

SCOPE AND OBJECTIVE:

Scope:

The program provides a basic paint application with a graphical user interface (GUI). It allows users to draw on a canvas using a mouse. Users can select different colors, adjust brush sizes, and save their creations. The program aims to provide a simple and intuitive drawing experience.

Adjectives:

1. **Basic:** The program offers fundamental drawing functionalities without complex features.
2. **Intuitive:** The user interface and interactions are designed to be easy to understand and use.
3. **Customizable:** Users can choose different colors and adjust the brush size to suit their preferences.
4. **Efficient:** The program allows users to create drawings quickly and easily.
5. **Interactive:** The application responds to user input in real-time, providing immediate feedback.
6. **User-Friendly:** The GUI is designed with the user's experience in mind, making it accessible and easy to navigate.
7. **Versatile:** While it's a simple paint program, it provides enough options (colors, sizes, saving) to cater to a range of user preferences.

DESCRIPTION

The Simple Paint Program is a basic drawing application that provides users with an intuitive platform to create digital drawings and designs. It offers a range of essential features for drawing and painting, allowing users to unleash their creativity. To develop a user-friendly paint program with a simple and intuitive interface. To provide basic drawing tools such as selecting colors, adjusting brush sizes, and saving the canvas. To create a versatile platform for users to express their creativity through digital art.

Features:

1. Drawing Canvas:

The program offers a resizable drawing canvas where users can freely create their digital artwork.

2. Brush Tools:

Users can draw using a customizable brush tool, with options to adjust the size for finer or broader strokes.

3. Color Selection:

The application allows users to choose from a wide range of colors through a color picker tool.

4. Saving Artwork:

Users can save their creations as PNG files to their local storage for future access.

5. Menu Options:

The program includes a menu with options for starting a new canvas, saving artwork, choosing colors, and adjusting brush sizes.

User Interface:

1. Canvas Area:

The central area of the application is dedicated to the drawing canvas, providing ample space for creativity.

2. Toolbars:

There are toolbars for selecting colors, adjusting brush size, and accessing other features like saving and starting a new canvas.

3. Menus:

Menus provide additional options for performing actions like saving artwork, changing colors, and adjusting brush sizes.

Technologies Used:

1. **Programming Language:** Python
2. **GUI Library:** tkinter
3. **Color Picker:** tkinter.colorchooser module

Future Enhancements:

1. **Undo/Redo Functionality:** Implement the ability to undo and redo actions.
2. **Shapes and Stamps:** Add predefined shapes and stamps for users to incorporate into their artwork.
3. **Advanced Tools:** Introduce more advanced drawing tools like gradients, patterns, and filters.
4. **Layer Support:** Enable users to work with multiple layers for complex compositions.

IMPLEMENTATION

```
import tkinter as tk
from tkinter.colorchooser import askcolor

class SimplePaint:
    def __init__(self, root):
        self.root = root
        self.root.title("Simple Paint Program")
        self.root.geometry("800x600")

        self.canvas = tk.Canvas(root, bg="white", width=800, height=600)
        self.canvas.pack(expand=tk.YES, fill=tk.BOTH)

        self.color = "black"
        self.line_width = 2
        self.old_x = None
        self.old_y = None

        self.canvas.bind("<B1-Motion>", self.paint)
        self.canvas.bind("<ButtonRelease-1>", self.reset)

        self.menu = tk.Menu(root)
        root.config(menu=self.menu)

        self.file_menu = tk.Menu(self.menu)
        self.menu.add_cascade(label="File", menu=self.file_menu)
        self.file_menu.add_command(label="New", command=self.new_canvas)
        self.file_menu.add_command(label="Save", command=self.save_canvas)

        self.color_menu = tk.Menu(self.menu)
        self.menu.add_cascade(label="Color", menu=self.color_menu)
        self.color_menu.add_command(label="Choose Color", command=self.choose_color)

        self.size_menu = tk.Menu(self.menu)
        self.menu.add_cascade(label="Size", menu=self.size_menu)
        self.size_menu.add_command(label="Increase Size", command=self.increase_size)
        self.size_menu.add_command(label="Decrease Size", command=self.decrease_size)

    def paint(self, event):
        x, y = event.x, event.y
        if self.old_x and self.old_y:
```

```

        self.canvas.create_line((self.old_x, self.old_y, x, y), width=self.line_width,
fill=self.color, capstyle=tk.ROUND, smooth=tk.TRUE, splinesteps=36)
        self.old_x = x
        self.old_y = y

def reset(self, event):
    self.old_x = None
    self.old_y = None

def choose_color(self):
    color = askcolor(color=self.color)[1]
    if color:
        self.color = color

def increase_size(self):
    self.line_width += 1

def decrease_size(self):
    if self.line_width > 1:
        self.line_width -= 1

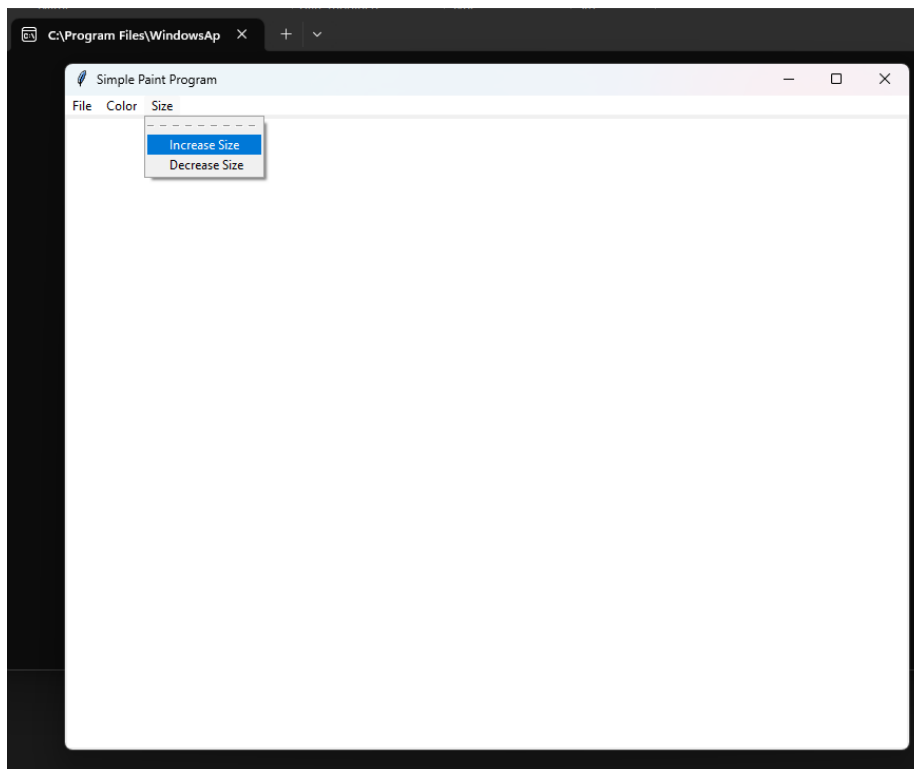
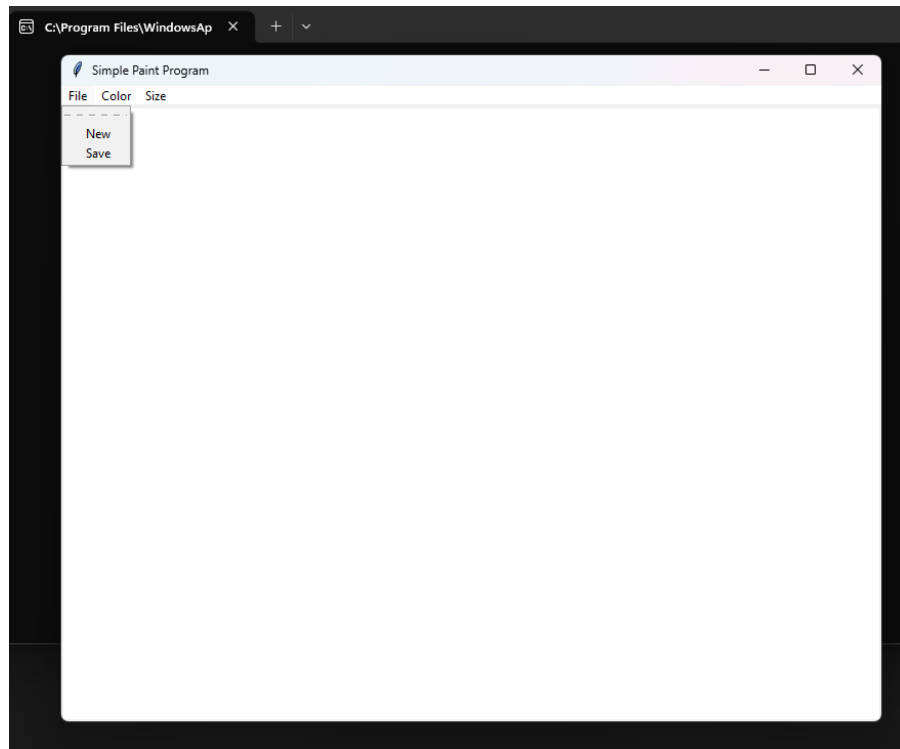
def new_canvas(self):
    self.canvas.delete("all")

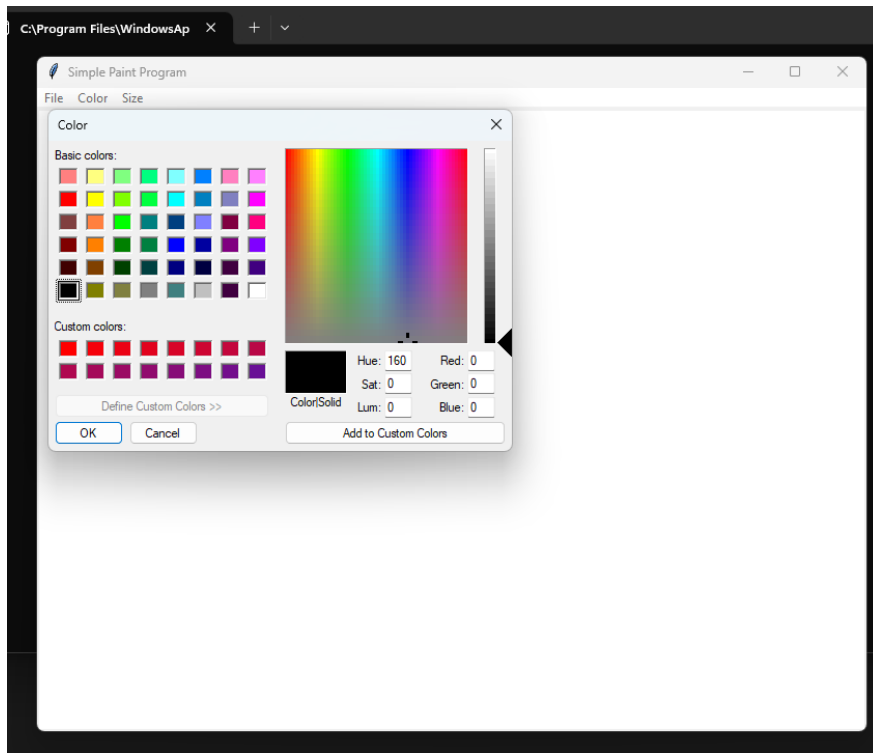
def save_canvas(self):
    file_path = tk.filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG
files", "*.png")])
    if file_path:
        self.canvas.postscript(file=file_path, colormode="color")

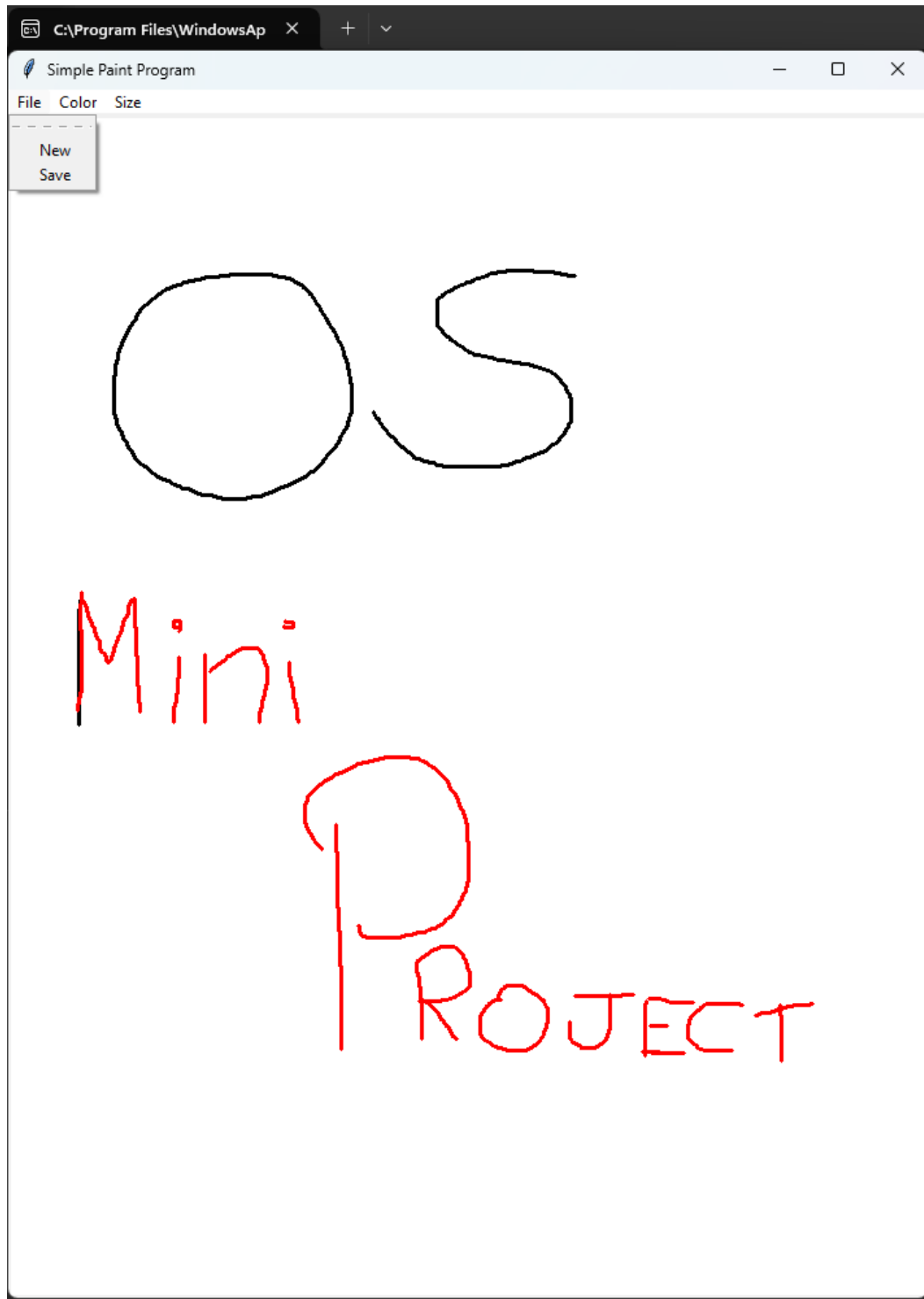
if __name__ == "__main__":
    root = tk.Tk()
    paint_app = SimplePaint(root)
    root.mainloop()

```

Output:





**Result:**

The Simple Paint Program provides a straightforward yet effective platform for digital drawing. With its intuitive interface and essential features, it caters to both beginners and casual artists looking for a versatile drawing tool. This project lays the foundation for potential future expansions and improvements to enhance the user experience even further.