

CMPE 496 HUMAN COMPUTER INTERACTION

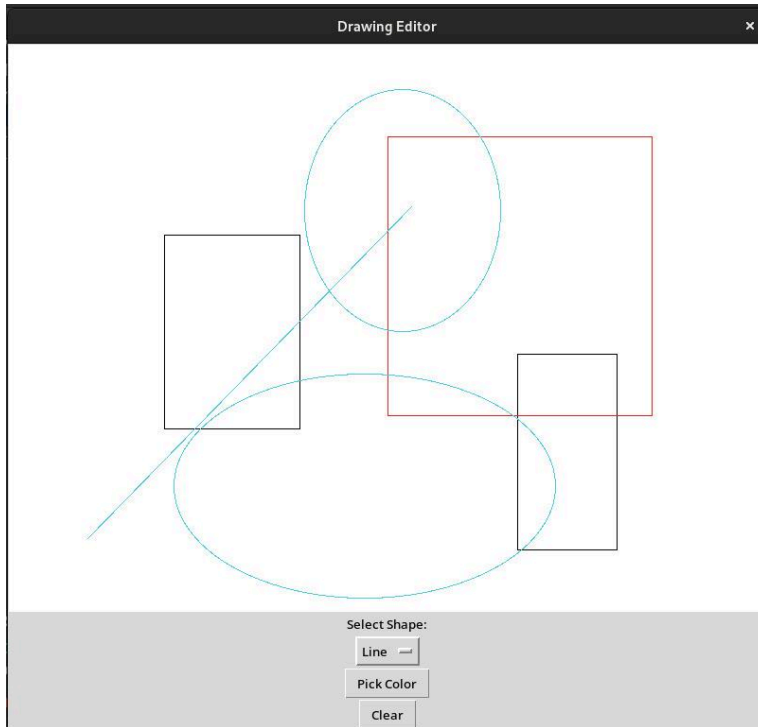
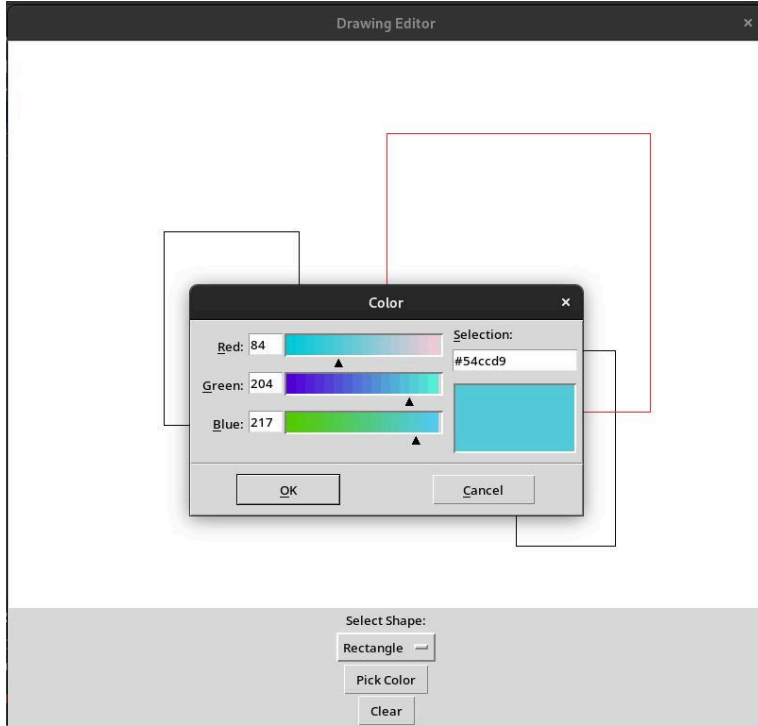
Homework #1 Graphic editor

Portfolio Summary Of This Design

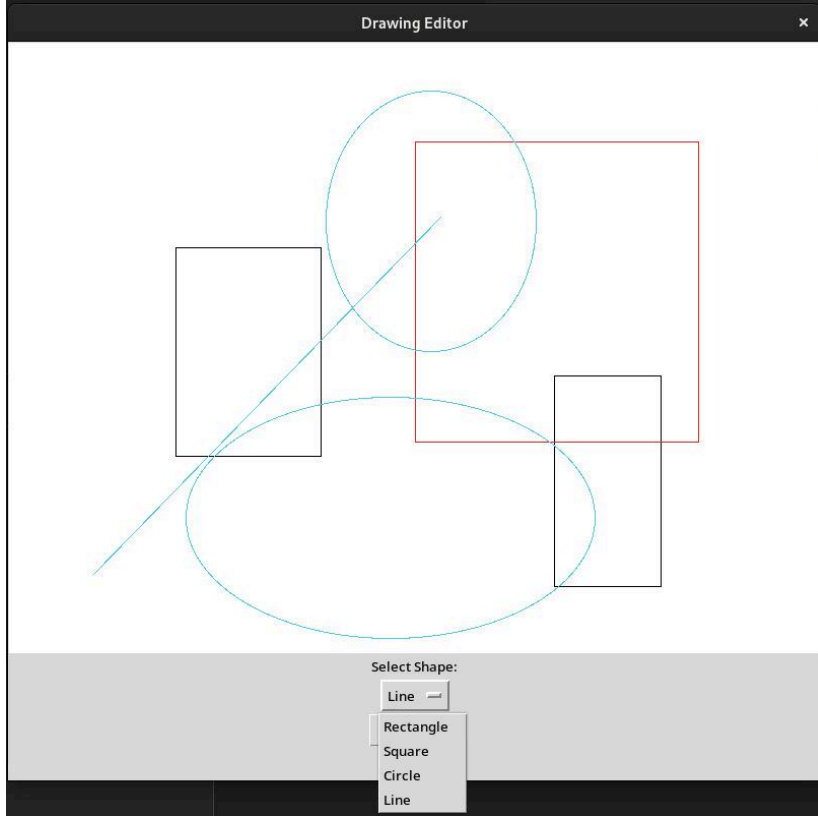
When the program is run, a graphical screen(GUI) called `DrawingEditor` opens. It provides a graphical user interface for drawing and editing various shapes. While there is a drawing section on the screen, 2 buttons, 1 label and 1 option menu are placed at the bottom. While drawing can be done with the left mouse button, the right button allows the drawn objects to be moved on the screen.

- There are two buttons available: one for picking colors and the other for clearing the canvas.
- An OptionMenu allows users to select the type of shape they want to draw, including rectangles, squares, circles, and lines.
- Users can interact with the canvas using both left and right mouse clicks. Left-clicking enables drawing shapes, while right-clicking initiates the process of moving existing shapes.
- Shapes are drawn with the currently selected color, which users can choose by clicking the "Pick Color" button and selecting from a color picker dialog.
- To move a shape, users can initiate the process by right-clicking on the shape they wish to move. This action selects the shape for movement.
- To clear a shape, users can initiate the process by middle-clicking (mouse middle clicking) on the shape (nearest shape) they wish to clear. The nearest object is deleted from the editor when the middle-button is pressed.
- With the ability to draw various shapes and select colors dynamically, users can express their creativity and design visually appealing compositions.
- The application is designed with simplicity and usability in mind, making it accessible to users of all skill levels.

ALİ ÖZDEN
YİĞİT SARIOĞLU
AHMET EREN ÖZTÜRKMEN



ALİ ÖZDEN
YİĞİT SARIOĞLU
AHMET EREN ÖZTÜRKMEN



API Documentation for DrawingEditor Control

The `DrawingEditor` control provides a graphical user interface for drawing and editing various shapes on a canvas using the Tkinter library in Python. Below is the API documentation describing the functionalities and methods available for developers to interact with the `DrawingEditor` control.

Class: DrawingEditor

Constructor: `__init__(master)`

- Parameters:
 - `master`: The Tkinter master window where the `DrawingEditor` control will be placed.
- Description: Initializes the `DrawingEditor` control with a Tkinter master window, sets up canvas for drawing, and binds mouse events for drawing, editing, and deleting shapes.

Methods:

`start_draw(event)`

- Parameters:
 - `event`: The mouse event when the drawing action starts.
- Description: Initiates the drawing action by capturing the initial coordinates of the mouse click.

`draw(event)`

- Parameters:
 - `event`: The mouse event during drawing action.
- Description: Draws the shape on the canvas based on the mouse motion.

`end_draw(event)`

- Parameters:
 - `event`: The mouse event when the drawing action ends.
- Description: Finalizes the drawing action, adds the drawn shape to the list of shapes, and resets drawing parameters.

delete_shape(event)

- Parameters:
 - `event`: The mouse event for deleting a shape.
- Description: Deletes a shape from the canvas when clicked.

delete_all()

- Description: Deletes all shapes from the canvas.

start_move(event)

- Parameters:
 - `event`: The mouse event when the moving action starts.
- Description: Initiates the moving action by capturing the initial coordinates of the mouse click and finding the shape to be moved.

move_shape(event)

- Parameters:
 - `event`: The mouse event during the moving action.
- Description: Moves the shape on the canvas based on the mouse motion.

end_move(event)

- Parameters:
 - `event`: The mouse event when the moving action ends.
- Description: Ends the moving action.

pick_color()

- Description: Opens a color picker dialog to choose a color for the next drawn shape.

Main Functionality

The `DrawingEditor` control allows users to:

ALİ ÖZDEN

YİĞİT SARIOĞLU

AHMET EREN ÖZTÜRKMEN

- Draw rectangles, squares, circles, and lines on the canvas.
- Pick colors for the shapes.
- Delete individual shapes by clicking on them.
- Delete all shapes on the canvas.
- Move shapes by clicking and dragging them.

Integration

To integrate the `DrawingEditor` control into your application:

1. Create a Tkinter master window.
2. Instantiate the `DrawingEditor` class with the master window.
3. Customize the UI elements as needed, such as shape selection, color picking, and clearing the canvas.
4. Run the Tkinter main event loop using `root.mainloop()`.

By following the provided API documentation, effectively utilize and extend the functionality of the `DrawingEditor` control.

How to Run

1- You could run the executable (exe file) by downloading from the cloud

<https://drive.google.com/file/d/1voVxHmAq2kjRidejw1FMiBb5XHeUFi0g/view?usp=sharing>

2 - You could run the program by executing this command in the terminal window:

-> python main.py