BTP600 Assignment 3 – Java Paint Program

Neal Paint, by: Alex Craig and Neil Guzman

System Requirements

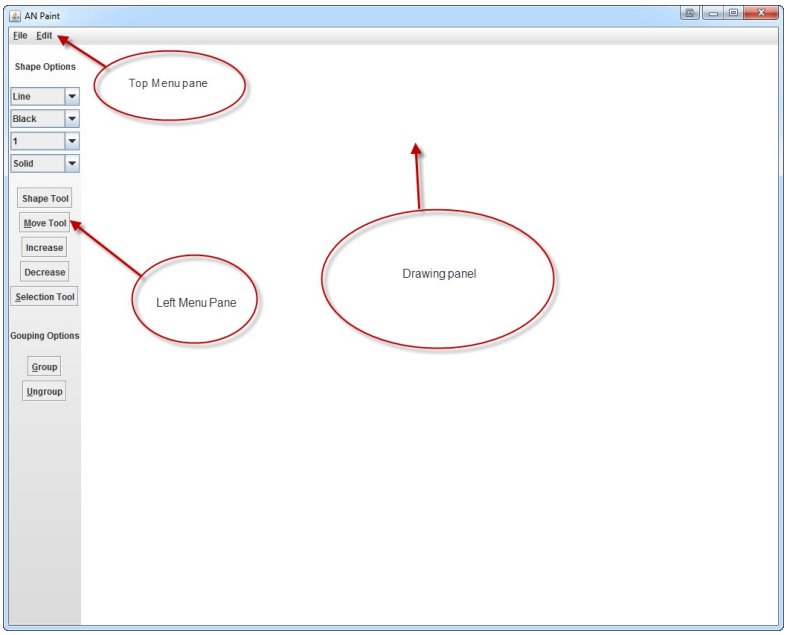
- Tested and working on Windows environment (did not try with OSX or *nix platforms)
- Latest version of the Java Runtime Environment (JRE)

Instructions

1. Run the program by:
    - Double-clicking the provided “ANPaint.jar” file, or
    - Executing the command: java -jar "ANPaint.jar" from the command line (be sure to have the correct path to “ANPaint.jar”

> *Note: If you receive a blank screen when starting up the program, resizing and/or closing and reopening the program fixes this problem. This is due to the screen resolution.*
>
> *If the program fails to load at all, please make sure you are running windows and have the latest JRE version installed.*

2. Once the program has started, feel free to select any of the commands available on the left menu bar or the top menu bar:



3. Feel free to then start testing out the program by clicking on a tool (e.g. the “Shape Tool”) and dragging from one point to another on the drawing panel.

Available Tools (shortcuts and description)

Shape

- Choose shape, color, weight, and type then drag from 1 point on the drawing panel to another point to draw shape with the specified attributes

Selection (Alt+s)

- Select the "Selection Tool" and drag the selection rectangle over shape(s) to select. Click on a blank space on the drawing panel to unselect

Move (Alt+m)

- Select shapes then choose the "Move Tool" to then click on a different point on the drawing panel to move the shape(s) there

Increase/Decrease

- Select shape(s) and then choose either command to increase or decrease the shape(s) in size

Group (Alt+g)

- Select shape(s) and then choose "Group" to group them together to be classified as a single shape

Ungroup (Alt+u)

- Select the group and then choose "Ungroup" to split up the group into their separate shapes

Save (Ctrl+s)

- Save whatever is on the drawing panel to a custom ".nap" file

Open (Ctrl+o)

- Specifically choose which ".nap" file to load onto the drawing panel

Exit (Ctrl+e)

- Exit the program

Copy (Ctrl+c)

- Selected shape(s) will be copied into a buffer

Paste (Ctrl+v)

- If there is something in the buffer, the shape(s) will be pasted onto the top-left corner of the drawing panel

Undo (Ctrl+z)

- Undo last command that was done

Redo (Ctrl+y)

- Redo last command that was undone

Patterns Implemented

Singleton

Class: AppWindow
Reason: This program only needs 1 instance of the drawing application running. AppWindow would contain a static instance of itself and a private constructor to make sure that only 1 instance is up during the lifetime of the program.

Abstract Factory

Package: Creators
Reason: We have a family of Shapes of related objects (Circle, Line, Rectangle, etc.). We decouple the creation of the classes so that the creator methods can call their specific product to create. The constructed object is returned to whatever called it.

Command

Packages: Commands, Invokers
Reason: We have a bunch of commands that work by having the invoking class work with the ActionListeners so that when a menu command or shortcut has been executed, the listener will invoke a call to a command corresponding to the command/shortcut used. The drawing panel acts as a receiver that contains all the implementations of the commands the current states of the drawing panel.

Composite

Package: BasicShapes
Reason: Mainly used for the Group class which can contain more groups and/or more shapes. The Shapes (Circle, Line, Rectangle, etc.) acts as leaves while the Group acts as a composite.

Template

Class: AppWindow
Reason: As the application is first initialized, the AppWindow constructor will call the several methods used to construct the application window. This will prevent duplication of code and will make it easy where everything is called to be constructed.