My Takeaways From Creating the Game of Life

Alex Yeh

February 11, 2023

1 Introduction

This assignment was to create “The Game of Life” using a two-dimensional grid of cells that represent a unvierse. It required the use of writing two c files, life.c and universe.c, and a Makefile which was used for compiling and linking. universe.c implements the universe ADT and life.c utilizes a main function to run the game.

2 What I Learned From This Assignment

One thing I learned from this assignment was how to use a struct. More specifically, I learned how to define a struct and how to use it in my other functions. One of the biggest things I learned was how the Game of Life works. I have never heard of it before doing this assignment. I learned how the game was essentially played on two-dimensional grid of cells to represent a unviverse and as the game progresses through generations, the cells in the grid are either alive or dead. To my understanding, there are only three rules to determine the state of the universe after each generation. The first rule is that if a cell is alive and has two or three neighbors, it will remain alive. The second rule is that if a cell is dead and has exactly three neighbors, it will become alive. The third rule is that all other cells die, either due to lonliness or overcrowding. Some other things I learned was how to allocate memory with malloc and calloc and how to use the fscanf function. Lastly, I learned about the ncurses library which I have never seen anything like it before. I found it really cool to create a window that utilizes a for loop to print to create an animation-like effect.

3 How I Made Use of the Ncurses Library

I used the ncurses library to create a window that would display the universe. In life.c, I set a boolean variable called “ncurses” to true. In my -n case, I set this variable to false. Outside my getopt while loop, I had an if statement to check if the ncurses variable was true. If it was true, I would initialize the screen, hide the cursor, clear the window, print an ’o’ if a cell was alive for each generation, refresh the window after the loop of printing was finished, and sleep for 50000 microseconds. An example command to run the program with the ncurses library would be "./life -s -i ./lists/101.txt". Running this command would simulate the Game of Life with the 101.txt file and display the universe in a window for each generation 100 times because the default number of generations is 100. All of this was able to be done with the use of the ncurses library.

4 Insights Obtained From Compiling/Linking From This Assignment

A very important insight I obtained from this assignment was the difference between compiling and linking and how to do both in a Makefile. At first, I put -lncurses in my CFLAGS variable in my Makefile. This caused an error because I was trying to compile the ncurses library. I then realized that I needed to put -lncurses in my LFLAGS variable because linking libraries doesn't occur in compilation. In my Makefile for the rule to compile life.c, I had "$(CC) -o $@ $$̂(LFLAGS)". This command compiles life.c with clang which is what CC is set to, creates an executable called life, and links the ncurses library. This is because life.o and universe.o look for functions in the ncurses library by calling $(LFLAGS).

5 Conclusion

Overall, I really enjoyed this assignment and learning about the Game of Life and other things I needed to utilize to create the game such as defining a struct, allocating memory, and using the ncurses library. I'm sure I will be utilizing these things in future assignments and classes so I'm glad I was able to learn by creating a game.