# AP Assignment - 1.1

Simone Passèra

Jul 01, 2024

## 1 Introduction

This programming assignment consisted of the implementation of the reduced version of the famous 15 Puzzle game, using Java Beans.

The system that I have implemented consists of the three Beans required by the specification (EightTile, EightController, EightBoard).

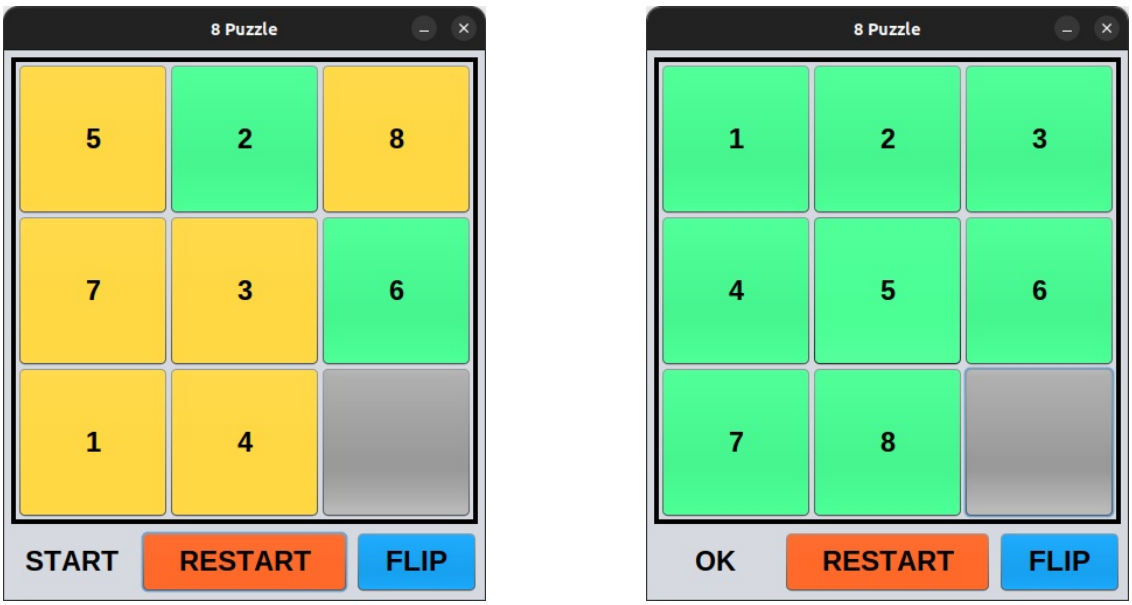


**Figure 1:** The 8 Puzzle game. `Left`: Random initial configuration. `Right`: Final configuration.

## 2 Code Implementation

### 2.1 EightTile Bean

The `EightTile` class extends JButton and implements the Serializable and PropertyChangeListener interfaces. It is designed to represent a tile in the 8 Puzzle game.

When a tile is clicked, the public *setLabelTile(9)* method is called, and fires a vetoable change event ("setLabelTile") to EightController, which will accept or reject the change. If the change is invalid, the tile flashes red for half a second, otherwise set the label to 9 (the hole).

The *propertyChange()* method handles various events:

- `"restartBoard"`: when the restart button is clicked, the EightBoard send this event to all tiles. It resets the tile's label and update the tile's appearance.

- `"holeMoved"`: when a tile becomes the new hole, its label is sent via this event from the EightController to all tiles. Only the tile representing the current hole will update its label.

- `"flip"`: when the flip button is clicked, the EightBoard send this event to the tile in position 1. The tile in position 1 implements the exchange with the tile in position 2 in this way. So, when the flip event is fired, the tile in position 1 sends its label via the "setLabel" event to the tile in position 2 (at this point the "flip" event is over). The tile in position 2 before updating its label, sends it to the tile in position 1 by generating the "setLabel" event.

- `"setLabel"`: this event is handled by the tiles in position 1 and 2 as a consequence of the "flip" event. Simply update the label with the label passed within the event. If the event is executed by the tile in position 2 then before updating its label it fires another "setLabel" event directed toward the tile in position 1.

## 2.2 EightController Bean

The `EightController` class extends JLabel and implements Serializable, VetoableChangeListener and PropertyChangeListener interfaces. It has to check that only legal moves are performed. The EightController to know the current configuration of the board, keeps in the private variable `holePosition`, the position of the tile representing the hole.

The *propertyChange()* method handle the "restartBoard" event. When the EightBoard fired the event, the EightContoller updates *holePosition* to the default value (9) and displays `START`.

The *vetoableChange()* method handles two events:

- `"setLabelTile"`: when a tile is clicked, it fires this event to the EightController, passing the position, the old and new label of the tile (9, the hole). The EightController check if the move is legal and if not, throws the *PropertyVetoException* exception and displays `KO`.

- `"flipRequest"`: The EightController simply checks whether the flip request can be executed, based on the position of the hole and if not, throws the *PropertyVetoException* exception, otherwise displays `FLIP`. The event is fired by the EightBoard as a result of clicking the flip button.

## 2.3 EightBoard Bean

The `EightBoard` class extends JFrame and implements Serializable interface. It is the main class of the application. The dashboard displays: the tiles inside a JPanel for look and feel only, the EightController and the two JButton, `restart` and `flip`.

In the constructor of the class, the links between the various beans are actually created, adding listeners for each instance of the beans. The creation of a PropertyChangeSupport (or VetoableChangeSupport) was avoided if the object offered native support for events.

This is the action performed when the following buttons are clicked:

- `RESTART` button: executes the private method *restartBoard()*, in which the new configuration is sent through the firing of the "restartBoard" event. I decided that in each new configuration the hole tile is in position 9.

- `FLIP` button: fires the "flipRequest" VetoableChange event to the EightController. If the controller does not veto the change, then it fires the "flip" event at the tile in position 1, which will swap its label with the tile in position 2 as mentioned above.