The 8 Puzzle report

DAVIDE ORSUCCI

January 30, 2025

The project has been structured in three files:

EightTile.java : EightTile is a java Bean which extends the JButton. The EightTile is made up of two properties: *label* and *position*. At the beginning of each game the tiles are reinitialized using the *restart* method which initializes the tiles using an array containing the permutation of the labels.

EightController.java : EightController is a java Bean that extends JLabel. It implements the logic of the game by saving only the position of the hole in the private property *holePosition*. EightController does the following actions:

- it vetoes the change if the tile is the hole or it is not adjacent to the hole, displaying “KO”. Otherwise it shows ”OK”. this is done using the functions *vetoableChange* and *isAdjent* which uses the position (not the label) of the tiles to check if the user's move is allowed. If the veto is positive, the action is performed by the tiles and the EightController updates the *holePosition* property.
- it is registered as an action listener of the flip button, so it can perform the flip of the first two tiles in certain conditions (see later).
- it is registered as an action listener of the restart button, so when the game is restarted, the *holePosition* is reinitialized and the EightController displays ”START” (as the beginning).This is done using the new hole position using the method *restart*.

EightBoard.java extends JFrame, it's the entry point of the program and creates the GUI using and manages the tiles and the buttons as requested.

1 The Flip Button

The flip button is created in the function *initComponents()* of EightBoard.java.

```java
flip = new javax.swing.JButton();
```

After the button is created by the call of *initComponents()* by the constructor *EightBoard()* the following fragment of code is executed

```
54 flip.setActionCommand("flip");
55 flip.putClientProperty("eightTile1", eightTile1);
56 flip.putClientProperty("eightTile2", eightTile2);
57 flip.addActionListener(eightController1);
```

line 54 : sets the action command of the action associated with the ”flip” component to the flip button.

line 55 and 56 : add the two client properties eightTile1 and eightTile2 to the flip component, each one with its own key so the 2 properties can be retrieved.

line 57 : Adds an action listener to the flip component.

The EightController class implements the ActionListener interface so it implements the *actionPerformed* method that will be called when the action occurs.

```
80 @Override
81 public void actionPerformed(ActionEvent ae) {
82     JButton button = (JButton) ae.getSource();
83     ...
84     if(button.getActionCommand().equals("flip")){
85         if(holePosition == 9){
86             EightTile t1 = (EightTile) button.
                   getClientProperty("eightTile1");
87             EightTile t2 = (EightTile) button.
                   getClientProperty("eightTile2");
88             String tempText = t1.getText();
89             t1.setText(t2.getText());
90             t2.setText(tempText);
91         }
92     }
93 }
```

line 82 : gets the source of the action performed

line 84 : checks if the action command of the button is ”flip”

line 85 : if yes, the EightController checks if the *holePosition* is 9

line 86-90 if *holePosition* is 9, the swap between the two EighTiles’s labels is performed, otherwise no.