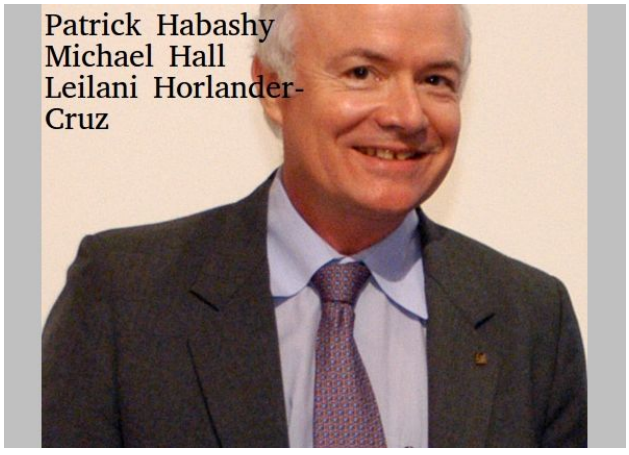
Team Information:
The Defenstrators - Team 3
Patrick Habashy - Tasks 1 and 5
Michael Hall - Tasks 3 and 4
Leilani Horlander-Cruz - Tasks 2 and 6

Problem and Significance:
We were tasked with building a game in C++ using FLTK to make a graphical interface. The game was to use FLTK buttons to allow the user to interact and a calculator from earlier in the semester to find the final score. These are significant tasks because it is very important to be able to build GUIs and to implement outside code into your own projects.

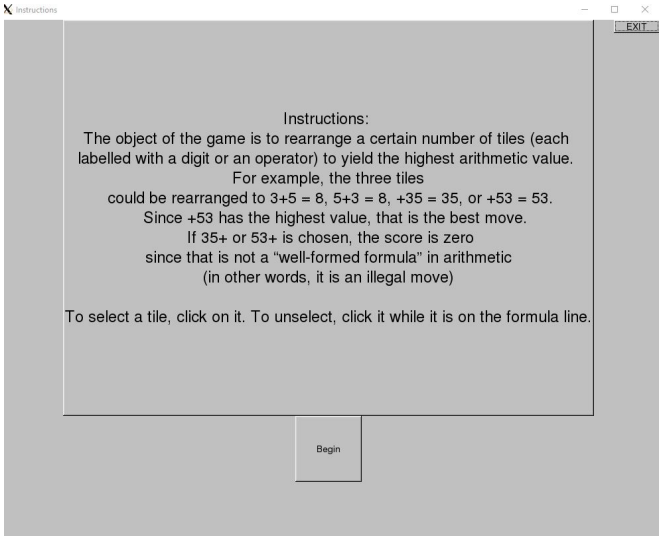
Restrictions and Limitations:
Some restrictions we had were that FLTK is very limited in resources and the syntax didn't always work out. Sometimes, we had issues and segmentation faults that were not descriptive in the least. FLTK is very specific, and leaves little room for error. However, working together and using each other's work for reference made it much more manageable.

Approach:
We took a very object oriented approach to this problem, making a Tile object that handled button callbacks, placement on the screen, and even drawing. Successfully implementing this class made drawing the game board as simple as a single function call, leaving us to focus on score calculation and error handling.

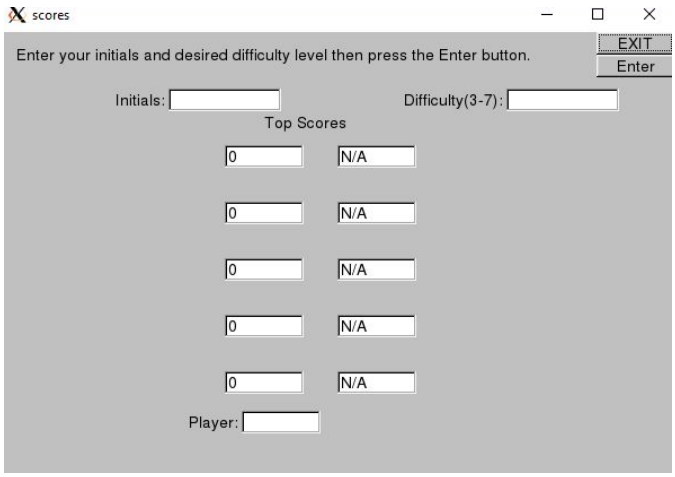
Sample Run:
Splash Screen:



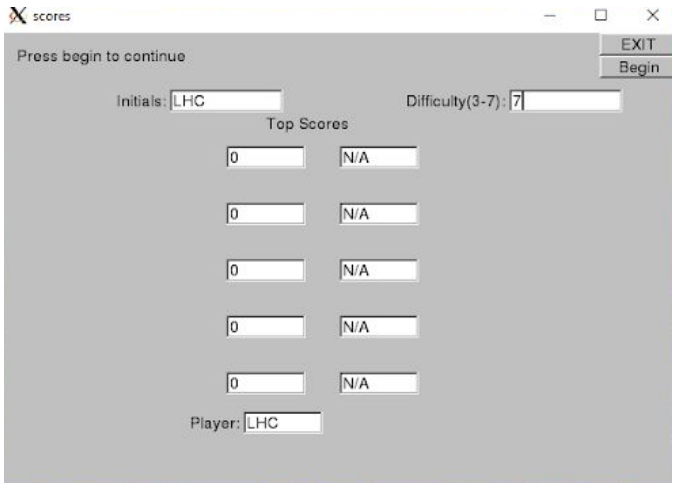
Instructions:



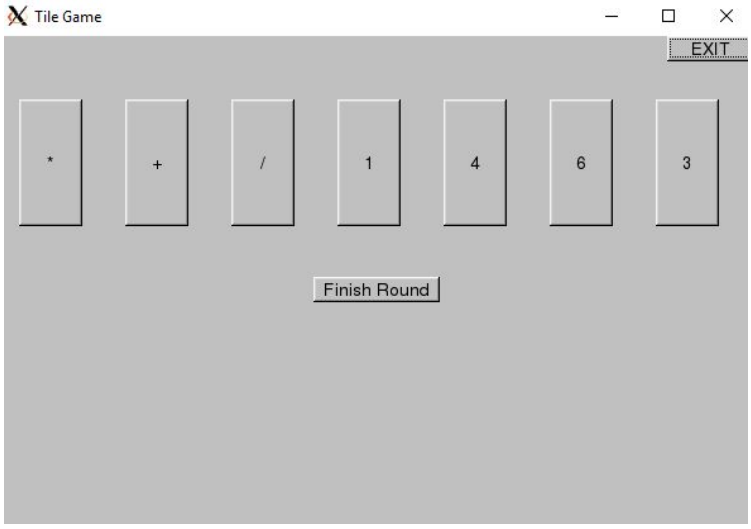
Difficulty and initials:



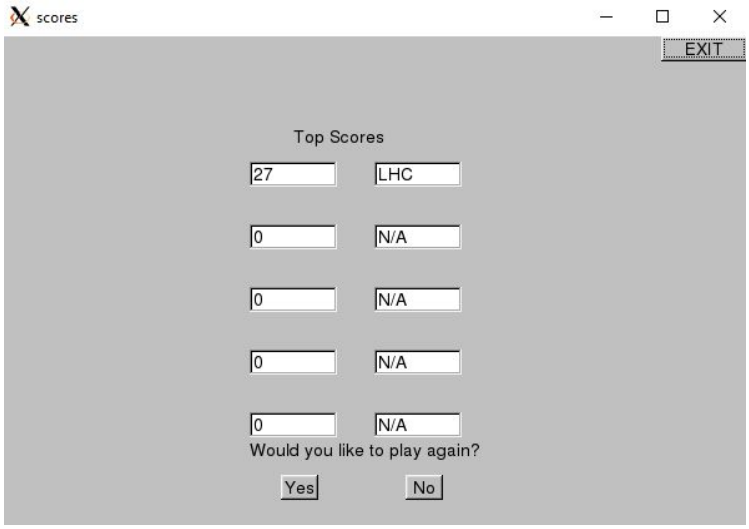
Display scores for difficulty level and initials:



Number of tiles matches difficulty level:



Display top 5 scores for level and ask to play again:



Results and Analysis:

Tying everything together went surprisingly smoother than expected. We were able to work independently and then later come together and wrap everything up. The program functions cleanly and neatly, with no strange errors or warnings. We were able to implement everyone's contributions with little to no modifications.

Conclusion:

For working independently and as a group, we were able to finish in a timely manner. We communicated efficiently, using FLTK's website and our class assignments. This was a really eye opening experience because usually we write our own code and collaborate in different ways. This was a very beneficial project because it is closely related to real world assignments and tasks. Being able to look at someone else's code is not always an easy task, so we had to

ensure we were writing clear, understandable code for not just our own eyes but for our partners as well.

Future Research:
The program could be improved in a number of ways. The obvious include adding the factorial operator and a round timer, but some other ways include adding music, making the GUI more aesthetically pleasing, and making the tile portion of the game happen in the same window as difficulty selection. Additionally, running the game in fullscreen would be an interesting improvement.

Instructions:
After loading the CD onto the computer, the user is to import the folder to their H: drive on the TAMU server. The user then should make sure Xming, X11, or XQuartz, depending on which one they have, is open and running, and then they should open Putty. The user should input the settings so that X11 forwarding is enabled under the SSH tab, and the host name should be build.tamu.edu. The user should then enter their login credentials as prompted in the console window. Once this has been done, the user should map to the folder they just imported from the CD by inputting

```
cd filename
```

Then, the user should input

```
g++ -std=c++17 *cpp -lfltk -lfltk_images && ./a.out
```

to compile and execute.

Bibliography
www.cplusplus.com