*Team: [Aliab Eman, 041-000-420]&[Matt Vecchio, 041-004-137]*

*Algonquin College | Computer Engineering Technology –Computing Science*

*A11 Game interface*

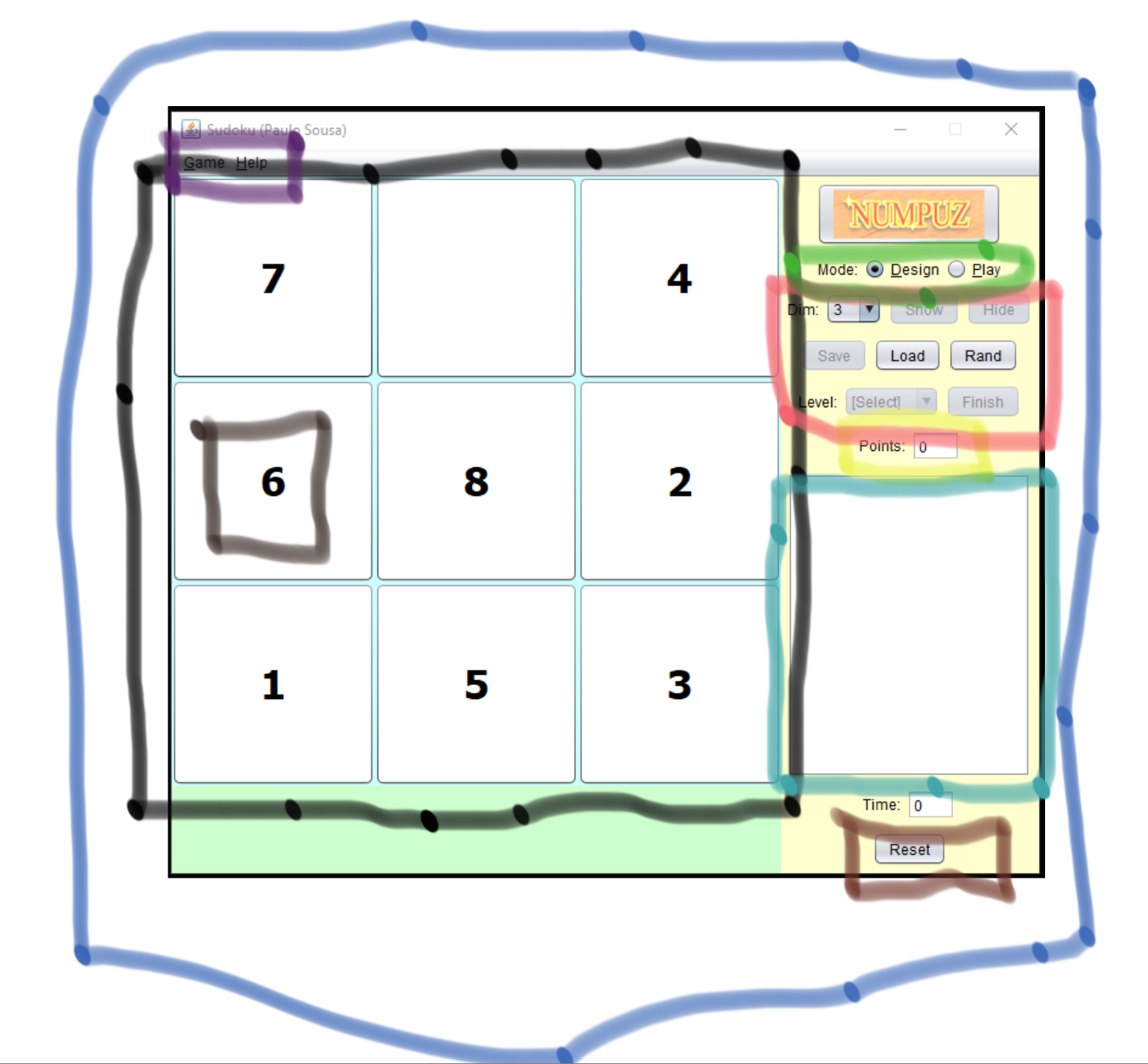
*CST8221-302*

*Presented to: Professor Paulo Sousa*

**Explanation**

Assignment 11 will serve as the template necessary for me to display to the user how to approach the creation of my own style of the game “NumPuz”. My understanding of the game is that we need to be able to arrange a disordered set of consecutive numbered elements from the smallest to largest number based on the dimensions of the game we allow the user to play. My plan will be to make a competitive play out of this experience for the players.

**1.1. Defining the Components**



**Figure 1. Dissecting Professor’s NumPuz Design into potential components**

List of Components

JFrame:

Highlighted in blue, I will have a JFrame instance to serve as the window frame that holds all the components to my NumPuz game. I want to either have a default size of 400x400px for the main JFrame or an adjustable frame, it will probably be easier to keep a default size instead of adjusting the frame because that would impact the components inside the JFrame. The close operation of this JFrame should stop the game upon clicking the exit button. My concern with this frame is if I will use the same frame continuously between different moments of the NumPuz API, or if I plan to have other JFrame instances pop-up according to the game mode that gets selected.

JLabel:

Highlighted in purple, I plan to have a JLabel in order to allow the player to make modifications to their game.

-JLabel can include the “Game” and “Help” menu buttons.

Inside the “Game” button, list of options:

-save game

-exit a game

-allow to switch game modes (from Design to Play mode, vice versa)

- load a game

Inside the “Help” button:

-We could have an instruction manual.

JPanel: Inside the JFrame should contain one or many JPanels with different types of layouts and components inside of them.

-I want my NumPuz game, the Sudoku board itself, to have its own panel either as a GridPane via JavaFX or a BoxLayout via Swing. The reason for a grid-layout is to simplify the arrangement of the elements (alphanumeric squares, or null square) on the game board

-I also want to have a separate JPanel for the attributes of the game. This container should be able to influence the NumPuz game by hosting buttons/ components for saving/ loading the game state, editing the design for the game, drop-down menus for levels, etc.

JTextArea:

-Can be used to keep the scores of previous players.

-Can be used for keeping the score of the current game

-Can be used to keep track of the time spent on a game session

JTextField:

-The text field will be needed for usernames for save states of the NumPuz game.

JComboBox:

-This component could be useful for selecting the dimensions for the NumPuz game. It could allow for the user to select the level of difficulty for the game to play.

JButton:

-Can be used on the JPanel outside of the game’s board. The JButtons should trigger actions that change the game:

-a reset button to reset the game board (either shuffle or restart the current game session)

-A load and a save button.

JRadioButton:

-Can be used to switch between the Design mode and the Play mode of the NumPuz game.

**1.2. User Manual**

Move-By-Click:

1. A sequence (or a “play”) is considered to have happened if the position of the null square in the game is changed
2. An alphanumeric square can only switch positions with the null square
3. An alphanumeric square cannot switch positions with other alphanumeric squares
4. Alphanumeric squares need to be to the north, south, east or west of the null square in order to be able to complete a sequence
5. To complete a sequence, click on any alphanumeric square proposed by rule 4
6. When a successful sequence is made, the position of the alphanumeric square, and the null square, are switched
   1. If the alphanumeric square is at position ‘a’ and the null square is at position ‘b’ before the sequence, after the sequence you will find the alphanumeric square to be in position ‘b’ and the null square to be in position ‘a’
7. Any diagonally positioned alphanumeric square (north-east, north-west, south-east, south-west) relative to the null square cannot complete a sequence with the null square
8. If an alphanumeric square is not part of the sequence, its position remains the same after the end of a sequence
9. A game is considered completed once:
   1. The alphanumeric squares are ordered from the top left of the game’s grid to the bottom right of the game’s grid
   2. The null square is at the position of the last element on the grid (right + bottom of the game’s grid)

References:

1. JavaFX Layout Panes. <https://www.tutorialspoint.com/javafx/javafx_layout_panes.htm>