

# Conway's game of life

Author: Michał Bronowski

## I) Goal:

Text-interface simulation of evolution determined by the initial state, requiring no further inputs (zero-player game).

## II) Content:

Project is containing 3 files: **code**, **binary file**(executable file), and **text file** named „**tab.txt**” which store the initial state. The board size is set to **10x10** cells, and the only way to change it is to change values of **int size = 10;** and the size of the arrays **tab[ ][ ]**, and **tab2[ ][ ]** in the code block.

Program is consist of interface (**BuildTab()**; function and **Main()**;) and other useful functions.

## II) Interface:

- 1) **Main()**; - At the beginning the board is fetched from the text-file, screen is cleaned, and the cursor is at the top-left corner. Then program in „**do while**” loop generate current state (look at the **BuildTab()**; function) and the possible actions. The process is repeated as long as the user wants to exit.
- 2) **BuildTab()**; - Function is printing the array **tab[ ][ ]** on the screen as a board. It doesn't change any value, but only read it. In the first row and column there are numbers of rows and column. Then the space is divided into **size x size** cells separated in a row by „|” sign (**ASCII: 0124**), and in a column by „\_” sign (**ASCII: 0095**). „0” are not displayed at the output board and „1” are red (not every operating system).

## IV) Other functions:

-) In the programs menu, we have following options:

- 1) Flip values of the cells (**SetCel()**;) )
- 2) Start (**Graj()**;) )
- 3) Reset (**Reset()**;) )
- 4) Set default values (**SetDef()**;) )
- 5) Set random values (**SetRan()**;) )
- 6) Exit

-) Description of each function:

- 1) Use **SetCel()**; function. User puts in which row (**a**) and column (**b**), he wants to change the value of the cell. Then program substitute those values to **tab[a][b]** and flip content of the cell (possible only 1 and 0. If **tab[a][b]** was equal 0, now it will be equal to 1). To go back to the main window, user enters 0, when program asks him if he wants to continue.
- 2) Use **Graj()**; function. In the beginning of the game, program demand the number of generations, we want to see. Given value is the number of iteration of the main loop. It analysis how many of live neighbor has the cell, and depends on that changes the value and copy it to the **tab2[ ][ ]**. The rules are exactly the same as

the Conways. After 1 repeat of the main function (each cell was analysed), program copy **tab2[ ][ ]** to **tab[ ][ ]** and print the board on the screen and informing us which generation it is. The program is set to wait 1 second after each iteration step, to show the changes.

- 3) Use **Reset();** function. It set every value of the **tab[ ][ ]** to 0.
- 4) Use **SetDef();** function. It opens previously defined textfile with reading attributes. Then using **fscanf();** copy array included in the file to **tab[ ][ ]**. At the end of the function, program close the file.
- 5) Use **SetRan();** function. User is asked how many percent of the board he wants to fill. For each cell, program generate random number (based on **srand();** which depends from the time fetched from system. If the random number is less than the given value (percentage of the board, which should be filled), the cell is set to 1. Otherwise it's 0.
- 6) Exit. It cleans the screen, set the cursor in the top-left corner of the screen, and finish the program.