



Paint Report

CS 221: Programming II

Team Members

Mahmoud Tarek Samir	(62)
Mostafa Mohamed Labib	(67)

Table of Contents

Introduction	2
Design Architecture	2
Implementation	2
UML Diagrams	3
Factory method design pattern	3
Command design pattern	4
MVC Model	4
User Guide	5
Introduction	6
Overview	6
User Interface	6
Instructions	7
Appendix : Plugins	12
References	13

Introduction

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 also include several built in, and possibly extensible set of geometric shapes, and classically, they allow the user to undo or redo any instructions to make the application more usable so we implemented a Vector Based Drawing Application.

This Guide here is to show how we managed to implement the code and there is a user Guide that shows how to use the program.

First we included the design Architecture supported with UML diagrams then there is a detailed sections to show how to use the program

Design Architecture

1. An MVC model based implementation.
2. Using Factory design pattern with creating shapes.
3. Using Command design pattern with undo and redo actions.

Implementation

With creating a class named **GUI** as a view, a class named **MainController** as a controller and the classes that implement the given interfaces as a model we managed to make an MVC model.

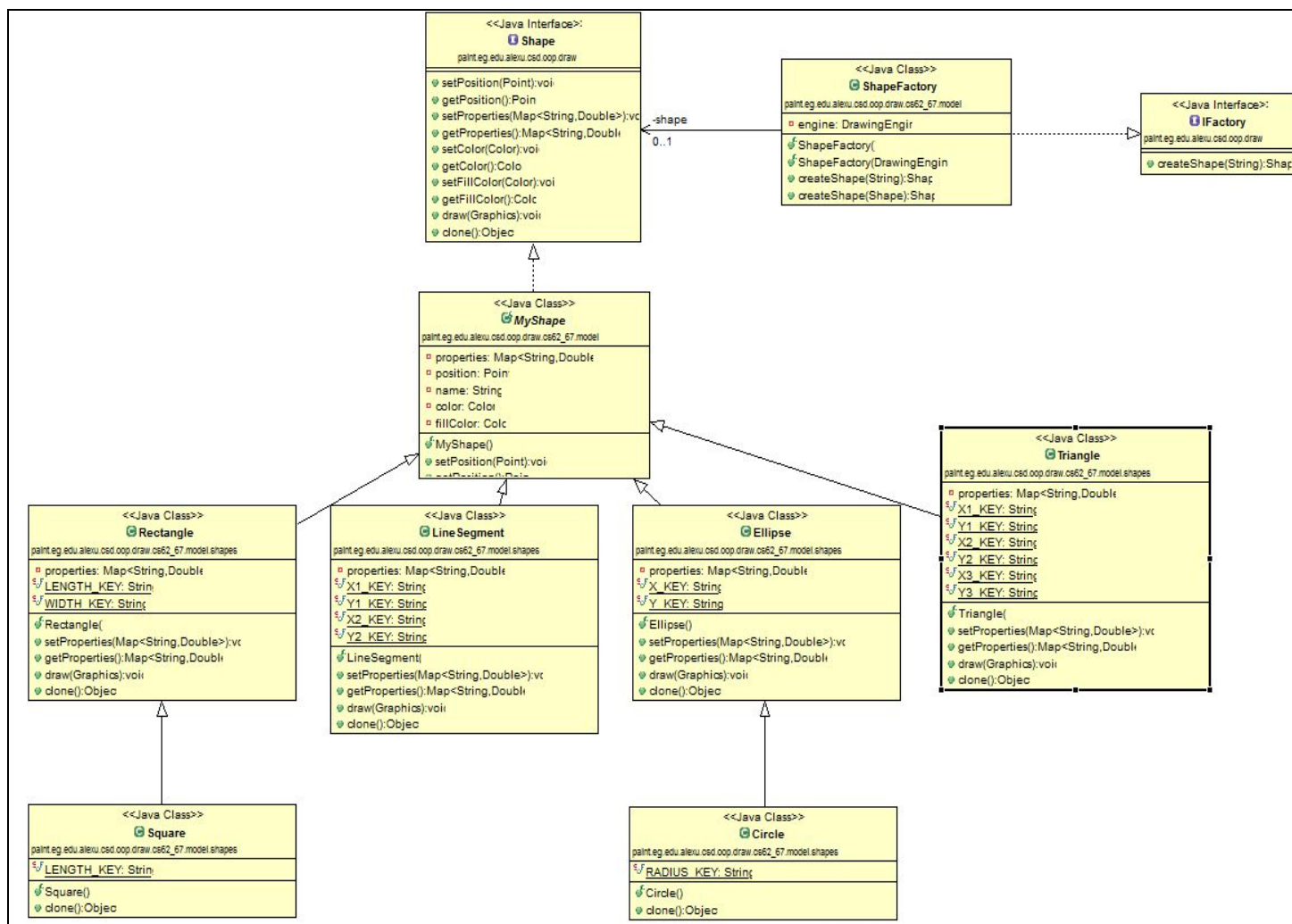
By implementing interface named with **Shape** by an abstract class named with **MyShape** then extend that class by different shapes classes and a factory class named with **ShapeFactory** we implemented Factory design pattern.

We created an extra interface called **ICommand** that was implemented by some classes that represent different actions and by using **DrawingEngine** interface that was implemented by **MyDrawingEngine** we managed to use Command design pattern.

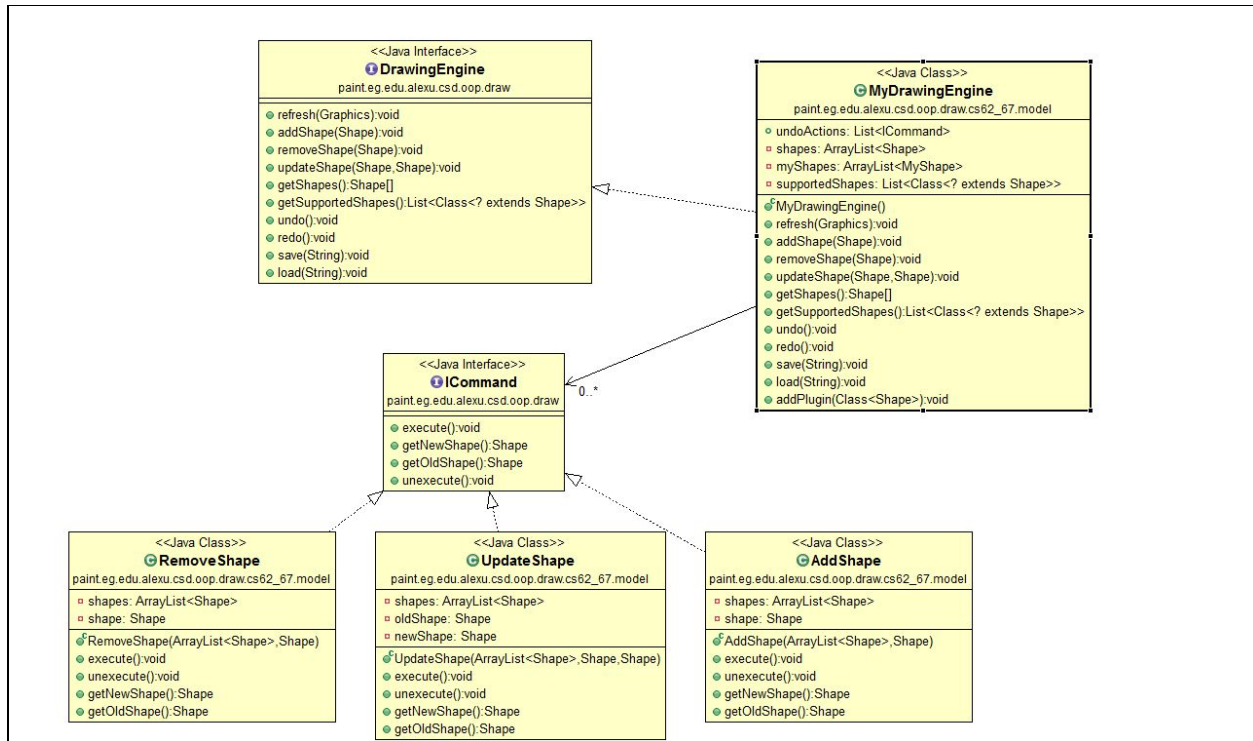
UML Diagrams

Here is the class diagrams for different implemented design patterns.

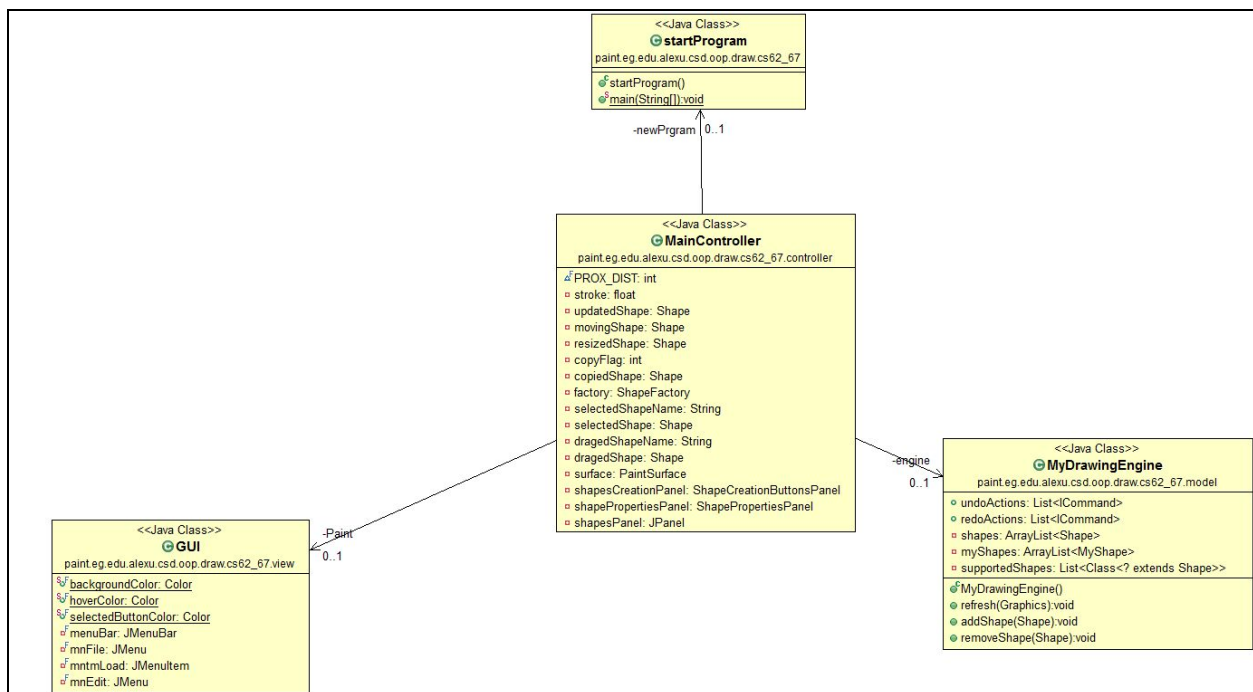
1. Factory method design pattern



2. Command design pattern



3. MVC Model



User Guide

About this Guide

This guide is divided into the following sections:

- Section 1 : "Introduction"
- Section 2 : "Describing the software" gives an overview of the key features and software Environment.
- Section 3 : "How to use this software"
- Appendix : "Plugins"

Who should use it

This guide is intended for users of different degrees of knowledge and experience with this program.

This guide assume that you have some knowledge of the operating system.

Typographical Conventions

This guide uses the following typographical conventions :

- Command and option names appear in **bold type**.
- Actions such as mouse actions appears in *italic type*.

1. Introduction

This guide is to define the functionality and features of this program and teach the user how to use it and do specific things that he want.

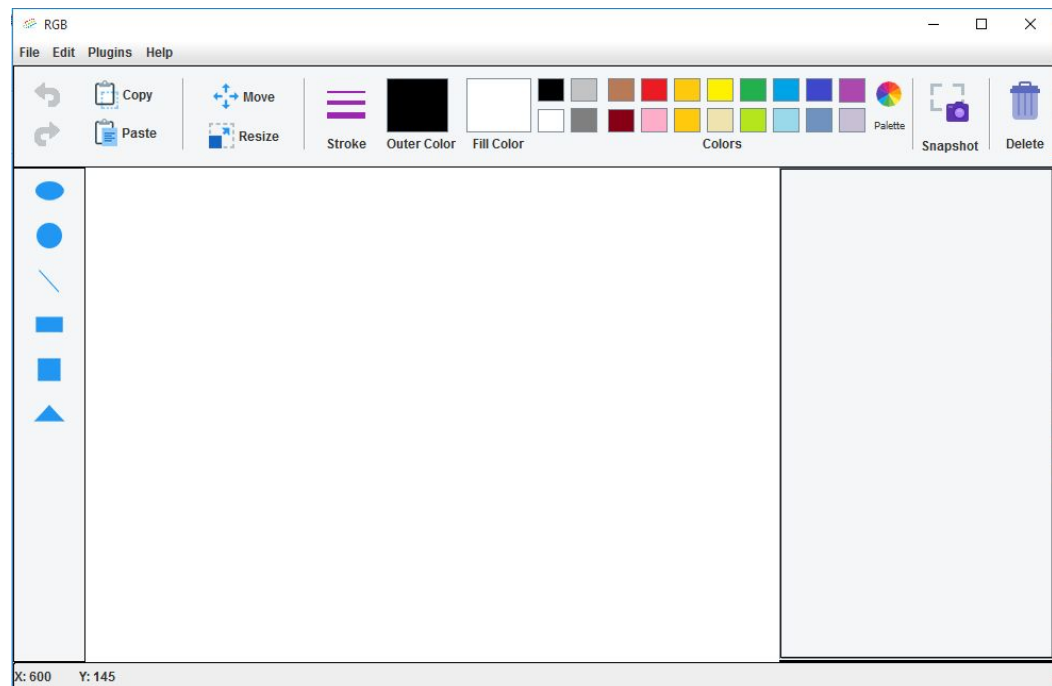
2. Overview

This program allows you to draw different shapes (Ellipse, Circle, Line Segment, Rectangle, Square and Triangle) and manipulate them, it supports many features :

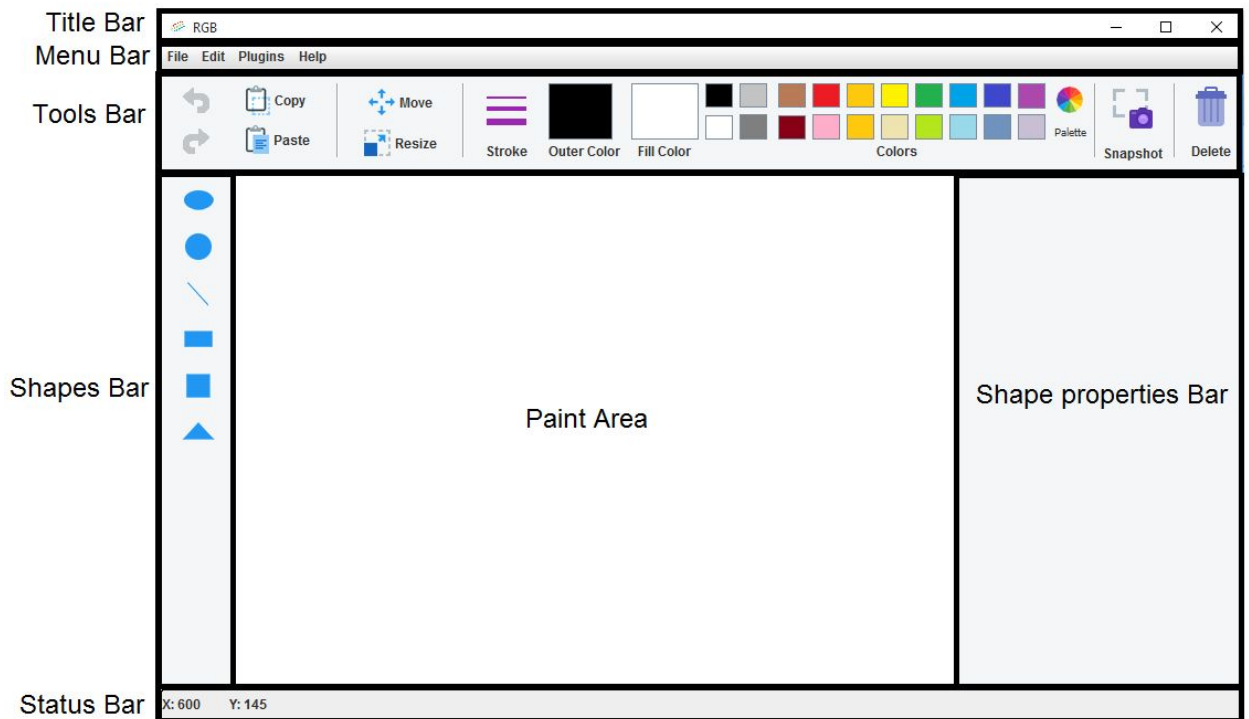
- Save & Load
- Undo & Redo
- Copy & Paste
- Move
- Resize
- Delete shapes
- Add plugins

3. User Interface

When you open the program, the following window will appear



Now, we will explore this Window. The following figure demonstrate the different parts of this window.



4. Instructions

4.1. Create Shape

To create new shape:

- Click on the shape that you want from **Shapes Bar**.



then

- Click on **Paint Area**, *drag* the mouse till you reach the appropriate shape size then *release* the mouse.

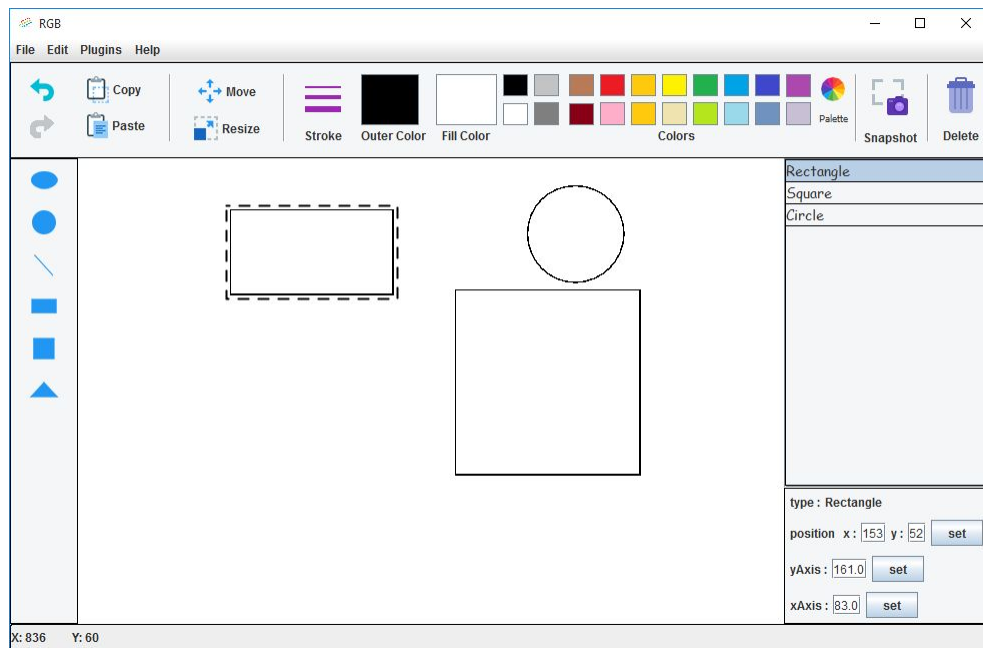
or

- Click on **Paint Area** then *release* the mouse.
- Edit shape properties from **Shape Properties Bar**.

4.2. Select Shape

To select shape you have drawn:

- Click on the shape from **Shape Properties Bar**.
- You will see this shape highlighted in **Paint Area**, the following figure illustrates this.



4.3. Move Shape

- *Select* the shape you want to move from **Shape Properties Bar** then
- *Click* on **Move** Button in **Tools Bar**.



- *Drag* the shape to the point you want to move it.
- or
- Edit the position property from **Shape Properties Bar**.

4.4. Resize Shape

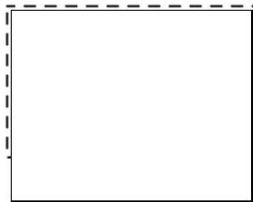
- *Select* the shape you want to resize from **Shape Properties Bar**.

Then

- *Click* on **Resize** Button in **Tools Bar**.



- Click on the shape from the side or corner you want to resize the shape from it in **Paint Area**.
- Drag this point till you reach the desired size.

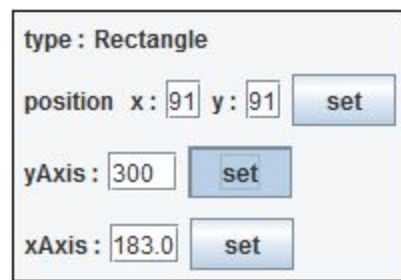


Or

- Edit the shape properties from **Shape Properties Bar**.

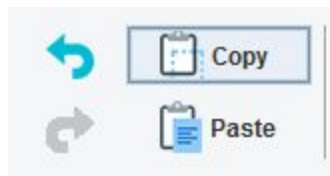
4.5. Edit Shape Properties

- Select the shape from **Shape Properties Bar**.
- Enter the new value for the property you want to change in its text field in **Shape Properties Bar**.
- Click on **Set** Button besides to this property.



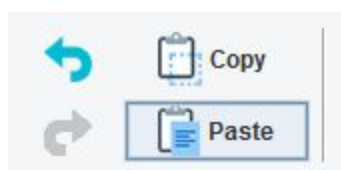
4.6. Copy & Paste Shape

- Select the shape.
- Click on **Copy** Button in **Tools Bar** or press **CTRL+F**.



Then

- Click on **Paste** Button in **Tools Bar** or press **CTRL+D**.

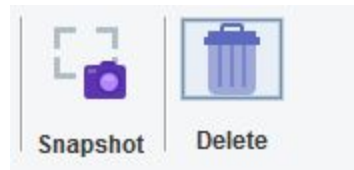


Or

- Click on the new desired position on **Paint Area**.

4.7. Delete Shape

- Select the shape.
- Click on **Delete** Button on **Tools Bar** or Press **Delete**.



4.8. Color & Fill Color Shape

- Select the shape.
- Click on **Outer Color** Button or **Fill Color** Button (as you want) in **Tools Bar**.
- Select the desired color from **colors pane** or from **the palette** in **Tools Bar**.



4.9. Stroke

- Select the shape.
- Click on **Stroke** Button to change current stroke from **Tools Bar**.



- Click on it once more again till reach the desired stroke.

4.10. Undo

- Click on **Undo** Button from **Tools Bar** or Press **CTRL+Z**.



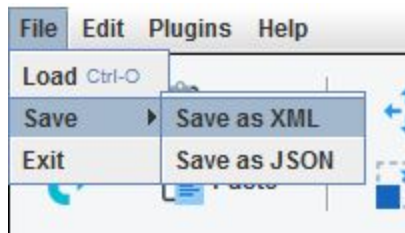
4.11. Redo

- Click on **Redo** Button from **Tools Bar** or Press **CTRL+Y**.



4.12. Save

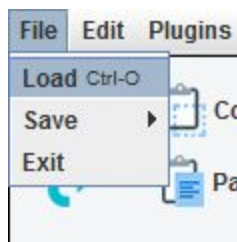
- Click on **File** menu.
- Select **Save** menu item.
- From pop-up menu, Select **Save as XML** or **Save as JSON** (as you want).



- Browse to navigate the desired path.
- Write the file name then Click on **Save** button.

4.13. Load

- Click on **File** menu.
- Click on **Load** menu item.



Or

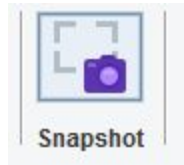
- Press **CTRL+O**.

Then

- Browse to navigate the file directory then *select* it.
- Click on **Open** button.

4.14. Take a Snapshot

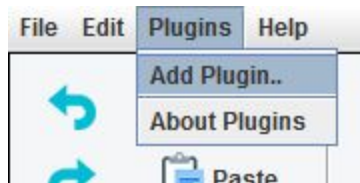
- Click on **Snapshot** Button in **Tools Bar**.



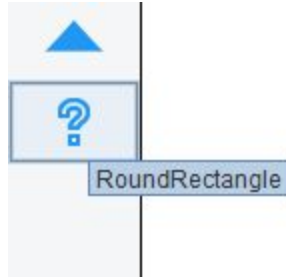
- From pop-up window, Browse to navigate the desired path.
- Write the image name then *Click* on **Save** button.

4.15. Add Plugin

- *Click* on **Plugins** menu.
- *Click* on **Add Plugin..** menu item.



- Browse to navigate the plugin (jar file) directory then *select* it.
- *Click* on **Open** button.
- The new shape will appear in the end of **Shapes Bar**.



5. Appendix : Plugins

The concept of dynamic class loading is widely spread in computer applications; it is an option that allows the user to extend application features at runtime. This Program provides this feature. Easily, you can add more shapes in addition to the Basic shapes that the program supports.

All you have to do is getting the source code for the new shape as a jar file and import it to the program by add plugin option as mentioned above in instructions section.

References

- <http://www.klariti.com/technical-writing/User-Guides-Tutorial.shtml>
- <https://www.tutorialspoint.com/swing/index.htm>
- https://sourcemaking.com/design_patterns
- <http://www.java2s.com/>
- <http://tutorials.jenkov.com/java-reflection/index.html>
- <https://docs.oracle.com/en/>