

# ASSIGNMENT 4 DESIGN DOCUMENT

Chris Moon

February 2023

## 1 Goal

Implement a matrix-style data type, with corresponding functions, allowing the user to simulate John Horton Conway's game of life.

## 2 Pseudocode

### RULES

- each element can be dead or alive
- use a boolean perhaps?
- dead cells with three alive cells bordering, revive
- cells with 4+ neighbors or less than 2 neighbors die

### UNIVERSE STRUCT

- there needs to be a matrix on which to run the simulation
- C doesn't have a matrix type
- create one with a struct (similar to an object)
- the struct has an int to represent rows, and one for columns
- use these values for accessing specific elements in the matrix
- the struct uses a double pointer "grid" of type boolean to keep track if a cell is dead or alive
- the struct also contains a variable to track if its a toroidal type or not

### UNIVERSE CREATE FUNCTION

- creates the universe data type
- allows users to use the matrix data structure, as well as specify the rows and columns
- takes rows and columns as parameters and uses calloc to allocate memory for a matrix
- calloc initializes array values at 0
- pretty simple: allocate an array of type boolean of size row, and then for all the rows, set the value to another array of type boolean of size column.
- returns a universe type

### UNIVERSE DELETE FUNCTION

- clears all arrays

- need to clear the inner arrays before the outer array, since the data type is a matrix (an array of arrays).

#### UNIVERSE RETURN ROWS

- takes a universe struct
- return the rows for that specific universe
- use u->rows (fine to use since it's inside the universe.c file)

#### UNIVERSE RETURN COLUMNS

- same as above, but return u->columns

#### UNIVERSE SET LIVE

- takes a universe and a row-col coordinate
- sets the boolean value there to true (alive)

#### UNIVERSE SET DEAD

- takes a universe and a row-col coordinate
- sets the boolean value there to false (dead)

#### UNIVERSE GET STATUS

- returns the live/dead status of a specific cell

#### UNIVERSE POPULATE

- uses file data
- files are set up in row-column form, like a matrix
- first two values = row and columns of total universe
- everything after is a coordinate pair for a live cell
- returns a new universe type, using the data from the file