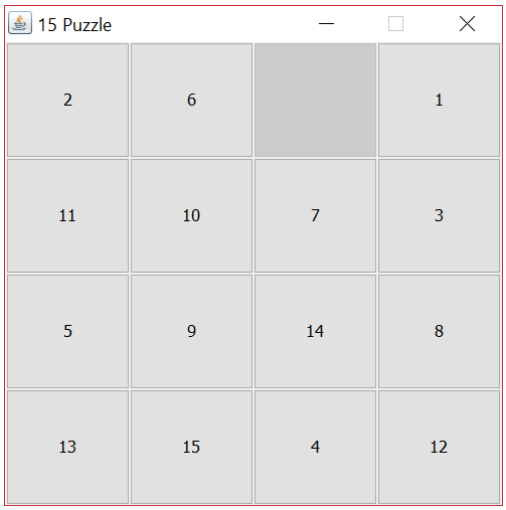
Object-Oriented Design
COMP 2000
Spring 2024

Lab 9

A 15-puzzle is a kind of puzzle made up of 15 tiles on a 4 × 4 grid. Thus, there are 15 tiles but 16 spaces, leaving one empty space. By moving one of the tiles surrounding the empty space to the empty space, the order of tiles can be changed. These tiles are typically numbered or have a picture printed on them such that one arrangement of the tiles puts all the numbers in ascending order or forms the picture. Thus, the goal of the puzzle is to move the tiles around until this ordering can be achieved.

A 15-puzzle is the name for this particular, common size of this puzzle, but an 8-puzzle is essentially the same, having 8 tiles on a 3 × 3 grid, and other sizes are possible as well. The goal of this lab is to complete a playable GUI for a 15-puzzle using the Java Swing library. It should look something like the following, although the tiles might be randomized differently.



1. **Create a project called Lab9 and create a package in your project called puzzle. Download Lab9.zip and add the two Java files it contains to the puzzle package. Read carefully the source code for both.**

2. **There are a whopping 20 TODO comments you must fill out inside the Puzzle class, and it would be redundant to list them here. READ CAREFULLY those TODO comments and ask questions before you begin to make sure you understand what you're doing. The indentation of the TODO comments is significant and indicates nesting of scopes in your solution. I recommend that you avoid edits in IntelliJ that automatically reformat these comments, since if you lose the indentation you will no longer understand the nesting structure of the solution.**

3. **The TODOs walk you through the process of representing the tiles in the GUI with 16 buttons (including an "empty" button with the number 0) and adding an ActionListener to each one so that it moves correctly. The hard work of figuring out which neighboring button is empty is done for you by calling a method. Note that it's important to add the buttons in order from 1 up to SIZE*SIZE - 1 (15 for a 15-puzzle) and then add 0 as the last button. We take a solved puzzle and then scramble it rather than randomly placing tiles because it is possible to create unsolvable puzzles.**

   Successfully completing the Puzzle class will result in a fully playable game that will announce when you have won.

4. **By adjusting the size variable in the main() method, you can play a 3 × 3 puzzle known as an 8-puzzle. You might wish to do so for debugging purposes, since it's time-consuming to try to beat the full 15-puzzle. (You could also play a 5 × 5 or 6 × 6, etc. if you want some fun.)**

5. **Deliverable: Zip up your entire `puzzle` package and email it to prof. stucki...**

   All work must be done individually. Never look at someone else's code. Please refer to the course policies if you have any questions about academic integrity. If you have trouble with the assignment, I am always available for assistance.