Life in True and False: Writing Process

Isabella Phung

January 2023

1 Introduction

This documents the difficulties and specific issues while putting together this small library for Conway's Game of Life. The DESIGN.pdf document elaborates in further detail the design process, structure, and pseudocode of the individual functions. The README.md file also continues additional information in regards to the compilation of these files.

2 Notable Difficulties

I had initially underestimated this program, thinking that it'd be easier and more straightforward than dealing with sorting algorithms. Little did I know that there would inumerable segementation faults ahead of me.
gdb was effectively required for me to work on this program. I got plenty of segmentation faults while working on this. I spent a lot of time placing breakpoints, stepping through my program and trying to figure out the odd values I was getting.
There are a lot of aspects of this program that I think could be polished. One of my big mistakes was mixing signed and unsigned values in my uv_census. Doing so causes a world of headaches and decreases clarity in the program. I had attempted to correct my mistake but it caused a domino effect of issues that caused more headaches than I was willing to handle. I had used signed values since neighbors along the top edge of the matrix would be -1 rather than a postive value. However saving that value doesn't even matter since the neighbor gets set to NULL anyways. The main issue that affected this was using mod for toroidal universes. When I had initially wrote it, I didn't think much about what issues mixing unsigned and signed values would cause; turns out mod doesn't work. It makes sense condisering that decimal values are represented differently between signed and unsigned values.
The way approached this was by checking every single value for it's neighbors but I might try thinking up of a more complex but efficient way of approaching things by focusing more on the live cells rather than every single cell. It would require some diagramming and thinking to figure out the furthest dead cell that could be influenced by a live cell.

There were some tips that I had completely forgotten about until I had to sit and debug. The spec explicitly stated that a good approach would be to have two universes that would work in tandem with one another. I had even taken notes regarding this, but it completely left my mind while I actually put everything together. It would take ages to plug in a test case and step through the many loops I had in my program, so every time an issue came up, a lot of patience was required to step through and figure out what was wrong.
It did take a hefty chunk of time to figure out how to accept user-inputted file names via the command line and how to parse through the file. There are so many different functions that sometimes have confusingly similar names: scanf, fscanf, and fopen are some that come to mind. It took a lot of Googling and testing to understand these different functions and how to best utilize them.
Although I already spent time before actually programming to figure out my general approach to handle finding neighbors, there were still numerous bugs, edge cases, and issues I hadn't thought of in my initial drafts. Programming always takes a lot of patience and time.

3 Resources

fscanf Documentation
fopen Documentation
strtoul Documentation
ncurses clear vs. erase
Command Line Argument example fprintf Examples
Reading in filenames from command line arguments
Differences in Modulo operator when juggling unsigned and signed integers
Understanding Null pointers with arrays