



UNIVERSITÀ DEGLI STUDI DI PISA

ADVANCED PROGRAMMING

A.A. 2023/2024

EightPuzzle Report

Author:

Francesco Kotopulos De Angelis

GitHub
Repository

October 9, 2024

1 Introduction

The assignment involves developing a Java application for the known Eight Puzzle game. Starting from a random configuration of the tiles the game consists of reaching the final sorted configuration moving each time one of the tiles adjacent to the current hole. The project is divided into three main Java Classes representing the main entities of the game. In the following sections, we will present in detail the requirements and the design choices for each component.

2 EightTile

The EightTile bean is the one representing each of the nine tiles of the board. The requirements for the project were to extend JButton and to have at least two private properties "position" and "label". Position is a constant while label is a property bound and constrained. The EightTile class provides two different constructors: the first one with no arguments for the Java beans specification and the second one with the initial position and label of the tile which is also responsible to instantiate a PropertyChangeSupport and a VetoableChangeSupport that will be used to fire events.

The "setLabel()" method is the one called by the board to try to swap the tiles. If veto parameter is set to True, the method fires a VetoableChangeEvent to ask the controller if the next move is legal or not. Otherwise, it will change directly the labels.

After the controller checks, if the move is vetoed, the method catches the VetoableChangeException and the current tile flashes in red to notify the user that the move is not legal. In case of a legal move, the tile fires two different events:

- PropertyChange "swap" event: notified to the board to change the label of the swapped tile;
- PropertyChange "hole" event: notified to the controller to update the new position of the current hole.

The last method called is the "updateTile()" which is responsible for changing the background and the text of the tiles according to the new labels.

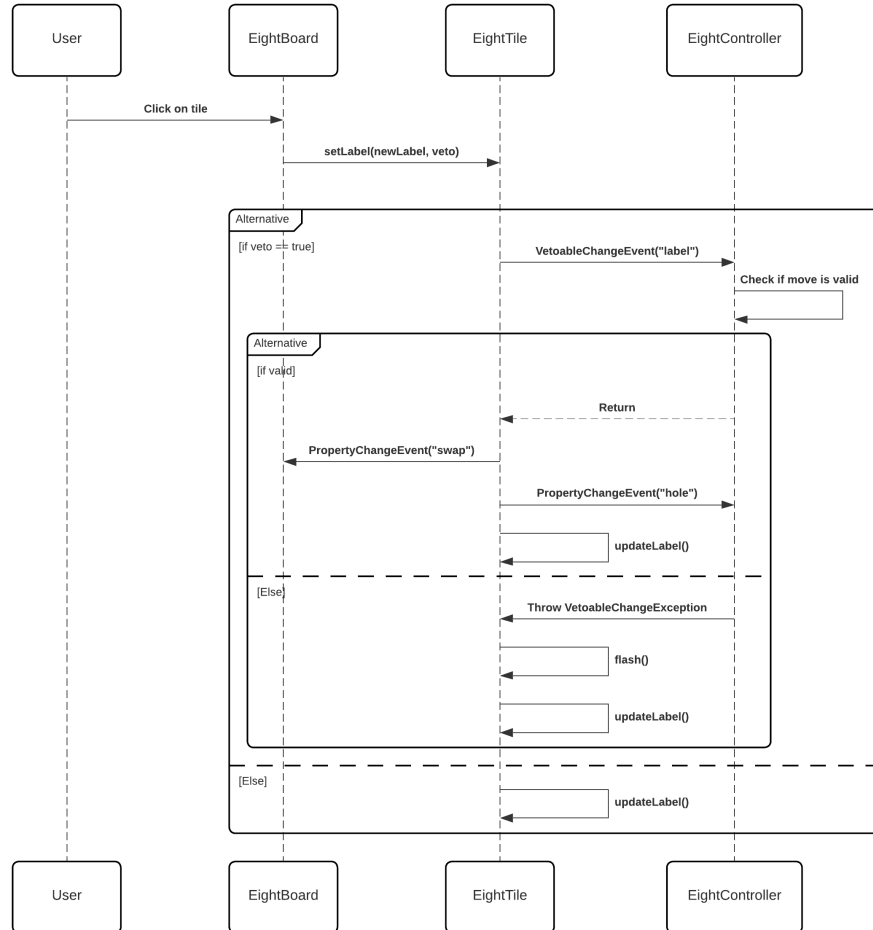


Figure 1: Tile Click

3 EightBoard

The EightBoard bean is the dashboard of the application. The class extends JFrame and defines a 3x3 grid with nine different EightTile objects and an EightController responsible for managing the logic of the game. The board has also two buttons "RESTART" and "FLIP" respectively to restart the game with a new permutation of the tiles and to implement the flip feature that will be described in the following chapters.

The class defines one constructor with no arguments that call two different methods: the first one instantiates all buttons, tiles and the controller while the second one initializes the Tiles array, registers the controller as a `VetoableChange` and `PropertyChange` listener for each tile and adds the board itself as a `propertyChangeListener` to manage the "swap" event.

4 EightController

The `EightController` class extends `JLabel` and is responsible for managing the logic of the `EightPuzzle` game. The constructor with no arguments for the Java beans specification sets the initial position of the hole and initializes the adjacency map that will be used to check if the two positions are adjacent or not. After initialization, the controller waits for `VetoableChange` or `PropertyChange` events coming from the tiles.

- **`VetoableChangeEvent` "label"**: checks if the clicked position is the hole and if this is not the case it checks the position of the current hole and the clicked tile in the adjacency map to decide if next move is legal or not;
- **`PropertyChangeEvent` "hole"**: it's responsible for updating the new hole position in the controller.

5 Flip operation

The Flip operation is a function required by the project to guarantee that if the final configuration cannot be reached from the starting one, the new one obtained by switching the labels of the tile in positions one and two will be solvable. This operation can be performed if and only if the current hole is in position 9. To implement this feature the `EightBoard` provides a button that when clicked asks the controller the position of the current hole. If the hole is in position 9 the function calls the `setLabel` method of tiles one and two with the `veto` parameter set to false to avoid the controller check.

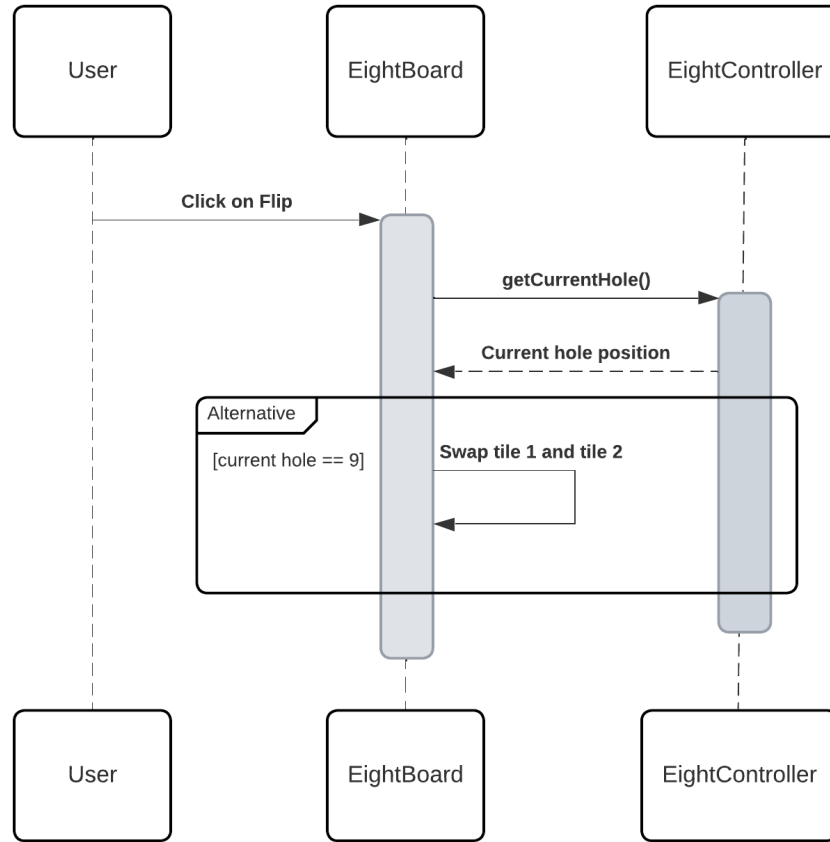


Figure 2: Flip Operation

6 Restart operation

The restart operation is a function required by the project to generate a new sequence of labels to be assigned to each tile. The implementation fills an array with numbers from one to nine and then shuffles it. This guarantees that labels are not repeated. With this random sequence, the `resetBoard` method assigns to each tile the new label calling the `setLabel` method and the `veto` parameter set to false.

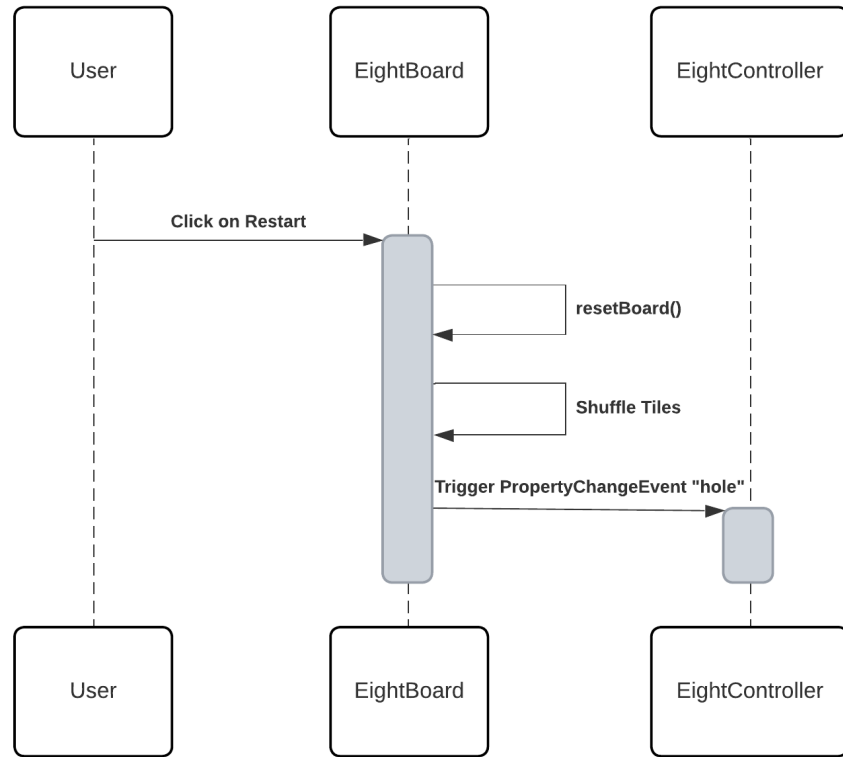


Figure 3: Restart Operation