

Karlova Univerzita
Matematicko-fyzikální fakulta
1. ročník, Informatika

Započtový program Pro Informatiky

«Game of life»

By Salohub Illia

Table of contents

Content

Task description	3
Project technologies.....	4
Main game classes	5
1.Menu Class.....	5
1.1 Technical Part.....	5
1.2 User part	5
2. Game Class.....	6
2.1 Technical Part.....	6
2.2 User Part	7
3. Settings Class.....	7
3.1 Technical Part.....	7
3.2 User Part	8
4. Rules Class.....	8
4.1 Technical Part.....	8
4.2 User Part	9
Test input for Game of life.....	10
Conclusions about the created game.....	11

Task description

Using the PyGame and Tkinter libraries in Python to make a simple cellular automaton, the main requirements is:

1. Make a game window with simple user input
2. Create an algorithm that will inherit game logic
3. Make an ability to customize the program (Game speed, window resolution, etc.)
4. Create window with game rules
5. Modernize game with some own rules (Loop field, age of living cells, etc.)

Project technologies

For creating simple Game of life using python 10.0, Tkinter 8.6 and Pygame libraries were chosen.

PyGame is the most popular game library that provides easy frame system for creating games on level of 3D games, third person 2D's, etc.

In Tkinter library there are a simple way to create windows with Labels, Scales, Buttons, etc.

Main game classes

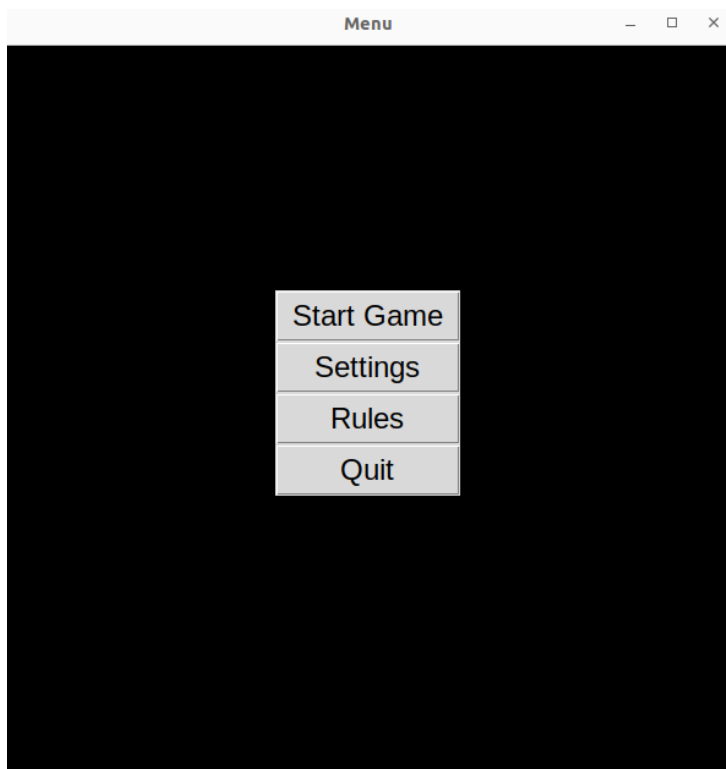
1.Menu Class

1.1 Technical Part

Code of class Menu you can find [here](#).

This class has a simple program navigation. Window field and navigation buttons are created in the beginning in function startMenu(). Every button has a onClick() action that will cause redirect to other windows and will start their classes.

1.2 User part



In this navigation window user can press buttons Game, Settings and Rules to redirect to related windows. Also, there is button Quit that will cause quit from the program.

2. Game Class

2.1 Technical Part

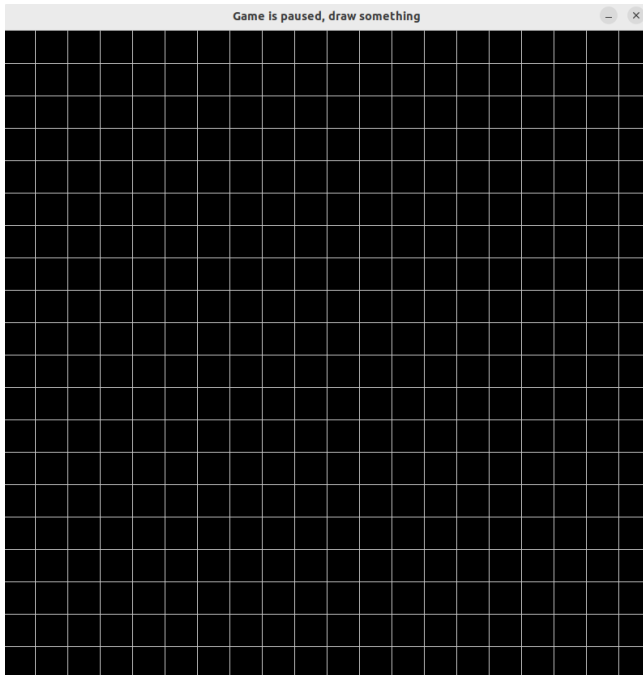
Code of class Game you can find [here](#).

This class has a simple PyGame field creating, user input at the beginning of program and then, after start of the game, program is running using own Game of life algorithm, that is in function algorithm ().

When this algorithm is called, at the beginning he copies a field cells array. After that the dynamic updating is running. For every cell program looks at 8 neighbors and counts number of living cells. With the basic rules, if cell is dead and have 3 live neighbors, in next generation it will be alive. After the algorithm has completed, game field will be updated. Efficiency of created algorithm is $O(N^2)$.

Field update rate depends on game speed, that user can set in Settings window.

2.2 User Part



When user opens this window, game will be paused. User can make any cell alive just by clicking on it. After pressing the Space button game will be started and it will be updated with rate that

user can set in Settings window. Now user can do anything with cells on this loop board. By clicking the Escape button user can back to main menu.

3. Settings Class

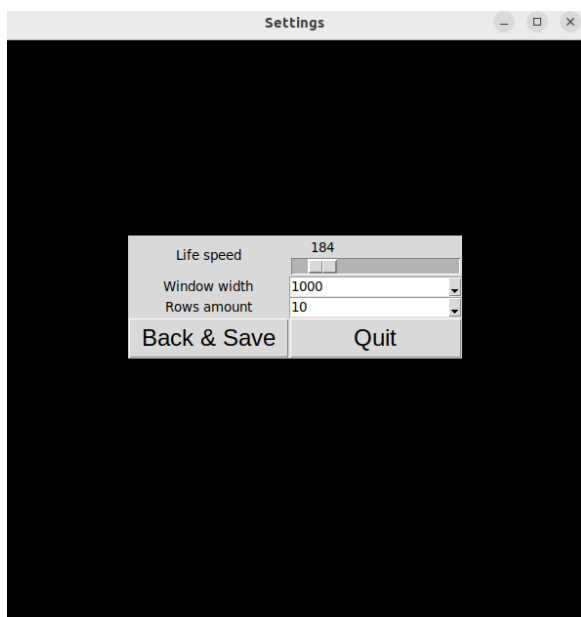
3.1 Technical Part

Code of class Settings you can find [here](#).

This class has a simple game customization. Window field and settings elements are created in the beginning in function `windowSettings()`. After editing of every setting new data will be saved to

settings.json file(Game class takes data from this file after starting the game). Function `valueChanged()` change the possible settings of Rows based on chosen Resolution value.

3.2 User Part



On this window user can change game window settings – set the resolution, number of rows and columns and change the life cycle speed. Changes will be saved after pressing Back & Save button.

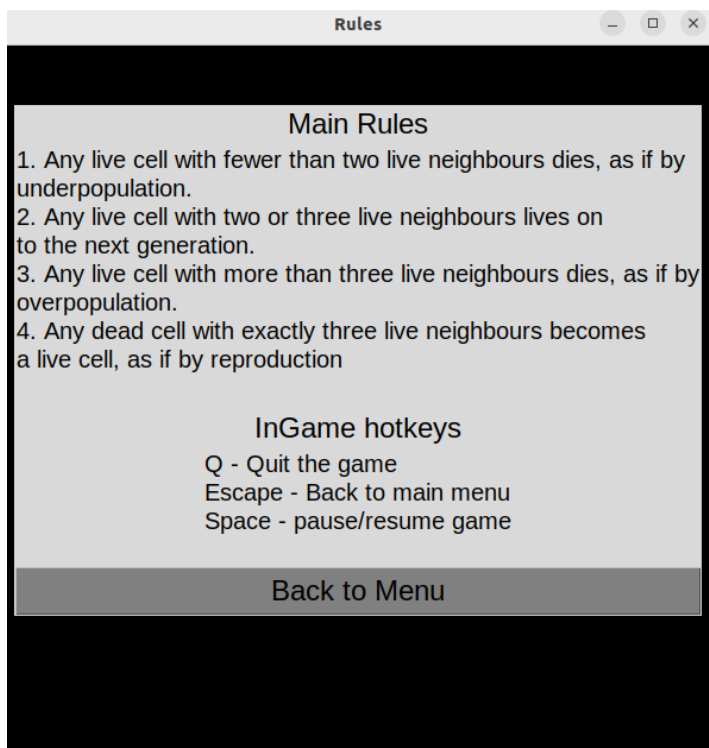
4. Rules Class

4.1 Technical Part

Code of class Settings you can find [here](#).

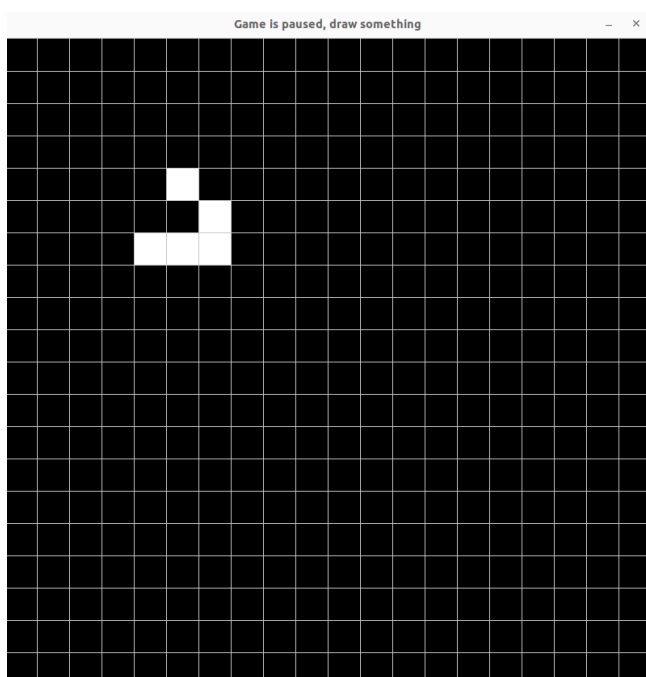
In this class there is simple tkinter window with a couple of labels with game rules and button with action Back to Menu.

4.2 User Part

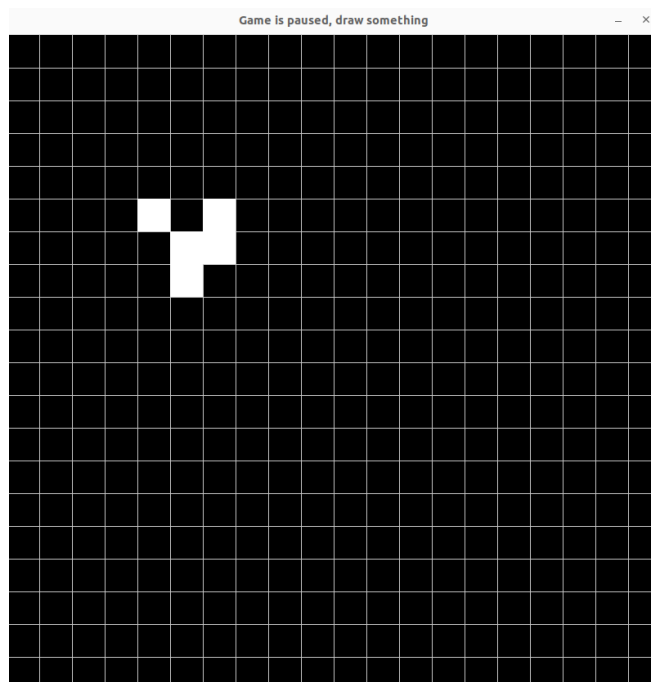


On this window user can read basic rules about game of life and read about simple hotkeys in game window.

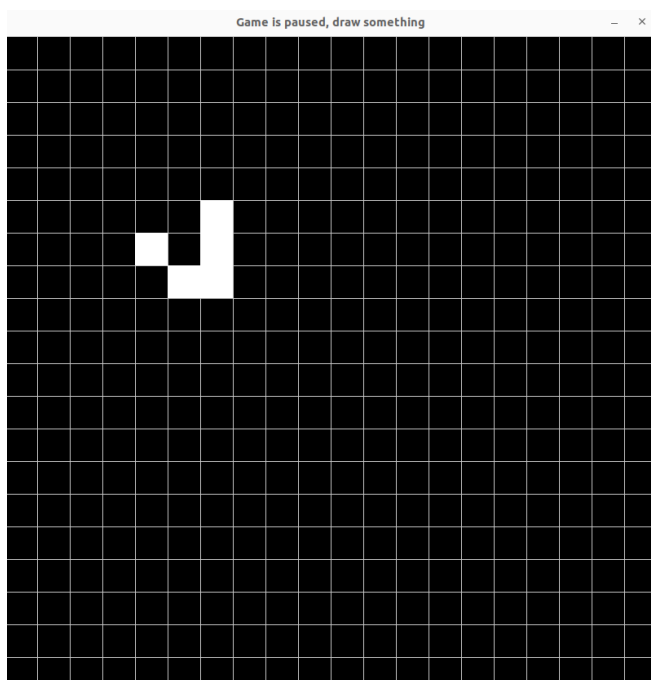
Test input for Game of life



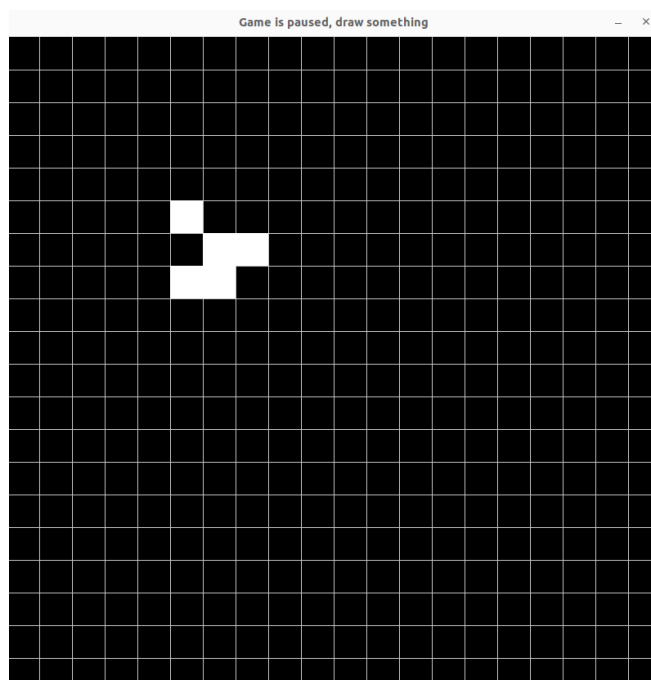
Input data



Generation 1



Generation 2



Generation 3

Conclusions about the created game

In the course of this work, the following tasks was performed:

- 1) Was explored methods of working with windows and frames in Python using Pygame and Tkinter
- 2) Game of life was made with an infinite field add-on
- 3) The game has been tested