

On the Origin of Binary

The Game of Life

Arnav Nepal

February 13, 2023

1 Introduction

In this assignment we were tasked with implementing John Horton Conway's *Game of Life* in C. The Game of Life (or just "life") is a simple game that operates under the following 3 rules:

1. any living cell with 2 or 3 neighbors survives into the next generation
2. any dead cell with exactly 3 neighbors comes alive
3. all other cells die

Life is a zero-player game where the initial state of the universe determines the evolution of the game and the user is unable to affect the game universe (except to stop the game). We are tasked with simulating the Game of Life in a finite universe. We will be implementing the game in both a normal finite universe (with edges) and a toroidal one.

2 Lessons Learned

This assignment served as a full introduction to working with structs in C. I learned a lot about how to work with structs, files, and pointers.

2.1 Structs, Files, and Integers : Universe.c

While we worked with structs a little bit in assignment 3 with the statistics module, this was our first introduction to actual implementing our own abstract data types (ADT) with structs. This took the shape of the "Universe" struct that determined the characteristics of the universe that the Game of Life was to be played on. I learned about how to implement an *opaque* ADT, where the only way to interact with the data within should be with user created functions, and not direct access. Furthermore, creating the Universe ADT also allowed me to learn how to work with files directly in C, in contrast to previous assignments where we only used standard input and output along with bash to work with files. File interaction in C mainly works through a

pointer that is set to the beginning of the file when `fopen()` is called on a `FILE*`. We then use `fprintf()`, `fputc()`, `fscanf()`, and other file scanning and editing function to “read from” or “write to” the file.

2.1.1 `uv_census()`

The hardest part of this assignment for me was implementing the toroidal case for `uv_census`, which scans adjacent cells of a particular cell to determine the number of neighbors. With a toroidal universe specified, on cells on the edge of the plane `uv_census` had to check cells on the opposite edge as well, since the universe “wrapped around” on itself. The math for this “wrap around”, while requiring a little thinking, was not too difficult. It helped that I drew (on paper) a diagram to visualize the math behind the toroidal case. However, I was stuck on the case for a while due faulty assumptions I made about the `mod (%)` operation in C. I believe that it would be equivalent to `mod` in mathematics, which can only output positive numbers. This is not the case in C, where it seems to work a little different and outputs different results for negative numbers (including negative results). Due to the fact that I was working with unsigned integers, this obviously caused problem. I was eventually able to resolve the problem after figuring out how `mod` worked in C and casting unsigned integers to signed integers so they would be capable of holding a negative number.

While implementing this function, I learned to *be very careful* when working with integers, especially when combining signed and unsigned integers. This is because C will not warn you of any problems if your code does unexpected things with integers, and it can be difficult to track down.

2.2 `Ncurses and Objects (ADTs) : life.c`

This was our first time working with ADTs in C. It was not too difficult to learn since it mostly amounts to calling the functions we implemented, which should do everything that is needed to be done. This is because we don’t want to access the ADT directly within `main()`. The harder part was actually implementing the ADT in `Universe.c`. The Game of Life was also not too difficult to implement as it is based on very simple rules. `Ncurses` was also not too difficult to work with, and I was able to implement a working printing logic by closely mirroring logic from my `uv_print()` (basically, printing character by character).

3 Conclusion

This was a fun assignment to bring together and see *come to life*. This assignment taught me the importance of writing things down and drawing out my ideas on paper. For me, seeing a diagram on paper ironically makes it much easier to implement the logic behind said diagram in code. It was also interesting learning about the *Game of Life*. I’ve heard about it before, but never really played around with it, so it was interesting to actually play around with it and learn about it’s significance in computer history.