Paint app

**use for javascript canvas**

**from tkinter import \***

**import tkinter.font**

**class PaintApp:**

**# Stores current drawing tool used**

**drawing\_tool = "line"**

**# Tracks whether left mouse is down**

**left\_but = "up"**

**# x and y positions for drawing with pencil**

**x\_pos, y\_pos = None, None**

**# Tracks x & y when the mouse is clicked and released**

**x1\_line\_pt, y1\_line\_pt, x2\_line\_pt, y2\_line\_pt = None, None, None, None**

**# ---------- CATCH MOUSE UP ----------**

**def left\_but\_down(self, event=None):**

**self.left\_but = "down"**

**# Set x & y when mouse is clicked**

**self.x1\_line\_pt = event.x**

**self.y1\_line\_pt = event.y**

**# ---------- CATCH MOUSE UP ----------**

**def left\_but\_up(self, event=None):**

**self.left\_but = "up"**

**# Reset the line**

**self.x\_pos = None**

**self.y\_pos = None**

**# Set x & y when mouse is released**

**self.x2\_line\_pt = event.x**

**self.y2\_line\_pt = event.y**

**# If mouse is released and line tool is selected**

**# draw the line**

**if self.drawing\_tool == "line":**

**self.line\_draw(event)**

**elif self.drawing\_tool == "arc":**

**self.arc\_draw(event)**

**elif self.drawing\_tool == "oval":**

**self.oval\_draw(event)**

**elif self.drawing\_tool == "rectangle":**

**self.rectangle\_draw(event)**

**elif self.drawing\_tool == "text":**

**self.text\_draw(event)**

**# ---------- CATCH MOUSE MOVEMENT ----------**

**def motion(self, event=None):**

**if self.drawing\_tool == "pencil":**

**self.pencil\_draw(event)**

**# ---------- DRAW PENCIL ----------**

**def pencil\_draw(self, event=None):**

**if self.left\_but == "down":**

**# Make sure x and y have a value**

**if self.x\_pos is not None and self.y\_pos is not None:**

**event.widget.create\_line(self.x\_pos, self.y\_pos, event.x, event.y, smooth=TRUE)**

**self.x\_pos = event.x**

**self.y\_pos = event.y**

**# ---------- DRAW LINE ----------**

**def line\_draw(self, event=None):**

**# Shortcut way to check if none of these values contain None**

**if None not in (self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt):**

**event.widget.create\_line(self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt, smooth=TRUE, fill="green")**

**# ---------- DRAW ARC ----------**

**def arc\_draw(self, event=None):**

**# Shortcut way to check if none of these values contain None**

**if None not in (self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt):**

**coords = self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt**

**# start : starting angle for the slice in degrees**

**# extent : width of the slice in degrees**

**# fill : fill color if needed**

**# style : can be ARC, PIESLICE, or CHORD**

**event.widget.create\_arc(coords, start=0, extent=150,**

**style=ARC)**

**# ---------- DRAW OVAL ----------**

**def oval\_draw(self, event=None):**

**if None not in (self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt):**

**# fill : Color option names are here http://wiki.tcl.tk/37701**

**# outline : border color**

**# width : width of border in pixels**

**event.widget.create\_oval(self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt,**

**fill="midnight blue",**

**outline="yellow",**

**width=2)**

**# ---------- DRAW RECTANGLE ----------**

**def rectangle\_draw(self, event=None):**

**if None not in (self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt):**

**# fill : Color option names are here http://wiki.tcl.tk/37701**

**# outline : border color**

**# width : width of border in pixels**

**event.widget.create\_rectangle(self.x1\_line\_pt, self.y1\_line\_pt, self.x2\_line\_pt, self.y2\_line\_pt,**

**fill="midnight blue",**

**outline="yellow",**

**width=2)**

**# ---------- DRAW TEXT ----------**

**def text\_draw(self, event=None):**

**if None not in (self.x1\_line\_pt, self.y1\_line\_pt):**

**# Show all fonts available**

**print(tkinter.font.families())**

**text\_font = tkinter.font.Font(family='Helvetica',**

**size=20, weight='bold', slant='italic')**

**event.widget.create\_text(self.x1\_line\_pt, self.y1\_line\_pt,**

**fill="green",**

**font=text\_font,**

**text="WOW")**

**def \_\_init\_\_(self, root):**

**drawing\_area = Canvas(root)**

**drawing\_area.pack()**

**drawing\_area.bind("<Motion>", self.motion)**

**drawing\_area.bind("<ButtonPress-1>", self.left\_but\_down)**

**drawing\_area.bind("<ButtonRelease-1>", self.left\_but\_up)**

**root = Tk()**

**paint\_app = PaintApp(root)**

**root.mainloop()**