

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

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

from time import strftime
import time
from timeit import default_timer
import pygame

from Notepad_project_tfe import *
from projet_tfe_calculatrice import *
from Paint import *

import os
from os import path

import getpass

class Desktop(object):
    ''' Création d'une application qui regroupe et permet de lancer les autres : - le Bloc-Note
                                           - La calculatrice
                                           - Le Paint'''

    def __init__(self):
        pygame.init()

        self.fen = Tk()
        self.fen.geometry("{0}x{1}".format(str(450), str(400)))
        self.fen.title("Desktop")
        self.fen.iconbitmap("images/image4.ico")
        self.color = "#282828"
        self.fen.configure(bg = self.color)
        self.fen.resizable(width=False, height=False)

        self.user = getpass.getuser() #Permet, sous window, d'avoir l'utilisateur actif, pratique pour définir le chemin à ouvrir lors

        # Espace app calculatrice
        self.pic= PhotoImage(file="images/image2.png")
        self.choixCalu = Button(self.fen, height= 75, width= 75, image=self.pic, relief=FLAT, command= self.launchCalc,bg= self.color)
        self.choixCalu.grid(row=2, column=1)

        self.labelcalc = Label(text="Calculatrice", bg = self.color, fg="#FDF1B8")
        self.labelcalc.grid(row = 3, column = 1)

        # Espace app Bloc note
        self.pic2= PhotoImage(file="images/image1.png")
        self.choixBloc = Button(self.fen, height= 75, width= 75, image=self.pic2, relief=FLAT, command= self.launchNotepad, bg= self.c
        self.choixBloc.grid(row=2, column= 2)

        self.labelbloc = Label(text="Bloc Note", bg = self.color, fg="#FDF1B8")
        self.labelbloc.grid(row = 3, column = 2)

        # Espace app Paint
        self.pic3= PhotoImage(file="images/image3.png")
        self.choixPaint = Button(self.fen, height= 75, width= 75, image=self.pic3, relief=FLAT, command= self.launchPaint, bg= self.cc
        self.choixPaint.grid(row=2, column= 3)

        self.labelPaint = Label(text="Paint", bg = self.color, fg="#FDF1B8")
        self.labelPaint.grid(row = 3, column = 3)

        # Horloge
        self.labelclock = Label(text="", bg = self.color, fg="#FDF1B8")
        self.labelclock.grid(row = 0, column = 1)

        # Couleur background
        self.fen.bind('<Button-3>', self.rightClick)

        # Search google
        self.entree = Entry(self.fen, width = 35, fg=self.color, relief = FLAT)
        self.entree.bind("<Return>", self.search) # Appuyer sur ENTER afin de faire une recherche google
        self.entree.grid(row = 1, column = 1, columnspan = 10)

        self.entree.insert(0, "Rechercher sur Google")
        self.entree.bind("<Button-1>", self.clearEntree)

        self.pic4= PhotoImage(file="images/googlem.png")
        self.labelGoogle = Label(self.fen, image = self.pic4, height = 50, width = 100, bg = self.color, fg="#FDF1B8" )
        self.labelGoogle.grid(row = 1, column = 0)

        # Musique
        self.frame1 = Frame(self.fen, borderwidth=5, bg = "#282828", highlightbackground = "#FDF1B8", highlightthickness=1)
        self.frame1.grid(row = 4, column = 1, columnspan = 5, pady = 10)

        self.slider = IntVar()
        self.bouttonVolume = Scale(self.fen, from_=0, to_=100, width = 5, length = 210, variable=self.slider, highlightbackground =
        self.bouttonVolume.grid(row=5, column=1, columnspan = 10)
        self.bouttonVolume.set(5)
```

```

self.bouttonAdd = Button(self.frame1, text = " Add music", relief=FLAT,command = self.loadMusic, bg= self.color, fg="#FDF1B8"
self.bouttonAdd.grid(row = 2, column = 0)

self.bouttonPlay = Button(self.frame1,text = "Play", relief=FLAT, command = self.playMusic, bg= self.color, fg="#FDF1B8" )
self.bouttonPlay.grid(row= 2, column = 1)

self.bouttonPause = Button(self.frame1, text="Pause", relief=FLAT,command = self.pauseMusic, bg= self.color, fg="#FDF1B8" )
self.bouttonPause.grid(row= 2, column = 2)

self.bouttonBack = Button(self.frame1, text="<<", relief=FLAT, command = self.changeMusicBack, bg= self.color, fg="#FDF1B8" )
self.bouttonBack.grid(row= 2, column = 4)

self.bouttonNext = Button(self.frame1, text=">>", relief=FLAT, command = self.changeMusicNext, bg= self.color, fg="#FDF1B8" )
self.bouttonNext.grid(row= 2, column = 5)

self.afficherMusique = Label(self.frame1, text = "", bg = self.color, fg="#FDF1B8")
self.afficherMusique.grid(row = 0, column=0, columnspan = 6)

# Minuteur
self.textClock = Label(self.frame1, text="", bg = self.color, fg="#FDF1B8")
self.textClock.grid(row = 1, column=0, columnspan = 6)

self.start = time.time()
self.stopChrono = False

self.musicQueue = []
self.currentlyPlayingSong = None

self.paused = False
self.start_time = None
self.stop_time = None
self.pause_time = 0

self.volume()
self.clock()
self.fen.mainloop()

def launchCalc(self):
    ''' Permet de lancer la calculatrice'''
    #self.fen.iconify()
    Calculatrice()

def launchNotepad(self):
    ''' Permet de lancer le Bloc-Note'''
    #self.fen.iconify()
    Notepad()

def clock(self):
    ''' Permet d'afficher l'heure actuelle selon les paramètre définit de l'ordinateur actif'''
    self.now = "0h 0min "
    self.now = time.strftime("%H:%M", time.localtime())
    self.labelclock.config(text=self.now)
    self.labelclock.after(200, self.clock)

def launchPaint(self):
    ''' Permet de lancer le Paint'''
    #self.fen.iconify()
    Paint()

def changeBackgroundColor(self):
    ''' Permet de changer la couleur du fond lorsqu'un CLIC DROIT + CHOIX COULEUR est fait'''
    self.color = askcolor(color=self.color)[1]
    self.fen.configure(bg = self.color)
    self.choixCalu.config(bg = self.color)
    self.labelcalc.config(bg = self.color)
    self.choixBloc.config(bg = self.color)
    self.labelbloc.config(bg = self.color)
    self.choixPaint.config(bg = self.color)
    self.labelPaint.config(bg = self.color)
    self.labelclock.config(bg = self.color)
    self.labelGoogle.config(bg = self.color)
    self.bouttonVolume.configure(bg = self.color, troughcolor = self.color)
    self.bouttonAdd.configure(bg = self.color)
    self.bouttonPlay.configure(bg = self.color)
    self.bouttonPause.configure(bg = self.color)
    self.bouttonBack.configure(bg = self.color)
    self.bouttonNext.configure(bg = self.color)
    self.afficherMusique.configure(bg = self.color)
    self.frame1.configure(bg = self.color)
    self.textClock.configure(bg = self.color)

def rightClick(self, *event):
    ''' Définit le clic droit'''
    # Menu clique droit
    self.fileRightClickMenu = Menu(self.fen, tearoff = 0)
    self.fileRightClickMenu.add_command(label="Changer les couleurs", command = self.changeBackgroundColor)

    # Récupérer les coords x et y du curseur
    self.ycoord = self.fen.winfo_pointery()
    self.xcoord = self.fen.winfo_pointerx()

```

```

# Ne plus associer le "Menu" de Tk à une barre fixe mais l'associer à la souris
self.fileRightClickMenu.tk_popup(self.xcoord, self.ycoord)

def clearEntree(self, *event):
    self.entree.delete(0, 'end')

def search(self, *event):
    if os.name == 'nt':
        ''' Permet de rechercher sur internet. Ne marche que sur Windows ... '''
        keyword = self.entree.get()
        keyword = keyword.replace(" ", "+")
        os.system("start https://google.com/search?q={}".format(keyword))
    else:
        print("Je n'ai pas trouvé comment lancer une recherche sur Linux. mais ça marche sur windows ! :D")

def changeTitle(self, musicName):
    ''' Permet de changer le nom de la musique écoutée'''
    if musicName:
        self.afficherMusique.config(text=path.basename(musicName))

def updateChrono(self):
    ''' Permet de lancer un chronomètre'''
    if not self.stopChrono:
        now = time.time() - self.start
        minutes, seconds = divmod(now, 60)
        self.str_time = "%02d:%02d" % (minutes, seconds)
        self.textClock.config(text=self.str_time)
        self.fen.after(1000, self.updateChrono)

def pauseChrono(self):
    ''' Met en pause le chronomètre '''
    self.pauze = time.time()
    self.stopChrono = True

def unpauseChrono(self):
    ''' Enlève la pause du chronomètre'''
    self.start += time.time() - self.pauze
    self.pauze = 0
    self.stopChrono = False
    self.updateChrono() #relancer le chrono là où il s'était arrêté

def resetChrono(self):
    ''' Permet de remettre le chrono à 0, utile lorsqu'on change de musique'''
    self.start = time.time()

def loadMusic(self, *event):
    ''' Permet de charger une musique et de la stocker dans une liste. La liste sera lu dans l'ordre dans lequel les musiques ont
    self.filename = askopenfilename(initialdir="C:/Users/%s/Music" % self.user, filetypes=[('mp3 files', '.mp3')])
    self.musicQueue.append(self.filename)
    pygame.mixer.music.load(self.filename)
    pygame.mixer.music.queue(self.filename)
    print(self.musicQueue)

def playMusic(self, *event):
    ''' Permet de jouer la dernière musique de la liste. Et de lancer l'application musique par la même occasion '''
    self.startTime = time.time()
    pygame.mixer.music.play()
    self.changeTitle(self.musicQueue[-1])
    self.updateChrono()

def pauseMusic(self, *event):
    ''' Permet de Mettre en pause et d'enlever la pause de la musique'''
    if self.paused:
        pygame.mixer.music.unpause()
        self.bouttonPause.config(text = "Pause")
        self.paused = False
        self.unpauseChrono()
    else:
        pygame.mixer.music.pause()
        self.bouttonPause.config(text = "Reprendre")
        self.paused = True
        self.pauseChrono()

def changeMusicNext(self, *event):
    ''' Permet de changer la musique en +1 de la liste. Donc d'aller à la suivante'''
    self.musicQueue = self.musicQueue[1:] + [self.musicQueue[0]]
    pygame.mixer.music.load(self.musicQueue[0])
    pygame.mixer.music.play()
    self.changeTitle(self.musicQueue[0])
    self.resetChrono()

def changeMusicBack(self, *event):
    ''' Permet de de changer la musique en -1 dans la liste. Donc d'aller à la précédente'''
    self.musicQueue = self.musicQueue[1:] + [self.musicQueue[0]]
    pygame.mixer.music.load(self.musicQueue[-1])
    pygame.mixer.music.play()
    self.changeTitle(self.musicQueue[-1])
    self.resetChrono()

def volume(self):
    ''' Permet de changer le volume de la musique. Par défaut il est à 5%'''

```

```
self.fen.after(100, self.volume)
volume = self.bouttonVolume.get()
pygame.mixer.music.set_volume(int(volume)/10.0)
```

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