Ex. No. : 7 Date: 08-11-2025

Register No.: 241701003 Name: AISHWARYA B

Create a Simple Graphics Application for Drawing Basic Shapes Using Python Tkinter

AIM:

Create a Simple Graphics Application to Draw Basic Shapes with Color Selection Using Tkinter

- □ **Drawing basic shapes**
- □ **Color selection**
- □ **Canvas clearing**

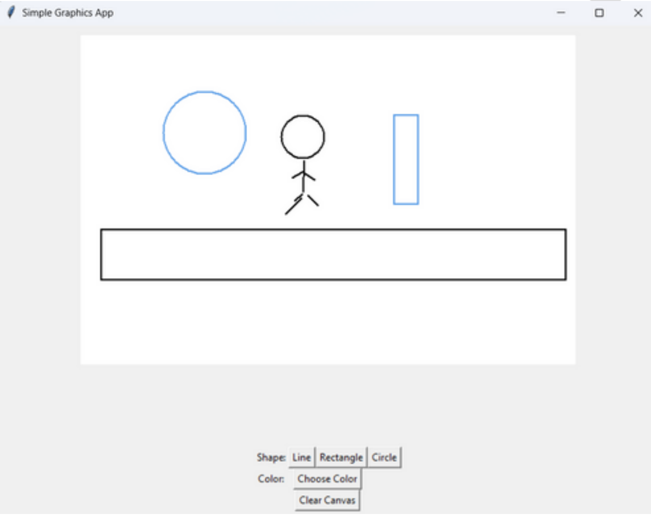
into a unified user interface using Python and suitable libraries.

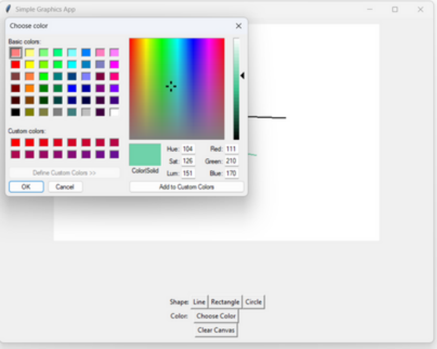
Procedure:

1. Use a Python GUI framework: **tkinter**.
2. Create a canvas widget to draw shapes and buttons for selecting shapes (line, rectangle, circle), choosing colors, and clearing the canvas.
3. Use mouse events to capture start and end points for drawing shapes on the canvas
4. Implement functions to set shapes, pick colors using a color chooser, and clear the drawing area.
5. Run the application main loop to enable user interaction with the graphics interface

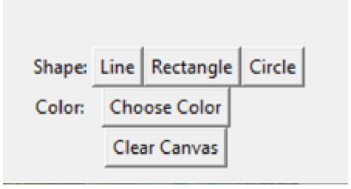
Program:

```
import tkinter as tk
from tkinter import colorchooser

class SimpleGraphicsApp:
  def __init__(self, root):
    self.root = root
    self.root.title("Simple Graphics App")
    self.root.geometry("800x600")


    self.shape = "line"
    self.color = "black"
    self.start_x = None
    self.start_y = None
self.canvas = tk.Canvas(root, bg="white", width=600, height=400)
        self.canvas.pack(side=tk.TOP, padx=10, pady=10)


        control_frame = tk.Frame(root)
        control_frame.pack(side=tk.BOTTOM, pady=10)


        tk.Label(control_frame, text="Shape:").grid(row=0, column=0)
        tk.Button(control_frame, text="Line", command=lambda: self.set_shape("line")).grid(row=0, column=1)
        tk.Button(control_frame, text="Rectangle", command=lambda: self.set_shape("rectangle")).grid(row=0, column=2)
        tk.Button(control_frame, text="Circle", command=lambda: self.set_shape("circle")).grid(row=0, column=3)


        tk.Label(control_frame, text="Color:").grid(row=1, column=0)
        tk.Button(control_frame, text="Choose Color", command=self.choose_color).grid(row=1, column=1, columnspan=2)


        tk.Button(control_frame, text="Clear Canvas", command=self.clear_canvas).grid(row=2, column=0, columnspan=4)


        self.canvas.bind("<ButtonPress-1>", self.on_mouse_press)
        self.canvas.bind("<ButtonRelease-1>", self.on_mouse_release)

    def set_shape(self, shape):
        self.shape = shape

    def choose_color(self):
        color = colorchooser.askcolor(title="Choose color")[1]
        if color:
            self.color = color
    def clear_canvas(self):
        self.canvas.delete("all")
    def on_mouse_press(self, event):
        self.start_x = event.x
        self.start_y = event.y
    def on_mouse_release(self, event):
        end_x = event.x
        end_y = event.y
        if self.shape == "line":
            self.canvas.create_line(self.start_x, self.start_y, end_x, end_y, fill=self.color, width=2)
        elif self.shape == "rectangle":
            self.canvas.create_rectangle(self.start_x, self.start_y, end_x, end_y, outline=self.color, width=2)
        elif self.shape == "circle":
radius = ((end_x - self.start_x)**2 + (end_y - self.start_y)**2)**0.5
            self.canvas.create_oval(self.start_x - radius, self.start_y - radius,
                                    self.start_x + radius, self.start_y + radius,
                                    outline=self.color, width=2)

if __name__ == "__main__":
    root = tk.Tk()
    app = SimpleGraphicsApp(root)
    root.mainloop()t
```





Hue: 104 Red: 111
Sat: 126 Green: 210
Lum: 151 Blue: 170



Result:

The simple graphics application using Tkinter allows users to draw lines, rectangles, and circles with selectable colors and clear the canvas successfully.