



Vector Based Drawing Application

Flying Colors

35- Salma Yehia

79- Nada Ayman

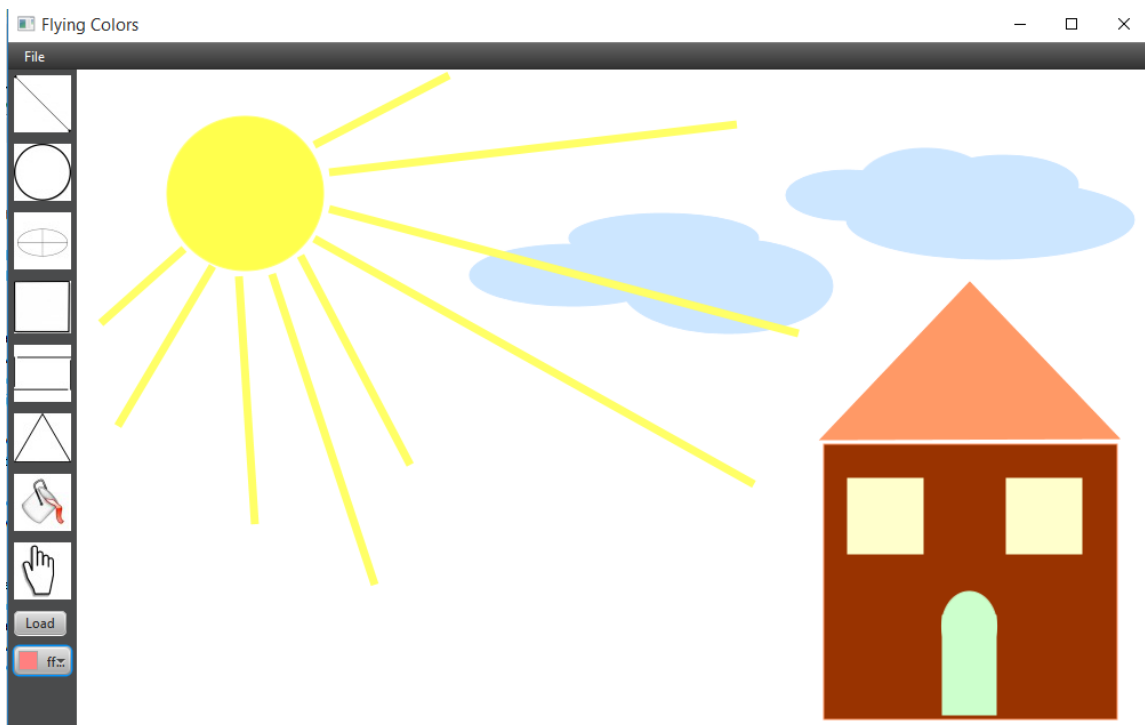
Flying Colors

A vector based drawing desktop application developed in **Java** applying all OOP principals (inheritance, polymorphism, encapsulation, etc.)

The GUI is implemented using **JavaFX**.

User Guide

The User Interface consists of three areas:



1. **Drawing area** : Where the user can draw and interact with shapes.
2. **Tools area** : Where the user can find all the supported tools in the program
 - **Line**: draws a line by clicking, dragging then releasing the mouse.
 - **Circle**: draws a circle by specifying the center, dragging to the desired radius then releasing the mouse.
 - **Oval**: draws an oval / ellipse by dragging the ellipse from its corner then releasing the mouse.

- **Triangle:** draws a triangle by specifying two points, the third point is assumed and the triangle appears, with the ability to modify that third point by dragging it.
- **Rectangle:** draws a rectangle by clicking to specify its corner then dragging then releasing the mouse when the desired size is reached.
- **Square:** draws a square by clicking to specify its corner then dragging then releasing the mouse when the desired size is reached.
- **Fill tool:** to be selected to choose the color to fill a closed shape.
- **Select tool:** used to disable drawing and start interacting with the shapes move, resize or color them it also deselects all the selected shapes.
- **Color picker:** pick colors.
- **Class loading button:** to load shapes at run time (Circle – Square).

3. Menu bar:

Has one menu “File” that has three options “New”, “Save” and “Open File”.

- New: the user is asked whether he wants to save the current file or not then a new file is opened.
- Save: save the current file giving the user to options (JSON file – XML file), the user can choose directory in which the file should be saved.
- Open file: Load an existing file (JSON – XML) and enable the user to modify it.

Keyboard:

Delete : Delete all selected shapes.

U : Undo.

R : Redo.

The Design

Package: “eg.edu.alexu.csd.oop.paint”: The main package that contains the whole project.

The Shapes Hierarchy:

All shapes extends one class called “TheShape”

TheShape Class:

Every shape has some common properties eg.

boolean `selected`, to denote whether the shape is selected or not.

Color `boarderColor`, the color of the boarder.

double `points[]`, array of doubles contains the coordinates of the shape
`points(Anchors)`.

Anchor `shapeAnchors[]`, array of anchors to enable the user to resize the shape.

Every shape has some common methods to manage the interaction with the shapes using the mouse, eg.

`(MousePressed - MouseDrag - MouseClick - MouseEntered -
MouseReleased)`

`Delete`

`SetEverything`

Some Abstract methods implemented for every shape

Create the anchors to resize the shape

```
public abstract void createControlAnchorsFor();
```

```
public abstract void moveWithAnchors();
```

```
public abstract void setCorners();
```

```
public abstract void updateAnchors(Group root);
```

LineSegmant Class extends TheShape Class

ClosedShap Class extends TheShape Class

public void setStyle(Color color) , To fill the shape with the chosen color.

Oval class extends ClosedShap Class:

CircleShape class extends Oval class:

Can be dynamically loaded at run time.

RectangleShape class extends ClosedShap:

Square class extends RectangleShape:

Can be dynamically loaded at run time.

TriangleShape class extends ClosedShap:

The Actions Hierarchy:

These classes are used to save every modification to maintain a stack of actions so the user can undo and redo any number of times.

All Action extends one class called "Action"

Classes extend the "Action" Class DeleteAction – DrawAction – ResizeAction - MoveAction – ColorAction.

The "Paint" Class extends Application:

Contains the GUI code in its start method.

The "Anchor" Class: controls the anchors appearance and actions

The "XML" Class:

for saving and loading XML files using **DOM** library.

The "JSON" Class:

For saving and loading JSON files using **json.simple** library.

The "MenuControl" Class:

Has the methods that are called when interacting with the menu bar

Style Sheet "Style.CSS":

Styling the user interface.

Some Design Decisions

- Stacks is used to implement (undo / redo) feature, making use of its FIFO property.
- No polygon class is needed, as it won't help reduce the code of ours.
- Linked list data structure is used to save all the shapes drawn – dynamic size -
- The circle and the square are dynamically imported.

UML Diagram

Created using “ObjectAid UM Explorer”

