

Design and Implementation

For my implementation of the update method I made a copy of the original board using memory from the heap to make use of its dynamism. I called a double loop that goes through the 2d array and checks each elements neighbors. It checks the elements neighbors by using a bunch of conditionals to determine its position in the matrix and by using a counter I add up all the elements of the neighbors which is 0 or 1. using this counter we can determine the number of neighbors which is just whatever the the value of the counter is. If we know that the element of the matrix is we can determine whether the value should be a 1 or 0 based on the rules for conway's game of life.

Challenges

There were a lot of challenges for implementing this method. The biggest challenge was probably writing assembly itself. Finding out what the equivalent C code for each few lines of assembly is was hard to understand. Therefore, I had to reference the instruction set for assembly and look at the power points very frequently.

Big O and Space Complexity

The number of of comparisons done in this program varies depending on the position of the element of the array. The number of comparisons mainly depends on the dimensions of the $m \times n$ array so the worst case big O complexity is $O(mn)$.