Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A21

Game MVC

Team:

[Akshit Sabharwal] - Id: [041025056] / [Erneste Iradukunda Senga] - Id: [040973461]

NumPuz Proposal

***This template is suggested (not mandatory) to answer A21 Specification.***

|  |  |
| --- | --- |
| **Part**  **1** | **GUI Definition** |

* 1. **MVC Details**

*Describe the way you can define the MVC components in your game.*

|  |  |  |
| --- | --- | --- |
| **Class** | **Object** | **function** |
| GameController | gameController | Responsible for logic or actions of the game play |
| GameView | displayLayout | Responsible to display or define the view of the game |
| GameModel | gameModel | Responsible for maintaining data of the game like time or points. This will act as a super class, it could be extended by other sub-classes based on features/data we need to implement |
| PuzzleGrid | drawPuzzle | Responsible for drawing our playable puzzle based on dimensions. |

* 1. **View Component**

*Describe how your interface should be organized using new components. Show the idea about your “top-down” organization.*

|  |  |  |
| --- | --- | --- |
| **Class** | **Object** | **function** |
| JFrame | *outerFrame* | Displays the outer frame of the application |
| JFrame | *popUpFrame* | *Displays the pop window for game finished game* |
| JPanel | puzzlePanel | *Displays game grid* |
| JPanel | buttonPanel | *Displays the game buttons* |
| JButton | Design | *Design mode* |
| JButton | Play | *Play mode* |
| JButton | Show | *Show solution button* |
| JButton | Hide | *Shuffles the grid* |
| JButton | Save | *Save the game play* |
| JButton | Load | *Shuffle the solution* |
| JButton | Rand | *Shuffle the grid* |
| JButton | Finish | *Finish the game* |
| JButton | Restart | *Reset the game and shuffle a new game* |
| JLabel | Mode | *Displays the label* |
| JLabel | Dim | *Displays the label* |
| JLabel | Type | *Displays the label* |
| JLabel | Moves | *Displays the label* |
| JLabel | Points | *Displays the label* |
| JLabel | Time | *Displays the label* |
| JTextField | *Tf\_text* | *Text field for writing the text to shuffle* |
| JTextField | tf\_moves | *Text field for displaying moves made* |
| JTextField | tf\_points | *Text field for displaying points* |
| JTextField | tf\_timer | *Text field for displaying the points* |
| JTextField | tf\_4 | *Text field* |
| JComboBox | comboBoxDim | *Dropdown box for dimensions* |
| JComboBox | comboBoxType | *Dropdown box for type - letters or numbers* |
| JWindow | splashScreen | *Displays the screen before the game starts* |
| JMenu | Menu | *Drop down menu* |
| JMenu | Color | *Allow to change the configurations of game* |
| JMenuItem | About | *Diplays a new window* |
| JMenuItem | Solution | *Show the solution* |
| JMenuItem | Exit | *Exit the game* |

* 1. **Controller Component**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object** | **Event** | **Method** | **Function** |
| ComboBoxType | addActionListener | actionPerfomed | *Shifts the type of puzzle whether numbers or letters* |
| Design | addActionListener | actionPerfomed | *Mode in which the player gets to adjust features of the game* |
| Play | addActionListener | actionPerfomed | *Mode for playing* |
| Show | addActionListener | actionPerfomed | *Show the solution* |
| Hide | addActionListener | actionPerfomed | *Shuffles the solution* |
| Save | addActionListener | actionPerfomed | *Save the game in a file* |
| Load | addActionListener | actionPerfomed | *Loade saved game* |
| Rand | addActionListener | actionPerfomed | *Shuffle the grid of the game* |
| Finish | addActionListener | actionPerfomed | *Finish the game play* |
| Restart | addActionListener | actionPerfomed | *Adjust or reset the feature of the game* *to start a new game* |
| ComboBoxDim | addActionListener | actionPerfomed | *Displays the dimensions of the game* |

*Describe aspects of your controller using, for example, one unique action command. Create the “map” to define functions with actions.*

* 1. **Model Component**

*Finally, what is your idea to define the model to be used in a “default” (randomized) game.*

|  |  |  |
| --- | --- | --- |
| **Object** | ***Value*** | **function** |
| drawPuzzle | 3X3 dim | paintComponent(), drawElement for drawing default 3X3 dimension numpuzzle |
| displayLayout | *Default layout* | configure(), redefineFrame() |
| gameModel | *Timer, Points, solution* | *getSolution, setTimer, setpoint, getPoints, getTimer* |
| gameController | *Controls Game Action* | start(), restart(), solution(), splashScreen(), save(), load() |

|  |  |
| --- | --- |
| **Part**  **2** | **Implementation Design** |

* 1. **Game Evolution**

The main layout remains the same as defined in A12.

*Added Features to Game*

-For Menu Bar: Added Color option, that will allow user to change the color for puzzle.

- Added Splash Screen before loading our original game.

- Added Points and Timer for the game

- Improved Code Strategy for implementing game.

Reason for adding changes:

1. To make better user experience like points or timer would make game more interesting.
2. Adding Color Configuration would allow users to play game in their preferred color.
3. Added Splash Screen, to display game logo before loading the main layout of our game.
   1. **Others DP**
4. Facade Pattern: This DP lets you create a simple class that can provide simple interface to user for handling background complex functionalities/actions which are not important for client or user. As GameController class in our game can be such a class which will have access to other classes and perform certain action based on user’s selection. [1]
5. Command Pattern: It is also known as action or transaction; it passes the command to invoker object, which in return looks for appropriate object which can handle that command or action and then executes that action. As we have several buttons like Save, Load, etc to perform action so this DP can also be implemented for our game. [2]

**References**

|  |  |
| --- | --- |
| [1] | "Facade," Refactoring Guru, [Online]. Available: https://refactoring.guru/design-patterns/facade. [Accessed 14 October 2022]. |
| [2] | "Command Pattern," Java T point, [Online]. Available: https://www.javatpoint.com/command-pattern. [Accessed 14 October 2022]. |

Algonquin College

Spring / Summer, 2022