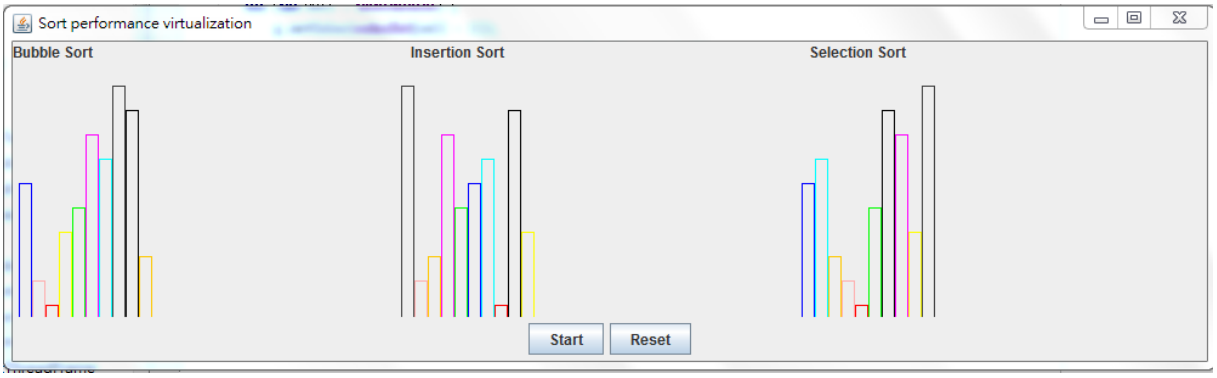
2012/11/06 Software Studio Lab 5

Honor Code

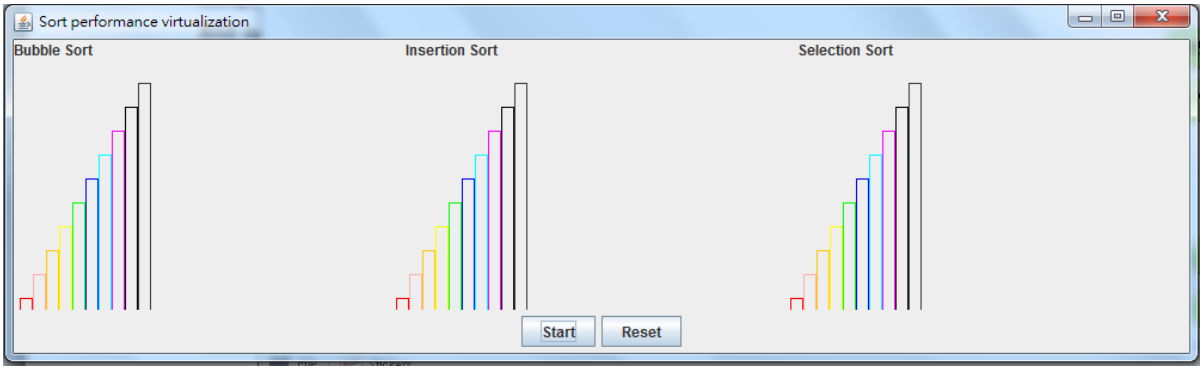
Any cheating will be handled seriously in compliance with the university rules. All assigned work is expected to be individual, except where explicitly written otherwise (e.g., term project). You are encourages to discuss with your classmates; however, what you hand in should be tour own work.

1. (40%) **Simultaneous Visualization of Sorting Algorithms.** Bubble sort, insertion sort and selection sort are classical sorting algorithms in computer science. Use Swing GUI to visualize the processes of bubble sort, insertion sort and selection sort. Each sorting algorithm has to sort one int array that contains 10 different integers from number 1 to 10. The 10 integers are randomly distributed in the array. Draw histograms to visualize the process of sorting from time to time. Also, you have to simultaneously execute the three sorting algorithms by using JAVA threads, and visualize the processes of the three algorithms in the same panel at the same time.

In the following sample, users can click the "Start" button to start executing the three algorithms and visualizing their execution:



The screenshot after the three algorithms are completed.

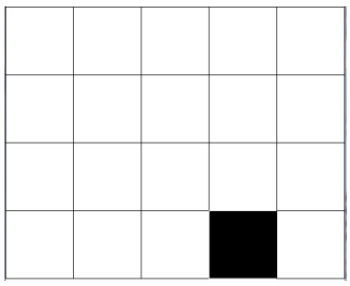


2. (60%) **Black/White Chess Board Revisited**. Last week, we programmed the first version of the black/white chess board as a standalone, single-player game. This time we want it make it a two-player game so that two players can collaborate to play the game. In this new version, two players can play together over socket communication to color cells of the chess board. Same to the previous version, the goal is to color all the cells black.

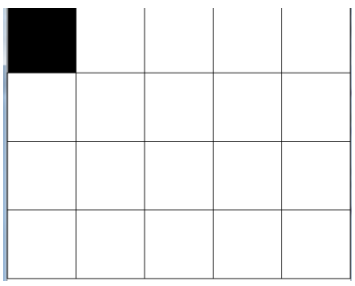
   Each of the two players will have access to one **5 (columns)*4 (rows)** chessboard. Choose one chess board to be the server, and another one to be the client. Program the server chess board appropriately so that it can wait for the client. Once the client connects to the server, the game starts. At the beginning, these two chess boards have to randomly choose one cell to be black. In the game, clicking a cell does not only change local chess board, but also modify the corresponding cell in the partner's chess board. Same to last week's assignment, when a player clicks one cell, this cell will switch the color (black to white or white to black). Then vertically adjacent and horizontally adjacent cells will also change the color. What's different now is that the adjacent cells of both chess boards will be changed.

For example, in the beginning

Server Chess Board

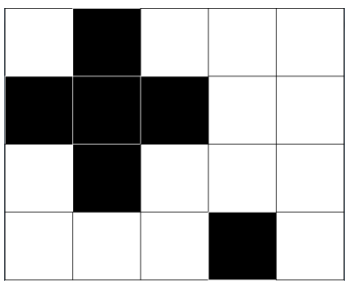


Client Chess Board

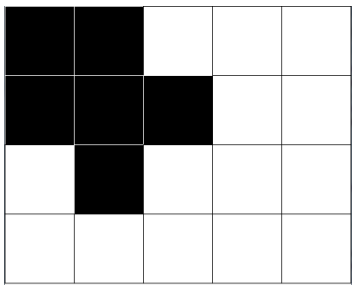


After Server click 2-2

2-2,1-2,2-2,1-3-2,2-3 switch color

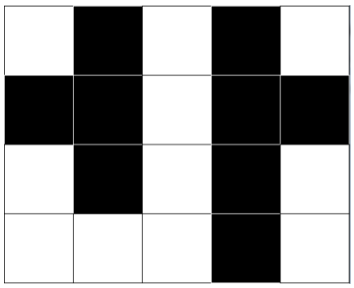


Client 2-2,1-2,2-2,1-3-2,2-3 also switch color



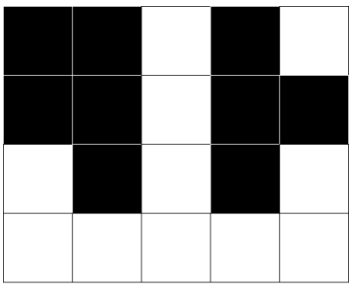
Server 4-2,3-2,5-2,4-1,4-3

also switch color

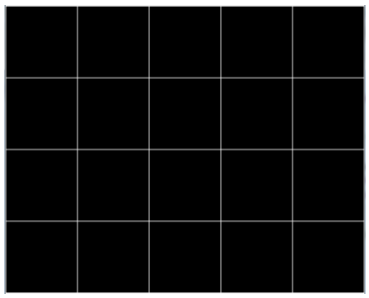


Client click 4-2

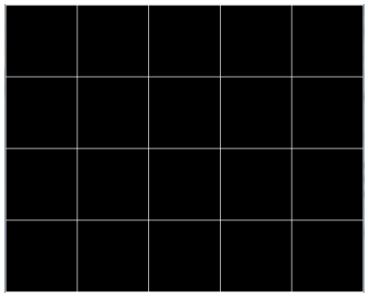
4-2,3-2,5-2,4-1,4-3 switch color



If server



and client

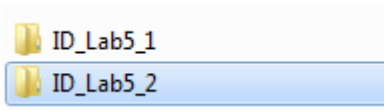


The game is end.

Tips : Using the JAVA Socket to pass data and use thread to retrieve data.

注意事項：

1. **Deadline: 2012/11/18 23:59 (11/19 00:00~23:59 交者，成績*0.8)**
2. 交作業時，請按照以下格式標示清楚題號：



   請在各題的資料夾放入**整個 project 資料夾、export 後的 runnable .jar file、Readme.txt**
   再將這些資料夾一起壓縮成”**學號_Lab5.zip**”，
   將壓縮檔上傳至 http://lms.nthu.edu.tw/ 軟體實驗的**作業**區
3. 每個.java 檔內，第一行請用註解加上學號、姓名及題號。
4. 程式碼務必要有**註解**，說明你解題的方法。(此項也列為評分標準之一)
5. Readme 請包含執行步驟(條列式)、遇到的困難及解決方法。