

# ITSC202 - Project 13-01

## The Game of Life

[https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

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 *neighbours*, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. Any live cell with fewer than two live neighbours dies, as if caused by under-population.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by over-population.
4. Any dead cell with exactly three live neighbours 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.

Assignment specification (please read carefully, un-implemented parts will cost points)

The game will consist of at least two source files, the main program, and the drawing backend.

The main program shall be named `liferun.c`

The game has what we call state – the life as it exists until next generation is calculated.

The state shall be implemented as two-dimensional array of short. The program shall use bit 15 of the state array to save future generation as it is being calculated.

Bit 14 shall be used for the current state of the game as it is displayed.

Bits 13 down to 0 shall save 14 previous generations of the game, bit 0 being the oldest.

The three functions below must be implemented in separate file `lifedrawXXX.c`, to facilitate different methods of displaying the game (i.e., being able to link different files defining these functions with the main program, without having to change the main program). XXX stands for the implementation method of the drawing method – one of `stdio`, `curses`, `png`, `opengl`.

**`int draw_init (int size_rows, int size_cols)`**

returns -1 if it can not initialize

**`int draw_life (int rows, int cols, short *state)`**

returns negative number if the drawing failed for some reason

**`void draw_finish()`**

always successful

Minimum size of the grid is 80x24.

The program may get some parameters on the command line. Option **- - random** means the seed should be randomly generated by reading /dev/urandom. A filename on the command line specifies a file from which the program will read the seed, one cell per line, row coordinate is first, e.g.:

```
12 23
3 45
```

...

Without any parameters, the program shall use hard-coded, predefined seed of your choice.

Organize your program well, define functions for tasks where appropriate and comment it well. Format and indent your program properly, curly brackets shall match in the same column, and use their own line.

The prettiest seed will get extra 5 bonus points.

The main program implementation is worth 60 points.

The drawing library (the file lifedraw.c) can be implemented in following ways:

using stdout, worth 20 points

using ncurses is worth 30 points

producing a series of png images is worth 40 points

using OpenGL is worth 60 points.

Making previous generations progressively fade out is worth 20 points.

Mandatory 15-minute presentation on how you have developed your code, and how it works, is worth 20 points. Without the presentation, your mark is limited to total of 80 points.

Some common ways to lose points:

Spaghetti code -10 points

Code without comments -10 points

Code not indented (M-x mark-whole-buffer, M-x indent-region) -10 points.

**Inability to describe and explain the code you developed in detail, and to describe the code at a level which demonstrates good understanding, deletes your mark for that code.**

You can write several versions of the lifedraw.c file, the points awarded are cumulative up to 150/100. Implementation of the main program is mandatory, i.e., the 60 points can not be earned by implementing all the drawing libraries.

The grading of this assignment has twice the weight of regular weekly assignments.

If you re-use any code from external sources, you must give credit for that code, and it will not be part of your mark. You still need to understand and be able to describe the functionality of that code in detail.

