**Digital White Board using Python Tkinter**

**Download the free source code from my website, PanjwaniSoftwares.com.**

Here's an explanation of each line of the code:

**from tkinter import \***

This line imports the Tkinter module, which provides the functionality to create graphical user interfaces in Python.

**root = Tk()**

**root.title("Digital Whiteboard")**

These lines create the main window of the application and set its title to "Digital Whiteboard".

**canvas = Canvas(root, width=800, height=600, bg="white")**

**canvas.pack()**

These lines create a canvas widget inside the main window with a width of 800 pixels, height of 600 pixels, and a white background color. The canvas is then packed into the main window, causing it to fill the available space.

**start\_x = None**

**start\_y = None**

These lines initialize the variables start\_x and start\_y to None. These variables will store the starting position of the mouse when drawing on the canvas.

**current\_color = "black"**

This line initializes the variable current\_color to "black". This variable will store the currently selected color for drawing on the canvas.

**def on\_button\_press(event):**

**global start\_x, start\_y**

**start\_x = event.x**

**start\_y = event.y**

This function is called when a mouse button is pressed. It updates the start\_x and start\_y variables with the current mouse position.

**def on\_move(event):**

**global start\_x, start\_y**

**if start\_x and start\_y:**

**canvas.create\_line(start\_x, start\_y, event.x, event.y, fill=current\_color, width=2)**

**start\_x = event.x**

**start\_y = event.y**

This function is called when the mouse is moved while a button is pressed. It draws a line on the canvas from the previous mouse position (start\_x, start\_y) to the current mouse position (event.x, event.y) using the currently selected color and a line width of 2 pixels.

**def on\_button\_release(event):**

**global start\_x, start\_y**

**start\_x = None**

**start\_y = None**

This function is called when a mouse button is released. It resets the start\_x and start\_y variables to None, indicating that drawing has stopped.

**def change\_color(new\_color):**

**global current\_color**

**current\_color = new\_color**

This function is called when a color button is clicked. It updates the current\_color variable with the newly selected color.

**def clear\_canvas():**

**canvas.delete("all")**

This function is called when the clear button is clicked. It clears the entire canvas by deleting all items on it.

**color\_palette = Frame(root)**

**color\_palette.pack(pady=10)**

These lines create a frame widget called color\_palette inside the main window. The frame is then packed into the main window with a vertical padding of 10 pixels.

**colors = ["black", "red", "green", "blue", "yellow"]**

**for color in colors:**

**button = Button(color\_palette, bg=color, width=5, command=lambda c=color: change\_color(c))**

**button.pack(side="left", padx=5)**

These lines create color buttons inside the color\_palette frame. Each button has a background color corresponding to the colors in the colors list. When a button is clicked, it calls the change\_color function with the respective color as an argument.

**eraser\_button = Button(root, text="Eraser", command=lambda: change\_color("white"))**

**eraser\_button.pack(pady=5)**

These lines create an eraser button inside the main window. When the eraser button is clicked, it calls the change\_color function with the color "white", effectively selecting the eraser.

**clear\_button = Button(root, text="Clear", command=clear\_canvas)**

**clear\_button.pack(pady=5)**

These lines create a clear button inside the main window. When the clear button is clicked, it calls the **clear\_canvas** function to clear the canvas.

**canvas.bind("<ButtonPress-1>", on\_button\_press)**

**canvas.bind("<B1-Motion>", on\_move)**

**canvas.bind("<ButtonRelease-1>", on\_button\_release)**

These lines bind the mouse events to the canvas. When the left mouse button is pressed, released, or moved, the respective functions (**on\_button\_press**, **on\_move**, **on\_button\_release**) are called.

**root.mainloop()**

This line starts the main event loop of the application, allowing it to respond to user interactions and updating the GUI accordingly.

I hope this explanation helps you understand the code. Let me know if you have any further questions!