WRITEUP - ASGN4

Viveka Agrawal

CSE13S, Winter 2023

In this assignment, I learned how to code and implement the Game of Life for both flat and toroidal universes. In my program, the Game of Life is played using 2 Universes A and B. Universe A represents the current state of the Universe. Universe B represents the next state of the Universe. Initially, I populated Universe A with all dead cells. Next, I populated Universe A with live cells using the row-column pairs given in infile. I executed the generate() function in my program to evolve the Universe from A to B using the 3 rules of the Game of Life. The Game of Life has 3 main rules: 1. Any live cell with 2 or 3 neighbors survives. 2. Any dead cell ith exactly 3 live neighbors becomes a live cell. 3. All other cells die, either due to loneliness or overcrowding. After each generation is run, the current state of the Universe is displayed on the screen using the ncurses function mvprintw(). After all the generations are run, the final state of the Universe is printed at the end of the program in the outfile or standard output (stdout).

I learned the concept of toroidal universes and how a toroidal universe looks like compared to a flat universe. A toroidal universe is kind of shaped like a donut where the columns and rows wrap around to the other side. In other words, a toroidal universe has no edges. A flat universe does have edges and therefore our program ends when a row-column pair is out-of-bounds.

I also learned about file processing in and how to read input files and write to output files in C. I learned how to use standard input (stdin) and standard output (stdout) in C. In the life amd64 program provided in Resources, if there is no input file or output file specified by the user, the help message is printed. As a result, I learned how to handle missing arguments for command line options like -i and -o in the program.

In addition, I learned how to define and implement abstract data types like the struct data type. I also learned how to write accessor and manipulator functions for the struct data type and learned how to use the functions from

the ncurses library to develop a text-based interface that displays the state of the Universe.

I learned how to use arrays of pointers and pointers to arrays of pointers to create a finite 2-D grid and how to traverse a 2-D grid by rows and columns by using arrays of pointers and pointers to arrays of pointers.

I also learned how to use the diff function to compare the output of my program vs the output from the life amd64 program provided in Resources. After comparing my results with the Professors, I found out that there is no difference between my program and the life amd64 program.

I learned how to use the functions from the ncurses library to develop a text-based interface that displays the state of the Universe. After each generation is run, I used the ncurses library to show the current state of the Universe. All of the live cells in the Universe are printed out using the ncurses function mvprintw().

I learned the UNIX linking process. I learned that UNIX links binaries from left to right. Linked libraries should always be linked at the end. When an undefined reference is encountered, it is expected to be defined later.