Contents

1 Overview / What does the program do?

2 Program Description / How does it do it?

3 Testing

Unit testing was done in order to iteratively develop the Game of Life algorithm, display function, and helper functions. Although the finished program assumes the size of the working area is effectively 64 by 64, in order to test the algorithm and game loop it was necessary to test with a smaller grid, which was chosen to be 3 by 3. This ensured that any logic errors were able to be dealt with swiftly. Since the births and deaths array initialized with a smaller size of 9 elements, tracing step by step in Mars was quicker and solved many off-by-one errors and missed branch statements.

For the algorithm, the first case tested was a single pixel in the middle of the grid, with the expected result of the pixel being set as dead (the color black) in births and deaths array and written back to the display. After getting the display function working, the width was increased to the final 64 pixels and a 4 pixel square was placed in the center, with the expected result of the square staying static. Next, the corners and edges of the display were tested again with the same square and with single pixels to make sure they were working correctly. Finally, a pattern was recreated from an online simulation (`https://bitstorm.org/gameoflife/`), the 10 Cell Row pattern, which ended up becoming a preset for the final program.

Tracing issues with the display function was not as straightforward as tracing the algorithm. The display could not be tested in its entirety at the same time as the algorithm because of the nature of the display. Since the display is continuous memory with 64 pixels being the cutoff point for displaying the next line, the 3 by 3 array simply showed up as 9 horizontal pixels. Although this helped in some regard with assuring the algorithm was working, it was not sufficient for testing whether the display function would work over the entirety of the 4096 pixel grid. The best course of action was then making sure the display function worked correctly at small sizes and gradually working up to the full grid size, where the only remaining possible issue would be having an off-by-one error. The bitmap display was tested at 9 pixels, 64 pixels or one full row, half the grid, and finally the full grid.

A planned feature of the program was the ability to take a user-specified width as the baseline for the working area; however, the options in the bitmap display only provide support for widths that are multiples of 64 (64, 128, 256, 512 and 1024), and any number provided in-between would have been displayed off-center. Another option provided in the bitmap display is a width and length option of each pixel, where increasing those values decrease the effective working space.