Project 3: Life's game

Introduction: *Life* is a computational puzzle that simulates the "up's and down's" of a particular population through the time. The puzzle's entities live in a two-dimensional world (namely, a checker's board of any size). Following a precise set of rules, the entities, appear, disappear or remain in the world, from one generation ("turn") to the next. The aim of this puzzle is that, given an initial arrangement of entities and a fixed number of generations (i.e., several pawns distributed over the board, changing status each turn), displays the configuration of the world in each generation.

Life's rule is in fact a very simplified **model** of a population, the (**discrete**) **simulation** represents its "life" (since the Big Bang to the Big Crunch ☺), and the visualization of the population status produces an interesting sequence of patterns (that is, some ceramics for your swimming pool's floor).

Description: Design and test a paper- based simulator for the Game of Life.

Goal: Practice strategy and algorithm design, handtracing, and simple prediction, of a well-defined problem.

Objectives:

1. Solve a typical programming exercise, but aiming for a paper- and-pencil execution.
2. Perform a *documented search* to find the information required to solve the problem.
3. Elaborate the **documentation** for the solution given to a particular problem. (That is, a formal document containing the problem specification, the analysis of resources, inputs and outputs, the design to transform inputs into outputs given only the resources, and the data set used for testing [i.e., board configurations: initial state and changes for several time units]).
4. Recognize (enlist) low-level problems (component mechanics "details") that somehow have to be addressed to implement the solution.
5. Determine the usefulness of the Basic Strategy for Algorithmic Problem Solving with respect to this project.
6. Find algorithmic thinking concepts within a concrete situation (see the *Checklist for AT fluency*).

Instructions:

By carefully following the last two stages of the General Strategy for Algorithmic Problem Solving (http://www.cs.jhu.edu/~jorgev/cs106/ProblemSolving.html), and the experience you may have had with the first ones, make a document with qualitative and comprehensive explanation of the problem, concepts involved, tools required, resources, data and results; along with an **user's manual**, a clear and simple description, written for an unknown third party, who will used to play the game with paper and pencil.

Use the questionnaires to guide your thoughts and the flowchart to guide your search. Keep a log of your work: for every stage, jot down comments and answers to the questions that best lead your work. Also record the time/date you start each stage.

Hints:

1. You do not have to elaborate a computer program as result of this project, but you can use the computer, at any stage, as a tool to make your work easier.
2. Life's rules, and examples, are hosted in several sites of the WWW.
3. You may think that your solution will be played on a checkers board.
4. If of any help, remember Excel…

Answer the following questions:

1) What algorithmic principles were applied while planning the Life's Game?
2) What non-computational concepts ("theory") were needed to perform this activity?
3) In what way this problem was abstract? In what way your solution is abstract or concrete?
4) What assumptions did you have to make to work out the problem?
5) How many components (modules) did you find in your model?
6) Describe the "mechanics" used by your model to keep track of the population status from one generation to the next.
7) What happens if the simulator does not remember the population status in the generation before to the current one?
8) What happens if the simulator does not remember the population status in three generation before to the current one?
9) Indicate why your plan for the simulator is just a model.
10) If possible, describe the "architecture" of your model by drawing a hierarchical scheme of its components (use a rectangle for each part.)
11) What are the basic actions (**primitives**) the entities living in the Life's world can perform?
12) What do you think are the basic actions ("primitives") the simulator of Life can perform?
13) Think of the population (**state**) visualization component, and explain why it can be regarded as black-box, and how its internals could work.