```
from tkinter import *
from tkinter.colorchooser import askcolor
import sys, os, string, time
from PIL import Image, ImageDraw
import tkinter

tk = tkinter
class Paint():

    DEFAULT_PEN_SIZE = 5.0
    DEFAULT_COLOR = 'black'

    def __init__(self):
        self.root = Tk()
        self.root.title("Simple Gui")


        self.loc = self.dragged = 0


        self.pen_button = Button(self.root, text='pen', command=self.use_pen)
        self.pen_button.grid(row=0, column=0)

        self.brush_button = Button(self.root, text='label', command=self.use_brush)
        self.brush_button.grid(row=0, column=1)

        self.color_button = Button(self.root, text='color',
command=self.choose_color)
        self.color_button.grid(row=0, column=2)

        self.eraser_button = Button(self.root, text='eraser',
command=self.use_eraser)
        self.eraser_button.grid(row=0, column=3)

        self.button = Button(self.root, text="save", command=self.save)
        self.button.grid(row=0, column=5)

        self.choose_size_button = Scale(self.root, from_=1, to=10,
orient=HORIZONTAL)
        self.choose_size_button.grid(row=0, column=4)


        self.c = Canvas(self.root, bg='white', width=1000, height=1000,
highlightthickness=10)
        a = self.c
        self.c.label = Listbox(self.root)
        self.c.label.insert(0, 'Object 1 has triangles')
        self.c.label.insert(1, 'Object 2 has schools')
        self.c.label.insert(2, 'Object 3 has courses')
        self.c.label.insert(3, 'Arrow 1 is getschool()')
        self.c.label.insert(4, 'Arrow 2 is showcourses()')
        self.c.label.insert(5, 'Arrow 3 is trianglewidth()')
        widget = Label(self.c, text='Attributes', fg='white', bg='black')
        self.c.create_window(100, 200, anchor=NW, window=widget)



        self.c.grid(row=1, columnspan=5,)
        self.c.create_window(10, 10, anchor=NW, window=self.c.label)
```

```
        self.c.create_line(575, 200, 315, 200, width=2, fill='black',
arrow='first', activefill='violet', tags="DnD")
        self.c.create_oval(400, 400, 200, 200, width=2, fill='green',
activefill='gray', tags="DnD")
        self.c.create_oval(700, 400, 500, 200, width=2, fill='blue',
activefill='red', tags="DnD")
        self.c.create_oval(550, 600, 350, 400, width=2,fill='orange',
activefill='yellow', tags="DnD")
        self.c.create_line(400, 415, 360, 380, width=2, fill='black', arrow='last',
activefill='blue', tags="DnD")
        self.c.create_line(505, 415, 540, 380, width=2, fill='black',
arrow='first', activefill='sky blue', tags="DnD")

        self.defaultcolor = a.itemcget(a.create_text(300, 415,
                                                     font=("Helvetica",
14), text="Object 1", tags="DnD"),
                                       "fill")
        a.create_text(600, 415,
                      font=("Helvetica", 14), text="Object 2", tags="DnD")
        a.create_text(450, 615,
                      font=("Helvetica", 14), text="Object 3", tags="DnD")
        a.create_text(450, 180,
                      font=("Helvetica", 14), text="Arrow 1", tags="DnD")
        a.create_text(402, 380,
                      font=("Helvetica", 14), text="Arrow 2", tags="DnD")
        a.create_text(500, 380,
                      font=("Helvetica", 14), text="Arrow 3", tags="DnD")


        self.setup()

        self.root.mainloop()

        a.tag_bind("DnD", "<ButtonPress-3>", self.down)
        a.tag_bind("DnD", "<ButtonRelease-3>", self.chkup)
        a.tag_bind("DnD", "<Enter>", self.enter)
        a.tag_bind("DnD", "<Leave>", self.leave)

        width = 400
        height = 300
        green = (0,128,0)
        center = height // 2
        white = (255, 255, 255)
        image1 = Image.new("RGB", (width, height), white)
        draw = ImageDraw.Draw(image1)
        draw.line([0, center, width, center], green)

    def save(self):
        filename = "my_drawing.jpg"
        self.image1.save(filename)


    def down(self, event):
        print
        "Click on %s" % event.widget.itemcget(tk.CURRENT, "text")
        self.loc = 1
        self.dragged = 0
        event.widget.bind("<Motion>", self.motion)
```

```
def motion(self, event):
    self.root.config(cursor="exchange")
    cnv = event.widget
    cnv.itemconfigure(tk.CURRENT, fill="blue")
    x, y = cnv.ax(event.x), cnv.ay(event.y)
    got = event.widget.coords(tk.CURRENT, x, y)


def leave(self, event):
    self.loc = 0


def enter(self, event):
    self.loc = 1
    if self.dragged == event.time:
        self.up(event)


def chkup(self, event):
    event.widget.unbind("<Motion>")
    self.root.config(cursor="")
    self.target = event.widget.find_withtag(tk.CURRENT)
    event.widget.itemconfigure(tk.CURRENT, fill=self.defaultcolor)
    if self.loc:  # is button released in same widget as pressed?
        self.up(event)
    else:
        self.dragged = event.time


def up(self, event):
    event.widget.unbind("<Motion>")
    if (self.target == event.widget.find_withtag(tk.CURRENT)):
         print
         "Select %s" % event.widget.itemcget(tk.CURRENT, "text")
    else:
         event.widget.itemconfigure(tk.CURRENT, fill="blue")
         self.master.update()
         time.sleep(.1)
         print
         "%s Drag-N-Dropped onto %s" \
         % (event.widget.itemcget(self.target, "text"),
            event.widget.itemcget(tk.CURRENT, "text"))
         event.widget.itemconfigure(tk.CURRENT, fill=self.defaultcolor)

def setup(self):
    self.old_x = None
    self.old_y = None
    self.line_width = self.choose_size_button.get()
    self.color = self.DEFAULT_COLOR
    self.active_button = self.pen_button
    self.c.bind('<B1-Motion>', self.paint)
    self.c.bind('<ButtonRelease-1>', self.reset)

def use_pen(self):
    self.activate_button(self.pen_button)

def use_brush(self):
```

```
        self.activate_button(self.brush_button)

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

    def use_eraser(self):
        self.activate_button(self.eraser_button, eraser_mode=True)

    def activate_button(self, some_button, eraser_mode=False):
        self.active_button.config(relief=RAISED)
        some_button.config(relief=SUNKEN)
        self.active_button = some_button
        self.eraser_on = eraser_mode

    def paint(self, event):
        self.line_width = self.choose_size_button.get()
        paint_color = 'white' if self.eraser_on else self.color
        if self.old_x and self.old_y:
            self.c.create_line(self.old_x, self.old_y, event.x, event.y,
                               width=self.line_width, fill=paint_color,
                               capstyle=ROUND, smooth=TRUE, splinesteps=36)
        self.old_x = event.x
        self.old_y = event.y

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


 #def main():
   # root = Tk()
   # root.geometry('800x600+10+50')
    #app = Paint(root)
    #app.mainloop()

if __name__ == '__main__':
    Paint()
```