Guillermo Alfaro
CSE13S Winter 2022

The Game of Life Design Document

Description:

This program will simulate The Game of Life by John Horton Conway. Given a starting point as a file, the game runs on its own. It's comprised of cells that are either dead or alive, any live cell with two or three live neighbors survives, any dead cell with exactly three live neighbors becomes a live cell, and all other cells die.

Files:

1. Universe.c:

    Implements the Universe ADT

2. Universe.h:

    Specifies the interface to the Universe ADT, provided file.

3. Life.c:

    Contains main()

4. Makefile:

    File to easily compile and remove any lingering files created.

5. README.md:

    Describes how to use the program and Makefile.

6. DESIGN.pdf:

    Describes the design process with enough detail to recreate this program given the same materials.

Pseudocode:

Universe.c

```
Struct universe
    Uint32_t rows
    Uint32_t cols
    Bool **grid
    Bool torodial;
*uv_create(rows, cols, torodial)
    Reserve size amount of calloc(rows, sizeof(uint32_t *)) = matrix
    For loop r = 0 r < rows r+ 1
        Reserve size amount of calloc(cols, sizeof(uint32_t));
    Set if toroidal
    Set rows
    Set cols
    Return universe
void uv_delete(Universe *u)
    For i < col i ++
            free(reserved amount of calloc)
    free(u)


Uint32_t uv_rows(Universe *u, text file name)
    Return u.rows;
```

```
Uint32_t uv_cols(Universe *u, text file name)

    Return u.cols;


Void uv_live_cell(universe *u, uint32_t r, uint32_t c)

    *u[r][c] = true;


Void uv_dead_cell(universe *u, uint32_t r, uint32_t c)

    *u[r][c] = false;


Bool uv_get_cell(universe *u, uint32_t r, uint32_t c)

    return *u[r][c]


Bool uv_populate(universe *u, FILE *infile)

    Infinite loop

        Break if EOF

        fscanf(% %/n, rvar, cvar)

        uv_live_cell(rvar, cvar)
```

Life.c

```
Main()

    Initialize bools/variables for

        -t      toroidal

        -s      silence ncurses
        -n      specify number of generations default 100
        -i      specify input file name, default stdin
        -o      specift output file to print final state of universe, default is stdout


Get opt switch statement to receive all inputs
        Assign each input to the respective variable
        Read file to get rows and columns and assign them to their variables
Universe A = uv_create(rows, cols, torodial)
Universe B = uv_create(rows, cols, torodial)
Universe swap
uv_populate(Universe A, textfile)


For (int x < n)
        While loop simulates the actual game of life
        Does all the calculations wether a cell is dead or alive
        Uses ncurses to prints to screen
        Creats updated universe with next generation
        and loops here
Close screen
Print final state of universe
```

```
uv_delete(a)

uv_delete(b)

Return 1;
```