Introduction

We have created a program that is able to simulate the cellular automaton developed by John Horton Conway named 'The Game of Life'. The game evolution is determined by its initial state and requires no further input. Every cell interacts with its eight neighbour pixels: cells that are horizontally, vertically, or diagonally adjacent. At each matrix update in time the following transitions may occur to create the next evolution of the domain:

- Any live cell with fewer than two live neighbours dies
- Any live cell with two or three live neighbours is unaffected
- Any live cell with more than three live neighbours dies
- Any dead cell with exactly three live neighbours becomes alive

Our program successfully computes the automaton for a given number of moves concurrently, using multiple threads.

Functionality and Design

Our solution was built up by initially creating a single threaded solution to the problem. This version iterates through the board bitwise, and for each bit gathers all of the 'neighbours' for the cells (the 8 directly adjacent cells). From this, the logic is applied and the cell is updated if necessary. This is repeated for the desired number of turns.

From this, we created a multi-threaded solution. We split the board up into strips and passed each strip to a worker. However, each worker would also need information from the lines directly above and below its strip of cells (called *halo lines*). We decided to pass these halo lines wrapped around the strips, so that the workers are able to calculate each cell correctly. Once they have completed their strip, they return it to the distributor function. The function reconstructs the world and begins the process again for the desired number of turns.

Experiments and Critical Analysis