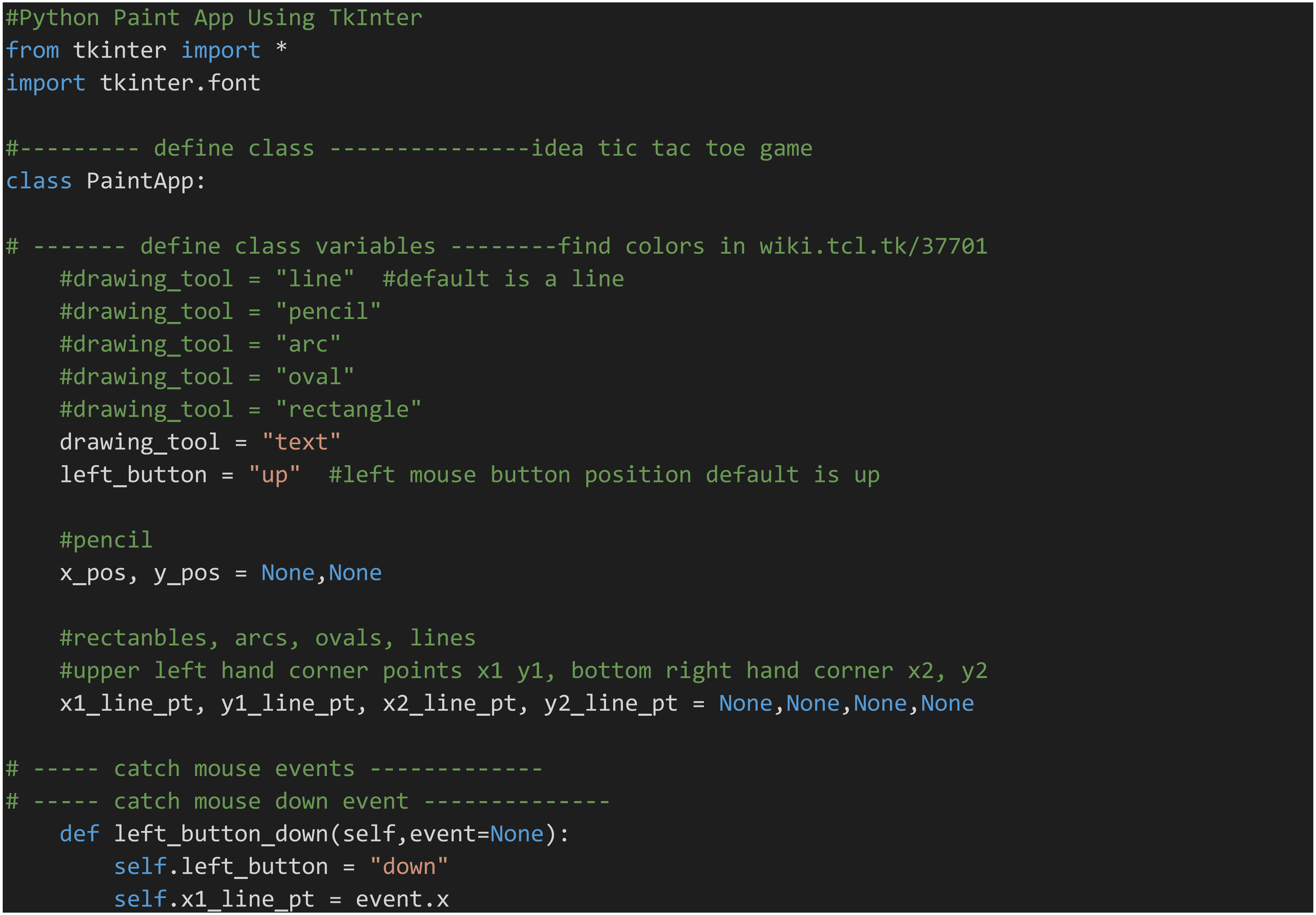
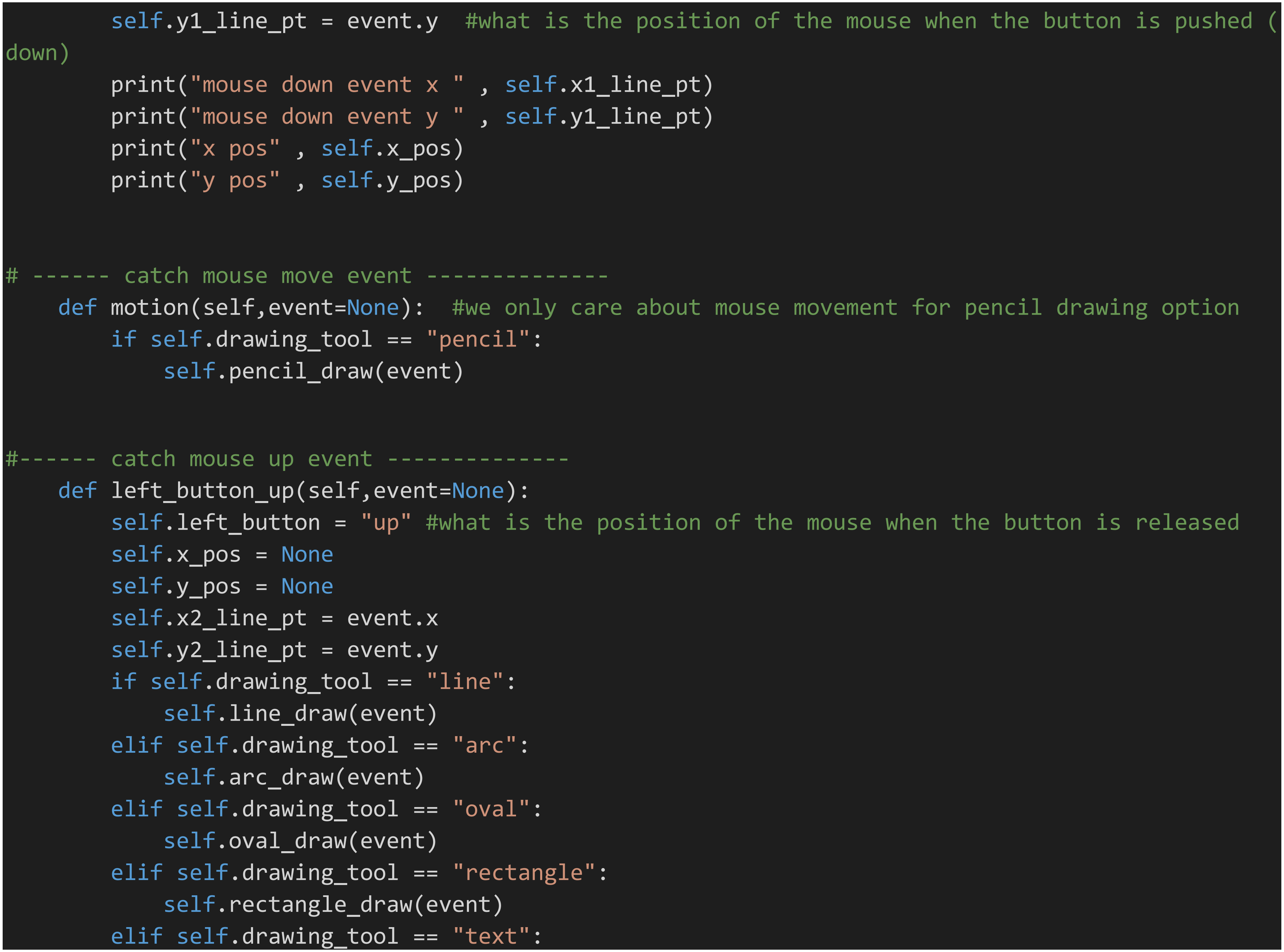
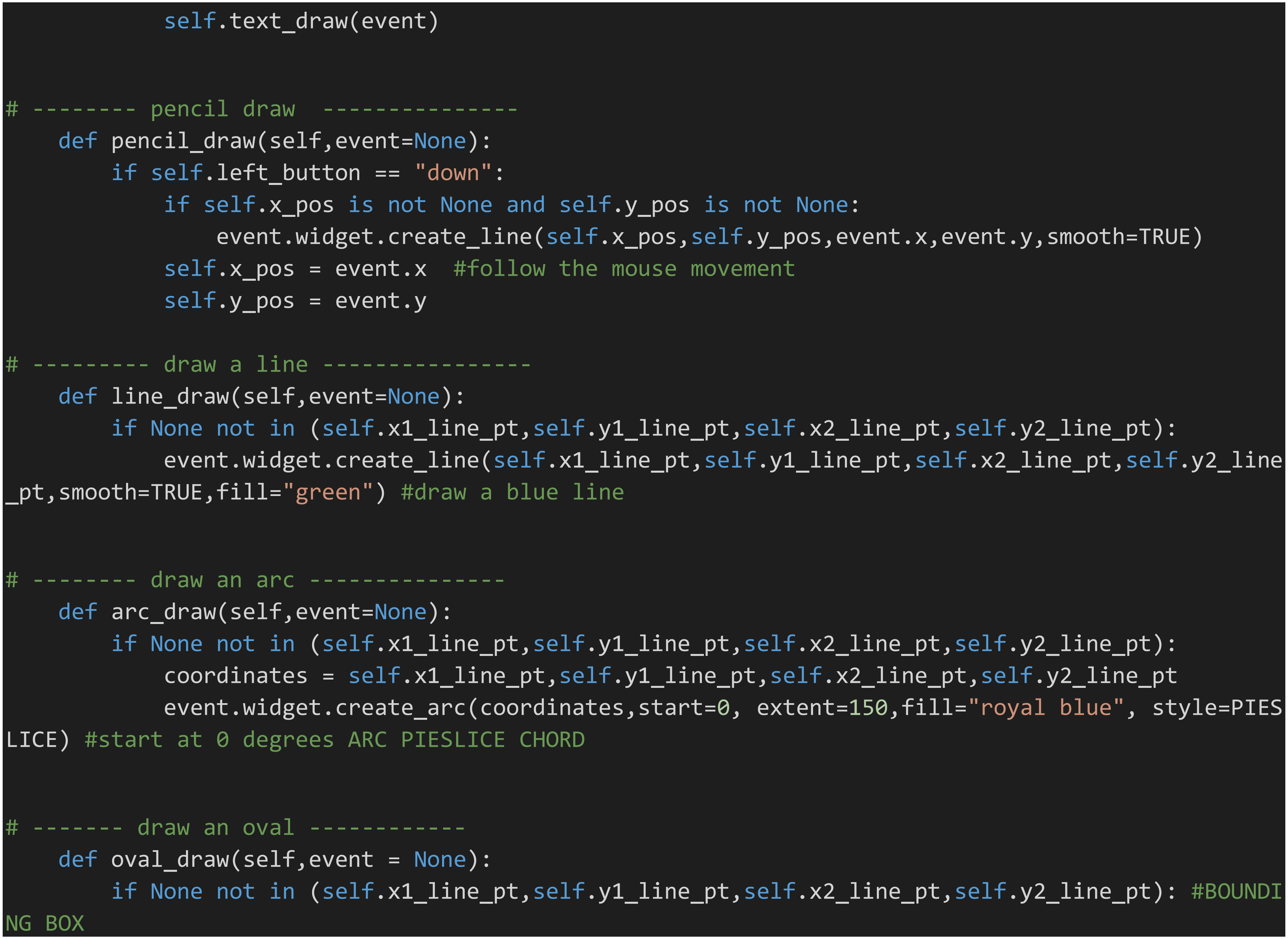
Basic Drawing on Canvass – In progress



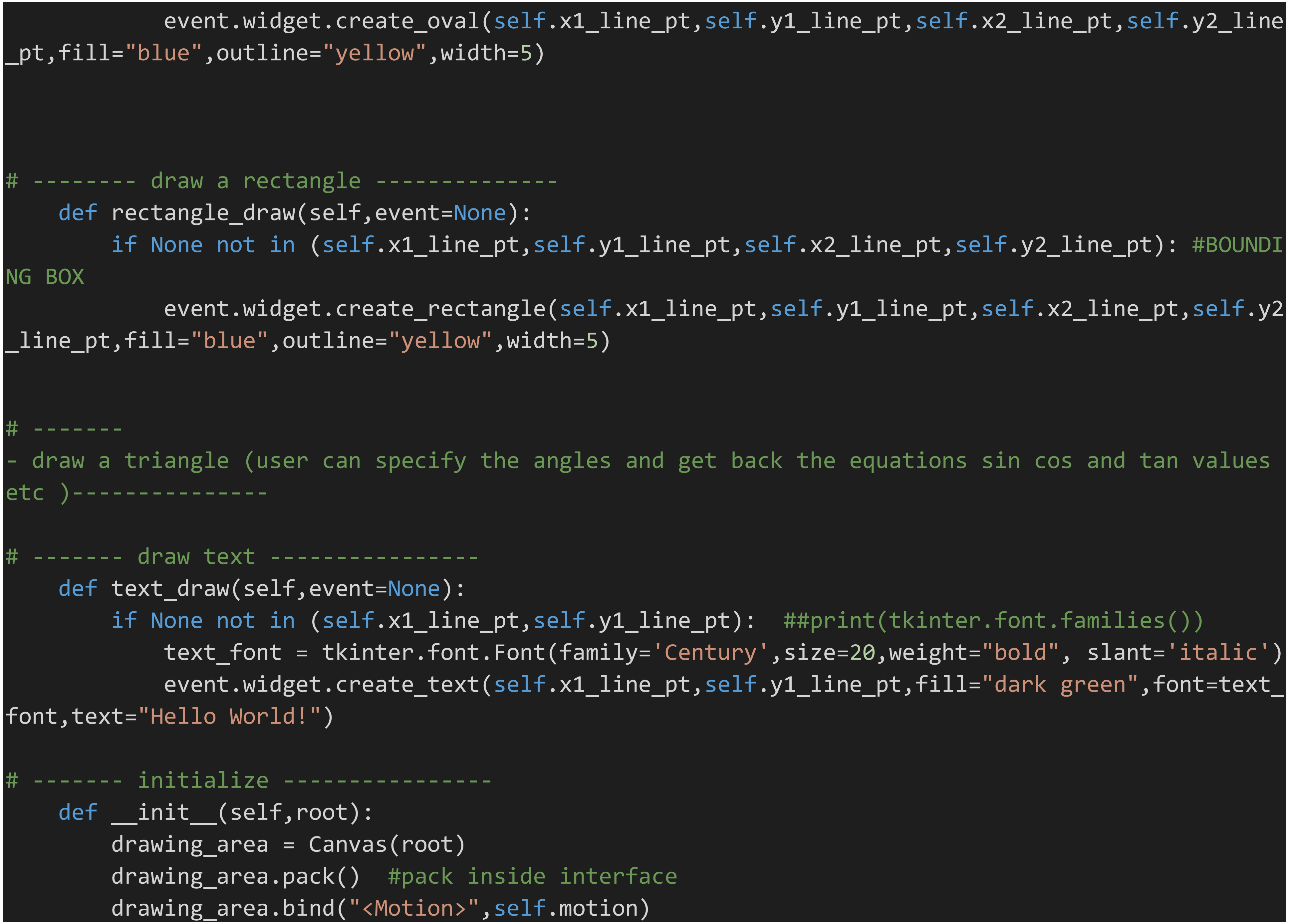
```
#Python Paint App Using TkInter
from tkinter import *
import tkinter.font

#--------- define class ---------------idea tic tac toe game
class PaintApp:

# ------- define class variables --------find colors in wiki.tcl.tk/37701
    #drawing_tool = "line"  #default is a line
    #drawing_tool = "pencil"
    #drawing_tool = "arc"
    #drawing_tool = "oval"
    #drawing_tool = "rectangle"
    drawing_tool = "text"
    left_button = "up"  #left mouse button position default is up

    #pencil
    x_pos, y_pos = None,None

    #rectanbles, arcs, ovals, lines
    #upper left hand corner points x1 y1, bottom right hand corner x2, y2
    x1_line_pt, y1_line_pt, x2_line_pt, y2_line_pt = None,None,None,None

# ----- catch mouse events -------------
# ----- catch mouse down event --------------
    def left_button_down(self,event=None):
        self.left_button = "down"
        self.x1_line_pt = event.x
```



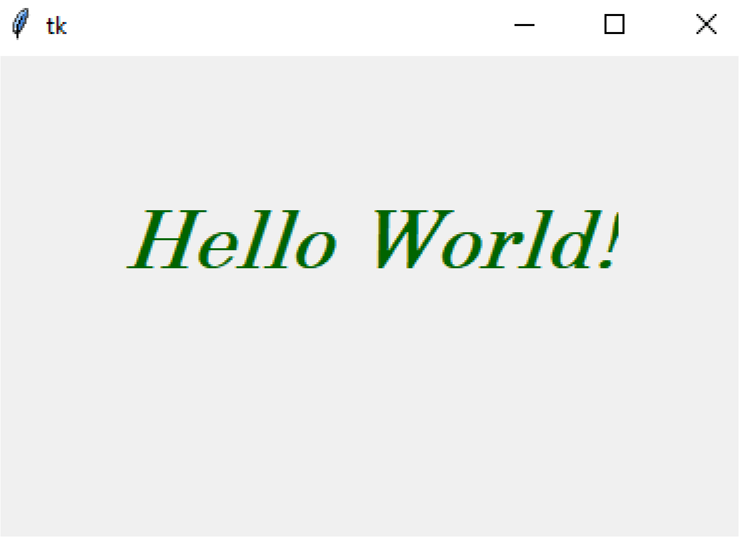
```
        self.y1_line_pt = event.y  #what is the position of the mouse when the button is pushed (
down)
        print("mouse down event x " , self.x1_line_pt)
        print("mouse down event y " , self.y1_line_pt)
        print("x pos" , self.x_pos)
        print("y pos" , self.y_pos)


# ------ catch mouse move event --------------
    def motion(self,event=None):  #we only care about mouse movement for pencil drawing option
        if self.drawing_tool == "pencil":
            self.pencil_draw(event)


#------ catch mouse up event --------------
    def left_button_up(self,event=None):
        self.left_button = "up" #what is the position of the mouse when the button is released
        self.x_pos = None
        self.y_pos = None
        self.x2_line_pt = event.x
        self.y2_line_pt = event.y
        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":
```



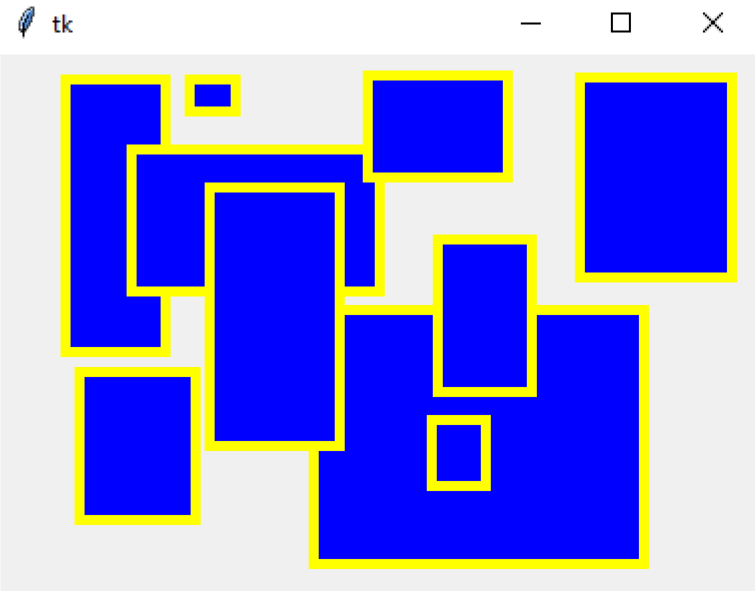
```
            self.text_draw(event)


# -------- pencil draw  ---------------
    def pencil_draw(self,event=None):
        if self.left_button == "down":
            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  #follow the mouse movement
            self.y_pos = event.y

# --------- draw a line ----------------
    def line_draw(self,event=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 a blue line


# -------- draw an arc ---------------
    def arc_draw(self,event=None):
        if None not in (self.x1_line_pt,self.y1_line_pt,self.x2_line_pt,self.y2_line_pt):
            coordinates = self.x1_line_pt,self.y1_line_pt,self.x2_line_pt,self.y2_line_pt
            event.widget.create_arc(coordinates,start=0, extent=150,fill="royal blue", style=PIES
LICE) #start at 0 degrees ARC PIESLICE CHORD


# ------- draw an 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): #BOUNDI
NG BOX
```

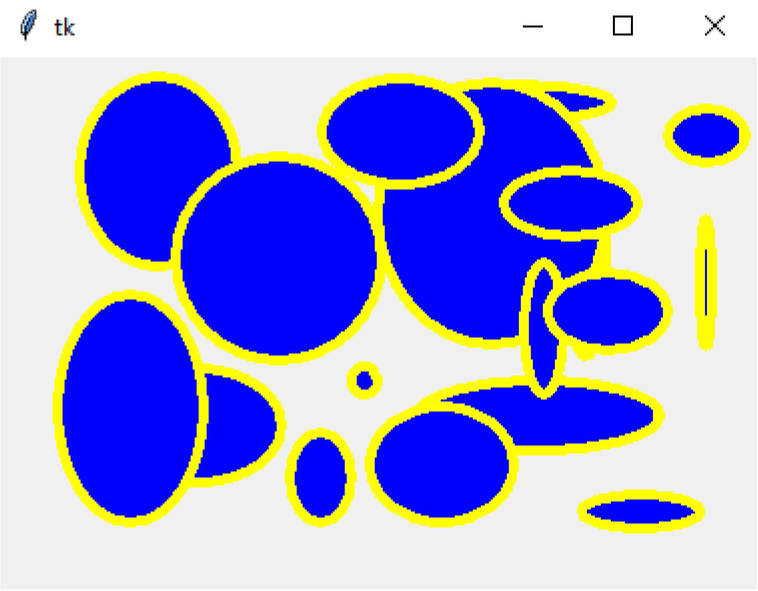


```
            event.widget.create_oval(self.x1_line_pt,self.y1_line_pt,self.x2_line_pt,self.y2_line
_pt,fill="blue",outline="yellow",width=5)



# -------- draw a 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): #BOUNDI
NG BOX
            event.widget.create_rectangle(self.x1_line_pt,self.y1_line_pt,self.x2_line_pt,self.y2
_line_pt,fill="blue",outline="yellow",width=5)


# -------
- draw a triangle (user can specify the angles and get back the equations sin cos and tan values 
etc )---------------

# ------- draw text ----------------
    def text_draw(self,event=None):
        if None not in (self.x1_line_pt,self.y1_line_pt):  ##print(tkinter.font.families())
            text_font = tkinter.font.Font(family='Century',size=20,weight="bold", slant='italic')
            event.widget.create_text(self.x1_line_pt,self.y1_line_pt,fill="dark green",font=text_
font,text="Hello World!")

# ------- initialize ----------------
    def __init__(self,root):
        drawing_area = Canvas(root)
        drawing_area.pack()  #pack inside interface
        drawing_area.bind("<Motion>",self.motion)
        drawing_area.bind("<ButtonPress-1>",self.left_button_down) #ButtonPress-1 is left mouse button press down
        drawing_area.bind("<ButtonRelease-1>",self.left_button_up)


root = Tk()
paint_app = PaintApp(root)
root.mainloop()
```



Hello World!




tk