



Faculté Pluridisciplinaire de Nador

Master Sciences des Données et Systèmes Intelligents

Module: Langage Python

Année universitaire : 2020-2021

Rapport de jeu morpion avec Python

Professeur Responsable:
Pr. El OUAHABI SAFAA

Réalisé par: MOUNIR DOUIRI

Objectif :

L'objectif est donc de créer en Python et à l'aide de la bibliothèque graphique Tkinter, un jeu de morpion, jouable à deux joueurs;

Le jeu du morpion se joue sur une grille 3x3 à deux joueurs. Un joueur fait des cercles et l'autre des croix dans les cases, à tour de rôle. Celui qui réussit à faire un alignement vertical, horizontal ou diagonal a gagné.

Le but de ce projet est compris comment créer les widgets avec la bibliothèque graphique Tkinter, et compris aussi utilisation de la gestionnaire d'événements, les fenêtres de dialogue,... .

Plateau de jeu :

Le plateau de jeu est constitué deux frames principal :

- Le première frame: contient deux zones de saisies afin de saisir les noms des joueurs par l'utilisateur (Champs obligatoires), et deux liste (Combobox) afin de choisir pour chaque joueur un symbole (X ou O), et contient aussi trois boutons (**Jouer** pour de démarrer le jeu, **Annuler** pour effacer toutes les informations et désactiver le jeu, et **Recommencer** pour redémarrer le jeu ou commencer une nouvelle match).
- Le deuxième frame: contient trois autres frames, le premier pour afficher des informations sur le premier joueur (nom, score,...), le deuxième pour afficher les informations sur le deuxième joueur, et le troisième frame contient les règles du jeu. Il contient aussi une simple grille de 3 cases x 3 cases dessinée sur un canevas Tinter (Rappel: il s'agit d'un widget correspondant à une "zone de dessin"), et chaque case contient un bouton (widget Button) qui prend comme une valeur X ou O.
- Le plateau de jeu contient aussi à la fin un bouton **Quitter** pour quitter le jeu, et deux Labels l'une avec la couleur **rouge** qui indique que le jeu est désactivé, et l'autre avec le noir qui indique que le jeu est activé.

Jeu du morpion

Players Name :

Player One : Symbol : X ▾

Player Two : Symbol : O ▾

Jouer **Annuler** **Recommencer**

Player One:

SCORE:

Player Two:

SCORE:

Règle du jeu: Chaque joueur choisit une forme de jeton. A tour de rôle, les joueurs placent un de leurs jetons dans une case du quadrillage. Le but est d'être le premier à compléter une rangée des ses trois points horizontalement, verticalement ou diagonale.

Quitter *Rouge: le jeu est désactive *Noir: le jeu est active

Les étapes de création du jeu :

Nous allons décrire comment construire ce jeu en les étapes suivantes:

Etape 1: Création de premier frame

Players Name :

Player One : Symbol : X ▾

Player Two : Symbol : O ▾

Jouer **Annuler** **Recommencer**

Alors tout d'abord on crée un frame appelé par exemple frame1, et nous l'avons place dans notre fenêtre avec la méthode place.

```
frame1 = Frame(Mafenetre,highlightbackground='black', highlightthicknes=3)
frame1 . place(x=10,y=10, height=120, width=616)           # placer sur la fenêtre
```

L'option **Mafenetre** indique la fenêtre, et **highlightbackground** indique Couleur de la mise en valeur du focus lorsque le cadre a perdu le focus, et **highlightthickness** pour spécifier Épaisseur de la zone de mise en valeur du focus.

Pour les options de la méthode place, **x** et **y** indique décalage horizontal et vertical (en pixels), et **height**, **width** indique hauteur et largeur en pixels.

Etape suivant est la création des widgets qui sont à l'intérieur de frame:

```
lbl = Label(frame1, text='Players Name :').place(x=2,y=2)

player1 = Label(frame1, text='Player One :').place(x=20,y=30)
player2 = Label(frame1, text='Player Two :').place(x=20,y=65)

p1 = Entry(frame1,highlightbackground='black', highlightthickness=1)
p1.place(x=100, y=30)    # p1 --> zone de saisie le nom de joueur 1
p1.bind('<Button-1>', ChangeColor)

p2 = Entry(frame1,highlightbackground='black', highlightthickness=1)
p2.place(x=100, y=65)    # p2 --> zone de saisie le nom de joueur 2
p2.bind('<Button-1>', ChangeColor)

SymbolPlayer1 = Label(frame1, text='Symbol :').place(x=470,y=30)
SymbolPlayer2 = Label(frame1, text='Symbol :').place(x=470,y=65)

X01 = ttk.Combobox(frame1,values=['X','0'],width=5) # liste pour choisir X ou 0 pour joueur 1
X01 . place(x=535, y=30)
X01 . current(0)          # la valeur par défaut est la valeur qui est dans l'index 0 (X)

X02 = ttk.Combobox(frame1,values=['X','0'],width=5) # liste pour choisir X ou 0 pour joueur 1
X02 . place(x=535, y=65)
X02 . current(1)          # la valeur par défaut est la valeur qui est dans l'index 1 (0)
```

Pour Combobox C'est une liste avec une barre de défilement incluse qui est présente dans l'extension ttk, c'est-à-dire il doit importer ttk à l'aide de l'instruction **from tkinter import ttk**.

Pour la création et l'emplacement des trois boutons a fait ça avec code suivant:

```
Buttuonplay = Button(frame1, text = 'Jouer', fg='#001386', command=go,font=('Arial Black',8))
Buttuonplay . place(x=230, y= 85, width=60)    # Buttuonplay ---> bouton Jouer
ButtuonReset = Button(frame1, text = 'Annuler', fg='#A11411', command=reset,font=('Arial Black',8))
ButtuonReset . place(x=310, y= 85, width=60)    # ButtuonReset ---> bouton Annuler
ButtonRecommencer=Button(frame1,text='Recommencer',fg='#1B740B',command=Recommencer,font=('Arial Black',8))
ButtonRecommencer . place(x=390, y=85)          # ButtonRecommencer ---> bouton Recommencer
```

L'option **command** pour appeler une fonction ou méthode lorsqu'on clique sur le bouton.

Le bouton **Jouer** appeler la fonction **go()**, le bouton **Annuler** appeler la fonction **reset()**, et le bouton **Recommencer** appeler la fonction **Recommencer()**.

Etape 2 : La création de deuxième frame (frame2): contient 3 autres frames (frame3, frame4, frame5).

```
frame1 = Frame(Mafenetre,highlightbackground='black', highlightthicknes=3)
frame1 . place(x=10,y=10, height=120, width=616)           # placer sur la fenêtre

frame2 = Frame(Mafenetre,highlightbackground='black', highlightthicknes=3)
frame2 . place(x=10,y=135, height=400, width=616)         # placer sur la fenêtre

frame3 = Frame(frame2, highlightbackground='blue', highlightthicknes=3)
frame3 . place(x=5,y=12.3, height=299.5, width=145)

frame4 = Frame(frame2, highlightbackground='green', highlightthicknes=3)
frame4 . place(x=460,y=12.3, height=299.5, width=145)

frame5 = Frame(frame2, highlightbackground='black', highlightthicknes=3)
frame5 . place(x=5,y=320, height=70, width=600)
```

Frame3 pour affiche des informations sur le joueur 1 (Player One),
Frame4 pour affiche des informations sur le joueur 2 (Player Two),
Frame5 pour affiche les règles du jeu.

Le deuxième frame contient aussi la grille, le coude pour créer la grille est:

```
Canevas = Canvas(frame2, bg="white")
Canevas.place(x=152, y=10, width=304, height=304)

# ici on créer les lignes qui delimite les colones et les cases
Canevas.create_line(0, 100, 300, 100, fill="red", width=4)
Canevas.create_line(0, 200, 300, 200, fill="red", width=4)
Canevas.create_line(0, 300, 300, 300, fill="red", width=4)
Canevas.create_line(0, 3, 300, 3, fill="red", width=4)

Canevas.create_line(3, 300, 300, -100000, fill="red", width=4)
Canevas.create_line(100, 300, 300, -100000, fill="red", width=4)
Canevas.create_line(200, 300, 300, -100000, fill="red", width=4)
Canevas.create_line(300, 302, 300, -100000, fill="red", width=4)
```

On utilise le widget canvas et on crée 4 lignes horizontales et 4 lignes verticales.

Puis on ajoute dans chaque case un bouton:

```

button1 = Button(frame2,bg='red',text='',font=('',60), state=DISABLED, command=check1)
button1 . place(x=157, y=15, height=93.5, width=94)

button2 = Button(frame2, bg='red', text='',font=('',60), state=DISABLED, command=check2)
button2 . place(x=254, y=15, height=93.5, width=98)

button3 = Button(frame2,bg='red',fg='white',text='', font=('',60), state=DISABLED, command=check3)
button3 . place(x=354, y=15, height=93.5, width=96)

button4 = Button(frame2,bg='red',text='', font=('',60), state=DISABLED, command=check4)
button4 . place(x=157, y=112, height=96, width=94)

button5 = Button(frame2, bg='red', text='',font=('',60) , state=DISABLED, command=check5)
button5 . place(x=254, y=112, height=96, width=98)

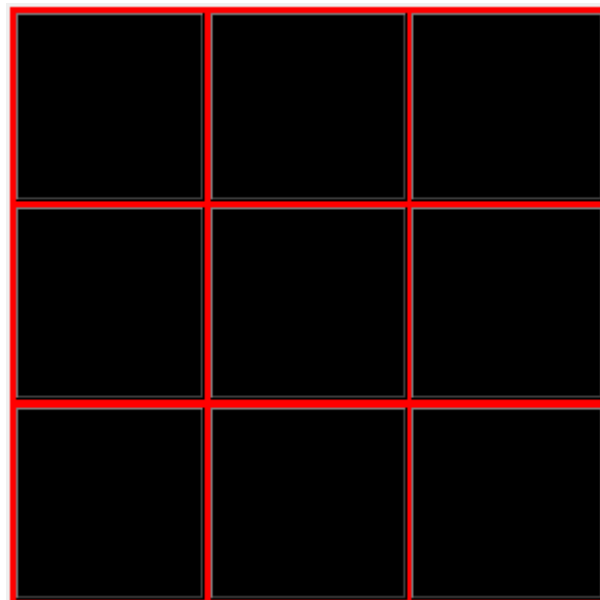
button6 = Button(frame2,bg='red',fg='white',text='', font=('',60), state=DISABLED, command=check6)
button6 . place(x=354, y=112, height=96, width=96)

button7 = Button(frame2,bg='red',text='', font=('',60), state=DISABLED, command=check7)
button7 . place(x=157, y=212, height=96, width=94)

button8 = Button(frame2, bg='red', text='',font=('',60), state=DISABLED, command=check8)
button8 . place(x=254, y=212, height=96, width=98)

button9 = Button(frame2,bg='red',fg='white',text='', font=('',60), state=DISABLED, command=check9)
button9 . place(x=354, y=212, height=96, width=96)

```



Le frame3 et frame4 contient les mêmes widgets: zone de saisie pour afficher le nom de chaque joueur, le score, et une image pour chacune avec un message quand le jeu terminé indique le joueur gagnant et Le joueur vaincu.

```
JR01 = Label(frame3,text='Player One: ', font=('',15) )
JR01 . place(x=1,y=1) # Label pour affiche le nom de joueur 1
ZP1 = Entry(frame3 ,highlightbackground='black', highlightthicknes=1,fg='blue',justify=CENTER,font=('Arial Black',13))
ZP1 . place(x=20,y=30,width=100,height=20) # zones de saisie nom joueur 1
ZP1 . config(state=DISABLED)

JR02 = Label(frame4,text='Player Two: ', font=('',15) )
JR02 . place(x=1,y=1)
ZP2 = Entry(frame4 ,highlightbackground='black', highlightthicknes=1,fg='green',justify=CENTER,font=('Arial Black',13))
ZP2 . place(x=20,y=30,width=100,height=20)
ZP2 . config(state=DISABLED)

img1 = PhotoImage(file='nr\m.png') # image no winner no loser
img2 = PhotoImage(file='loser.png') # image loser
img3 = PhotoImage(file='winner.png') # image winner
img = PhotoImage(file='BG.PNG') #image vide

imp1 = Label(frame3,image=img,bd=0)
imp1 . place(x=5,y=70,width=130,height=150)

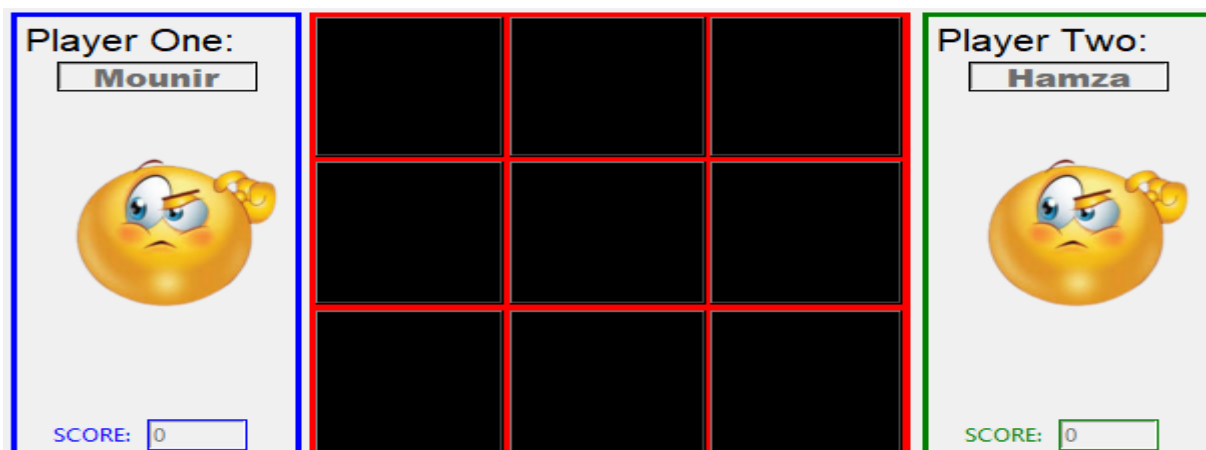
imp2 = Label(frame4,image=img,bd=0)
imp2 . place(x=5,y=70,width=130,height=150)

SCP1 = Label(frame3,text='SCORE: ', fg='blue')
SCP1 . place(x=15,y=270) # score player one
ZSCP1 = Entry(frame3 ,highlightbackground='blue', highlightthicknes=1)
ZSCP1 . place(x=65,y=270,width=50) # zone de saisie score player one
ZSCP1 . config(state=DISABLED)

SCP2 = Label(frame4,text='SCORE: ', fg='green')
SCP2 . place(x=15,y=270) # score player two
ZSCP2 = Entry(frame4 ,highlightbackground='green', highlightthicknes=1)
ZSCP2 . place(x=65,y=270,width=50) # zone de saisie score player two
ZSCP2 . config(state=DISABLED)

wnr = Label(frame3,text='',font=('Arial Black',10),justify=CENTER,width=14)
wnr . place(x=5,y=220) # pour msg dans la fin de jeu pour player one

lsr = Label(frame4,text='',font=('Arial Black',10),justify=CENTER,width=14)
lsr . place(x=5,y=220) # pour msg dans la fin de jeu pour player two
```



Pour le frame5 contient les règles du jeu:

```
frame5 = Frame(frame2, highlightbackground='black', highlightthickness=3)
frame5 . place(x=5,y=320, height=70, width=600)

var = StringVar()
var . set('Chaque joueur choisit une forme de jeton. A tour de rôle, les joueurs placent un de leurs\n'
        ' jetons dans une case du quadrillage. Le but est d'être le premier à compléter une rangée \n'
        ' des ses trois points horizontalement, verticalement ou diagonale.')

regle_jeu = Label(frame5,text='Règle du jeu: ',justify=LEFT,font=('Arial Black',13))
regle_jeu . place(x=0,y=0)
rg = Label(frame5,textvariable=var,justify=LEFT)
rg . place(x=120,y=7)
```

Règle du jeu: Chaque joueur choisit une forme de jeton. A tour de rôle, les joueurs placent un de leurs jetons dans une case du quadrillage. Le but est d'être le premier à compléter une rangée des ses trois points horizontalement, verticalement ou diagonale.

Etape 3:

A la fin de la fenêtre on créé un bouton **Quitter** qui appelle la méthode quitter pour quitter le jeu, et aussi deux remarques indiquent si le jeu est actif ou désactivé.

```
qtr = Button(Mafenetre,text='Quitter',fg='red',font=('Arial Black',10), command=quitter)
qtr.place(x=10,y=538,width=80,height=30)

Jact = Label(Mafenetre,text='*Rouge: le jeu est désactive', fg='red').place(x=250,y=540) # jeu active
Jdesct = Label(Mafenetre,text='*Noir: le jeu est active').place(x=470,y=540) # jeu désactive
```



Le code source de la méthode quitter():

```
def quitter():
    Mafenetre.quit()
    Mafenetre.destroy()
```

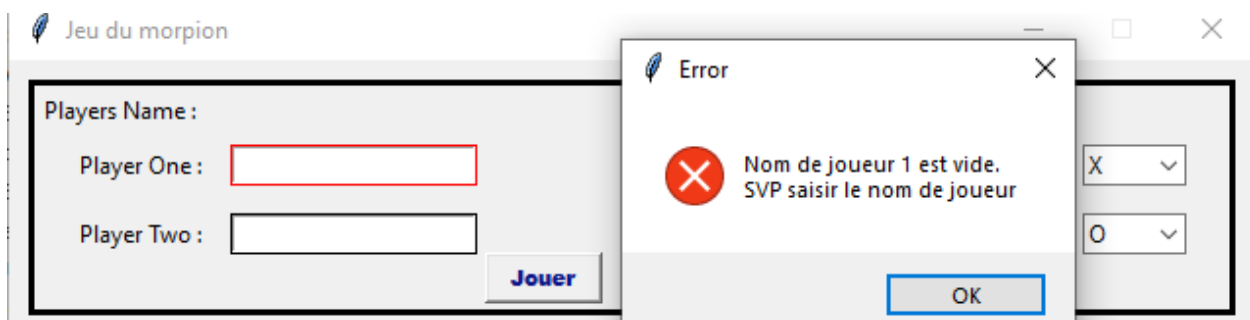
La méthode tkinter **quit** quitte le gestionnaire d'événements **mainloop**, et **destroy** détruit tous les widgets incorporés et quitte ensuite la **mainloop**.

Pour la fonction go():

```
def go():
    global etatWidgets, temp, p1, p2, X01, X02, Buttonplay, imp1, imp2, img1, ZSCP1, ZSCP2
    if (p1.get()==' ' or p2.get()==' '):
        if(p1.get()==' '):
            p1.config(highlightbackground='red', highlightthickness=1)
            showerror('Error', 'Nom de joueur 1 est vide.\nSVP saisir le nom de joueur')
        else:
            p2.config(highlightbackground='red', highlightthickness=1)
            showerror('Error', 'Nom de joueur 2 est vide.\nSVP saisir le nom de joueur')
    elif (X01.get()==X02.get()):
        showerror('Error', 'Impossible les deux joueurs jouer avec le même symbole.\n'
            'SVP chose X for player one and O for player two or vice versa')
    else:
        p1.config(state=DISABLED)
        p2.config(state=DISABLED)
        X01.config(state=DISABLED)
        X02.config(state=DISABLED)
        etatWidgets = 0 # P1,P2,X01,X02 sont désactive

    button1.config(text='', state=NORMAL, bg='black')
    button2.config(text='', state=NORMAL, bg='black')
    button3.config(text='', state=NORMAL, bg='black')
    button4.config(text='', state=NORMAL, bg='black')
    button5.config(text='', state=NORMAL, bg='black')
    button6.config(text='', state=NORMAL, bg='black')
    button7.config(text='', state=NORMAL, bg='black')
    button8.config(text='', state=NORMAL, bg='black')
    button9.config(text='', state=NORMAL, bg='black')
```

La première partie de la fonction indique que si le nom de joueur 1 (Player One) ou joueur 2 (player two) est vide, ou si les deux joueurs ont le même symbole, le programme doit mettre en rouge le cadre de la zone de saisie et affiche un message d'erreur, pour ça on a utilisé l'alerte **showerror** qui existe dans le module **tkinter.messagebox**, C'est-à-dire tout d'abord il faut importer ce module.



Pour la deuxième partie indique si toutes les informations sont complètes, une fois on clique sur le bouton Jouer le programme désactive les zones de saisies pour les noms des joueurs et aussi les listes des symboles (X, O),

Et active les 9 boutons de chaque case de la grille (par défaut on a défini les boutons à état désactivé), et aussi met en couleur noir les 9 boutons de la grille.

L'option **state** : indique état de widget (active ou désactive)

La suite de la fonction **go()**:

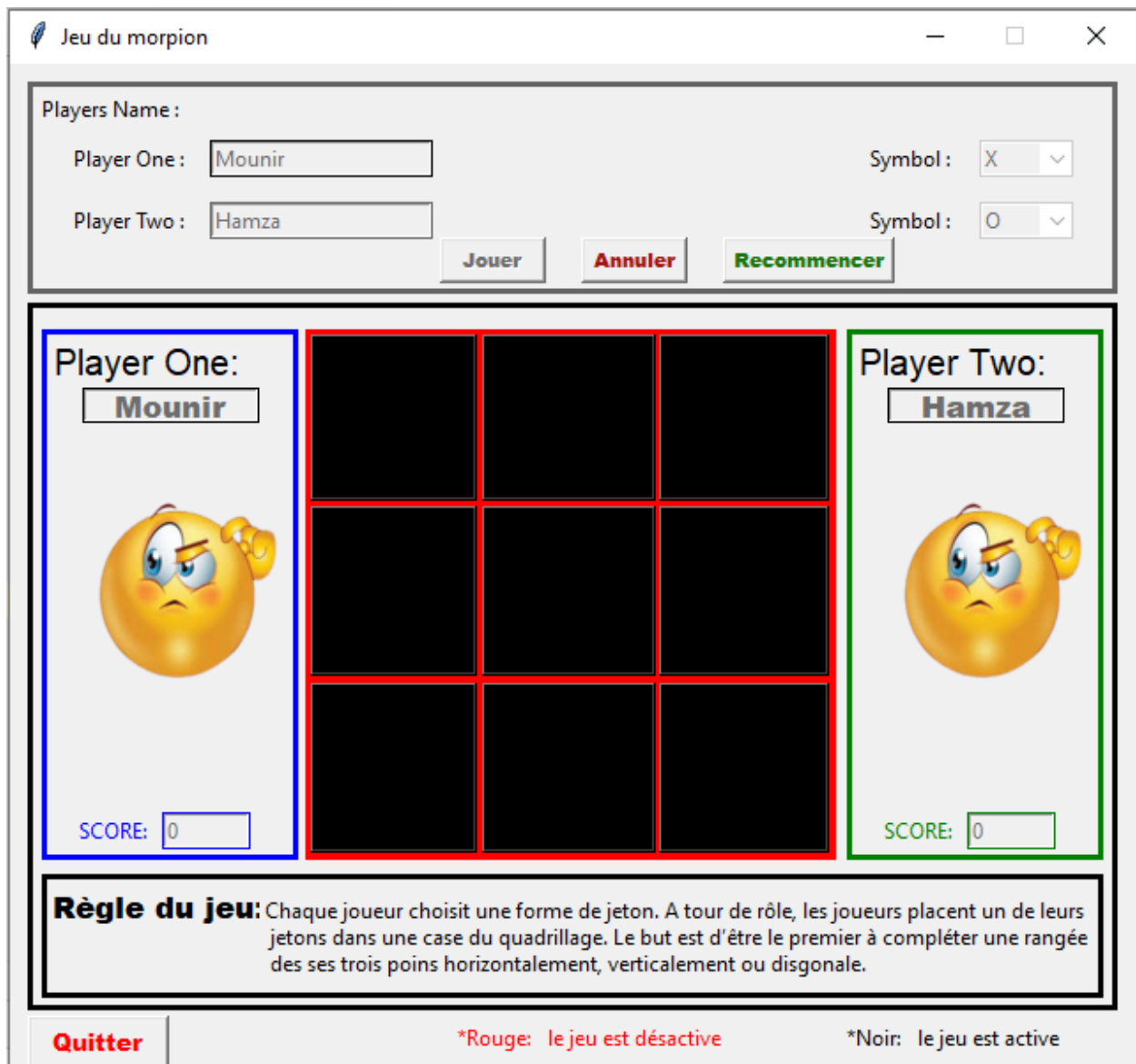
```
temp = 1                                #Symbole de joueur X

imp1 . config(image=img1)
imp2 . config(image=img1)

nm1=p1.get()
nm2=p2.get()
ZP1.config(state=NORMAL)
ZP2.config(state=NORMAL)
ZP1.insert(0,nm1)
ZP2.insert(0,nm2)
ZP1.config(state=DISABLED)
ZP2.config(state=DISABLED)

ZSCP1.config(state=NORMAL)
ZSCP2.config(state=NORMAL)
ZSCP1.insert(0,0)
ZSCP2.insert(0,0)
ZSCP1.config(state=DISABLED)
ZSCP2.config(state=DISABLED)
```

La fonction go() permet aussi d'afficher un image qui indique état de joueur au début du jeu (ni Heureux ni Triste, Normal) pour chacun joueur, et permet aussi d'afficher les noms des joueurs et la valeur par défaut de score pour les joueurs(0).



Pour la fonction **reset()**:

```
def reset():
    global etatWedgets, Bouttonplay, L1, L2, L3, C1, C2, C3, G1, G2, img, imp1, imp2, ZP1, ZP2, ZSCP1, ZSCP2
    global X01, X02, SCORE1, SCORE2

    if (etatWedgets == 0):
        p1.config(state=NORMAL)
        p2.config(state=NORMAL)
        X01.config(state=NORMAL)
        X02.config(state=NORMAL)
        etatWedgets=1
```

```

p1 . delete(0,END)
p2 . delete(0, END)
X01 . current(0)
X02 . current(1)
button1 . config(text='', state=DISABLED, bg='red')
button2 . config(text='', state=DISABLED, bg='red')
button3 . config(text='', state=DISABLED, bg='red')
button4 . config(text='', state=DISABLED, bg='red')
button5 . config(text='', state=DISABLED, bg='red')
button6 . config(text='', state=DISABLED, bg='red')
button7 . config(text='', state=DISABLED, bg='red')
button8 . config(text='', state=DISABLED, bg='red')
button9 . config(text='', state=DISABLED, bg='red')
Buttuonplay.config(state=NORMAL)

L1 = ['', '', ''] # ligne1
L2 = ['', '', ''] # ligne2
L3 = ['', '', ''] # ligne3

C1 = ['', '', ''] # column 1
C2 = ['', '', ''] # column 2
C3 = ['', '', ''] # column 3

G1 = ['', '', ''] #diagonale 1
G2 = ['', '', ''] #diagonale 2
imp1.config(image=img)
imp2.config(image=img)
wnr.config(text='')
lsr.config(text='')

ZP1.config(state=NORMAL) # active la zone de saisie qui affiche nom player one
ZP2.config(state=NORMAL) # active la zone de saisie qui affiche nom player two
ZP1.delete(0, END) # supprimer nom de player one
ZP2.delete(0, END) # supprimer nom de player two
ZP1.config(state=DISABLED) # desactive la zone de saisie qui affiche nom player one
ZP2.config(state=DISABLED) # desactive la zone de saisie qui affiche nom player two

ZSCP1.config(state=NORMAL) # active la zone de saisie qui affiche le score player one
ZSCP2.config(state=NORMAL) # active la zone de saisie qui affiche le score player two
ZSCP1.delete(0, END) # supprimer score de player one
ZSCP2.delete(0, END) # supprimer score de player two
ZSCP1.config(state=DISABLED) # desactive la zone de saisie qui affiche le score player one
ZSCP2.config(state=DISABLED) # desactive la zone de saisie qui affiche le score player two
SCORE1=0 # le variable qui calcule score de player one
SCORE2=0 # le variable qui calcule score de player two

```

La fonction reset permet de supprimer toutes les informations des joueurs (les informations qui affiche dans les frames 1 et 3 et 4 : nom de joueur, symbole, score, image,...), et désactive tous les 9 boutons la grille.

Mais à condition si la variable **etatWedgets=0** c'est-à-dire si les wedgits de frame1 sont désactivé.

En général la méthode reset permet d'annuler le jeu, et demande de remplir à nouveau les informations qui sont dans le frame 1.

Pour la méthode Recommencer():

```
def Recommencer():
    global temp, etatWedgets, L1, L2, L3, C1, C2, C3, G1, G2, img1, imp1, imp2, wnr, lsr
    if (etatWedgets==0):
        temp = 1
        button1.config(text='', state=NORMAL)
        button2.config(text='', state=NORMAL)
        button3.config(text='', state=NORMAL)
        button4.config(text='', state=NORMAL)
        button5.config(text='', state=NORMAL)
        button6.config(text='', state=NORMAL)
        button7.config(text='', state=NORMAL)
        button8.config(text='', state=NORMAL)
        button9.config(text='', state=NORMAL)
        L1 = ['', '', ''] # vidé la ligne 1
        L2 = ['', '', ''] # vidé la ligne 2
        L3 = ['', '', ''] # vidé la ligne 3
        C1 = ['', '', ''] # vidé column 1
        C2 = ['', '', ''] # vidé column 2
        C3 = ['', '', ''] # vidé column 3
        G1 = ['', '', ''] #vidé diagonale 1
        G2 = ['', '', ''] #vidé diagonale 2

        imp1.config(image=img1) # image pour affiche état des joueur 1 dans état en jeu
        imp2.config(image=img1) # image pour affiche état des joueur 2 dans état en jeu
        wnr.config(text='') # vidé le Un message de félicitations pour joueur 1
        lsr.config(text='') # vidé le Un message de félicitations pour joueur 2
```

Si les wedgits de frame1 sont désactivés (**etatWedgets=0**) c'est-à-dire le jeu est active, la fonction Recommencer vidé les valeurs de 9 boutons de chaque case de la grille, pour commencer un autre match avec les mêmes joueurs.

Pour la fonction **ChangeColor()**:

```
def ChangeColor(event):
    p1.config(highlightbackground='black', highlightthicknes=1)
    p2.config(highlightbackground='black', highlightthicknes=1)
```

Prend en paramètre un event, et permet de met en noir le cadre de la zone de saisie le nom de joueurs 1 et 2 à chaque fois en clique sur l'une de ces zones.

```

p1 = Entry(frame1,highlightbackground='black', highlightthicknes=1)
p1.place(x=100, y=30)    # p1 --> zone de saisie le nom de joueur 1
p1.bind('<Button-1>', ChangeColor)

p2 = Entry(frame1,highlightbackground='black', highlightthicknes=1)
p2.place(x=100, y=65)    # p2 --> zone de saisie le nom de joueur 2
p2.bind('<Button-1>', ChangeColor)

```

La méthode bind() permet de lier un événement avec une fonction,

Pour '**<Button-1>**' indique si on clique une seul fois sur widget P1 ou P2, appeler la fonction ChangeColor.

On a chaque case de la grille contient un bouton et chaque bouton appelé une fonction (**Check**).

Pour la fonction **check()**:

```

def check1():
    global temp,L1,C1,G1,X01,p1,p2,imp1,imp2,img2,img3,SCORE1,SCORE2
    if (temp == 1): # temp=1 indique le tour pour le player qui jouer avec X
        button1.config(state=DISABLED) # désactive le bouton
        button1.config(text='X')      # la valeur de bouton prend X
        temp = 0 # temp=1 indique le tour pour le player qui jouer avec 0
        L1[0]='X' # première valeur de ligne 1 prend X
        C1[0]='X' # première valeur de column 1 prend X
        G1[0]='X' # première valeur de diagonale 1 prend X

    elif (temp == 0): # temp=0 indique le tour pour le player qui jouer avec 0
        button1.config(state=DISABLED) # désactive le bouton
        button1.config(text='0')      # la valeur de bouton prend 0
        temp = 1 # temp=1 indique le tour pour le player qui jouer avec X
        L1[0]='0' # première valeur de ligne 1 prend 0
        C1[0]='0' # première valeur de column 1 prend 0
        G1[0]='0' # première valeur de diagonale 1 prend 0

```

On a défini par défaut le joueur qui début le jeu est le joueur qui jouer avec le symbole X.

Si le tour de jeu avec le joueur qui joue avec X, une fois on clique sur le bouton, ce bouton doit désactive et prend comme valeur la valeur X, et on passe le tour pour le joueur qui joue avec O. puis on stocke la valeur qui prend le bouton à sa position, pour voir à la fin le joueur qui gagner.

On fait la même chose si **temp= 0** c'est-à-dire, si le tour avec le joueur qui joue avec le symbole O.

Suite de la fonction check : cas si le tour avec le symbole **X**

```
if ((L1[0]=='X' and L1[1]=='X' and L1[2]=='X') or (C1[0]=='X' and C1[1]=='X' and C1[2]=='X') or
    (G1[0]=='X' and G1[1]=='X' and G1[2]=='X')):

    selc=X01.get() # recuperer le symbole de jeu pour player 1
    name1=p1.get() # recuperer le nom de joueur pour player 1
    name2=p2.get() # recuperer le nom de joueur pour player 2
    if(selc=='X'):
        showinfo('Resultat','Mabrouuuuuk, Le joueur '+name1+' est gagner!')
        imp1.config(image=img3)
        imp2.config(image=img2)
        wnr.config(text='Congratulations !')
        lsr.config(text='Hard luck... !')
        SCORE1=SCORE1+1
        ZSCP1.config(state=NORMAL)
        ZSCP1.delete(0,END)
        ZSCP1.insert(0,SCORE1)
        ZSCP1.config(state=DISABLED)
```

Si L1=[X,X,X] ou C1=[X,X,X], ou G1=[X,X,X] alors le joueur qui joue avec le symbole X est gagner dans la fonction doit afficher un message indique que le joueur est gagner, et afficher une image et un message de félicitations pour chaque joueur. Puis incrément le score de joueur qui gagner.

Suite de fonction:

```
else:

    showinfo('Resultat','Mabrouuuuuk, Le joueur '+name2+' est gagner!')
    imp1.config(image=img2)
    imp2.config(image=img3)
    lsr.config(text='Congratulations !')
    wnr.config(text='Hard luck... !')
    SCORE2 = SCORE2 + 1
    ZSCP2.config(state=NORMAL)
    ZSCP2.delete(0, END)
    ZSCP2.insert(0, SCORE2)
    ZSCP2.config(state=DISABLED)

    button1.config(state=DISABLED)
    button2.config(state=DISABLED)
    button3.config(state=DISABLED)
    button4.config(state=DISABLED)
    button5.config(state=DISABLED)
    button6.config(state=DISABLED)
    button7.config(state=DISABLED)
    button8.config(state=DISABLED)
    button9.config(state=DISABLED)
```

Si la valeur de **selc='X'** c'est-à-dire que le joueur 1 qui gagner car **selc** prend la valeur de symbole de première joueur.

else indique si **selc != 'X'** c'est-à-dire **selc='O'**, donc dans ce cas le deuxième joueur qui gagner.

Et à la fin on désactive les autres boutons qui sont disponible pour indiquer que, le jeu est terminé.

Suite de fonction: cas si le tour avec le symbole ○

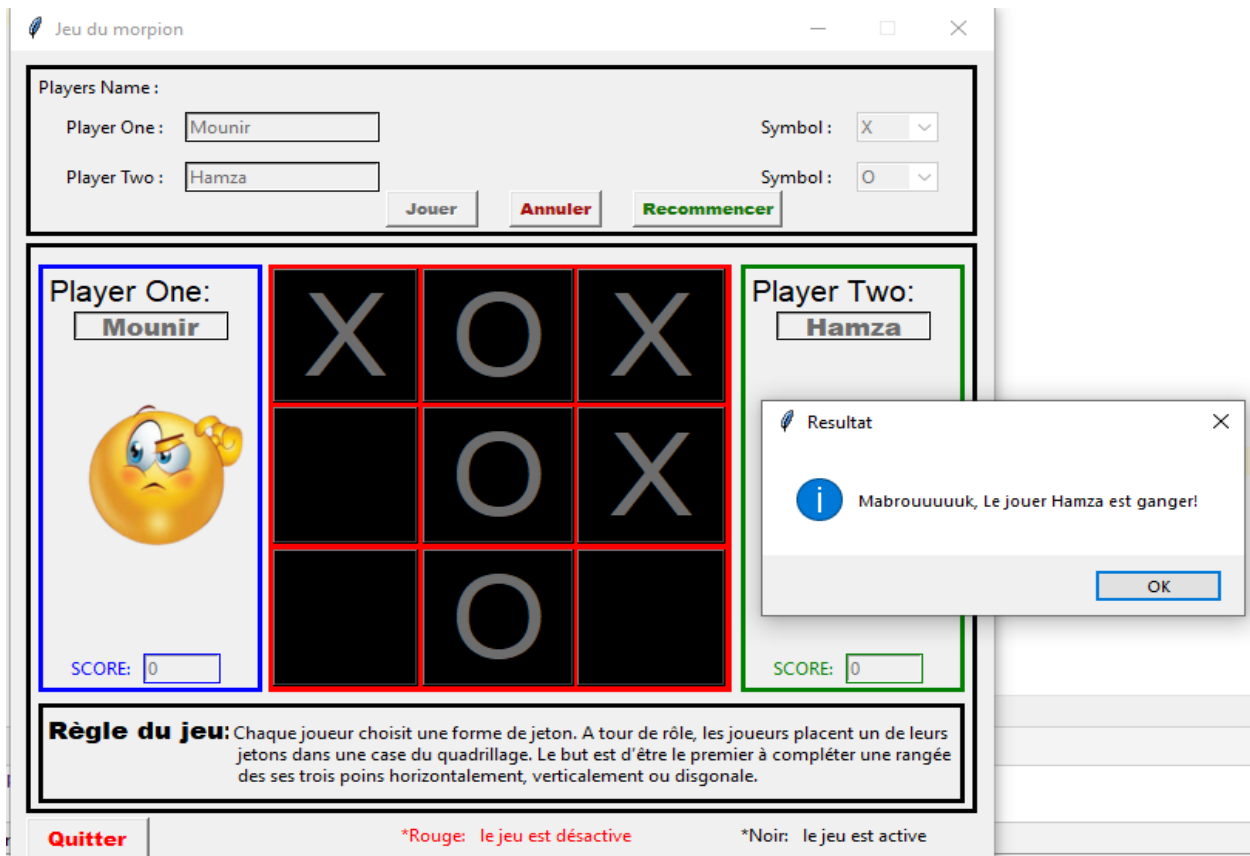
```
if ((L1[0]=='0' and L1[1]=='0' and L1[2]=='0') or (C1[0]=='0' and C1[1]=='0' and C1[2]=='0') or
    (G1[0]=='0' and G1[1]=='0' and G1[2]=='0')):

    selc = X01.get()
    name1 = p1.get()
    name2 = p2.get()
    if (selc == '0'):
        showinfo('Resultat', 'Mabrouuuuuk, Le joueur ' + name1 + ' est gagner!')
        imp1.config(image=img3)
        imp2.config(image=img2)
        wnr.config(text='Congratulations !')
        lsr.config(text='Hard luck... !')
        SCORE1 = SCORE1 + 1
        ZSCP1.config(state=NORMAL)
        ZSCP1.delete(0, END)
        ZSCP1.insert(0, SCORE1)
        ZSCP1.config(state=DISABLED)
    else:
        showinfo('Resultat', 'Mabrouuuuuk, Le joueur ' + name2 + ' est gagner!')
        imp1.config(image=img2)
        imp2.config(image=img3)
        lsr.config(text='Congratulations !')
        wnr.config(text='Hard luck... !')
        SCORE2 = SCORE2 + 1
        ZSCP2.config(state=NORMAL)
        ZSCP2.delete(0, END)
        ZSCP2.insert(0, SCORE2)
        ZSCP2.config(state=DISABLED)

    button1.config(state=DISABLED)
    button2.config(state=DISABLED)
    button3.config(state=DISABLED)
    button4.config(state=DISABLED)
    button5.config(state=DISABLED)
    button6.config(state=DISABLED)
    button7.config(state=DISABLED)
    button8.config(state=DISABLED)
    button9.config(state=DISABLED)
```

On répète les mêmes étapes qui ont vu avec le cas de tour avec symbole X

On répète toutes les étapes précédentes avec tous les boutons c'est-à-dire, on crée 9 fonctions (check1, check2, ..., check9), et on l'appelle chacun dans un bouton.



On clique sur le bouton Ok :

