Alexandria University, Faculty of Engineering
CC272: Programming 2

Spring 2016
Project
Due: Friday, 6 May 2016

An Object Oriented Drawing and Painting Application

Objectives:

Upon completion of this assignment, you will be able to:

- Abstract and model entities from reality in the form of classes
- Design an object oriented model for geometric shapes
- Draw a UML class diagram that represents your model
- Apply the OOP concepts of inheritance and polymorphism to your design
- Create an advanced GUI with 2D Graphics capabilities

Part 1: Geometric Shapes Data Model

Description:

Geometric shapes belong to different groups (ex: Elliptical Shapes, Polygons, Sectors…etc…). Members of these different groups are related to each other in the sense that they share common properties. In order to be able to implement an efficient and object oriented drawing application. It is essential to design a model that takes these relations into consideration.

Tasks:

- Design an object oriented model that covers the following geometric shapes: Line Segment, Circle, Ellipse, Triangle, Rectangle and Square.
- Identify all required Design patterns that can help to facilitate the implementation of the Application
- Draw a UML Class diagram that represents your model, showing all the

classes’ attributes and methods.

- Apply the concepts of inheritance and polymorphism to your design.

Part 2: Drawing and Painting Application

Description:

Drawing and painting applications are very popular and have a huge user base. They generally offer a big number of features that includes but is not limited to: Drawing, Coloring, and Resizing. They allow the user to undo or redo any instructions so as to make the application more usable.

Tasks:

- Implement your design from part 1 in an OOP language.
- Design and implement a GUI that allows the following functionalities for the user on all the shapes defined in part 2: Draw, Color, Resize, Move, copy and Delete.
- Implement your application such that it would allow the user to undo or redo any action performed.
- The cursor should be used to select the location of a shape while drawing it, or moving it to another location, for more accurate control on the shape parameters (ex: size), dialog boxes could be used, or you’re free to implement it in a more user friendly way of your choice.

Bonus Features:

- Two extra features could be added to the application: Rotate and Copy, in case of rotate, the select shape should be rotated about its center with a user defined number of degrees. As for the copy, a shape could be selected for copy, and pasted in any other spot of the drawing canvas.
- A “history” list of actions could be maintained so as to allow the user to return to a given stage, undoing all the actions performed since the selection action, this is considered as an extension for the undo/redo feature in part 2.

Deliveries & Notes:

- ✓ You should write the program using **java language**.
- ✓ You should work in a **group of three**.
- ✓ Your code should be **clean**, **readable** and **commented**.
- ✓ You should deliver a report that includes the relationships between entities you are modeling **(UML)**, their attributes, how encapsulation has been followed, user guide, as well as snapshots for your GUI.
- ✓ **Bonus** marks will be 10% of the total mark.
- ✓ Send the source code and the report to **ssp.oop.2016@gmail.com** **using subject [Project_group_number].**
- ✓ Late submission won't be accepted.
- ✓ Delivering a copy will be awfully penalized for both parties, so delivering nothing is so much better than delivering a copy.

Tutorials:

- https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html
- http://docs.oracle.com/javase/tutorial/uiswing/learn/settingup.html
- https://www.youtube.com/watch?v=LFr06ZKIpSM
- https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
- https://netbeans.org/kb/docs/java/gui-functionality.html#Exercise_1
- http://www3.ntu.edu.sg/home/ehchua/programming/java/j4a_gui.html

Good Luck