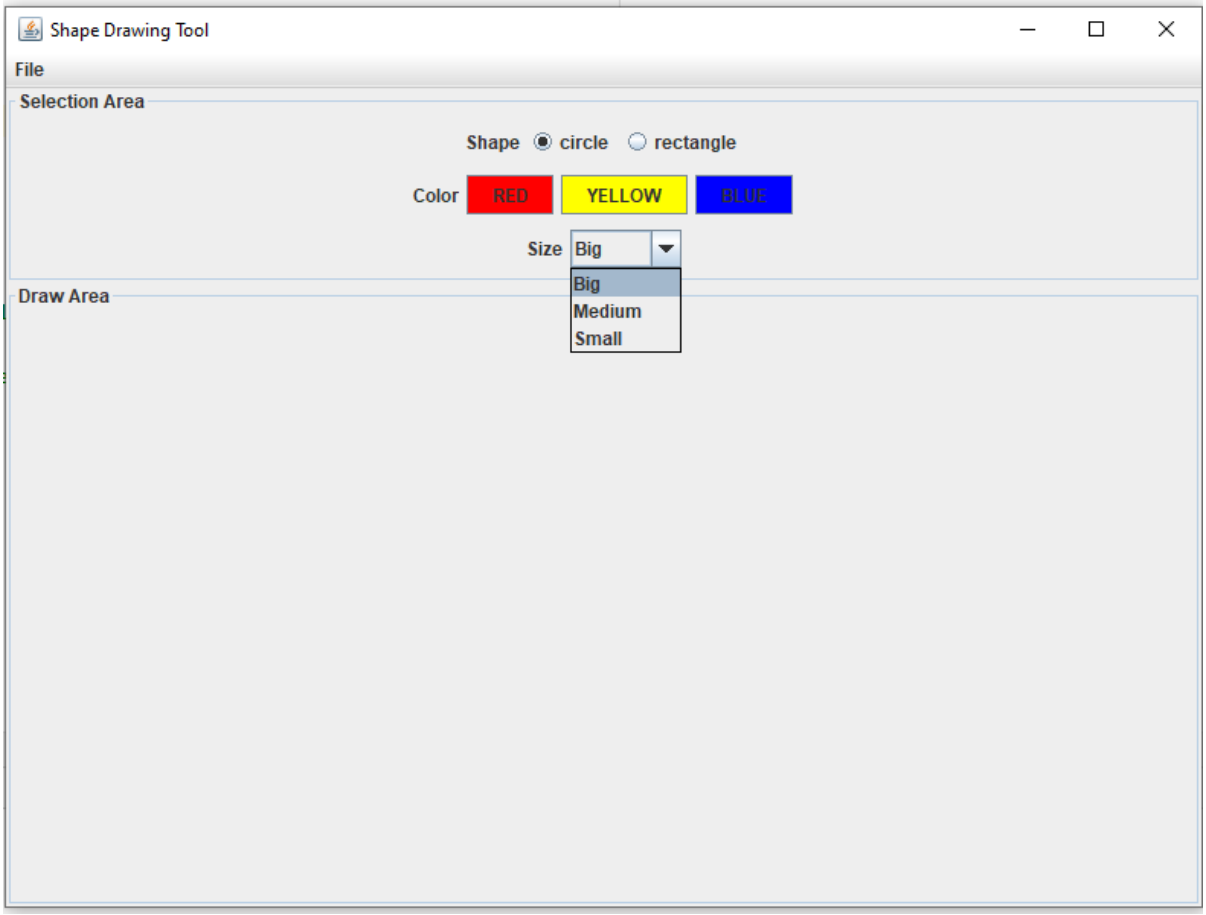
AP/ITEC 2610 Fall 2023 Section D
Assignment 2

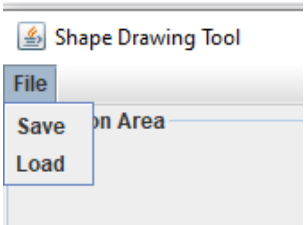
Task Description

Please complete a Java application to draw different shapes with the specified color and size, subject to the following requirements.

1. GUI requirements

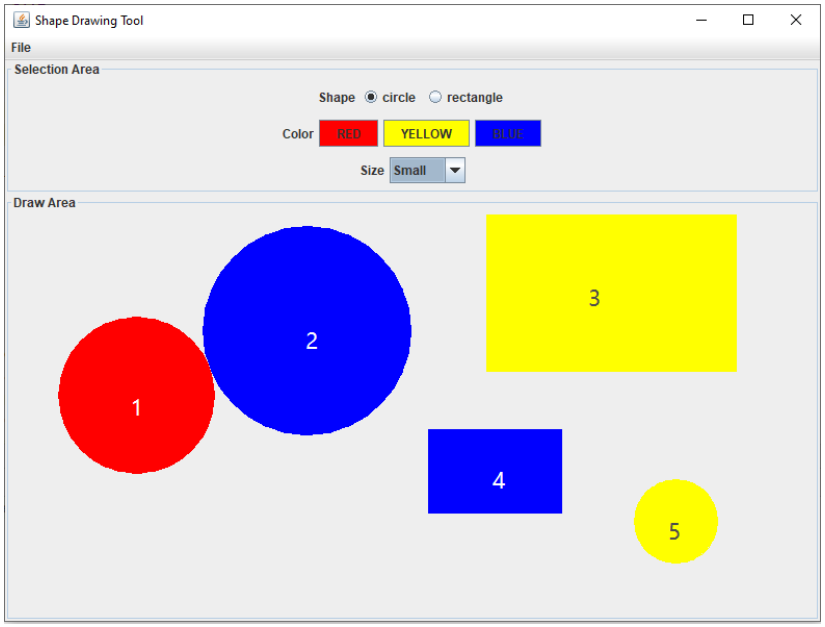


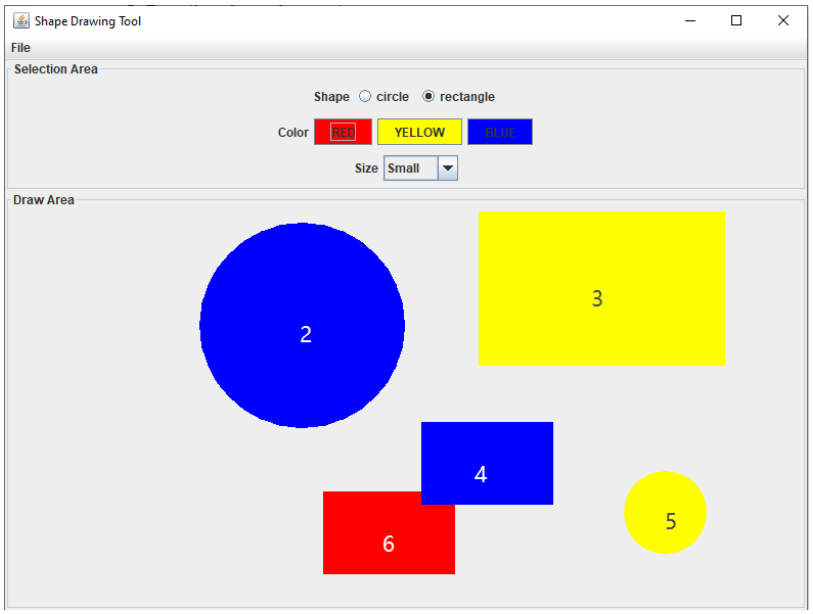
1) The GUI should be designed as above and all the components in it should be included. Do NOT add other components.
2) The layout should be consistent with the above screenshot.
3) The sizes of the frame and each panel can be decided by yourself, as long as they are reasonable.
4) The “File” menu should contain two items: Save and Load.



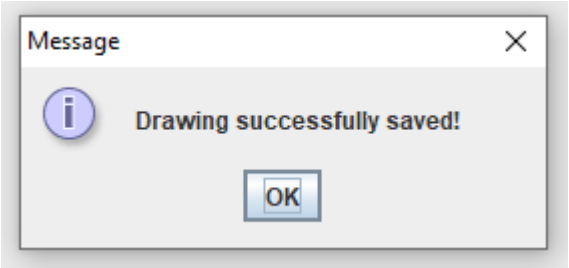
2. Functional requirements

1) Users can start with selecting the shape, color, and size in the “Selection Area”. Then, they can use the mouse to click any position in the “Draw Area” to draw a selected shape with the specified color and size. The position clicked is used as the center of the shape.
2) At any time, only up to 5 shapes can be shown on the Draw Area. As shown in the pictures below, after finishing the 5 shapes (Shapes 1 to 5), if the user draws another shape (Shape 6), the first shape (Shape 1) will be discarded and not shown any more. The same rule applies to the subsequent shapes (with Shape 7 replacing Shape 2, 8 replacing 3, and so on). (Hint: you can use an array to store these 5 shapes and an index to identify the position in the array for the current shape.)





3) The sizes of Big, Medium, Small for each shape can be arbitrarily specified as long as they look reasonable.
4) The shapes are drawn in the order they are added. That is, if there is an overlap between Shapes 2 and 3, then Shape 3 is drawn over Shape 2.
5) When selecting File->Save, the properties of the current drawn shape (including the type, colors, size and center) should be written to a file (named “shapeDrawing.txt”) in text format and have a pop-up window showing the message “Drawing successfully saved!” like the picture below. After that, the Draw Area should be cleared.



6) When selecting File->Load, the last file you saved will be read and the shape should be shown in the “Draw Area”. At the same time, pop up a window to show the message “Drawing successfully loaded!” If there is no saved drawing, pop up a window to show the message “No saved drawing!”
7) Please properly handle the checked exceptions (e.g., I/O exceptions) during the whole process.

What to submit:

Please complete the application according to the requirements and comment the code properly. **You have to design your own classes, keeping the OOP design principles in mind. Name your driver class (the class with the `main` method) `DrawingTool`.**

➔ Submit a single zipped archive named A2.zip containing all your source files. Nothing else (e.g., the shapeDrawing.txt file) should be submitted.

➔ DO NOT submit .class files! .java files ONLY. DO NOT use other archival formats such as .RAR or .7z.

Important Notes

(1) Your assignment will be given a zero mark if only the compiled files (.class files) are submitted. Please make sure to submit the source files (.java files).

(2) Please make sure to use Eclipse IDE as I will be testing your code on Eclipse IDE. If your code is written in a right manner, I will be able to run it on the Eclipse IDE installed on my computer. Remarking requests like “...but the code works on my computer/in my IDE” will not be entertained. No exceptions. It should run as you have seen me running examples on Eclipse IDE in the classroom.

Marking Scheme

Only the completed GUI with all defined functionalities will get 100/100 marks