Synchronization Strategy

Version 1:

1 mutex lock for each row of the grid (so numRows locks). This protects the row from being changed by another thread when one thread is already doing that. If there are more rows than threads, the threads will divide up the number of rows based on the max number of threads.

Version 2:

There's a 2d array of mutex locks, to protect each cell on the grid, so that when a thread randomly chose a cell to change, it wouldn't have a conflict with another thread.

Difficulties

Making it so a low number of threads could take care of a much larger number of rows, as in determining what rows each thread was responsible for.

Writing to the pipe and opening it at the right point.

Current Limitations

If a user enters more than 100 characters for a bash command, everything crashes because my buffer for reading from the pipe can only hold 100 characters.

If a user types "end " and another command after that without pressing enter, the cellular automation exits, but the bash script does not.

The display for number of live threads is inaccurate, the variable is changed appropriately, but the decrement happens when a thread joins, and rendering is after all the threads have joined again.