cse13s asgn4 DESIGN.pdf

Lucas Lee; CruzID: luclee

1/28/2022

1 Program Details

In this program we will make a game of life. The game uses a 2x2 grid as an ADT, and the game follows three rules.

1. live cells stay alive if two or three neighbors are alive.
2. dead cells come to life if exactly three neighbors are alive.
3. all other cells die.

2 Files and Pseudocode

1. universe.c
   (a) creates the 2D universe that the game will be played in. Contains functions:
   (b) uvcreate(rows, cols, toroidal) - creates the universe grid. Use calloc to initialize the array to 0's. if the boolean toroidal is true then return a pointer to Universe *.
   (c) uvdelete(Universe *u) gets rid of the created universe that is passed into the function as a parameter. Use free() here to avoid memory leaks
   (d) uvrows(Universe *u) returns number of rows inside of the universe.
   (e) uvcols(Universe *u) return the number of columns in the specified universe

(f) uvlivecell(Universe *u, uint32 r, uint32 c) sets the cell at row r and column c to live. use booleans to mark live and dead cells, true is live, false is dead. if the row and column don't exist in the universe given, then don't change anything.

(g) uvdeadcell(Universe *u, uint32 r, uint32 c) sets the cell at row r and column c to dead. Use false to mark the cell as dead. Do nothing is r and c don't exist in the Universe.

(h) uvgetcell(Universe *u, uint32 r, uint32 c) returns the boolean value at row r and column c. Return false if r and c don't exist.

(i) ucpopulate(Universe *u, FILE *infile) creates universe using the given infile. Line 1 of the file should be the number of rows and columns separated with whitespace. Every other line in the file is the row and column of the live cells in the universe for that file. fscanf() reads row-column pairs for use in the universe.

(j) uvcensus(Universe *u, uint32 r, uint32 c) returns number of adjacent live cells to the one given at row r and column c. If toroidal is set to true, then also consider the other side of the universe grid as adjacent if the live cell is at the edge of the universe. Otherwise do not consider this option.

(k) uvprint(Universe *u, FILE *outfile) prints the universe with the live cells being 'o' and the dead cells being '.'. Use fputc() or fprintf to specify the outfile as the place to print the universe.

2. universe.h

   (a) header file for universe.c that is going to be included in the file containing the main() function.

3. life.c

   (a) contains main() function and runs the game of life simulation code using universe.h.

4. Makefile

   (a) use CC = clang, CFLAGS = Wall, Wextra, Wpedantic, Werror

   (b) make, make life, make all must make the life executable file

   (c) make clean removes the files that the compiler generates

   (d) make format formats all c and h files

5. README.md

    (a) describes how to use the program, run the program, and compile the program. Also includes errors in the code.

6. DESIGN.pdf

    (a) describes the design process, and how to make the desired functions.