Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A21

Computer Science Challenge

Lab Professor / Lab Session:

[Paulo Sousa]

Team:

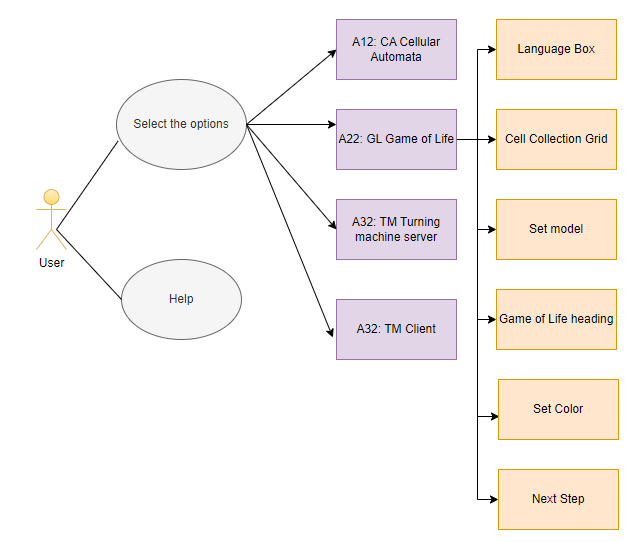
[Meet Kotadiya] - Id: [041070018] / [Dhanush Bajaj] - Id: [041067058]

CS Challenge 2: Game of Life

|  |  |
| --- | --- |
| **Part**  **1** | **Implementing GL** |

* 1. **Example UC Model (2pt)**

**UC Diagram**



**Actors table** :

|  |  |
| --- | --- |
| **Actors** |  |
| **User** | The person who interacts with the application to set the initial conditions, start the simulation, stop the simulation, adjust the speed, and select the cellular automaton type. |

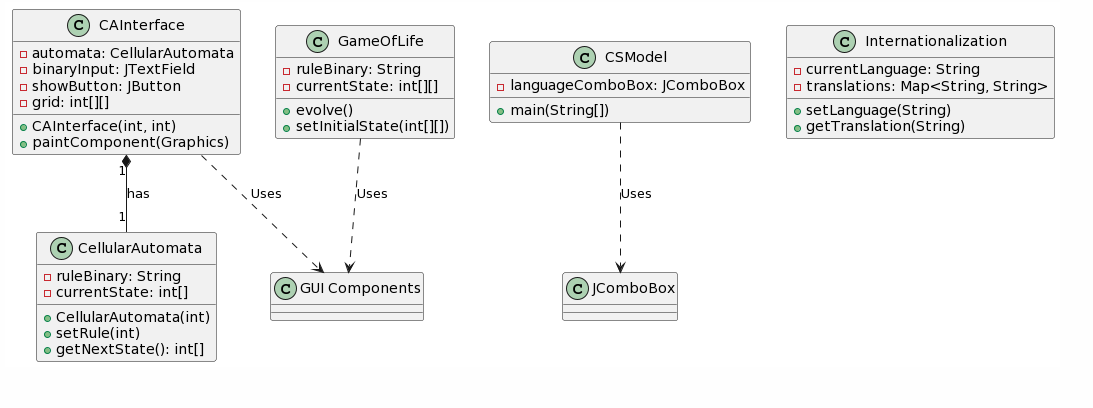
**UC table**:

|  |  |
| --- | --- |
| **Use Cases** |  |
| **Help** | The 'Help' button provides assistance to the user by offering detailed explanations about the functionalities of the application. If a user is unsure about how to operate the application or what a certain feature does, pressing 'Help' will provide the necessary guidance. |
| **Select an Option** | This is the step where the user is given the opportunity to choose from a list of four available games. The list of options includes 'CA Cellular Automata', 'GL Game of Life', 'TM Turing Machine Server', and 'TM Client'. |
| **Option 1: A12: CA Cellular Automata** | Choosing this option will start the Cellular Automata game. Cellular Automata is a simulation game involving cells that evolve over time according to a set of predefined rules. |
| **Option 2: A22: GL Game of Life** | Selecting this option will start the Game of Life. The Game of Life is a cellular automaton devised by mathematician John Conway. It is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input from humans. |
| **Option 3: A32: TM Turning machine server** | Choosing this option will start the Turing Machine server. A Turing Machine is a mathematical model of computation that defines an abstract machine that manipulates symbols on a strip of tape according to a table of rules. |
| **Option 4 : A32: TM Client** | Selecting this option will start the Turing Machine client. This option allows the user to interact with the Turing Machine server, sending inputs and receiving outputs. |
|  |  |
|  |  |

**Class Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name** | **Attributes** | **Methods** | **Relationships** |
| CAInterface | automata: CellularAutomata binaryInput: JTextField showButton: JButton grid: int[][] | CAInterface(int, int) paintComponent(Graphics) | Has-a relationship with CellularAutomata |
| CellularAutomata | ruleBinary: String currentState: int[] | CellularAutomata(int) setRule(int) getNextState(): int[] | None specified |
| CSModel | languageComboBox: JComboBox | main(String[]) | Uses relationship with JComboBox |
| GameOfLife | ruleBinary: String currentState: int[][] | evolve() setInitialState(int[][]) | Uses relationship with GUI Components |
| Internationalization | currentLanguage: String translations: Map<String, String> | setLanguage(String) getTranslation(String) | Used-by relationship with main classes |

**Details**



* 1. **Visual Components (1pt)**

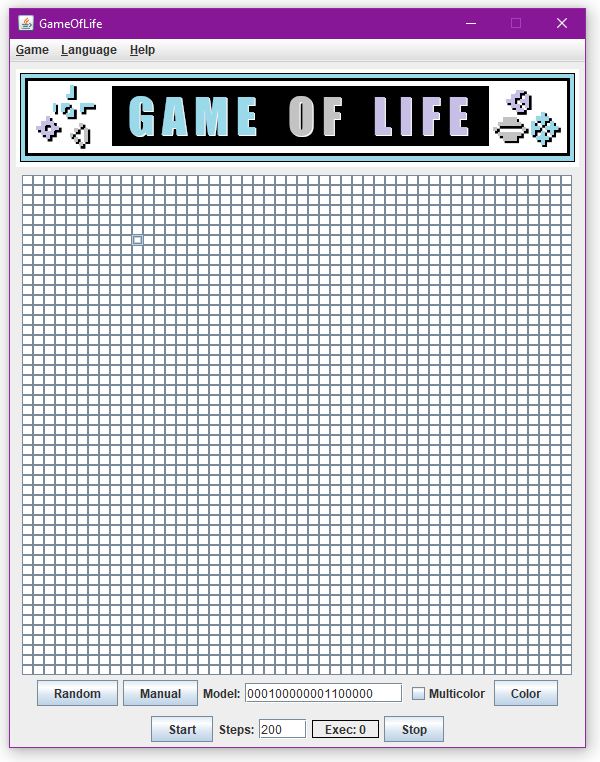
**GL Implementation**

***1. GL Window:***

*The main interface for the Game of Life implementation.*

*Properties:*

* ***Title****: A title bar, potentially showing an image or icon representing the game.*
* ***Menus****: Drop-down or side menus to select various options. This includes:*
  + ***Internationalization Menu****: Options to change the language of the game interface. At a minimum, this should provide options for English and at least one other language.*
  + ***About Menu****: A menu or section that provides basic information about the game, its rules, and potentially its history or background.*
* ***GL Grid Area****: A visual representation of the Game of Life grid, where cells can be alive or dead and will evolve over time based on the game's rules.*
* ***Input Area****: An input section where users can provide an initial configuration for the Game of Life grid. This configuration is given as an 18-bit binary sequence.*
* ***Color Options****: Options for users to select and apply colors to the Game of Life grid. This can include buttons for predefined colors and options for users to select custom colors.*
* ***Execution Options****: Controls for users to start, stop, or step through the evolution of the Game of Life grid. This can include:*
  + ***Start Button****: Initiates the evolution of the grid based on the game's rules.*
  + ***Stop Button****: Pauses the evolution of the grid.*
  + ***Step Button****: Advances the grid by one step or generation.*



**FINAL SUGGESTIONS**

* *Intuitive Design: Strive for a user-friendly and intuitive interface. The controls should be self-explanatory, and the user should be able to understand the game's state at a glance.*
* *Feedback: Provide feedback to the user. For instance, when they change the simulation speed or pause the game, there should be clear indications on the interface reflecting these changes.*
* *Documentation: Provide tooltips or a help section in multiple languages to guide users on how to interact with the application..*

**References**

1. Wolfram, S. (2002). A new kind of science. Wolfram media.
2. Gardner, M. (1970). The fantastic combinations of John Conway's new solitaire game "life". Scientific American, 223(4), 120-123.
3. ChatGPT

|  |  |
| --- | --- |
|  | * ***NOTE****: Even if you use any AI tool (ex: ChatGPT), report here, including the references used.* |

Algonquin College

Fall, 2023