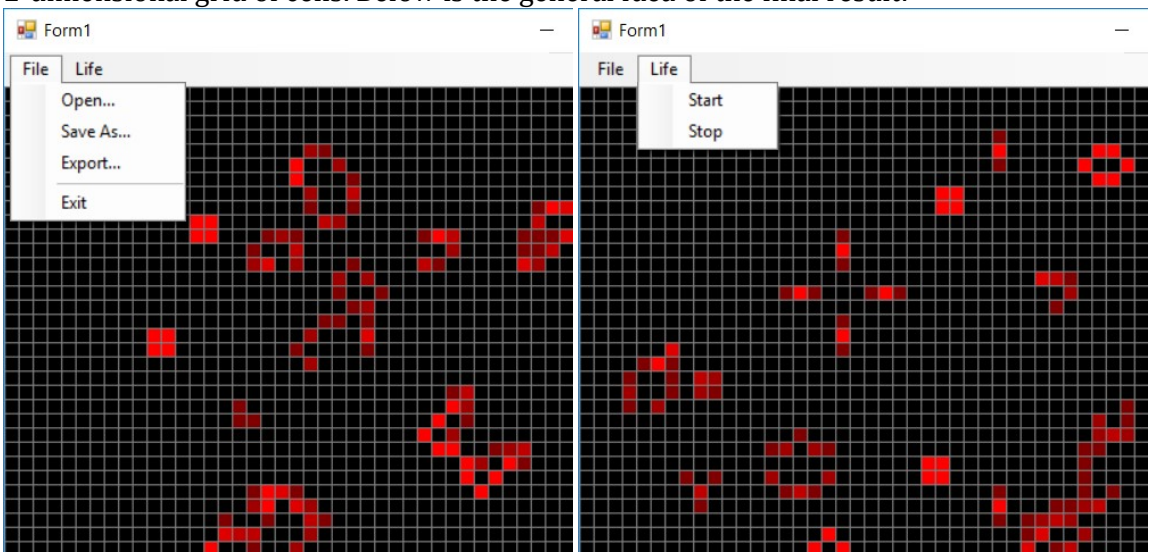
ADA University
School of Information Technologies and Engineering
CSCI 2408: Computer Graphics
Spring Semester, 2018

Homework 1: Bitmap images

Time – not important. Only life – important.

Instructions:

- **Write a program** (Windows Forms App with C#) using Microsoft® Visual Studio® (Community Edition) that will simulate the **Game of Life**[1] using consecutive application of simple rules to create new generations from initial population that live on the orthogonal 2-dimensional grid of cells. Below is the general idea of the final result:



- **Your submission** must be a .ZIP archive of the Visual Studio® solution/project folder. Make sure that it contains all the necessary files, so the grader can successfully compile and run your program.
- **The submission will be graded** based on the following criteria:
  - Correctly functioning code 60%
  - Well documented code 20%
  - Neat program code 10%
  - Solid use of OOP approach 10%
- **Documentation** is required to be in a form of commentary in the program code. It must explain the use of different parameters. It must include all references and cite any models, images, ideas, or algorithms that you did not develop yourself.
- **Minimum requirements** include the following:
  - The user should be able to enter the initial population by clicking the cells. For example, left click makes the clicked cell *alive* (populated), and right click makes it *dead* (unpopulated).
  - The simulation will apply the four rules below to the cells and update the image after regular periods of time.

[1] The *Game of Life* is a cellular automaton devised by the British mathematician *John Horton Conway* in 1970. Watch its simulation and read more about it at a Stanford site: http://web.stanford.edu/~cdebs/GameOfLife/

  - The simulation will show a 50x50 grid with everything outside considered as dead.
  - The user should be able to start or stop the simulation animation, as well as run it one step (*tick*) a time.
  - By default, the opening window should be well positioned whatever is the size of the screen.
  - The user should be able to export the animation as an animated GIF file, specifically in the original GIF87a format[2].
- **The evolution process** goes as follows: The initial pattern constitutes the *seed* of the system. At each step in time, the following transitions rules are applied:
  1. Any alive cell with fewer than 2 alive neighbors dies, as if caused by underpopulation.
  2. Any alive cell with 2 or 3 alive neighbors lives on to the next generation.
  3. Any alive cell with more than 3 alive neighbors dies, as if by overpopulation.
  4. Any dead cell with exactly 3 alive neighbors becomes an alive cell, as if by reproduction.

  The first *generation* is created by applying the above rules simultaneously to every cell in the seed – births and deaths occur simultaneously, meaning that only the 'values' of cells in the previous generation are considered while applying the rules. They continue to be applied repeatedly to create further generations.

- **Bonus(es): Optional features** are listed below. You will have to do at least one to get the maximum mark. The others will get you extra 10% points each:
  - Let users save the current generation into a file and load the initial seed from a file.
  - Add interface elements to let users change at least two of the following parameters easily: color(s) of alive or dead cells, delay between two ticks in milliseconds, size of each cell in pixels, width and height of the grid.
  - The color of an alive cell should be different based on its age (number of ticks it has been alive). Add parameters to define the color of alive cells at the lowest and highest age (the color would not change after the highest age). Think how to calculate colors that smoothly change between these two colors.
  - The color of a dead cell, which was once alive, should be slightly different from the default color for a dead cell.
  - Make your program open the GIF file it has generated with default application.
  - Create a short online video explaining your program and demonstrating how it works. Publish your video online as a public post on Facebook or/and any other social media with hash-tags including #ADAUniversity, #CSCI2408 and #GameOfLife. Add the URL of the post to your program code documentation.

- **GOOD LUCK!**

- **Plagiarism** is using others' ideas and words without clearly acknowledging the source of that information. To help you recognize what plagiarism looks like and what strategies you can use to avoid it, see Indiana University's *Plagiarism: What It Is and How To Recognize and Avoid It* at http://www.indiana.edu/~wts/pamphlets/plagiarism.shtml

[2] Read about it here: http://www.fileformat.info/format/gif/egff.htm Search the internet for more detailed file specification, if needed.

- **Academic misconduct will not be tolerated** and will be referred to the ADA Honor Council. Penalties for misconduct will be a zero for this assignment, a fail grade in the course, and/or other disciplinary action that may be applied by the ADA Honor Council.