to GUI, splash screen

Problem:

The goal of this project was to gain experience in working with teams to solve a problem. The task was to create a basic calendar program, similar to a basic Google Calendar or Outlook. It should be able to add and remove appointments and display all these appointments properly on the calendar, as well as switch between months. There should be a file that contains all the appointments, and should be able to read in appointments without a year as a yearly recurring event, such as holidays.

This is a great beginning project to introduce working in teams, as it encourages writing of code that is easy to read and understand, as well as combining your code with that of another person. It encourages teamwork and collaboration, and with several minds working together, the best solution to the problem can be found more easily.

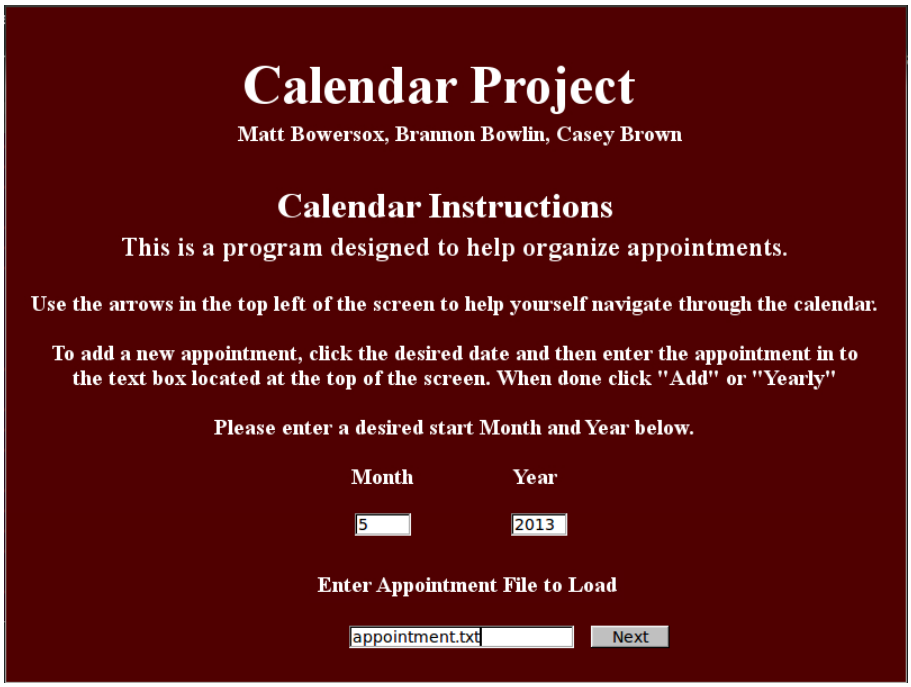
Restrictions:

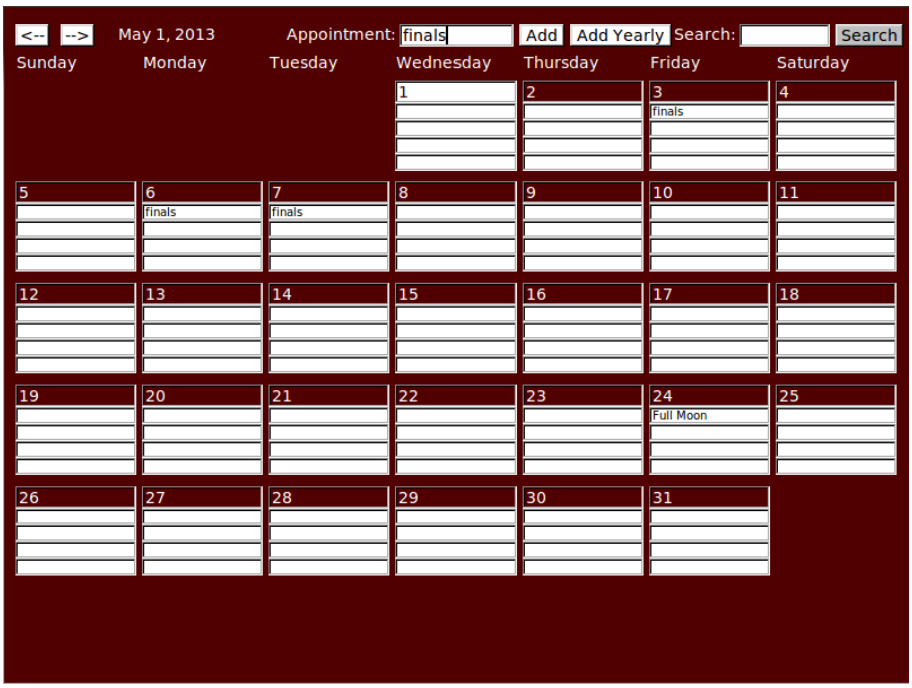
There were few limitations on this project. We were asked to use FLTK as the GUI toolkit, and each function should be fewer than 25 lines(minus whitespace or curly braces), or one terminal window. Apart from these, pretty much anything was available for use, including code from external sources with proper credit given.

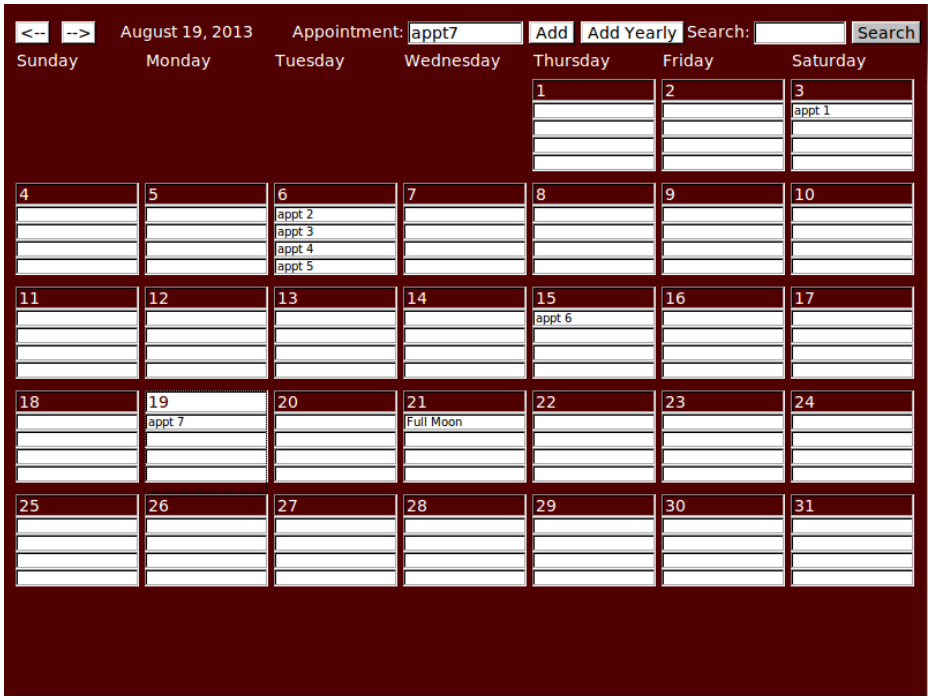
Approach:

The main approach used was to start with what we could do independently, and then work to combine it all into one working program. File I/O could be done without a GUI, a GUI could be built without needing the dates, and a logic as to how to combine the two could be created, given the header files as a base, and so each team member went to work on these separate endeavors. Once each member had finished his own part, we met up several times to put it all together and got it working properly.

Sample Run:


Calendar Project
Matt Bowersox, Brannon Bowlin, Casey Brown
Calendar Instructions
This is a program designed to help organize appointments.
Use the arrows in the top left of the screen to help yourself navigate through the calendar.
To add a new appointment, click the desired date and then enter the appointment in to the text box located at the top of the screen. When done click "Add" or "Yearly"
Please enter a desired start Month and Year below.
Month
Year
5
2013
Enter Appointment File to Load
appointment.txt
Next


<--
-->
May 1, 2013
Appointment: finals
Add
Add Yearly
Search:
Search
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
1
2
3
finals
4
5
6
finals
7
finals
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Full Moon
25
26
27
28
29
30
31



Results and Analysis:

The final project works just as we expected it would. It executes quickly and keeps track of all appointments without error. We have implemented search a search function as well as the occurrences of full moons and recurring events. It can keep track of multiple appointment files, and at the beginning of execution will prompt the user to select one of these files to display and make changes to.

Conclusions:

In creating this program, we have shown how to work in teams to get stuff done. Individually, it would have been difficult to make this happen, but when each person focuses on a specific problem, that problem is able to be solved more effectively, and together with the other parts, the whole program made much better. We have also learned how to utilize header files to integrate another's code even if it is not yet fully implemented, and we learned more about classes and how they relate to one another and can be used by one another for a greater purpose, such as the "chrono" class being an integral part of the "appt" class. Being limited to just 25 lines of code encouraged the writing of clean, highly organized code, as well as being mindful of how many lines it takes to accomplish a certain task and working to be efficient in writing our code.

Future Research:

This program could be further improved by adding more information to each appointment object, such as the start and end time of each event, or the option to have it repeat weekly or

monthly, as opposed to just yearly. It could also be improved by showing multiple appointment files at once on the calendar, and adding or deleting whole appointment files during runtime. But as a whole, it works quite well as it is for a basic calendar program, quick and simple, like a physical calendar but on a computer.

Listing:

The files necessary for the program are as follows:

- appointments.txt
- Calendar_window.cpp/.h
- chrono.cpp/.h
- enterapptfile.jpg
- fileIO.cpp/.h
- Graph.cpp/.h
- GUI.cpp/.h
- main.cpp
- Point.h
- Popup_window.cpp/.h
- splashimage.jpg
- std_lib_facilities_3.h
- Window.cpp/.h

Bibliography:

- Stroustrup, Bjarne. *Programming Principles and Practice Using C++*, Pearson Education 2009.
- Various authors, multiple pages used as reference(no code copied), cplusplus.com
- Forum, multiple threads used as reference(no code copied), Stackoverflow.com