

CMPE 496 PROJECT 1 - DRAWING EDITOR

Çağrı Gülbeycan - 2020400369

July 10, 2023

Contents

1	Introduction	2
2	How to Run the Code	2
2.1	To Run the Code with Command Line	2
3	Functionality	2
3.1	Creating Shapes	2
3.2	Resizing Shapes	2
3.3	Moving Shapes	2
3.4	Deleting Shapes	2
3.5	Changing Colors	3
3.6	Saving the Drawing	3
3.7	Removing the Last Item	3
3.8	Shortcuts	3
4	Images	4
5	Conclusion	4

1 Introduction

This report presents an overview and analysis of the code for a drawing application developed using PyQt5. The application allows users to draw various shapes such as circles, rectangles, lines, and hexagons on a graphical canvas. It provides features such as changing the brush color, setting the pen size, clearing the canvas, resizing shapes, moving shapes, deleting shapes, and more.

2 How to Run the Code

2.1 To Run the Code with Command Line

Run the command in the directory where "DrawingEditor.py" is located.

```
python3 DrawingEditor.py
```

3 Functionality

3.1 Creating Shapes

The application provides a combo box to select the shape to be drawn (circle, rectangle, line, or hexagon). When the user right-clicks on the canvas, the selected shape is created at the clicked position using the specified brush color and pen size. The shapes are added to the scene and can be moved, resized, and selected.

3.2 Resizing Shapes

Shapes can be resized by clicking and dragging their boundaries. For circles and rectangles, the size is adjusted based on the distance between the mouse position and the shape's center. For hexagons, the size is calculated by determining the distance between the mouse position and the shape's position and then recalculating the vertices of the hexagon accordingly.

3.3 Moving Shapes

Shapes can be moved by clicking and dragging them to a new position on the canvas. To move a shape, left-click on the shape and drag it to the desired location.

3.4 Deleting Shapes

Shapes can be deleted by selecting them and pressing the Delete key on the keyboard. To delete a shape, select it by left-clicking on it, and then press the Delete key.

3.5 Changing Colors

The application provides options to change the brush color and the pen color. These options open color dialogs that allow the user to select a desired color. The selected color is then applied to the inner parts or boundaries of the shapes as per the user's choice.

3.6 Saving the Drawing

One can save the drawing in PNG or in JPEG format. Users can click on "Save Image" in the file menu or can press `ctrl+S` in order to save the drawing as an image file.

3.7 Removing the Last Item

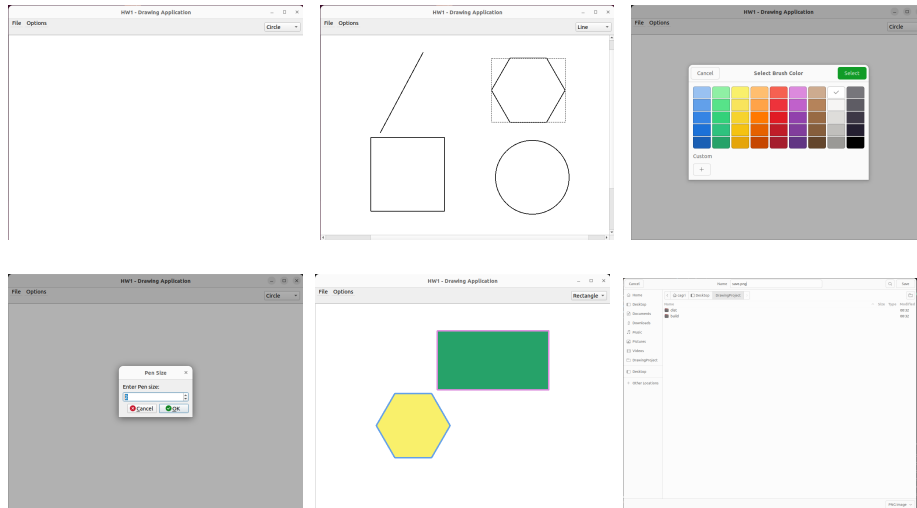
The last added shape can be removed by pressing `ctrl+Z` or using menubar.

3.8 Shortcuts

The following shortcuts are available in the drawing application:

- **Ctrl+N**: Clear the canvas.
- **Ctrl+Z**: Remove the last shape added.
- **Ctrl+D**: Paint the selected shapes with the brush color.
- **Ctrl++**: Increase the size of the selected shapes.
- **Ctrl+-**: Decrease the size of the selected shapes.
- **Ctrl+Esc**: Exit the application.
- **+**: Increase the pen size.
- **-**: Decrease the pen size.
- **Shift++**: Increase the boundary size of selected shapes.
- **Shift+-**: Decrease the boundary size of selected shapes.
- **Delete**: Delete the selected shape(s).
- **Ctrl+S**: Save the file as an image.

4 Images



5 Conclusion

The drawing application developed using PyQt5 provides a user-friendly interface for creating and manipulating shapes on a canvas. The code implementation covers a range of features, including shape creation, resizing, moving, deleting, color customization, and various shape manipulation operations.

The application is tried to be developed considering the aspects of usability engineering. Firstly, I tried to develop an intuitive and easy-to-understand interface, allowing users to quickly grasp the available functionalities. The combo box for selecting shapes, color dialogs for changing colors, and keyboard shortcuts for common operations contribute to a streamlined user experience.

The application also provides visual feedback and interactive elements that enhance usability. When creating shapes, users can immediately see the shape being drawn on the canvas. Resizing shapes through click-and-drag interaction and moving shapes by dragging them offer natural and expected behaviors.

Furthermore, the availability of keyboard shortcuts for frequently used actions improves efficiency and enables power users to perform tasks more quickly. The clear labeling of the menu options and the use of standard conventions (such as using the Delete key to delete shapes) also contribute to the overall usability of the application.

In conclusion, the drawing application serves as a successful case study in applying usability engineering principles to create a user-friendly software system.