

Design Doc

Manikanta Illuri - milluri

February 10, 2023

Initial Design

Objective: The objective of this assignment is to create the Game of Life. o

Deliverables:

- Makefile
- README.md
- DESIGN.pdf
- WRITEUP.pdf
- Universe.c
- universe.h
- Life.c

Universe.c

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

- This function's objective is to create a universe with three parameters; row, col, and bool value. This function will first have a while loop that runs while the condition $i < \text{number of rows}$ is present. Within the loop An if will double check if the i value is greater than rows, if so it will break. Else it will use calloc to allocate memory for the universe it is generating. It will then increase the iteration and return the value of u which is a variable to the universe.

void uv_delete(Universe *u)

- This function will return nothing as it is a void function. It will take the parameters of the universe pointer. It will free() to free the universe and set it to NULL.

uint32_t uv_rows(Universe *u)

- This will return the rows

uint32_t uv_cols(Universe *u)

- This will return the cols

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

- This function will take the pointer for the universe u, rows and cols. The if statement inside will check if the rows and cols are less than the universe r and c vals. Then it will set the universe values for r and c to true. Else it will return nothing.

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

- This function will take in the parameters of the universe and a pointer and row and cols. Then it will check if the row and cols values are greater than what the universe has, if so it will set the universe values to false. Else it will return nothing.

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

- The universe's function will take the universe as a pointer and the rows and cols. Then it will check if they are greater than the r and c values, Then it will return false. Else it will return the row and col values of the universe.

bool uv_populate(Universe *u, FILE *infile)

- This will populate the universe with the parameters of the infile and the universe. The function will have a while loop that will be fun while the conviction !feof(infile) is false. Within it will have two variables r and c that stand for the row and col values. Then if the statement is scanned it ifscan will read the file and check if the rows and cols are greater than the ones inside the universe. Then it will return false. Else it will call uv_live_cell.

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

- This function will take in 3 parameters of the universe and the rows and cols. I will have a counter variable that will be set to 0. Then 8 pairs of elements are initialized as mentioned in the asgn doc. Upon initializing these sets I will have a for loop that runs while i < 8. Within the loop I will initialize 4 variables of type uint32_t which will be the

moves for ro and cols. An if will run while u->toroidal is true. And within it rows will be calculated by using $(row + u->rows) \% u->rows$ and cols will be calculated and stored in the value by $(cols + u->cols) \% u->cols$. Then and other if outside the above one will run will the value of rows is greater than 0. Within this the counter will increase if uni is already set with r and c values. Outside the loop counter is returned.

void uv_print(Universe *u, FILE *outfile)

- This function will print the “o” or “.” based on its next loop iteration through the cells in the universe to the outfile.

Life.c

This function will contain the main function that will enable all the switch cases. I will predefine things such as in_file and out_file and use bool for silence. I will also have the switch cases enables

- T
 - Will create the universe as a toroidal
- S
 - Will not have an animation
- N
 - Set number of generations
- I
 - Will make a in file
- O
 - Will make an outfile
- H
 - Will give a help message
- Default
 - Will also return the exit program.

I will have a if the in file is NULL and will return a message. There will also be a check to see if the input is a number and if the count is not equal to 2 as only two inputs are taken. I will

also have an if to create uni A and will check if there was any memory allocation problem. I Will do the above for uni B. I will also have a check to see if the universe is getting populated or not. I will have a check to see if silent is not then will initscr(). I will also have a while loop that iterates while gen is less than generatiosna dn will use the ? to switch between uni A and uni B. I will iterate the cells with a nested loop and print using nested for loop. Will also have a nested for loop to iterate through cells and place them in another universe. Will close files and if they are not NULL and delete the uni.