



ÉCOLE CENTRALE LYON

UE INFORMATIQUE
CONCEPTION ET PROGRAMMATION OBJET
RAPPORT

Rapport du BE 5 - Le jeu du pendu

Élèves :

Mohamed BOUCHAFAA
Lucas BESSON

Enseignant :

Charles-Edmond BICHOT

28 mars 2022

Table des matières

1	Présentation du jeu à coder	2
2	Diagramme de classes UML	2
3	Présentation du code basique du jeu	3
4	Présentation du code des améliorations	3
4.1	Apparence modifiable du jeu	3
4.2	Système de fin de partie	4
4.3	Bouton Triche	4
4.4	Système de pseudos	4
5	Annexe - Code Python	5

Préambule

Afin de pouvoir profiter pleinement de l'expérience proposée, nous vous demandons d'installer le module Pygame sur python, pour cela il suffit de lancer la commande `pip install pygame` dans le terminale python de Spyder ou dans une fenêtre de terminal Anaconda (Menu démarrer → Anaconda 64bit → Anaconda PowerShell Prompt) :

`pip3 install pygame` (ou `pip install pygame`).

N.B : Si l'installation de la librairie pygame n'est pas possible ou ne se fait pas correctement il est toujours possible de lancer le jeu via un deuxième fichier .py nommé "pendu_sans_musique.py".

1 Présentation du jeu à coder

Le jeu du pendu consiste en la recherche d'un mot choisi aléatoirement lettre par lettre avec un nombre d'échecs limité (ici 10 échecs maximum). Un bouton de triche est rajouté au jeu, ce bouton permet au joueur de supprimer son échec précédent. Un système d'identification par pseudo permet également aux joueurs de connaître leur nombre de victoires et leur score (ratio du nombre de lettres trouvées sur le nombre de lettres cliquées).

2 Diagramme de classes UML

On représente les différentes classes utilisées ainsi que leurs attributs et leurs méthode dans un diagramme de classes UML.

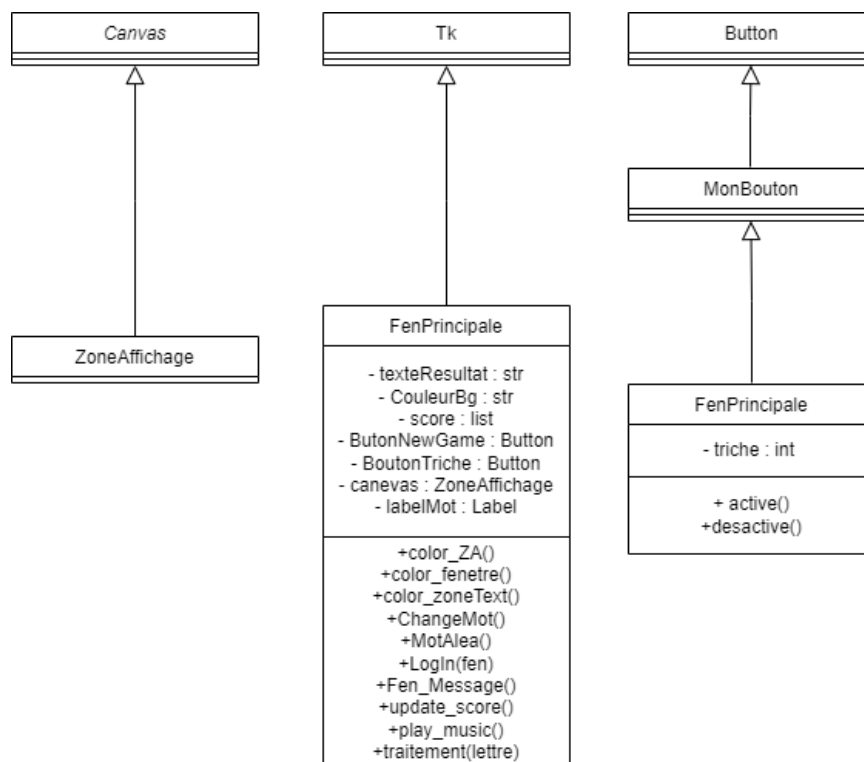


FIGURE 1 – Diagramme de classes UML

3 Présentation du code basique du jeu

Le jeu respecte le cahier des charges imposé dans les instructions relatives au BE5. En compilant le code on doit obtenir une interface similaire à celle-ci :

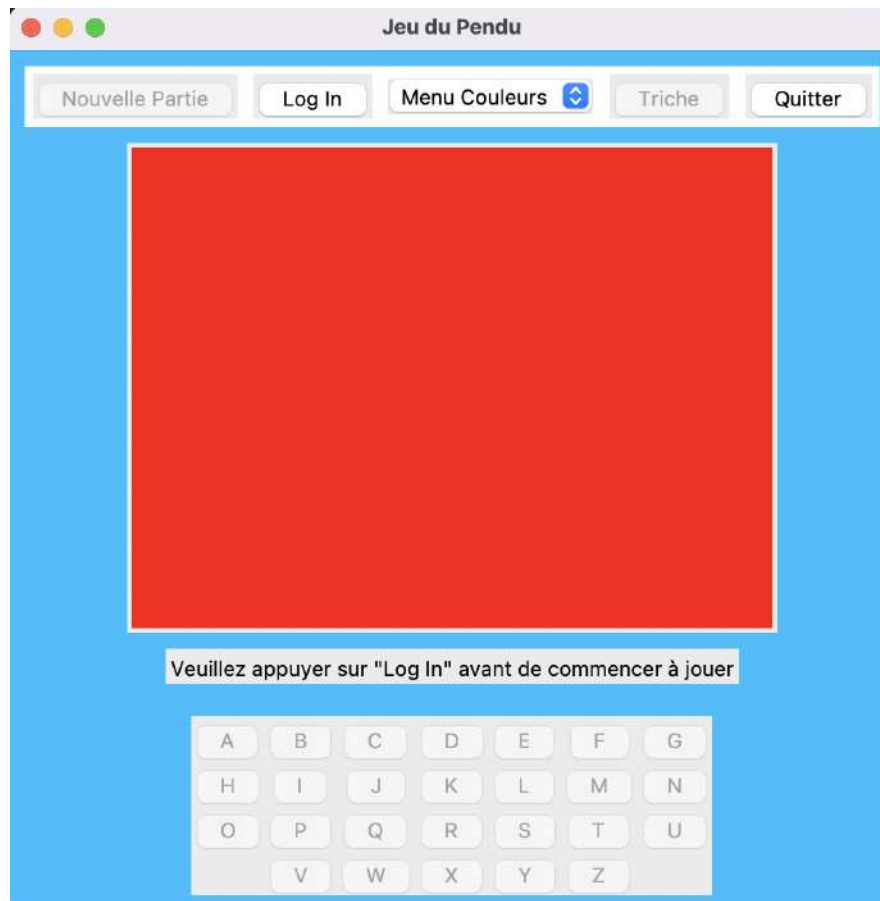


FIGURE 2 – Interface du jeu

4 Présentation du code des améliorations

Il est demandé au joueur de s'identifier avant de commencer à jouer afin de lui accorder des statistiques (nombre de partie joueur, nombre de victoire et score cumulé).

4.1 Apparence modifiable du jeu

Le jeu possède un menu déroulant contenant des boutons "couleurs" qui permettent de personnaliser les couleurs des différentes fenêtres de la scène, on peut modifier le fond où apparaît le pendu, le fond du texte ainsi que le fond de la fenêtre principale. On utilise pour cela différentes méthodes faisant appelle à un objet de la classe colorchooser de Tkinter qui permet de proposer toutes les couleurs.

4.2 Système de fin de partie

Lorsque le joueur a remporté une partie un message s'affiche et le félicite d'avoir gagné. Si le joueur a triché pour réussir, la triche sera reconnue et aucun point ne sera donné au joueur et la partie sera considérée comme "perdue" en terme de statistiques. Si le joueur n'a pas réussi à trouver le mot en moins de 10 essais alors un message de fin de partie s'affichera et le joueur devra relancer une nouvelle partie (via le bouton Nouvelle Partie) pour pouvoir rejouer.

4.3 Bouton Triche

Le bouton triche appelle une fonction undo qui diminue de 1 le compteur d'échecs de la partie et qui cache la dernière forme du dessin du pendu qui est apparue si il y a déjà eu un échec lors de la partie.

Le bouton triche n'est disponible que si le joueur a déjà fait au moins une erreur et que la partie n'est pas encore finie (c'est à dire que le joueur n'a pas perdu ou gagné) sinon il est grisé (disabled). L'utilisation du bouton triche change également le message de fin de jeu. Il est à noter que l'utilisation du bouton triche ne fait pas gagner de points au joueur étant donné qu'il a triché.

4.4 Système de pseudos

Au lancement de l'application, l'utilisateur doit s'identifier (ou s'inscrire) et il sera alors enregistré dans la base de donnée. Pour cela un bouton "LogIn" a été codé et permet de faire apparaître une fenêtre où il est possible d'entrer son pseudo (au clavier). Si le pseudo (ou identifiant) de l'utilisateur est déjà dans la base de donnée alors il sera reconnu et pourra améliorer ses anciens scores. Sinon on crée un nouveau joueur avec un nouvel ID et des statistiques vierges.

Il est possible de changer d'utilisateur (ou d'en créer un nouveau) sans avoir à ré ouvrir l'application. L'utilisateur n'est reconnu qu'avec son pseudo, il n'y pas de système de mot de passe.

Enfin, à chaque fin de partie les statistiques du joueur sont affichées dans la console python comme suit :

```
[Running] python -u "/Users/bouchafaa/Documents/Centrale 1A/S6/informatique/
-----
Votre score est :
nombre de partie jouées : 3
nombre de partie gagnées : 1
Score cumulé : 0.47058823529411764
```

FIGURE 3 – Récapitulatif des scores

5 Annexe - Code Python

Voici le code du programme :

```
1 from tkinter import *
2 from tkinter import colorchooser
3 import tkinter
4 #from tkinter import _EntryIndex
5 from tkinter.tix import DisplayStyle
6 from turtle import left
7 from formes import *
8 from random import randint
9 import sqlite3
10 import pygame
11
12
13
14 class ZoneAffichage(Canvas):
15     def __init__(self, parent, largeur, hauteur, couleur):
16         Canvas.__init__(self, parent, width=largeur, height=hauteur)
17         self.configure(bg=couleur)
18
19
20 #=====
21 class MonButton(Button):
22     def __init__(self, fenetre, nom, etat, methode):
23         Button.__init__(self, fenetre, text=nom)
24         self.config(state=etat)
25         self.lettre=nom
26         self.config(command=self.desactive)
27         self.clic=False
28         self.methode=methode
29         self.__triche=0
30
31     def active(self):
32         self.config(state=NORMAL)
33     def desactive(self):
34         self.config(state=DISABLED)
35         self.clic=True
36         self.methode(self.lettre)
37
38
39 class FenPrincipale(Tk):
40     def __init__(self):
41         Tk.__init__(self)
42         self.title("Jeu du Pendu")
43         self.configure(bg='deep sky blue')
44         self.ChangeMot()
45         self.__texteResultat = StringVar()
46         self.__CouleurBg="red"
47         pygame.mixer.init()
```

```

48     self.__score=[0,0,0]
49     self.__compteurP=0
50     self.play_music()
51     #=====
52     #=====-- Partie Data Base
53     #=====
54     self.__conn=sqlite3.connect("penduu.db")
55
56     #=====
57     #=====-- Partie Bouton en haut
58     #=====
59     #On Cre la fenetre des boutons nouvelle partie et quitter
60     Frame1 = Frame(bg='white')
61     Frame1.pack(side=TOP, padx=10 ,pady=10)
62
63     self.__ButonNewGame = Button(Frame1,bg="white",text="Nouvelle Partie")
64     self.__ButonNewGame.pack(side=LEFT, padx=5, pady=5)
65     self.__ButonNewGame.config(command=self.MotAlea)
66     self.__ButonNewGame.config(state=DISABLED)
67
68     self.__BoutonNewP = Button(Frame1, bg="white",text="Log In")
69     self.__BoutonNewP.pack(side=LEFT, padx=5, pady=5)
70     self.__BoutonNewP.config(command=self.Fen_Message)
71
72     MNU_OngletA = tkinter.Menu ( Frame1 , text = "Menu Couleurs" )
73     MNU_OptionA = tkinter.Menu ( MNU_OngletA )
74     MNU_OptionA.add_command ( label = "Couleur zone Pendu" , command =
75         self.color_ZA )
76     MNU_OptionA.add_command ( label = "Couleur arrire plan" , command =
77         self.color_fenetre )
78     MNU_OptionA.add_command ( label = "Couleur zone Texte" , command =
79         self.color_zoneText )
80     MNU_OngletA [ "menu" ] = MNU_OptionA
81     MNU_OngletA.pack(side=LEFT, padx=5, pady=5)
82
83     self.__BoutonTriche = Button(Frame1,bg="white",text="Triche")
84     self.__BoutonTriche.pack(side=LEFT, padx=5, pady=5)
85     self.__BoutonTriche.config(command=self.undo)
86     self.__BoutonTriche.config(state=DISABLED)
87
88     boutonQuitter = Button(Frame1,bg="white", text='Quitter')
89     boutonQuitter.pack(side=LEFT, padx=5, pady=5)
90     boutonQuitter.config(command=self.destroy)
91

```

```

92      #===== - Partie Zone Affichage du pendu
93      #on cre la zone d'affichage du pendu
94
95      self.__canevas=ZoneAffichage(self, 400,300,self.__CouleurBg)
96      self.__canevas.pack(side=TOP)
97      # Base, Poteau, Traverse, Corde
98      self.listeDesRect=[
99      Rectangle(self.__canevas, 50, 270, 200, 26, "brown"),\
100     Rectangle(self.__canevas, 87, 83, 26, 200, "brown"),\
101     Rectangle(self.__canevas, 87, 70, 150, 26, "brown"),\
102     Rectangle(self.__canevas, 183, 67, 10, 40, "brown"),\
103     Ellipse(self.__canevas, 188, 120, 17, 22, "black"),\
104     Rectangle(self.__canevas, 175, 143, 26, 60, "black"),\
105     Rectangle(self.__canevas, 133, 150, 40, 10, "black"),\
106     Rectangle(self.__canevas, 203, 150, 40, 10, "black"),\
107     Rectangle(self.__canevas, 175, 205, 10, 40, "black"),\
108     Rectangle(self.__canevas, 191, 205, 10, 40, "black")]
109
110
111
112
113
114
115     #===== - endroit o le mot s'affiche
116     #on cre la fenetre du mot qu'on doit dcouvrir
117     self.__Frame2 = Frame(bg="red")
118     self.__Frame2.pack(side=TOP, padx=10 ,pady=10)
119     self.__texteResultat.set('Veuillez appuyer sur "Log In" avant de
120     commencer jouer')
121     self.__labelMot =
122         Label(self.__Frame2,fg="black",textvariable=self.__texteResultat)
123     self.__labelMot.pack()
124
125     #===== - On cre ici les boutons pour
126     jouer =====
127     #On Cre la fenetre du clavier
128     Frame3 = Frame()
129     Frame3.pack(side=TOP, padx=10 ,pady=10)
130
131     self.liste_bouton=[MonButton(Frame3,
132         chr(ord('A')+i),DISABLED,self.traitement) for i in range(0,26)]
133     i=0
134     for bouton in self.liste_bouton:
135         bouton.grid(row=i//7, column=i%7)
136         if i>=21:
137             bouton.grid(row=i//7, column=(i%7) +1)
138         i=i+1

```



```
136
137 #===== - Partie menu couleur
138     def color_ZA(self):
139         colours=colorchooser.askcolor(title="Tkinter Color Chooser")
140         self.__canevas.configure(bg=colours[1])
141
142     def color_fenetre(self):
143         colours=colorchooser.askcolor(title="Tkinter Color Chooser")
144         self.configure(bg=colours[1])
145
146     def color_zoneText(self):
147         colours=colorchooser.askcolor(title="Tkinter Color Chooser")
148         self.__labelMot.configure(bg=colours[1])
149
150
151     def ChangeMot(self):
152         f = open('mots.txt', 'r')
153         s = f.read()
154         self.__mots = s.split('\n')
155         f.close()
156
157 #===== - Code du NewGame et triche
158     def MotAlea(self):
159         self.__score[2]=0
160         self.__gg=0
161         self.__triche=0
162         self.__cpt=0
163         self.__cptWin=0
164         self.__BoutonTriche.config(state=DISABLED)
165         nb = randint(1, len(self.__mots))
166         self.__lemot=self.__mots[nb]
167         #print(self.__lemot)
168         self.__MotHidden=''*len(self.__mots[nb])
169         self.__texteResultat.set('le mot est : '+self.__MotHidden)
170         self.__mot_affiche=list(self.__MotHidden)
171         for r in self.listeDesRect:
172             r.setState("hidden")
173         for i in range(0,26):
174             self.liste_bouton[i].config(state=NORMAL)
175         self.__compteurP+=1
176
177     def undo(self):
178         self.listeDesRect[self.__cpt - 1].setState("hidden")
179         self.__cpt-=1
180         if self.__cpt<=0:
181             self.__BoutonTriche.config(state=DISABLED)
182             self.__triche=1
183
```

```
184 #===== Partie code de la fenetre Log In
    =====
185
186 def LogIn(self, fen=[]):
187     self.__compteurP=0
188     global entry
189     self.__pseudo= self.__entryLogIn.get()
190     curseur = self.__conn.cursor()
191     a = "SELECT Pseudo FROM Joueurs WHERE Pseudo =
        '{}'.format(self.__pseudo)
192     curseur.execute(a)
193     liste_P=curseur.fetchall()
194     b = "SELECT ID FROM Joueurs"
195     curseur.execute(b)
196     liste_ID=curseur.fetchall()
197     self.__texteResultat.set('Veuillez appuyer sur "Nouvelle Partie" pour
        commencer jouer')
198     if self.__pseudo==' ' or self.__pseudo==' ':
199         self.__labelMsg.configure(text='Veuillez entrer un pseudo valide')
200     else:
201         if liste_P == []:
202             self.__labelMsg.configure(text='Bienvenue parmi nous, {} \n Vous
                pouvez fermer cette fenetre et commencer
                jouer'.format(self.__pseudo))
203             if liste_ID==[]:
204                 self.__tempID=1
205                 self.__newID=[1]
206             else:
207                 self.__tempID=int(liste_ID[-1][0]) + 1
208                 self.__newID=[self.__tempID]
209             c = "INSERT INTO Joueurs
                VALUES('{}'.format(self.__tempID, self.__pseudo)
210             curseur.execute(c)
211             self.__conn.commit()
212             z = "INSERT INTO Partie
                VALUES('{}'.format(self.__tempID, 0, 0, 0)
213             curseur.execute(z)
214             self.__conn.commit()
215
216             self.__FrameMsg.unbind('<Return>')
217             self.__boutonEntrer.config(state=DISABLED)
218             self.__ButonNewGame.config(state=NORMAL)
219         else:
220             b = "SELECT ID FROM Joueurs WHERE Pseudo =
                '{}'.format(self.__pseudo)
221             curseur.execute(b)
222             self.__newID=curseur.fetchone()
223             self.__labelMsg.configure(text='Re-Bienvenue parmi nous,
                {} \n Vous pouvez fermer cette fenetre et commencer
                jouer'.format(self.__pseudo))
```

```

224         self.__FrameMsg.unbind('<Return>')
225         self.__boutonEntrer.config(state=DISABLED)
226         self.__ButonNewGame.config(state=NORMAL)
227
228
229     def Fen_Message(self):
230         self.__FrameMsg=Toplevel(self)
231         self.__FrameMsg.config(bg='cyan')
232         self.__FrameMsg.title("Log In")
233         self.__FrameMsg.geometry("750x250")
234
235         self.__boutonEntrer = Button(self.__FrameMsg,bg="white", text='Entrer')
236         boutonQuitter = Button(self.__FrameMsg,bg="white", text='Fermer')
237         boutonQuitter.config(command=self.__FrameMsg.destroy)
238
239         self.__labelMsg=Label(self.__FrameMsg, text="Veuillez entrer votre
                pseudo", font=("Courier 22 bold"))
240         self.__labelMsg.pack()
241         self.__entryLogIn= Entry(self.__FrameMsg, width= 40)
242         self.__entryLogIn.focus_set()
243
244
245
246         self.__entryLogIn.pack()
247
248
249         self.__boutonEntrer.pack(side=TOP, padx=5, pady=5)
250         boutonQuitter.pack(side=TOP, padx=5, pady=5)
251
252         self.__FrameMsg.bind('<Return>',self.LogIn)
253         self.__boutonEntrer.config(command=self.LogIn)
254         #self.__ButonNewP.config(state=DISABLED)
255
256 #=====Methode pour mettre jour le score dans la data
        base =====
257     def update_score(self):
258         curseur = self.__conn.cursor()
259         t = "SELECT NbPartie,NbWin,Score FROM Partie WHERE ID =
                '{}'.format(self.__newID[0])
260         curseur.execute(t)
261         temp_score=curseur.fetchone()
262         if temp_score[0]==None:
263             temp_score=[0,0,0]
264         self.__score[0]=temp_score[0]+1
265         self.__score[1]=temp_score[1]
266         if self.__gg==1:
267             self.__score[1]+=1
268         self.__score[2]+=temp_score[2]
269         a = "UPDATE Partie SET NbPartie = '{} WHERE ID =
                '{}'.format(temp_score[0]+1,self.__newID[0])

```

```

270         curseur.execute(a)
271         self.__conn.commit()
272         a = "UPDATE Partie SET NbWin = '{}' WHERE ID =
           '{}'.format(self.__score[1],self.__newID[0])
273         curseur.execute(a)
274         self.__conn.commit()
275         a = "UPDATE Partie SET Score = '{}' WHERE ID =
           '{}'.format(self.__score[2],self.__newID[0])
276         curseur.execute(a)
277         self.__conn.commit()
278         #=====-- Partie code des musiques et
           sons-----
279
280         def play_music(self):
281             pygame.mixer.music.load("menu.mp3")
282             pygame.mixer.music.play(loops=100)
283
284         #=====--Fonction traitement
           -----
285
286         def traitement(self,lettre):
287             lettres_dumot=list(self.__lemot)
288             if self.__cpt<0:
289                 self.__cpt=0
290             #On compte le nombre de fois que le joueur choue  trouver une bonne
                lettre
291             if lettre not in lettres_dumot:
292                 self.listeDesRect[self.__cpt].setState("normal")
293                 self.__cpt+=1
294                 self.__BoutonTriche.config(state=NORMAL)
295
296
297
298             #Si le joueur trouve une bonne lettre on l'affiche :
299             for i in range(0,len(lettres_dumot)):
300                 if lettres_dumot[i] == lettre:
301                     self.__mot_affiche[i]=lettre
302                     self.__texteResultat.set('le mot est : '+'
                        ''.join(self.__mot_affiche))
303                     self.__cptWin+=1
304             #Cette partie du code permet de faire une fin de partie
305             if self.__cpt==10:
306                 for bouton in self.liste_bouton:
307                     bouton.config(state=DISABLED)
308                 self.__texteResultat.set('C\'est une fin de game pour vous. Le mot
                    tait {}'.format("".join(lettres_dumot)))
309                 self.__BoutonTriche.config(state=DISABLED)
310                 self.update_score()
311                 print("-----")
312                 print("Votre score est :")

```

```
313         print("nombre de partie joues : {}".format(self.__score[0]))
314         print("nombre de partie gagnes : {}".format(self.__score[1]))
315         print("Score cumul : {}".format(self.__score[2]))
316     if self.__cptWin==len(lettres_dumot):
317
318         self.__score[2]=self.__cptWin/(self.__cptWin + self.__cpt)
319         for bouton in self.liste_bouton:
320             bouton.config(state=DISABLED)
321         if self.__triche==0:
322             self.__texteResultat.set('Bravo vous avez gagn! Le mot tait
323             bien {}'.format("".join(lettres_dumot)))
324             self.__BoutonTriche.config(state=DISABLED)
325             self.__gg=1
326             self.update_score()
327             print("-----")
328             print("Votre score est :")
329             print("nombre de partie joues : {}".format(self.__score[0]))
330             print("nombre de partie gagnes : {}".format(self.__score[1]))
331             print("Score cumul : {}".format(self.__score[2]))
332         if self.__triche==1:
333             self.__texteResultat.set("A vaincre sans pril on triomphe sans
334             gloire. En trichant vous ne gagnez pas "+\
335             "de points.\nRejouez sans tricher pour en gagner.\nLe mot
336             tait {}".format("".join(lettres_dumot)))
337             self.__score[2]=0
338             self.__BoutonTriche.config(state=DISABLED)
339             self.update_score()
340             print("-----")
341             print("Votre score est :")
342             print("nombre de partie joues : {}".format(self.__score[0]))
343             print("nombre de partie gagnes : {}".format(self.__score[1]))
344             print("Score cumul : {}".format(self.__score[2]))
345
346
347
348
349
350 if __name__ == "__main__":
351     fen = FenPrincipale()
352     fen.mainloop()
```