

Game of Life

By: Mehak Ahuja and Sriya Akula

Description

- This program creates the Game of Life by Conway and implements the MVC model. It contains 4 classes:
 - LifeView
 - LifeModel
 - LifeFrame
 - LifeController
- Our program also includes user input, and the user can choose the speed of the program and the number of generations.
- It can also find different patterns while the program is running and count the number of patterns that show up

Model Class

- The model class deals with the rules of the game, and ensures that the life status of both the cells and its neighbors. It's main functions are:
 - `updateLife()`: This function identifies if a cell should be dead or alive based on the number of neighbors around it.
 - `countAliveNeighbours(int x, int y)`: This method checks on the neighbors of the cells wherever they are on the board to see if they are alive.
 - `matchesPattern(int[][] pattern)`: Goes through the grid while it's running and checks to see if patterns are found. If they are found, they are counted.

View Class

The view class is the UI part of the game. It contains the color and structure of the game. It has 3 components -

The background color, the color structure of the grids, the color and structure of the cells

The background color that we chose was pink while the grid lines and cells were black. It gives a very aesthetic view to the game.

The methods in the view class are - `paintComponent`, `drawGrid`, and `drawCells`

Controller Class

The controller class is a link between the model and view class. It connects both the classes for proper functioning of the program.

It contains the code for telling the user the current generation that the game is on. It also implements the `Thread.sleep();` function and it is set to 300. This means that before each generation there will be a pause for 300 milliseconds.

The controller class also contains the method to identify patterns in the game. We implemented the game to recognize four patterns which are - glider, tumbler, spaceship, and 10 cell row pattern.