CANJALE JUNIOR
Lucas - TP A1- TD1 - RT2

R309 – Programmation Evenementielle
TP2

Guide d'Utilisation et Description du Code:

Funcionalités:

-> Création des icônes (Multiples)

```
def client_pc():
    canvas.create_image(random.randint(250, 350),random.randint(70, 400),image=image2, tags='pc')
```

```
    imageFile = ("r.png")
    image1 = ImageTk.PhotoImage(Image.open(imageFile))

    imageFile = ("pc.png")
    image2 = ImageTk.PhotoImage(Image.open(imageFile))

    imageFile = ("sw.png")
    image3 = ImageTk.PhotoImage(Image.open(imageFile))

    imageFile = ("voip.png")
    image4 = ImageTk.PhotoImage(Image.open(imageFile))
```

Le code ci-dessus, sert à "ouvrir" une image qui est stockée sur notre disque dur et utilisée dans notre programme.

-> Déplacement (avec la souri)

```
def left(e):
    x, y = -50, 0
    canvas.move(imageFile, x, y)

def right(e):
    x, y = 50, 0
    canvas.move(imageFile, x, y)

def up(e):
    x,y = 0,-50
    canvas.move(imageFile, x, y)

def down(e):
    x,y = 0, 50
    canvas.move(imageFile, x, y)
```

Dans cette partie, vous attribuez des événements aux coordonnées auxquelles vous voulez que l'image se déplace. Les événements doivent indiquer une action.

```
#-----------------------------------------------------
def move_pc(e):
    global imageFile
    imageFile = ImageTk.PhotoImage(Image.open('pc.png'))
    canvas.create_image(e.x, e.y, image=imageFile)

def move_tel(e):
    global imageFile2
    imageFile2 = ImageTk.PhotoImage(Image.open('voip.png'))
    canvas.create_image(e.x, e.y, image=imageFile2)

def move_sw(e):
    global imageFile3
    imageFile3 = ImageTk.PhotoImage(Image.open('sw.png'))
    canvas.create_image(e.x, e.y, image=imageFile3)

def move_r(e):
    global imageFile4
    imageFile4 = ImageTk.PhotoImage(Image.open('r.png'))
    canvas.create_image(e.x, e.y, image=imageFile4)
#-----------------------------------------------------
```

Ensuite, chaque fonction de création d'images, au lieu d'attribuer des coordonnées statiques, attribue les coordonnées des événements déjà créés ci-dessus, ce qui me permet de déplacer les images. PS : Le déplacement des images se fait une par une, c'est-à-dire que la fonction déplacement prend toujours la dernière image créée (je n'ai pas pu résoudre ce bug).

-> Raccourci (avec le clavier)

Cette fonctionnalité permet au client d' ajouter d'une façon plus facile et rapide les icônes de réseaux dans le logiciel. Les raccourcis par chaque icône sont:

s -> Pour ajouter un switch; p -> Pour ajouter un PC

r -> Pour ajouter un routeur; t -> Pour ajouter un téléphone

```
#-----------------------------------------------------
#                       Raccourci
root.bind_all("<KeyPress-s>", lambda x: switch_add()) #Raccourci pour les switch's
root.bind_all("<KeyPress-p>", lambda x: client_pc())  #Raccourci pour les pc's
root.bind_all("<KeyPress-r>", lambda x: router_add()) #Raccourci pour les routeurs
root.bind_all("<KeyPress-t>", lambda x: client_tel()) #Raccourci pour les telephones
#-----------------------------------------------------
```

-> Création de lignes

```
def lignes():
    canvas.create_line(15, 50, 200, 50, width=5, fill="red")
```

-> Suppression des icônes (par click)

Cette fonction permet de supprimer une icône dans le programme avec un clic gauche de la souris, à partir de la propriété find_closest() elle permet de supprimer l'icône la plus proche de la souris.

```
def click(e):
    x,y = e.x,e.y
    c = canvas.find_closest(x,y)
    if c:
        canvas.delete(c[0])
```

-> Remplacement des icônes (par un menu pop up)

Pour le remplacement des icônes, j'ai procédé de la manière suivante :

Étape 1: J'ai d'abord téléchargé de nouvelles images et les ai importées.

```
imageF=("swws.png")
img1 = ImageTk.PhotoImage(Image.open(imageF))
```

Étape 3: J'ai créé une fonction dans laquelle j'utilise la propriété canvas.itemconfig pour pouvoir remplacer l'image désirée, pour cela j'ai utilisé un tag pour chaque image et la fonction créée précédemment (fonction qui importe les nouvelles images).

```
def changeSW():
    canvas.itemconfig("Switch", image=img1)
```

Étape 3: J'ai ensuite créé un petit menu pour chaque icône et j'ai attribué la fonction de substitution dans la propriété de la commande.

```
changeS = Menu(canvas, tearoff=0)
changeS.add_command(label="Changer Icon", command=lambda: changeSW())
```

Etape 4: Enfin une fonctionnalité pop up, elle permet d'ouvrir un petit menu en cliquant sur le côté droit de la souris sur l'icône désirée et ainsi nous pouvons remplacer l'image.

```
def pop_up(e):
    x,y = e.x, e.y
    changeS.tk_popup(x,y)
    changeS.post(e.x_root, e.y_root)
```