```
######################
# Thomas Guldentops  #
# TFE - Programmation#
# Juin 2018          #
######################

from tkinter import *
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
from tkinter.colorchooser import askcolor
import getpass

class Touches(object):
    ''' redéfinission de la classe boutton de Tkinter. Permet de pouvoir jouer et placer les bouttons un peu plus facilement. Voir col
    def __init__(self, fenetre, color, def_color):
        self.color = color
        self.fen = fenetre
        self.boutton = Button(self.fen, bg = color, width=2, command=lambda:def_color(color))

    def grid(self, ligne, colonne):
        self.boutton.grid(row=ligne, column=colonne+3, padx=0)

class Paint(object):
    def __init__(self, fen_largeur=504, fen_hauteur=590):
        self.fen = Tk()
        self.fen.geometry("{0}x{1}".format(str(fen_largeur), str(fen_hauteur)))
        self.fen.title("Paint")
        self.fen.iconbitmap("images/image3.ico")
        self.fen.resizable(width=False, height=False)
        self.fen_largeur = fen_largeur
        self.fen_hauteur = fen_hauteur
        self.fen.configure(bg = "#282828")
        self.user = getpass.getuser()

        # initialisation des variables
        self.paint_color_background = "#282828"
        self.paint_color = "#FDF1B8"
        self.x = 0
        self.y = 0

        # Frame 1 colors
        self.frame1 = Frame(self.fen, borderwidth=0, relief=GROOVE, bg = "#282828")
        self.frame1.grid(row = 0, column = 1,padx=2, pady=2)

        # frame 2 canvas
        self.frame2 = Frame(self.fen, borderwidth=0, relief=GROOVE, bg = "#282828")
        self.frame2.grid(row = 2, columnspan = 10)
        self.can = Canvas(self.frame2, bg = "#282828", width=500, height=500)
        self.can.grid(row = 5)

        self.lab = Label(self.fen, text="Couleurs", bg = "#282828", fg = "#FDF1B8")
        self.lab.grid(row= 1, column =1, columnspan = 1)

        self.choose = Scale(self.fen, from_=1, to=20, orient=HORIZONTAL,bg = "#282828", troughcolor = "#282828", fg  = "#FDF1B8", reli
        self.choose.grid(row=0, column=2, columnspan = 1)
        self.chooseLab = Label(self.fen, text= "Taille du pinceau", bg = "#282828", fg = "#FDF1B8")
        self.chooseLab.grid(row = 1, column = 2, columnspan = 1)

        self.bouttonColor = Button(self.fen, text = "Changer les couleurs", command = self.color)
        self.bouttonColor.grid(row = 0, column =1, columnspan = 1)
        self.can.bind('<B1-Motion>', self.paint)
        self.can.bind('<ButtonRelease-1>', self.reset)
        self.can.bind('<B3-Motion>', self.erase)
        self.can.bind('<ButtonRelease-3>', self.reset)
        self.can.bind('<Button-2>', self.fillBackground)


        self.colors = []
        self.current_num = 0
        self.max_column = 6
        self.max_row = 2

        #self.creationBoutton()
        self.fen.mainloop()

    '''def constrboutton(self, i):
        self.b = Button(self.frame1, bg=i, width= 1, height = 1,relief = GROOVE,command = lambda:self.color(i))
        return self.b

    def creationBoutton(self):
        self.buttons = []
        self.bg = ["#FDF1B8",
                   "#000000",
                   "#e6e6fa",
                   "#670002",
                   "#8a2be2",
                   "#044aea",
                   "#99cccc",
                   "#c39b9b",
                   "#034b59",
                   "#33eee0",
                   "#c714df",
                   "#2f5ac6",
                   "#670002",
                   "#93afad"]

        self.pos = [[0, 0], [0, 1], [0, 2], [0, 3], [0, 4], [0, 5], [0, 6], [1,0], [1, 1], [1, 2], [1, 3], [1, 4], [1, 5], [1, 6]]
```

```
        for i in self.bg:
            self.buttons.append(self.constrboutton(i))
        for j in range(len(self.bg)):
            self.buttons[j].grid(row=self.pos[j][0], column=self.pos[j][1])

'''
    def color(self):
        ''' Permet de changer de couleur et de crée une mémoire des couleurs en créant des bouttons. En cliquant sur les bouttons préc
        self.paint_color = askcolor(color=self.paint_color)[1]
        boutton = Touches(self.fen, self.paint_color, self.bouttoncolor)
        self.colors.append(boutton)

        posx = self.current_num % self.max_column
        posy = int(self.current_num / self.max_column)
        self.colors[self.current_num].grid(ligne = posy, colonne = posx)
        self.current_num += 1

    def bouttoncolor(self, color):
        self.paint_color = color

    def paint(self, event):
        '''Création de l'evenement qui va créer de sligne en suivant la souris lorsque le clic gauche est activé.'''
        if self.x and self.y:
            self.can.create_line(self.x, self.y, event.x, event.y,
                                 width=self.choose.get(), fill=self.paint_color, capstyle=ROUND)
        self.x = event.x
        self.y = event.y

    def reset(self, event):
        ''' Permet de relever la souris et donc de ne plus déssiner lorsque que le clique gauche n'est plus enfoncé'''
        self.x, self.y = 0, 0

    def erase(self, event):
        if self.x and self.y:
            self.can.create_line(self.x, self.y, event.x, event.y,
                                 width=self.choose.get(), fill=self.paint_color_background, capstyle=ROUND)
        self.x = event.x
        self.y = event.y

    def fillBackground(self, event):
        ''' permet de changer la couleur du canvas de dessin'''
        self.paint_color_background = self.paint_color
        self.can.delete(ALL)
        self.can.configure(bg = self.paint_color_background)


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