PAINTER APPLICATION REPORT

CS221 (Programming - 2)

Ahmed Khalil Mohammed Yakout (08)
Saeed Hamdy Mahmoud Hassan (31)

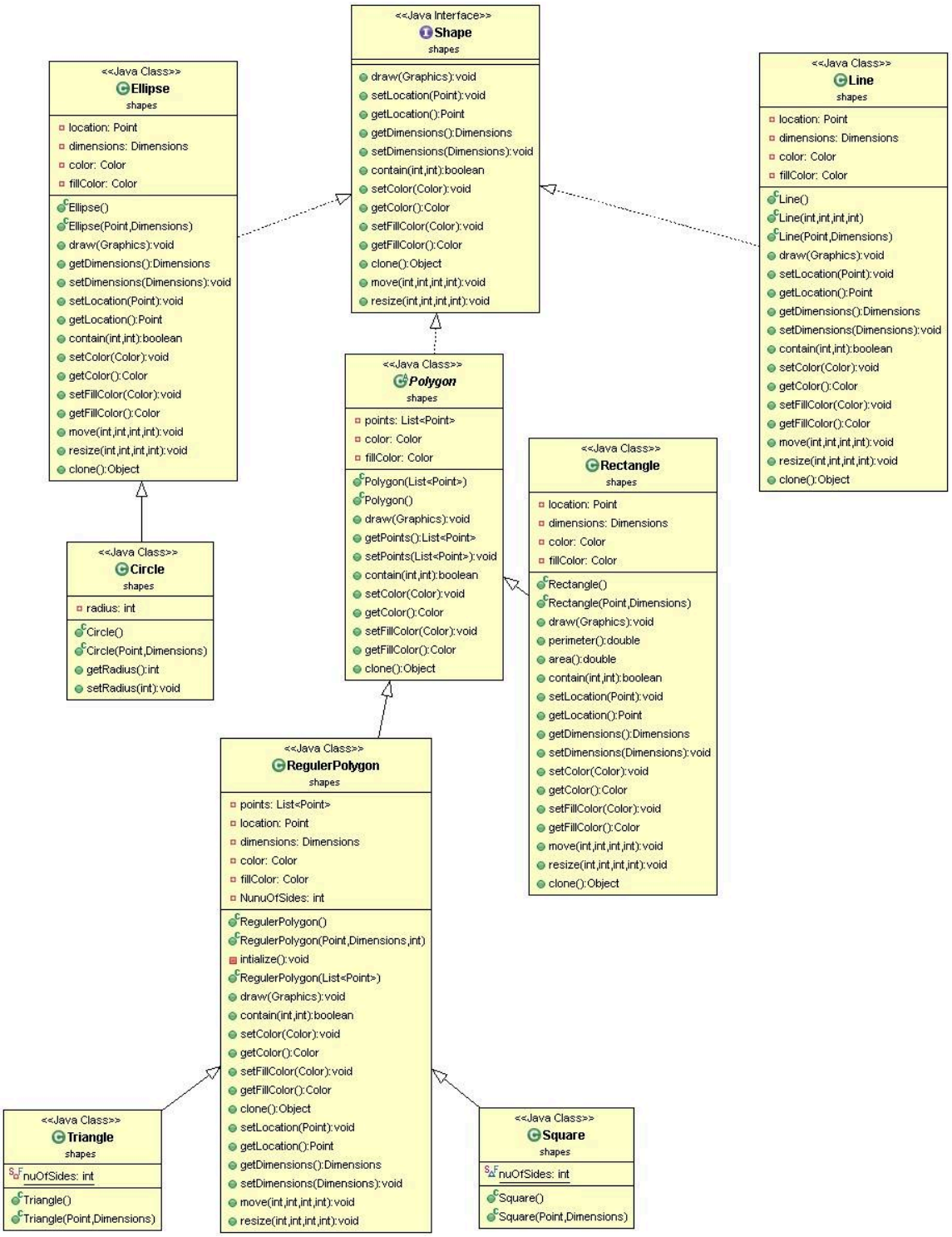
Including User Guide

TABLE OF ONTENTS

DESIGN

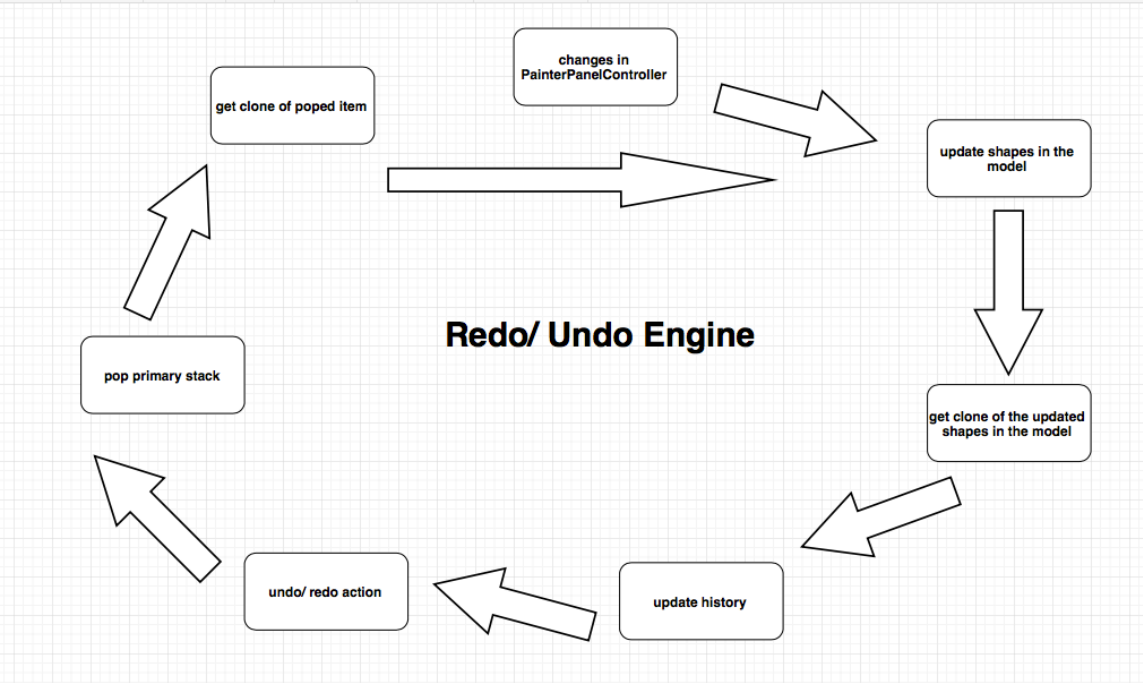
UML

UML diagram for the shapes.

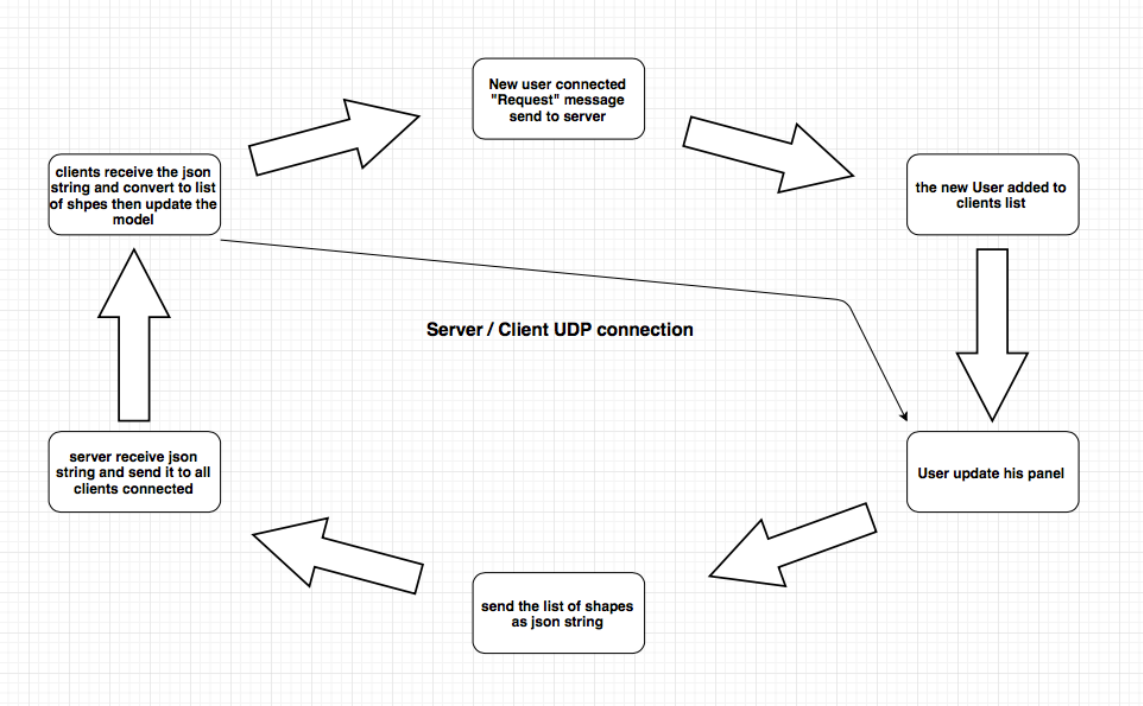


Redo/ undo engine

redo/undo actions were implemented using command design pattern.



Server-client UDP connection



Why swing not JavaFX?

"JavaFx handles rendering tasks for you automatically, rather than handling them manually. This is one of most important ways that JavaFx improves on Swing. As you may know, in Swing or the AWT, you must call the repaint() method to cause the window to be repainted. Furthermore , your application needs to store the window contents, redrawing them when painting is required. JavaFx eliminates this tedious mechanism because it keeps track of what you display in a scene and redisplays that scene as needed. This is called *retained mode*. With this approach there is no need to call a method like repint()"

- from: Herbert Schildt, Java The complete Reference 9th edition chapter 34.

As quoted above we found that – at least in this assignment – that using Swing and handling everything manually will give us a high level of understanding of what is going on when the app is running.

XStream and Jettison driver

This app uses these libraries for saving and loading the paint.

- XStream is a simple library to serialize objects to XML and back again, and Jettison driver uses Jettison StAX parser to read and write data in JSON format.

MVC

Design pattern

Some of the design pattern used in this app:

- Factory design pattern
- Singleton design pattern
- Delegation design pattern
- Command Design pattern
- Interface Design pattern

FEATURES

MVC Architecture

The implementation is based on the famous MVC pattern.

JNLP (Java Network Launch Protocol)

“JNLP enables an application to be launched on a client desktop by using resources that are hosted on a remote web server. Java Plug-in software and Java Web Start software are considered JNLP clients because they can launch remotely hosted applets and applications on a client desktop.” [source]

One important feature of JNLP that uses can get the last updated version directly from the server.

Multi-User

You can share your painter canvas with any one connected to the same Wi-Fi network in just 2 steps.

Dynamic extension

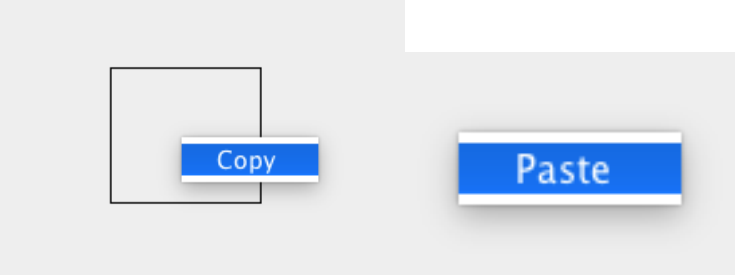
You can extends the tool bar with other shapes by implementing Shape interface.

Group Selection

You can select multiple shapes and move, resize, fill, delete them.

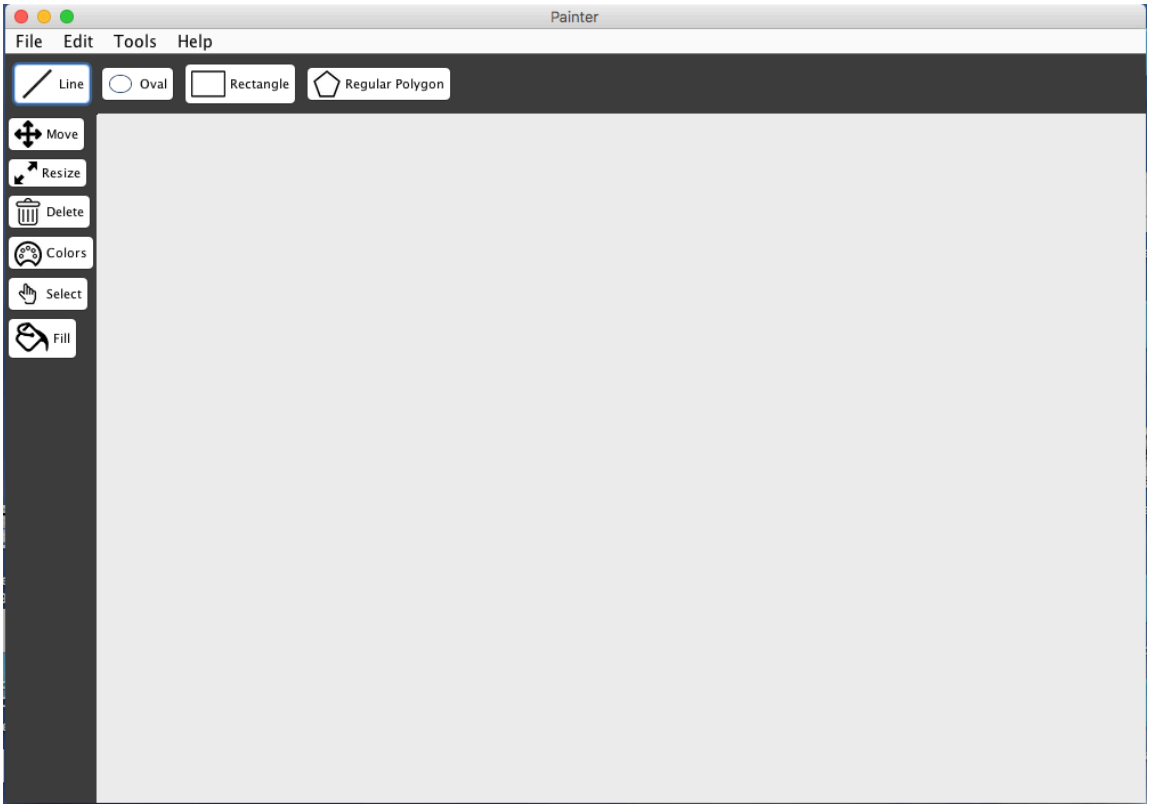
Copy / Paste the shape

You can right click the shape and copy it, then paste it any where you like.



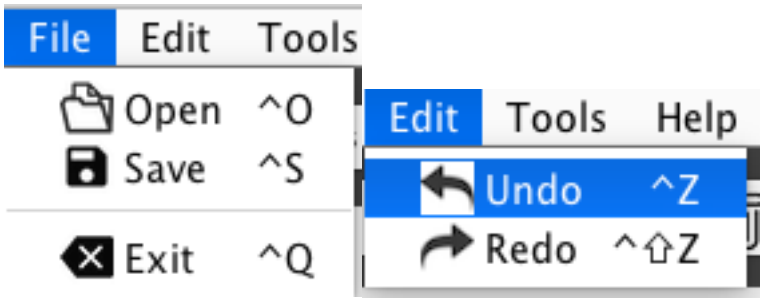
User-friendly

It's super easy to use with powerful GUI!



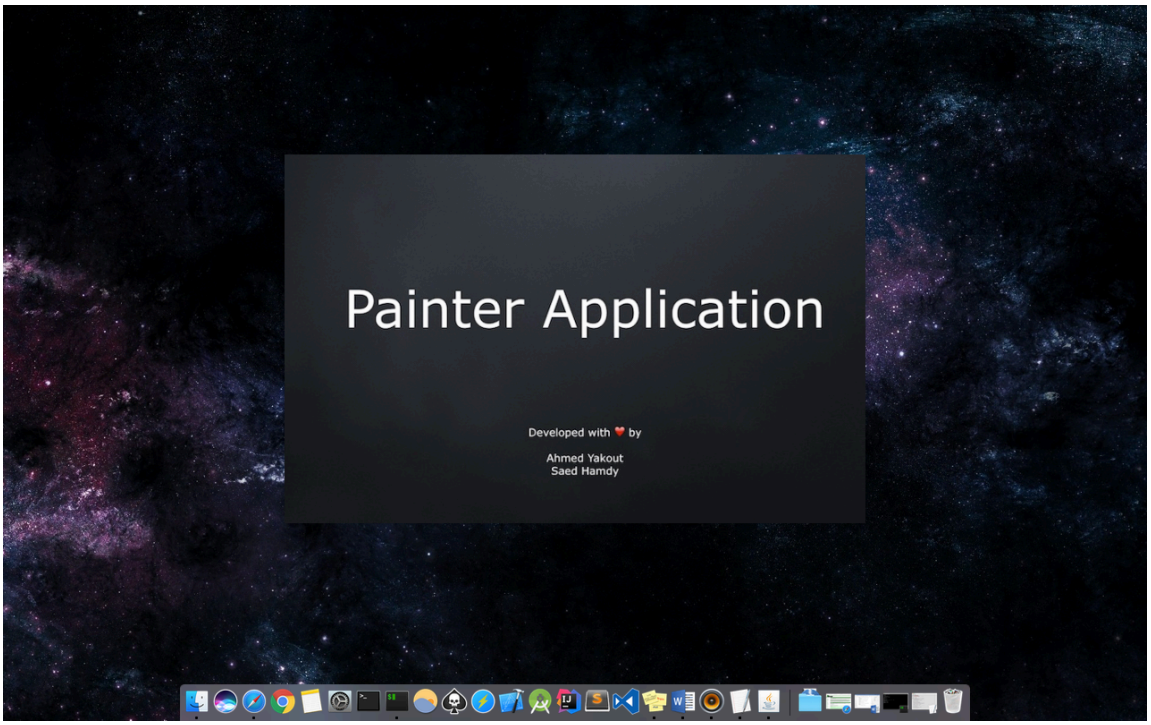
Shortcuts

Shortcuts for some operations like "undo", "redo", "open" and "save", to ease access them.



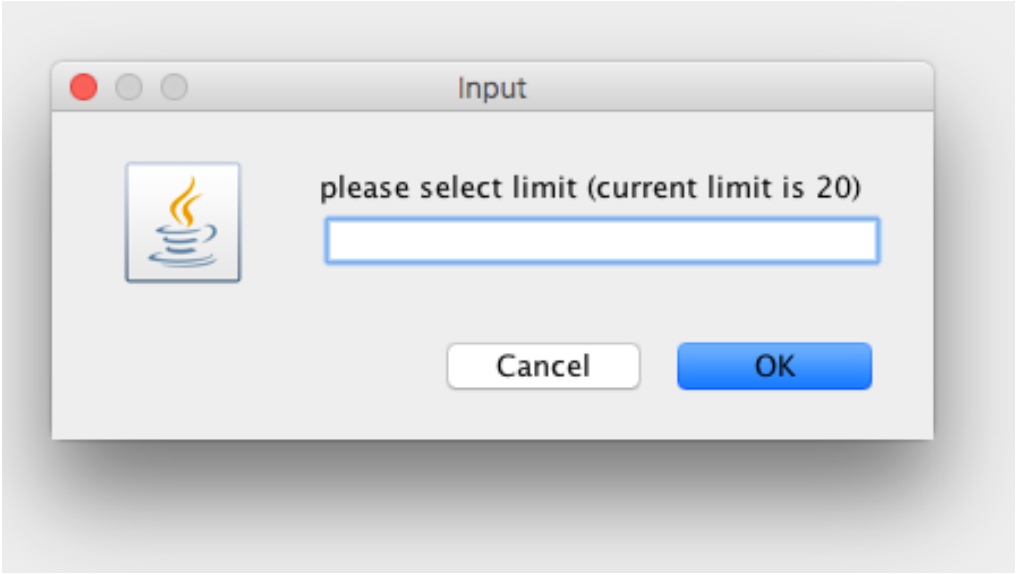
Splash Screen

Until the app is loading a beautiful Splash Screen will appear.



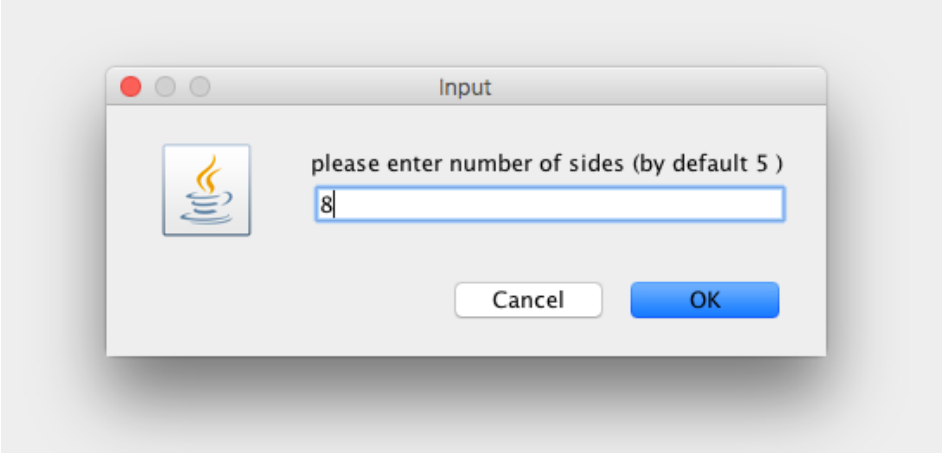
History limit

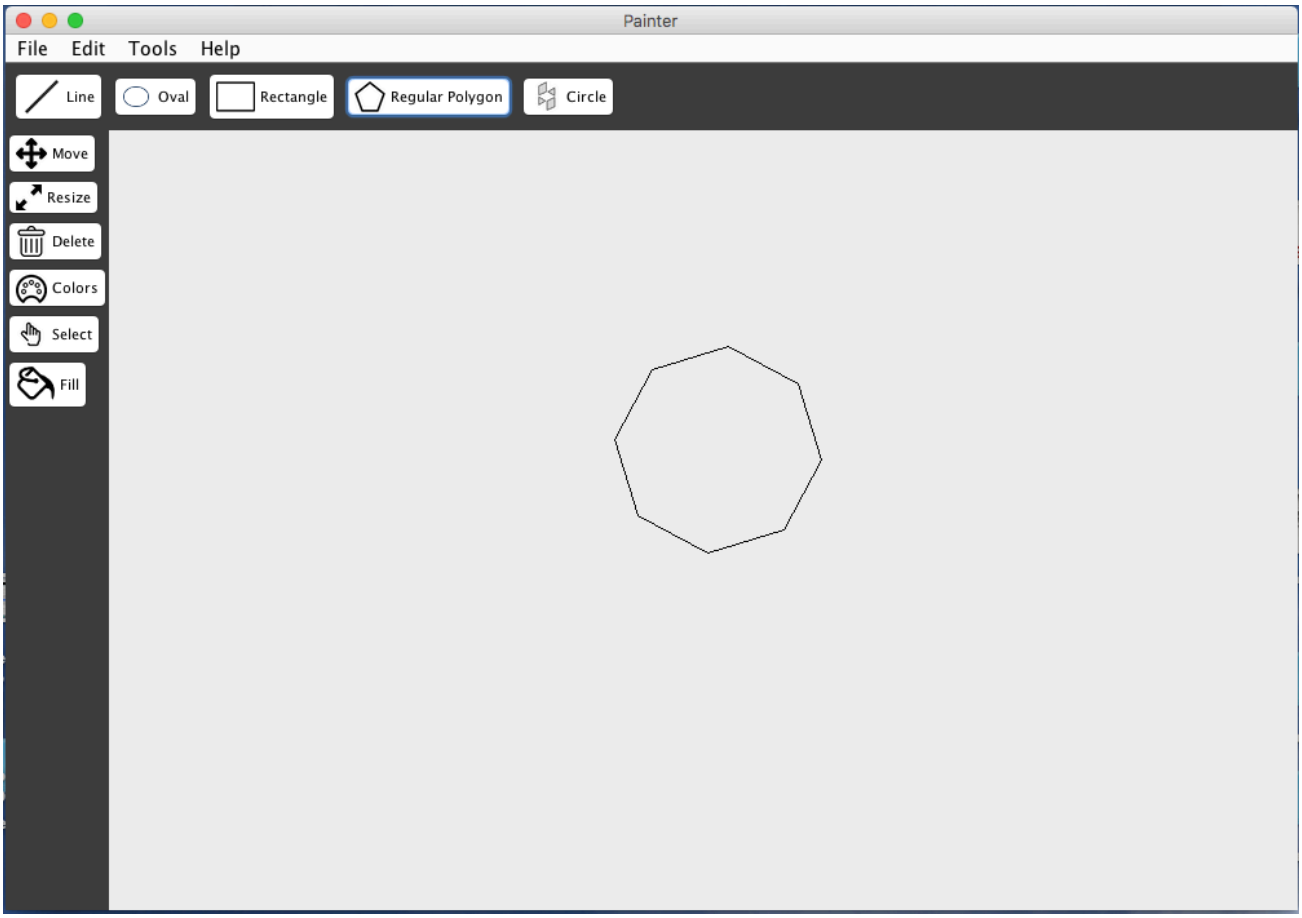
You can set the history limit to save your changes from tools menu. The default size of the history is 20.



Regular polygon

You can draw any regular polygon shape by specifying number of sides. (Default are 5 sides).





USER GUIDE

JNLP

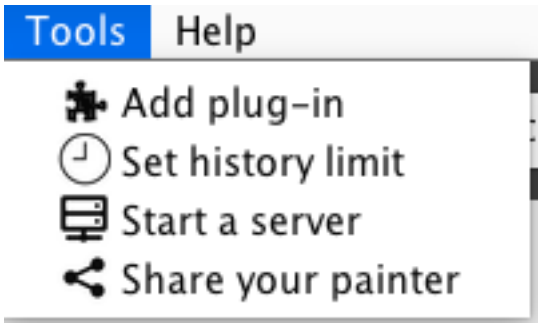
To start JNLP your have to first start a server, the easiest way to start a web server is to run this command in terminal on the same directory where JNLP files located “python –m SimpleHTTPServer [port number]” on Unix OS or “python –m http.server [port number]” on windows OS.

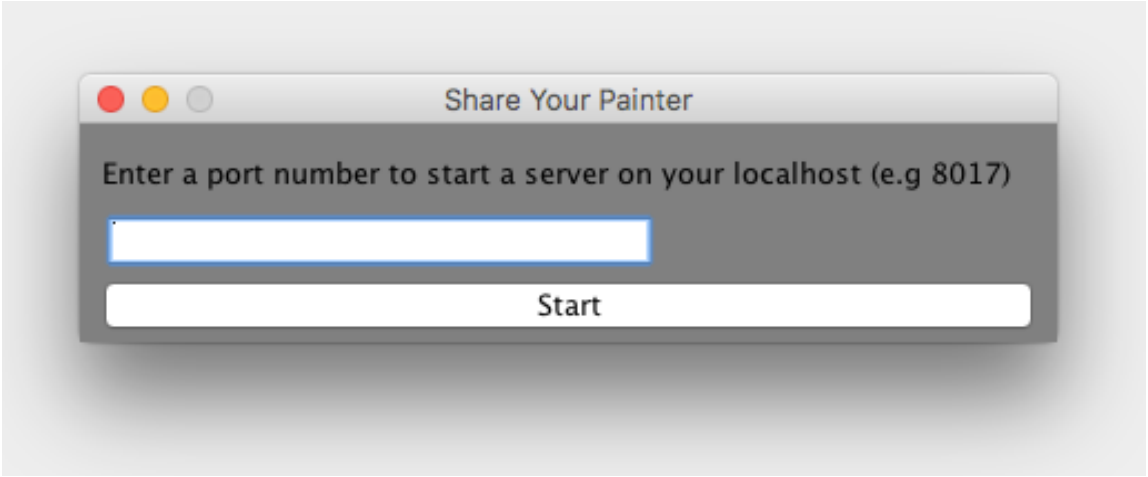
Finally you need a little change in the .jnlp file, change the codebase from codebase=”http://6b42825f.ngrok.io” to codebase="http://localhost:[port number]", then you are ready to launch the java application from the new server.

- Note that by default this app uses “ngrok” (some kind of port forwarding) to make the server available over the internet.

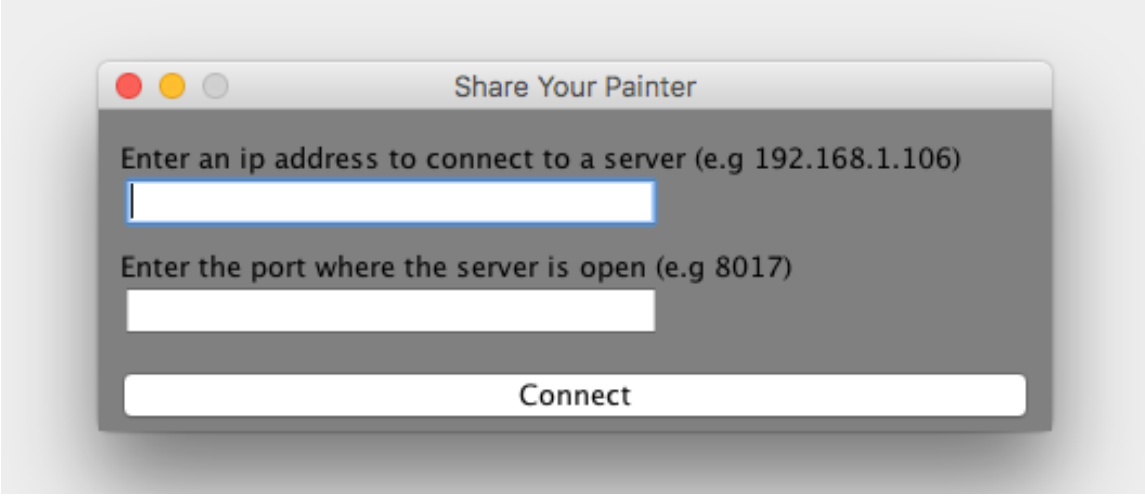
Multi-User feature

1. From the menu bar select tools -> “Start a server”, this will open a window to choose a port for your server, after clicking connect this will run a UDP sever on the port you entered.
   - The application uses UDP connection because it’s faster than TCP, although it’s more reliable to use TCP.





2 You need to connect to the server you just successfully made in the above step, from tools -> “Share your painter”, enter the address of the machine the server running on and enter the same port from step one.



For example let’s assume you have two machines connected to the same Wi-Fi network, machine A (address: 192.168.1.107) and machine B (address: 192.168.1.106), you need to start a server in any machine, and both machine to that server.

Dynamic extension

To create plug-in you have to implement the Shape interface and make sure to add your files in a package with the name “shapes” then compile your files and open the application then from tools -> “Add plug-in” which will open a file chooser window, choose the .class file of the new shape. A new button will be successfully added to the toolbar.

The Shape interface:

```
package shapes;

import java.awt.Graphics;
import java.awt.Color;
import java.io.Serializable;

/**
* @author YSteam
*/
public interface Shape extends Serializable {
void draw(Graphics g);

void setLocation(Point location);

Point getLocation();

Dimensions getDimensions();

void setDimensions(Dimensions dimensions);

boolean contain(int x, int y);

void setColor(Color color);

Color getColor();

void setFillColor(Color color);

Color getFillColor();

Object clone() throws CloneNotSupportedException;

void move(int x1, int y1, int x2, int y2);

void resize(int x1, int y1, int x2, int y2);
}
```