Mylo Lynch

Darrel Long

Cse 13s

2/5/23

The Game of Life

Deliverables:

Universe.c: implements Universe Abstract Data Type.

Universe.h: specifies the interface to the Universe ADT. provided unmodifiable header file

life.c: contains main() and may contain any other functions necessary to complete implementation

Rules of Existence

1. Any *live* cell (marked with a true boolean or a 1) with two or three live neighbors survives.

2. Any dead (marked with false or 0) cell with exactly three live neighbors becomes *a living cell*.

3. If a cell has less than two or more than three neighbors it will die from either loneliness or overcrowding. One cell can have up to 8 neighbors (up, down, left, right, and diagonally).

Universe.c

Write these functions in order to make the game of life:

Universe *uv_create(uint32_t rows, uint32_t cols, bool toroidal)

void uv_delete(Universe *u)

uint32_t uv_rows(Universe *u)

uint32_t uv_cols(Universe *u)

void uv_live_cell(Universe *u, uint32_t r, uint32_t c)

void uv_dead_cell(Universe *u, uint32_t r, uint32_t c)

bool uv_get_cell(Universe *u, uint32_t r, uint32_t c)

bool uv_populate(Universe *u, FILE *infile)

uint32_t uv_census(Universe *u, uint32_t r, uint32_t c)

void uv_print(Universe *u, FILE *outfile)

life.c

This file contains the main function that will use ncurses to run the game of life, as well as unistd.h's usleep() function to add a delay (so the game of life doesn't happen instantaneously).

Loop through columns while looping through rows while looping through generations in order to recreate the game of life. In the deepest nested loop, implement the rules of the game using if and else if statements.

Makefile

```
CC      = clang
CFLAGS  = -Wall -Wextra -Wpedantic -Werror -g
TARGET  = life

all: $(TARGET)

life: universe.c
        $(CC)   -o sorting universe.c -lncurses -lm

sorting.o: *.c
        $(CC)   $(CFLAGS) -c *.c

format:
        clang-format -i -style=file *.[ch]

clean:
        rm -f life *.o
```