COMP2396 - Assignment 4

Due: **15 April, 2019, 23:55**

Introduction

This assignment tests your skills on writing simple Java **GUI** program, using Java **graphics** and **events**.

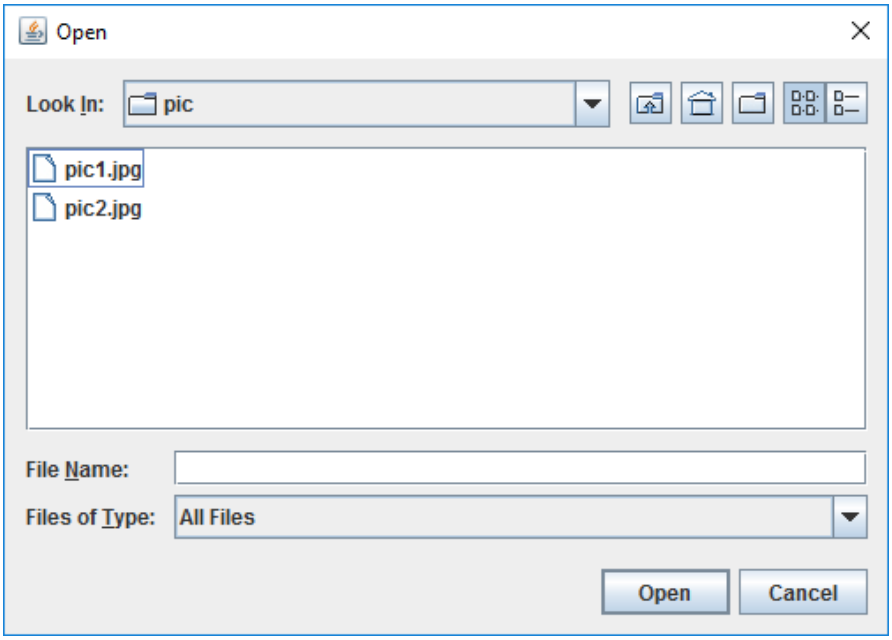
You need to write a puzzle program **JavaPuzzle.java**, which allows the user to select an image as a puzzle and the user can then play with the selected images.

This assignment will be evaluated on both **functionality** and **program design**. You can get part of the full marks if you implement some of the features.

You are expected to use Java **GUI graphics** to display the user interface of this assignment.

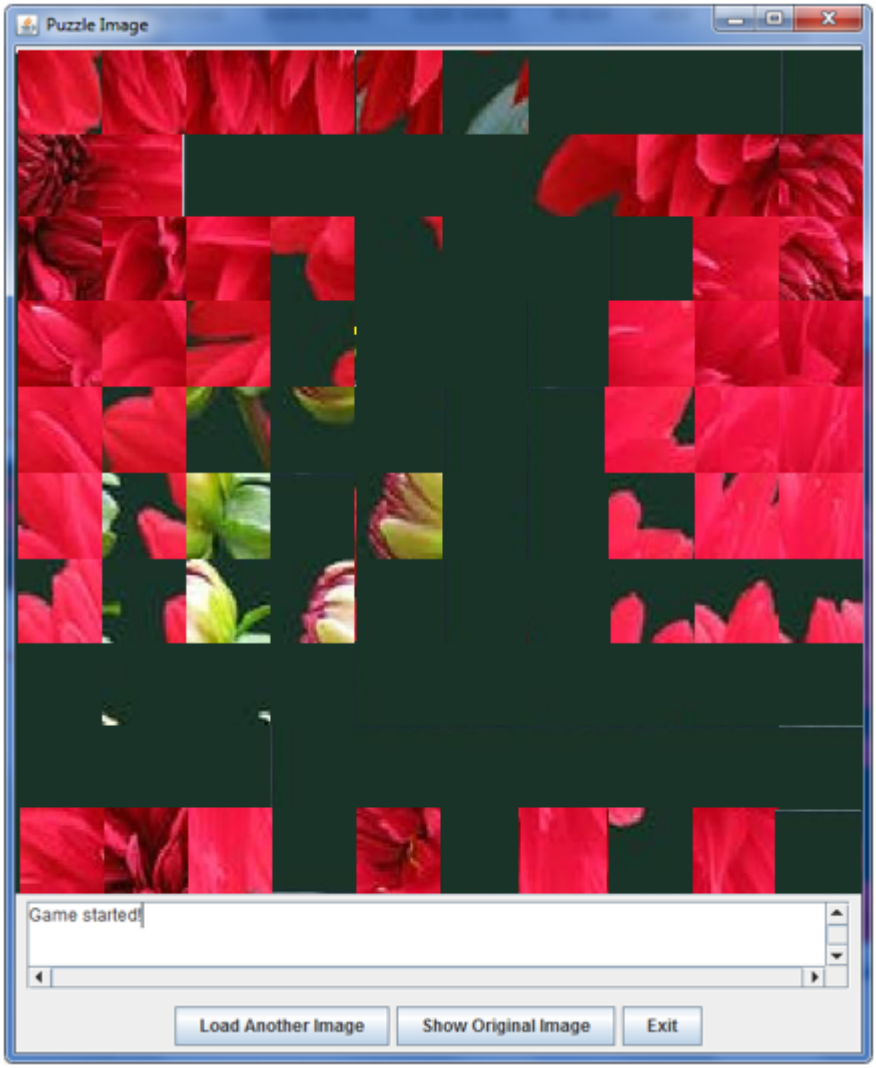
You are also required to write **JavaDoc** for all non-private classes and non-private class members. Programs without JavaDoc will not be marked.

Part I Select an Image



When **JavaPuzzle** is executed, a file chooser should be presented to ask for an image file.

Part II Display the Image and randomize the image



The image is loaded from the selected file and is divided into blocks of 80*80 pixels in size displayed as shown in above figure. The blocks of the image are randomized so that the user will not see the original image. If the image file fails to load, the program should generate an error message and ask the user to select another image file. Each image block should keep its original position information so that the original image can be reconstructed.

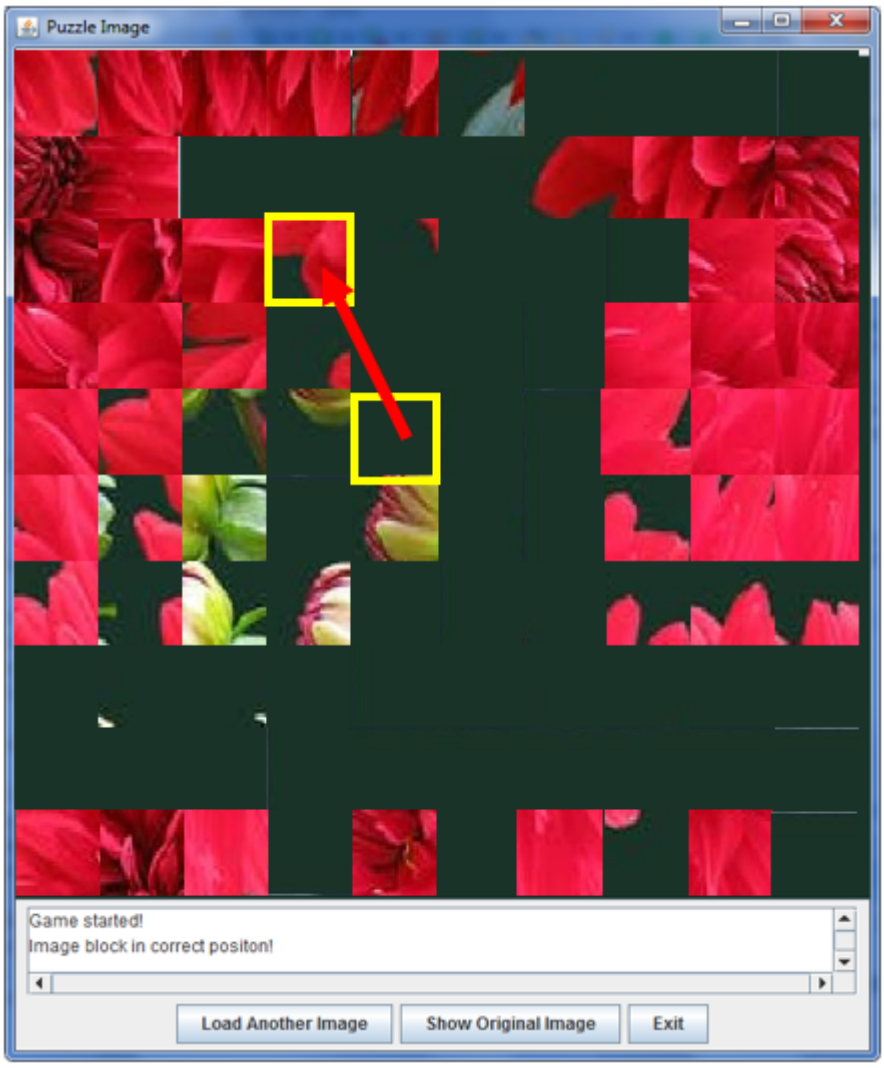
Your main UI should contain the following:

1. A JPanel for the image block display
2. A JTextArea for display message
3. Three JButtons: load image, show the original image and exit.

You are not required to follow exactly the size of the JTextArea and JButton objects as in the the above figure, but should look similar to it.

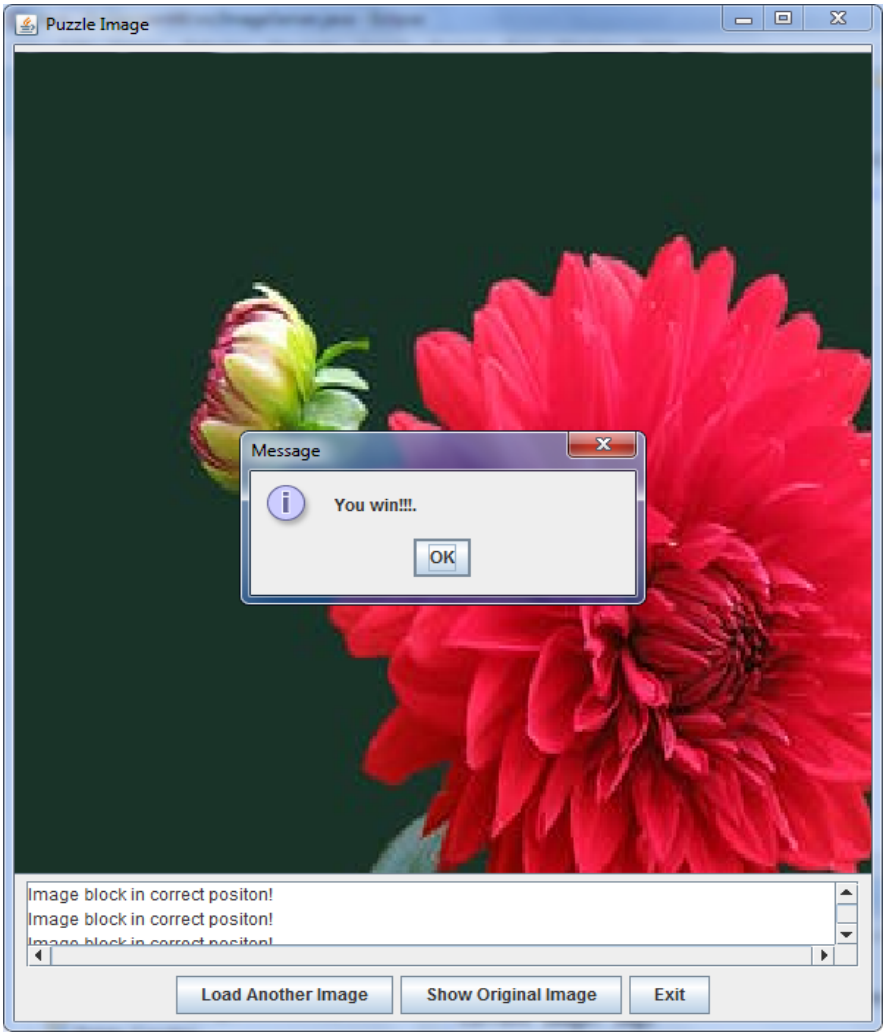
The image should be **resized and scaled** to fit into a canvas of 800*800 pixels in size. If the new image fails to load, an error message should be displayed and the old image is retained.

Part III Drag and Drop the image to the right position



The movements of image block are controlled manually by pressing and releasing the mouse. If an image block is currently at (i, j), the user has to press the mouse at (i, j) and drag it to the target image block position, then the selected image block and the target image block will be swapped.

Note that the image block is not required to be with the cursor during dragging. Make sure it causes a move when you release it at a proper position, and updates the image blocks in the panel. After swapping, if any image block is placed on its original correct position, a message should be displayed.



If the destination image block is in the correct position, drag and drop another image block to that position will NOT swap their position, and a message should be displayed. The game is finished when all image blocks are in correct position.

Marking

80% marks are given to the **functionality**.

- You may add additional classes, instant variables and methods to the project
- You will get part of the full marks if you implement some of the features.
- A program that can run normally without throwing exceptions during runtime gets higher marks.

20% marks are given to the **program design.**

- You should make good use of inheritance, polymorphism, layout, painter and event handler.
- You can check it by avoiding code duplication as much as possible.
- **Economy is valuable in coding: the easiest way to ensure a bug-free line of code is not to write the line of code at all.**

Programs without JavaDoc will not be marked.
Programs with compilation error will not be marked.

Submission:

Please submit all source files (*.java) in a single compressed file (in .zip or .7z) to Moodle. **Do not submit *.class* file.**