

# Contemplating Life and the Universe(.c)

Angela Shen

February 2023

## 1 Abstract

There are many ways to conceptualize the entirety of the universe. One example is the famous Shakespearean quote is "All the world's a stage", describing the world as a stage with individuals as players. Oscar Wilde has satirized this quote, saying "The world is a stage, but the play is badly cast". Fortunately, if anything were to be badly cast in `universe.c`, it would probably cause an error at compile time. This clearly demonstrates that `universe.c` and `life.c` are far superior to the real thing.

## 2 Introduction

Conway's Game of Life is one way to visualize a universe using individual "cells" on a grid. Like biological cells, these cells might "live" or "die" based on how many neighboring cells it has.

**Makefile** - Compiles an executable of and creates multiple object files (`life.c`'s dependencies). For a list of these `c` files, `.c` below.

**README.md** - Details the process of building any necessary files, the command line options for any executables, and any errors or bugs.

**DESIGN.pdf** - Contains the pseudo code and descriptions of each `c` file.

**WRITEUP.pdf** - This document. Describes the assignment in its entirety and discusses the results.

**life.c** - Implements the Game of Life. Accepts user input from an input file or stdin to populate the initial universe, then goes through multiple generations of the universe and outputs the final result to the output file or stdout.

**universe.c** - Implements the functions for the universe structure that are required for `life.c`.

**universe.h** - Contains the declarations of the functions in `universe.c`.

## 3 What I learned

Overall, this assignment gave me a much better understanding of C as a whole, since I had to review about half of *The C Programming Language* book just for this assignment (Specifically Chapters 5, 6, and 7).

### 3.1 Pointers and Structures

Despite the previous assignment containing both pointers and structures, I still had a lot of struggles with both of these concepts. I have had multiple tutors correct me on my syntax and understanding of pointers. Syntax was especially confusing for me because it was sometimes hard to find specific examples of what I had questions about (for example, I was unsure if `u->grid[some int]` was valid syntax). This took an embarrassing amount of time, but it was definitely well spent.

## 3.2 Input and Output

I think this part was the most time consuming since we have never directly learned about fscanf and using files in c. In addition, a lot of the information about scanning, printing, and accessing files is split between Chapter 7 and Appendix B in the book. I spent a lot of time reading the book and the individual manual pages for each function. Now, I have a better understanding of how opening, reading, writing to, and closing files works.

## 3.3 Memory Allocation and Valgrind

This assignment was also good review for memory allocation. It was a good demonstration of how to properly free arrays and pointers (free what is inside of an array, structure, or pointer first, then free the pointer itself). Additionally, I got more practice with using gdb and valgrind to debug and check for memory leaks, though it was confusing at first since ncurses always has memory leaks.

## 4 Conclusion

I want to clarify none of this complaining. I struggled a lot but I completely understand that this is on me and I think the growth from my efforts made it worth it. I like not being taught everything directly and having to figuring some concepts, but I think the straightforward way I write these makes it seem like I despise it entirely.

Thus far been trying to grasp the basic concepts of C, but now that I have read and reviewed most of *The C Programming Language* for this and previous assignments, I think I am starting to get a better understanding of the language and will hopefully be able to apply this knowledge more effectively in the future.