PROGRAMMING INTERVIEW PROJECT

Please complete the following programming project using the language of your choice. Your solution should demonstrate your thought process and approach to writing "complete, deliverable code", as well as your understanding of the language's syntax, conventions, and paradigms.

Conway's Game of Life (http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life#Rules)

> The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead. Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:
>
> > Any live cell with fewer than two live neighbors dies, as if caused by under-population.
> >
> > Any live cell with two or three live neighbors lives on to the next generation.
> >
> > Any live cell with more than three live neighbors dies, as if by overcrowding.
> >
> > Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.
>
> The initial pattern constitutes the seed of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a tick (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.

There should be some way to pass in a tab-delimited file containing the list of coordinates of the initial live cells. There should also be some way to specify the number of generations to run, after which the program produces a similar tab-delimited format describing the end-state of the grid.

Sample input file:

```
# this is a basic glider, centered at the grid origin
# (first column is x-coord, second column is y-coord)
-1      -1
0       -1
0       1
1       -1
1       0
```

Your solution should be provided as a zip/gzip file containing all files and instructions for compiling (if necessary), running tests (if any), and running the program. Do not include third party libraries or toolchain components that you reference (if any), but do note which ones are used and how to acquire them.