

```

1  import pygame
2  import time
3  import random
4  import sqlite3
5  import update2
6
7  import lvl
8  import tileset
9
10 class play:
11     """
12     classe permettant de creer le personnage du joueur et la fenetre pygame
13     """
14     def __init__(self):
15         global joueur
16         global login
17         self.height=736
18         self.width=512
19
20         self.win = pygame.display.set_mode((self.width,self.height))#Permet de
21         stocker la fenetre
22
23
24
25         starters=[pygame.transform.scale(pygame.image.load("./BattleScene/front/001.gif"), (int(512/3),int(512/3))),pygame.transform.scale(pygame.image.load("./BattleScene/front/004.gif"), (int(512/3),int(512/3))),pygame.transform.scale(pygame.image.load("./BattleScene/front/007.gif"), (int(512/3),int(512/3)))]
26
27         pokeball=pygame.transform.scale(pygame.image.load("pokeball.png"), (int(512/3),int(512/3)))
28         chosen=25#le pokémon choisi par défaut est pikachu
29         otime=time.time()#relève le temps au lancement du programme
30         tl=15#temps restant à l'utilisateur pour choisir son pokémon
31         while tl>0:#donne 15 secondes au joueur pour choisir son pokémon de départ
32             for event in pygame.event.get():
33                 if event.type == pygame.QUIT:
34                     global run
35                     run=False
36                 self.win.fill((255,255,255))
37                 tl=15+otime-time.time()
38                 self.win.blit(pygame.font.SysFont("monospace", 50).render(str(int(tl)), 1, (0,0,0)), (240, 0))
39                 pygame.draw.rect(self.win,(0,0,0),((40,60, 432, 50)))
40                 pygame.draw.rect(self.win,(255,255,255),((50,70, 412, 30)))
41                 pygame.draw.rect(self.win,(255,0,0),((50,70, (412/15)*tl, 30)))
42                 self.win.blit(pygame.font.SysFont("monospace", 20).render("Choisissez votre pokémon de départ:", 1, (0,0,0)), (0,150))
43                 if (736/2-512/6<pygame.mouse.get_pos()[1]<736/2+512/6 and pygame.mouse.get_pos()[0]<512/3) or chosen==1:
44                     self.win.blit(starters[0], (0, int(736/2-512/6)))
45                 else:
46                     self.win.blit(pokeball, (0, int(736/2-512/6)))
47
48                 if (int(736/2-512/6)<pygame.mouse.get_pos()[1]<int(736/2+512/6) and 512/3<pygame.mouse.get_pos()[0]<512*2/3) or chosen==4:
49                     self.win.blit(starters[1], (512/3, int(736/2-512/6)))
50                 else:
51                     self.win.blit(pokeball, (512/3, int(736/2-512/6)))
52
53                 if (int(736/2-512/6)<pygame.mouse.get_pos()[1]<int(736/2+512/6) and 512*2/3<pygame.mouse.get_pos()[0]) or chosen==7:
54                     self.win.blit(starters[2], (512*2/3, int(736/2-512/6)))
55                 else:
56                     self.win.blit(pokeball, (512*2/3, int(736/2-512/6)))
57
58                 if pygame.mouse.get_pressed()[0]==1:
59                     if 736/2-512/6<pygame.mouse.get_pos()[1]<736/2+512/6:
60                         if pygame.mouse.get_pos()[0]<512/3:
61                             chosen=1#pokémon sélectionné
62                         elif pygame.mouse.get_pos()[0]<512*2/3:

```

```

61             chosen=4
62         else:
63             chosen=7
64         if pygame.mouse.get_pressed()[2]==1:
65             chosen=0
66         pygame.display.update()
67         pygame.time.delay(10)
68
69     #Interaction avec la base de donnée
70     conn = sqlite3.connect('db.db')
71     c = conn.cursor()
72     #S'assure que les tables contenant les données de la partie soient vides
73     c.execute("DELETE FROM pnj")
74     c.execute("DELETE FROM personnage")
75     c.execute("DELETE FROM sac")
76     c.execute("DELETE FROM objets")
77     c.execute("DELETE FROM equipe")
78     c.execute("DELETE FROM pokemon")
79     c.execute("DELETE FROM attaquespokemon")
80
81     #attribue un numéro d'équipe au joueur
82     team=1
83     for row in c.execute("SELECT * FROM equipe"):
84         team+=1
85     self.numteam=team
86     team=str(team)
87     c.execute("INSERT INTO equipe(idEquipe) VALUES ('+team+')") #Créer l'équipe
    du joueur
88
89     #attribue son pokémon de départ au joueur
90     for row in c.execute("SELECT * FROM basepokemon WHERE id =
    "+str(chosen)):#Récupère les données du pokemon generique corrspondant à
    notre starter
91         base=row
92     npokemon=1
93     for row in c.execute("SELECT * FROM pokemon"):
94         npokemon+=1
95     npokemon=str(npokemon)
96     c.execute("INSERT INTO pokemon VALUES
    ('+npokemon+', "+str(base[0])+", '"+base[1]+"' , '"+base[2]+"' , 5, "+str(base[3])+",
    "+str(base[3])+", "+str(base[4])+", "+str(base[5])+", "+str(base[6])+", "+str(base
    [7])+", "+team+")") #Attribue les données génériques a notre pokemon de départ
97
98     #attribue sa première attaque au pokémon de départ du joueur
99     nattack=1
100    for row in c.execute("SELECT * FROM attaquespokemon"):
101        nattack+=1
102    c.execute("INSERT INTO attaquespokemon VALUES
    ('+str(nattack)+", 59, "+npokemon+")")
103    #attribue son sac au joueur
104    sac=1
105    for row in c.execute("SELECT * FROM sac"):
106        sac+=1
107    self.nsac=sac
108    sac=str(sac)
109    c.execute("INSERT INTO sac VALUES ('+sac+')")
110    c.execute("INSERT INTO objets VALUES (NULL, 'Pokeball', 0, "+sac+")")
111    c.execute("INSERT INTO objets VALUES (NULL, 'Superball', 0, "+sac+")")
112    c.execute("INSERT INTO objets VALUES (NULL, 'Hyperball', 0, "+sac+")")
113    c.execute("INSERT INTO objets VALUES (NULL, 'Masterball', 0, "+sac+")")
114    c.execute("INSERT INTO objets VALUES (NULL, 'Potion', 0, "+sac+")")
115    c.execute("INSERT INTO objets VALUES (NULL, 'SuperPotion', 0, "+sac+")")
116    c.execute("INSERT INTO objets VALUES (NULL, 'HyperPotion', 0, "+sac+")")
117    c.execute("INSERT INTO objets VALUES (NULL, 'PotionMax', 0, "+sac+")")
118    c.execute("INSERT INTO objets VALUES (NULL, 'Rappel', 0, "+sac+")")
119    c.execute("INSERT INTO objets VALUES (NULL, 'RappelMax', 0, "+sac+")")
120    conn.commit()
121    conn.close()
122
123
124
125

```

```

126 self.x = -300#position x du joueur sur la map en pixels
127 self.y = 0#position y du joueur sur la map en pixels
128 self.vel = 3#vitesse de déplacement
129
130 self.tileset = pygame.image.load("tileset.png").convert_alpha()#image
    contenant toutes les parties de la map
131
132 self.level=lvl.niv#contient la disposition de la map
133
134 self.xs=len(self.level[0])*32#largeur de la map en tuiles
135 self.ys=len(self.level)*32#hauteur de la map en tuiles
136
137 self.xtj=abs(self.x)/32+self.width/64#position x du joueur sur la map en
    tuiles
138 self.ytj=abs(self.y)/32+self.height/64#position y du joueur sur la map en
    tuiles
139 self.lt=[0,0]
140 self.collision=tileset.tilescollision#contient un liste qui détermine si une
    tuile est traversable ou non
141
142 self.trainers = pygame.image.load("characters.png").convert_alpha()#charge
    l'image contenant les personnages du jeu
143 self.trainers = pygame.transform.scale(self.trainers,
    (self.trainers.get_width()*2, self.trainers.get_height()*2))
144
145
146 self.textbox=pygame.image.load("./BattleScene/textbox.png")
147
148 self.textbox=pygame.transform.scale(self.textbox,(self.textbox.get_width()*2,s
    elf.textbox.get_height()*2))
149 self.message=False#Vrai si il y à un message à afficher
150
151 self.eact={'[2, 2]':'print("ceci est un sapaing")','[4,
    20]':'self.pickpokeball(ftt)'}#dictionnaire contenant les interaction avec
    la tuile choisie
152 self.way=""#initialisation de la direction du personnage du joueur à nulle
153
154 #génère les PNJ (personnages non joueurs)
155 self.pnjs=[]
156 for i in range(0,len(self.level)):
157     for j in range(0,len(self.level[i])):
158         for k in range(0,len(self.level[i][j])):
159             if len(self.level[i][j][k])==3:
160
161                 self.pnjs.append(createpnj(i,self.win,[j*32,i*32],self.level[i
162                     ][j][k][2][0],self.level[i][j][k][2][1],self.level[i][j][k][2
163                         ][2],self.vel,self.level[i][j][k][2][3]))
164
165
166 def move(self):
167     """
168     methode permettant de déplacer le personnage sur la map et de le faire
    interagir avec l'environnement
169     """
170     global joueur
171     global login
172     keys = pygame.key.get_pressed()
173     #En fonction de flèche enfoncée, si la vois est libre dans la direction de
    la flèche, déplace le personnage du joueur dans direction
174     #playertilepos contient la tuile du personnage à afficher
175     if keys[pygame.K_LEFT]:
176         self.way="L"
177         if self.x<0 and self.hocollision(self.way):
178             self.x += self.vel
179             if int((time.time()*4)%2)==0:
180                 playertilepos=[0,1344]
181             else:
182                 playertilepos=[0,1472]
183         else:
184             playertilepos=[0,1408]
185         self.message=False
186     elif keys[pygame.K_RIGHT]:

```

```

183     self.way="R"
184     if self.x>self.width-self.xs+50 and self.hocollision(self.way):
185         self.x -= self.vel
186         if int((time.time()*4)%2)==0:
187             playertilepos=[64,1344]
188         else:
189             playertilepos=[64,1408]
190     else:
191         playertilepos=[64,1280]
192     self.message=False
193 elif keys[pygame.K_UP]:
194     self.way="U"
195     if self.y<0 and self.hocollision(self.way):
196         self.y += self.vel
197         if int((time.time()*4)%2)==0:
198             playertilepos=[128,1280]
199         else:
200             playertilepos=[64,1472]
201     else:
202         playertilepos=[0,1280]
203     self.message=False
204 elif keys[pygame.K_DOWN]:
205     self.way="D"
206     if self.y>self.height-self.ys+21 and self.hocollision(self.way):
207         self.y -= self.vel
208         if int((time.time()*4)%2)==0:
209             playertilepos=[128,1408]
210         else:
211             playertilepos=[128,1472]
212     else:
213         playertilepos=[128,1344]
214     self.message=False
215 else:
216     if self.way=="L":
217         playertilepos=[0,1408]
218     elif self.way=="R":
219         playertilepos=[64,1280]
220     elif self.way=="U":
221         playertilepos=[0,1280]
222     else:
223         playertilepos=[128,1344]
224
225 #efface la fenetre
226 self.win.fill((255,255,255))
227
228 #détermine la tuile sur laquelle le joueur se trouve
229 self.xtj=abs(self.x)/32+self.width/64
230 self.ytj=abs(self.y)/32+self.height/64
231
232 #si la touche E est pressée
233 if keys[pygame.K_e]:
234     if self.ce:#si l'utilisateur peut appuyer sur e
235         self.ce=False
236         xf=self.x#coordonnée x de la tuile en face du joueur en pixels
237         yf=self.y#coordonnée y de la tuile en face du joueur en pixels
238         if self.way=="L":
239             xf+=20
240             yf-=16
241         elif self.way=="R":
242             xf-=20
243             yf-=16
244         elif self.way=="U":
245             yf+=10
246         else:
247             yf-=32
248         self.yft=int(-yf/32+self.height/64)#xf coordonnée x de la tuile en
249         face du joueur
250         self.xft=int(-xf/32+self.width/64)#yf coordonnée y de la tuile en
251         face du joueur
252         for i in range(0,len(self.pnjs)):#Interaction avec un PNJ
253             if int(self.xft)==self.pnjs[i].pos[0]/32 and
254                 int(self.yft)==self.pnjs[i].pos[1]/32:

```

```

252         if self.pnjs[i].active:
253             if self.pnjs[i].trainer:
254                 self.pnjs[i].duel()
255             else:
256
257                 objet=random.choice(['Pokeball','Superball','Hyperball',
258                                     '','Masterball','Potion','SuperPotion','HyperPotion','P
259                                     otionMax','Rappel','RappelMax'])#objet à donner
260                 quant=random.randint(1,3)#quantité d'objets à donner
261                 self.message=["Prend ces "+str(quant)+str(objet),"Tu
262                             en aura plus besoin que moi"]
263
264                 conn = sqlite3.connect('db.db')
265                 c = conn.cursor()
266                 #ajouter les objets dans la base de donnée
267                 for row in c.execute("SELECT quantité FROM objets
268                                     WHERE refsac="+str(joueur.nsac)+" AND
269                                     Nom='"+objet+"'"):
270                     row=row[0]+quant
271                     c.execute("DELETE FROM objets WHERE
272                             refsac="+str(joueur.nsac)+" AND Nom='"+objet+"'")
273                     c.execute("INSERT INTO objets VALUES
274                             (NULL, '"+objet+"', '"+str(row)+"', '"+str(joueur.nsac)+
275                             "')")
276                 conn.commit()
277                 conn.close()
278                 self.pnjs[i].active=False
279             else:
280                 self.message="Je n'ai rien pour toi!"
281                 ftt=self.level[self.yft][self.xft]
282                 for i in range (len(ftt)):
283                     if str(ftt[i]) in self.eact:
284                         act=self.eact[str(ftt[i])]
285                         exec(act)
286         else:
287             self.ce=True
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303

```

```

304     for i in range(0,len(self.pnjs)):#affiche et fait bouger les pnjs qui se
        situent au dessus le joueur
305         if int(self.xtj)-19<self.pnjs[i].pos[0]/32<int(self.xtj)+20 and
            int(self.ytj)-22<self.pnjs[i].pos[1]/32<int(self.ytj)+23:
306             trainertilepos=self.pnjs[i].move([self.x,self.y])
307             if self.pnjs[i].pos[1]-32+self.y<=self.height/2-32:
308                 self.win.blit(self.trainers,
                    (self.pnjs[i].pos[0]-16+self.x,self.pnjs[i].pos[1]-32+self.y),(tra
                    inertilepos[0],trainertilepos[1],64,64))
309
310
311     self.win.blit(self.trainers,
        (self.width/2-32,self.height/2-32),(playertilepos[0],playertilepos[1],64,64))#
        affiche le personnage du joueur
312
313
314     for i in range(0,len(self.pnjs)):#affiche et fait bouger les pnjs qui se
        situent sous le joueur
315         if int(self.xtj)-19<self.pnjs[i].pos[0]/32<int(self.xtj)+20 and
            int(self.ytj)-22<self.pnjs[i].pos[1]/32<int(self.ytj)+23:
316             trainertilepos=self.pnjs[i].move([self.x,self.y])
317             if self.pnjs[i].pos[1]-32+self.y>self.height/2-32:
318                 self.win.blit(self.trainers,
                    (self.pnjs[i].pos[0]-16+self.x,self.pnjs[i].pos[1]-32+self.y),(tra
                    inertilepos[0],trainertilepos[1],64,64))
319
320     for j in [int(self.ytj)-1,int(self.ytj),int(self.ytj)+1]:#dessine les tiles
        superposables autour du joueur
321         for i in [int(self.xtj)-1,int(self.xtj),int(self.xtj)+1]:
322             for k in range(0,len(self.level[j][i])):
323                 if
                    self.collission[self.level[j][i][k][1]][self.level[j][i][k][0]][1]=
                    =1 or
                    (self.collission[self.level[j][i][k][1]][self.level[j][i][k][0]][1]
                    ==2 and (j==int(self.ytj+0.5) or j==int(self.ytj+0.8))):
324                     self.win.blit(self.tileset, (i*32+self.x,
                        j*32+self.y),(self.level[j][i][k][0]*32,self.level[j][i][k][1]
                        *32,32,32))
325
326     #Vérifie si un PNJ est encore apte au combat sinon déclare les joueur
        vainqueur
327     last=True
328     for i in self.pnjs:
329         if i.trainer:
330             if i.active:
331                 last=False
332
333     if last:
334         global run
335         run=False
336         won=True
337         while won:
338             for event in pygame.event.get():
339                 if event.type == pygame.QUIT:
340                     run=False
341                     won=False
342             self.win.fill((255,255,255))
343             self.win.blit(pygame.font.SysFont("monospace", 100).render("Gagné",
                1, (0,0,0)), (100, 100))
344             pygame.display.update()
345
346
347
348
349
350
351
352
353     update2.update2(login,'win')
354
355
356

```

```

357
358
359
360
361
362
363
364
365         #affiche un message si il y à lieu
366     if self.message!=False:
367         self.win.blit(self.textbox, (0, 0))
368         if type(self.message)==list:
369             self.win.blit(pygame.font.SysFont("monospace",
370             15).render(self.message[0], 1, (0,0,0)), (20,20))
371             self.win.blit(pygame.font.SysFont("monospace",
372             15).render(self.message[1], 1, (0,0,0)), (20,40))
373         else:
374             self.win.blit(pygame.font.SysFont("monospace",
375             15).render(self.message, 1, (0,0,0)), (20,20))
376     pygame.display.update()
377     if pygame.mouse.get_pressed()[0]==1:
378         if self.cc:
379             self.cc=False
380             self.message=False
381         else:
382             self.cc=True
383
384 def hocollosion(self,direction):#regarde si la direction vers laquelle le joueur
385     veut se diriger est libre
386     """
387     methode permettant de valider ou non le fait que le joueur puisse avancer
388     dans une direction
389     prend en paramètre direction permetant d'indiquer la direction dans laquelle
390     le joueur va
391     """
392     global joueur
393     global login
394     if direction=="L":
395         dx=[-0.4]
396         dy=[0.4,0.8]
397     elif direction=="R":
398         dx=[0.4]
399         dy=[0.4,0.8]
400     elif direction=="D":
401         dx=[-0.2,0.2]
402         dy=[1]
403     else:
404         dx=[-0.2,0.2]
405         dy=[0.2]
406     sblood=""
407     for j in range(0,len(dy)):
408         for i in range(0,len(dx)):
409             xtja=int(self.xtj+dx[i])
410             ytja=int(self.ytj+dy[j])
411             let=False
412             blood=[]
413             for k in range(0,len(self.level[ytja][xtja])):#tiles potential hitbox
414                 blood.append(self.level[ytja][xtja][k])
415
416             for k in range(0,len(self.pnjs)):#pnj hitbox
417                 if int(self.xtj)-19<self.pnjs[k].pos[0]/32<int(self.xtj)+20 and
418                 int(self.ytj)-22<self.pnjs[k].pos[1]/32<int(self.ytj)+23:
419                     if
420                         self.pnjs[k].pos[0]<(self.xtj+dx[i])*32<self.pnjs[k].pos[0]+32
421                         and
422                         self.pnjs[k].pos[1]<(self.ytj+dy[j])*32<self.pnjs[k].pos[1]+32
423                     :
424                         if direction=="L":
425                             sblood+="L"
426                         elif direction=="R":

```

```

418         sblocd+="L"
419     elif direction=="D":
420         sblocd+="D"
421     else:
422         sblocd+="D"
423
424     for k in range(0,len(blocd)):#rassemble les directions bloquées
425         sblocd+=self.collison[blocd[k][1]][blocd[k][0]][0]
426
427
428     if (direction=="L" and "R" not in sblocd) or (direction=="R" and "L" not in
sblocd) or (direction=="U" and "D" not in sblocd) or (direction=="D" and "U"
not in sblocd):
429         return True
430
431 def pickpokeball(self,ft):#récupere la pokeball au sol et attribue les items
432     """
433     methode permetant de ramasser une pokeball et son contenu pour l'y ajouter à
l'inventaire
434     prend en paramètre ft pour la tuile en face du personnage
435     """
436     global joueur
437     global login
438     del self.level[self.yft][self.xft][len(self.level[self.yft][self.xft])-1]
439
440     objet=random.choice(['Pokeball','Superball','Hyperball','Masterball','Potion',
'SuperPotion','HyperPotion','PotionMax','Rappel','RappelMax'])
441     quant=random.randint(1,3)
442     print("Vous avez trouvé ",quant,objet)
443
444     conn = sqlite3.connect('db.db')
445     c = conn.cursor()
446
447     for row in c.execute("SELECT quantité FROM objets WHERE
refsac="+str(joueur.n sac)+" AND Nom='"+objet+"'"):
448         row=row[0]+quant
449         c.execute("DELETE FROM objets WHERE refsac="+str(joueur.n sac)+" AND
Nom='"+objet+"'")
450         c.execute("INSERT INTO objets VALUES
(NULL, '"+objet+"', '"+str(row)+"', '"+str(joueur.n sac)+"'")")
451
452     conn.commit()
453     conn.close()
454
455
456
457 class createpnj:
458     """
459     classe permetant de créer un PNJ
460     prend comme paramètre :
461     num qui est le numero du PNJ
462     fenetre qui est la fenetre pygame
463     pos qui est la position du PNJ à l'origine
464     trainer qui determine si le pnj est un deresseur ou non
465     ran qui détermine la taille du cercle decrit par le pnj
466     direction qui determine la direction dans laquelle est tournée le pnj
467     vitesse qui determine la vitesse du pnj
468     clas qui determine les type du pnj
469     """
470     def __init__(self,num,fenetre,pos,trainer,ran,direction,vitesse,clas):
471         global joueur
472         global login
473         self.number=num#numéro de l'équipe du PNJ dans la DB
474         self.teamnumber=0
475         self.fen=fenetre
476         self.pos=pos#coordonnées du PNJ
477         self.trainer=trainer#détermine si je PNJ est un deresseur
478         self.range=(ran)*32#détermine la taille du tour que le PNJ décrit
479         self.direction=direction#détermine la direction dans laquelle le pnj est
tourné
480         self.ranstate=0

```



```

481 self.vit=vitesse/5#vitesse de déplacement
482 self.tclass=clas#type du PNJ
483 self.xtp=abs(self.pos[0])/32#coordonnée x du PNJ
484 self.ytp=abs(self.pos[1])/32#coordonnée y du PNJ
485 self.active=True#si le PNJ peut ineragir
486 def move(self,coop):
487     """
488     méthode permettant au PNJ de se déplacer
489     prend comme argument coop de type list d'entiers contenant les coordonnées
490     du joueur
491     """
492     global joueur
493     global login
494     """
495     méthode permettant au PNJ de se déplacer
496     prend comme argument coop de type list d'entiers contenant les coordonnées
497     du joueur
498     """
499     self.xtp=abs(self.pos[0])/32#coordonnée x de la tuile du PNJ
500     self.ytp=abs(self.pos[1])/32#coordonnée y de la tuile du PNJ
501     tilepos=[self.tclass[0]*3*64+128,self.tclass[1]*64*4+64]
502     if self.ranstate<self.range and self.active:
503         if self.collisionp(self.direction[0],coop):
504             if self.direction[0]=="L":
505                 self.pos[0]-=self.vit
506                 self.ranstate+=self.vit
507                 if int((time.time()*4)%2)==0:
508                     tilepos=[self.tclass[0]*3*64,self.tclass[1]*64*4+64]
509                 else:
510                     tilepos=[self.tclass[0]*3*64,self.tclass[1]*64*4+3*64]
511             elif self.direction[0]=="R":
512                 self.pos[0]+=self.vit
513                 self.ranstate+=self.vit
514                 if int((time.time()*4)%2)==0:
515                     tilepos=[self.tclass[0]*3*64+64,self.tclass[1]*64*4]
516                 else:
517                     tilepos=[self.tclass[0]*3*64+64,self.tclass[1]*64*4+2*64]
518             elif self.direction[0]=="U":
519                 self.pos[1]-=self.vit
520                 self.ranstate+=self.vit
521                 if int((time.time()*4)%2)==0:
522                     tilepos=[self.tclass[0]*3*64+2*64,self.tclass[1]*64*4]
523                 else:
524                     tilepos=[self.tclass[0]*3*64+64,self.tclass[1]*64*4+3*64]
525             elif self.direction[0]=="D":
526                 self.pos[1]+=self.vit
527                 self.ranstate+=self.vit
528                 if int((time.time()*4)%2)==0:
529                     tilepos=[self.tclass[0]*3*64+2*64,self.tclass[1]*64*4+2*64]
530                 else:
531                     tilepos=[self.tclass[0]*3*64+2*64,self.tclass[1]*64*4+3*64]
532             else:
533                 if self.direction[0]=="L":
534                     tilepos=[self.tclass[0]*3*64,self.tclass[1]*64*4+128]
535                 elif self.direction[0]=="R":
536                     tilepos=[self.tclass[0]*3*64+64,self.tclass[1]*64*4]
537                 elif self.direction[0]=="U":
538                     tilepos=[self.tclass[0]*3*64,self.tclass[1]*64*4]
539             else:
540                 if self.direction[1]==True:
541                     if self.direction[0]=="L":
542                         self.direction[0]="U"
543                     elif self.direction[0]=="R":
544                         self.direction[0]="D"
545                     elif self.direction[0]=="U":
546                         self.direction[0]="R"
547                     elif self.direction[0]=="D":
548                         self.direction[0]="L"
549                 else:
550                     if self.direction[0]=="L":
551                         self.direction[0]="D"
552                     elif self.direction[0]=="R":

```

```

551         self.direction[0]="U"
552     elif self.direction[0]=="U":
553         self.direction[0]="L"
554     elif self.direction[0]=="D":
555         self.direction[0]="R"
556     self.ranstate=0
557     return(tilepos)
558
559 def collisionp(self,direction,xyp):
560     """
561     méthode permettant au PNJ vérifier si le joueur est devant
562     prend comme argument xyp de type list d'entiers contenant les coordonnées du
563     joueur et direction de type chaine de caractère contenant la direction du PNJ
564     """
565     global joueur
566     global login
567     """
568     méthode permettant au PNJ vérifier si le joueur est devant
569     prend comme argument xyp de type list d'entiers contenant les coordonnées du
570     joueur et direction de type chaine de caractère contenant la direction du PNJ
571     """
572     dx=0#décalage x de la tuiles en face du PNJ
573     dy=0#décalage y de la tuiles en face du PNJ
574     vx=[0,0]#décalages x des la tuiles en face du PNJ
575     vy=[0,0]#décalages y des la tuiles en face du PNJ
576     if direction=="L":
577         dx=0.5
578         vx[0]=64
579     elif direction=="R":
580         dx=-0.5
581         vx[1]=64
582     elif direction=="D":
583         dy=-1
584         vy[1]=64
585     else:
586         dy=1
587         vy[0]=64
588     if
589     -xyp[0]+self.fen.get_width()/2+16*dx-48-vx[1]<self.pos[0]<-xyp[0]+self.fen.get
590     _width()/2+16*dx+16+vx[0] and
591     -xyp[1]+self.fen.get_height()/2+16*dy-16-vy[1]<self.pos[1]<-xyp[1]+self.fen.ge
592     t_height()/2+16*dy+16+vy[0]:
593         if self.trainer:
594             self.duel()
595         else:
596             return True
597
598 def duel(self):
599     global login
600     """
601     méthode permettant au PNJ d'engager le combat contre le joueur
602     """
603     fight=initfight(self.fen,self)
604     fight.fight()
605
606 def createteam(self,typ,lvl):
607     global joueur
608     global login
609     """
610     méthode permettant de créer un équipe au PNJ
611     pernant comme argument typ de type chaine de caractère contenant le style
612     de dresseur et lvl de type entier pour le lvl du pokémon de l'équipe du PNJ
613     """
614     conn = sqlite3.connect('db.db')
615     c = conn.cursor()
616
617     team=1
618     for row in c.execute("SELECT * FROM equipe"):
619         team+=1
620     self.teamnumber=team
621     team=str(team)

```

```

616 c.execute("INSERT INTO equipe(idEquipe) VALUES (" + team + ")") #Cr  er l'  quipe
    du pnj
617 for row in c.execute("SELECT * FROM basepokemon WHERE id =
    " + str(random.randint(1,151))) : #R  cup  re les donn  es du pokemon
618     base=row
619     npokemon=1
620     for row in c.execute("SELECT * FROM pokemon") :
621         npokemon+=1
622     npokemon=str(npokemon)
623     c.execute("INSERT INTO pokemon VALUES
        (" + npokemon + ", " + str(base[0]) + ", " + str(base[1]) + ", " + str(base[2]) + ", " + str(lvl) + ", " + str(
        base[3]) + ", " + str(base[3]) + ", " + str(base[4]) + ", " + str(base[5]) + ", " + str(base[6]) + "
        , " + str(base[7]) + ", " + team + ")") # + ", " + str(base[7]) #####Attribue les donn  es
        g  n  riques a notre starter
624     nattack=1
625     for row in c.execute("SELECT * FROM attaquespokemon") :
626         nattack+=1
627     c.execute("INSERT INTO attaquespokemon VALUES
        (" + str(nattack) + ", 1, " + team + ")") #attribue la premi  re attaque au pokemon du pnj
628
629     conn.commit()
630     conn.close()
631
632
633
634
635 class initfight:
636     """
637     classe permettant d'initialiser un combat
638     prenant comme arguments fen qui est la fen  tre pygame et pnjo qui est le PNJ
        adversaire
639     """
640     def __init__(self, fen, pnjo) :
641         global joueur
642         global login
643         self.win=fen
644         self.pnj=pnjo
645
646
647         self.Pteam=[] #  quipe du joueur
648         self.Bteam=[] #pokemon sauvage/  quipe du pnj
649         #r  cup  re toutes les donn  es concernant le joueur et le PNJ concern  
650         conn = sqlite3.connect('db.db')
651         c = conn.cursor()
652         nbpoke=0
653         for row in c.execute("SELECT * FROM pokemon WHERE appartientequipe =
            " + str(joueur.numteam)) :
654             nbpoke+=1
655             self.Pteam.append([])
656             for i in range(0, len(row)) :
657                 self.Pteam[nbpoke-1].append(row[i])
658         for i in range(0, len(self.Pteam)) :
659             self.Pteam[i].append([])
660             for row in c.execute("SELECT Attaques_idAttaques FROM attaquespokemon
                WHERE appartientpokemon = " + str(i+1)) :
661                 self.Pteam[i][len(self.Pteam[i])-1].append(row[0])
662         for i in range(0, len(self.Pteam)) :
663             for j in range(0, len(self.Pteam[i][len(self.Pteam[i])-1])) :
664                 for row in c.execute("SELECT * FROM attaques WHERE idAttaques =
                    " + str(self.Pteam[i][len(self.Pteam[i])-1][j])) :
665                     self.Pteam[i][len(self.Pteam[i])-1][j]=[]
666                     for k in range(0, len(row)) :
667                         self.Pteam[i][len(self.Pteam[i])-1][j].append(row[k])
668         self.bag=[]
669         for i in
            ['Pokeball', 'Superball', 'Hyperball', 'Masterball', 'Potion', 'SuperPotion', 'Hyper
            Potion', 'PotionMax', 'Rappel', 'RappelMax'] :
670             for row in c.execute("SELECT * FROM objets WHERE refsac =
                " + str(joueur.nsac) + " AND Nom=" + i + ")") :
671                 self.bag.append([row[1], row[2]])
672         if self.pnj!=False:
673             if pnjo.teamnumber==0: #verifie si le pnj poss  de une   quipe, si non la

```

```

        créer
674         moy=0
675         for i in range(0,len(self.Pteam)):
676             moy+=self.Pteam[i][4]
677             moy=int(moy/(i+1))
678             self.pnj.createteam('normal',abs(int(random.randint(moy-10,moy)))+1)
679
680     nbpoke=0
681     for row in c.execute("SELECT * FROM pokemon WHERE appartientequipe =
"+str(self.pnj.teamnumber)):
682         nbpoke+=1
683         self.Bteam.append([])
684         for i in range(0,len(row)):
685             self.Bteam[nbpoke-1].append(row[i])
686     for i in range(0,len(self.Bteam)):
687         self.Bteam[i].append([])
688         for row in c.execute("SELECT Attaques_idAttaques FROM
attaquespokemon WHERE appartientpokemon = "+str(i+1)):
689             self.Bteam[i][len(self.Bteam[i])-1].append(row[0])
690     for i in range(0,len(self.Bteam)):
691         for j in range(0,len(self.Bteam[i][len(self.Bteam[i])-1])):
692             for row in c.execute("SELECT * FROM attaques WHERE idAttaques =
"+str(self.Bteam[i][len(self.Bteam[i])-1][j])):
693                 self.Bteam[i][len(self.Bteam[i])-1][j]=[]
694                 for k in range(0,len(row)):
695                     self.Bteam[i][len(self.Bteam[i])-1][j].append(row[k])
696     else:
697         moy=0
698         for i in range(0,len(self.Pteam)):
699             moy+=self.Pteam[i][4]
700             moy=int(moy/(i+1))
701             poke=random.randint(1,151)
702             for row in c.execute("SELECT * FROM basepokemon WHERE id="+str(poke)):
703                 print(row)
704
705         self.Bteam.append([0,row[0],row[1],row[2],abs(int(random.randint(moy-10,moy)))+1,row[3],row[3],row[4],row[5],row[6],row[7],joueur.numteam,[2,
'Absorb', 20, 100, 'grass', 10]])#gen du pokemon a randomiser
706
707
708
709
710
711     #Chargement des images
712     for i in range(0,len(self.Pteam)):
713         poke=str(self.Pteam[i][1])
714         img=pygame.image.load("./BattleScene/icons/"+poke+".png")
715
716         self.Pteam[i].append(pygame.transform.scale(img,(img.get_width()*2,img.get_height()*2)))
717         while len(poke)<3:
718             poke='0'+poke
719             img=pygame.image.load("./BattleScene/back/"+poke+".gif")
720
721         self.Pteam[i].append(pygame.transform.scale(img,(img.get_width()*3,img.get_height()*3)))
722
723     for i in range(0,len(self.Bteam)):
724         poke=str(self.Bteam[i][1])
725         img=pygame.image.load("./BattleScene/icons/"+poke+".png")
726
727         self.Bteam[i].append(pygame.transform.scale(img,(img.get_width()*2,img.get_height()*2)))
728         while len(poke)<3:
729             poke='0'+poke
730             img=pygame.image.load("./BattleScene/front/"+poke+".gif")
731
732         self.Bteam[i].append(pygame.transform.scale(img,(img.get_width()*2,img.get_height()*2)))

```

```

731 self.bg=pygame.image.load("./BattleScene/Font/Grass 2.png")
732 self.bg=pygame.transform.scale(self.bg, (self.win.get_width(),
733 self.bg.get_height()))
734
735 self.menu=pygame.image.load("./BattleScene/battlemenu.png")
736 self.attackmenu=pygame.image.load("./BattleScene/attackmenu.png")
737 self.textbox=pygame.image.load("./BattleScene/textbox.png")
738 self.textbox=pygame.transform.scale(self.textbox, (self.win.get_width(),96))
739 self.fightselect=pygame.image.load("./BattleScene/fightselect.png")
740 self.optionselect=pygame.image.load("./BattleScene/optionselect.png")
741 self.attackselect=pygame.image.load("./BattleScene/attackselect.png")
742 self.cancelselect=pygame.image.load("./BattleScene/cancelselect.png")
743 self.pokemenu=pygame.image.load("./BattleScene/pokemenu.png")
744 self.pokeframe=pygame.image.load("./BattleScene/pokeframe.png")
745 self.pokeframeselect=pygame.image.load("./BattleScene/pokeframeselect.png")
746
self.activepokeframeselect=pygame.image.load("./BattleScene/activepokeframesel
ect.png")
747 self.emptyattackslot=pygame.image.load("./BattleScene/emptyattackslot.png")
748 self.hpbox=pygame.image.load("./BattleScene/hpbox.png")
749 self.attacksbg=pygame.image.load("./BattleScene/attacksbg.png")
750 self.menubag=pygame.image.load("./BattleScene/menubag.png")
751 self.items=pygame.image.load("./BattleScene/items.png")
752
self.blankbottomscreen=pygame.image.load("./BattleScene/blankbottomscreen.jpg"
)
753
754
755
756 self.wmenu="menu"#indique le menu choisi par le joueur
757 self.select=""#variable tempon contenant le dernier menu survolé par
l'utilisateur
758 self.sel="soin"#indique la position du sac, si il est en mode soins ou balls
759 self.bsel=""#indique la position temporaire du sac, si il est en mode soins
ou balls
760 self.cc=True#Vrai si l'utilisateur peut cliquer
761 self.tsoin=False#Vrai si l'utilisateur veut utiliser un objet de soin
762
def fight(self):
763     """
764     methode permettant l'affichage et le déroulement du combat
765     """
766     global joueur
767     global login
768     global run
769     replace=False
770     keepfight=True
771     while keepfight and run:
772         for event in pygame.event.get():
773             if event.type == pygame.QUIT:
774                 keepfight = False
775                 run=False
776
777
778
779         self.win.fill((0,0,0))
780         self.win.blit(self.bg, (0, -50))
781
782
783         #affiche toutes les données relatives au pokemon adverse
784
785         self.win.blit(self.Bteam[0][len(self.Bteam[0])-1],(385-self.Bteam[0][len(s
elf.Bteam[0])-1].get_width()/2,135-self.Bteam[0][len(self.Bteam[0])-1].get
_height()))
786         self.win.blit(self.hpbox, (0, 20),(516,0,240,60))
787         self.win.blit(pygame.font.SysFont("monospace",
20).render(self.Bteam[0][2], 1, (0,0,0)), (5, 30))#a remplacer
788         lvl=str(self.Bteam[0][4])
789         if len(lvl)==1:#Affiche le lvl
790             self.win.blit(self.hpbox, (164, 36),(self.numbers(lvl[0]),166,16,14))
791         elif len(lvl)==2:
792             self.win.blit(self.hpbox, (164, 36),(self.numbers(lvl[0]),166,16,14))

```

```

792         self.win.blit(self.hpbox, (181, 36), (self.numbers(lvl[1]), 166, 16, 14))
793     else:
794         self.win.blit(self.hpbox, (164, 36), (self.numbers(lvl[0]), 166, 16, 14))
795         self.win.blit(self.hpbox, (181, 36), (self.numbers(lvl[1]), 166, 16, 14))
796         self.win.blit(self.hpbox, (198, 36), (self.numbers(lvl[2]), 166, 16, 14))
797
798     if self.Bteam[0][5]>self.Bteam[0][6]/2:
799         self.win.blit(self.hpbox, (100,
800             58), (0, 166, 96*self.Bteam[0][5]/self.Bteam[0][6], 14))
801     elif self.Bteam[0][5]>self.Bteam[0][6]/4:
802         self.win.blit(self.hpbox, (100,
803             58), (0, 182, 96*self.Bteam[0][5]/self.Bteam[0][6], 14))
804     else:
805         self.win.blit(self.hpbox, (100,
806             58), (0, 198, 96*self.Bteam[0][5]/self.Bteam[0][6], 14))
807
808     #affiche toutes les données relatives au pokemon allié
809
810     self.win.blit(self.Pteam[0][len(self.Pteam[0])-1], (100-self.Pteam[0][len(self.Pteam[0])-1].get_width()/2, 280-self.Pteam[0][len(self.Pteam[0])-1].get_height()))
811     self.win.blit(self.hpbox, (270, 140), (516, 82, 240, 82))
812     self.win.blit(pygame.font.SysFont("monospace",
813         20).render(self.Pteam[0][2], 1, (0, 0, 0)), (300, 150)) #a remplacer
814     lvl=str(self.Pteam[0][4])
815     if len(lvl)==1: #Affiche le lvl du pokémon allié
816         self.win.blit(self.hpbox, (458, 156), (self.numbers(lvl[0]), 166, 16, 14))
817     elif len(lvl)==2:
818         self.win.blit(self.hpbox, (458, 156), (self.numbers(lvl[0]), 166, 16, 14))
819         self.win.blit(self.hpbox, (475, 156), (self.numbers(lvl[1]), 166, 16, 14))
820     else:
821         self.win.blit(self.hpbox, (458, 156), (self.numbers(lvl[0]), 166, 16, 14))
822         self.win.blit(self.hpbox, (475, 156), (self.numbers(lvl[1]), 166, 16, 14))
823         self.win.blit(self.hpbox, (492, 156), (self.numbers(lvl[2]), 166, 16, 14))
824
825     hp=str(self.Pteam[0][5])
826     if len(hp)==1: #affiche les hp du pokémon allié
827         self.win.blit(self.hpbox, (428, 195), (self.numbers(hp[0]), 166, 16, 14))
828     elif len(hp)==2:
829         self.win.blit(self.hpbox, (428, 195), (self.numbers(hp[1]), 166, 16, 14))
830         self.win.blit(self.hpbox, (410, 195), (self.numbers(hp[0]), 166, 16, 14))
831     else:
832         self.win.blit(self.hpbox, (428, 195), (self.numbers(hp[2]), 166, 16, 14))
833         self.win.blit(self.hpbox, (410, 195), (self.numbers(hp[1]), 166, 16, 14))
834         self.win.blit(self.hpbox, (392, 195), (self.numbers(hp[0]), 166, 16, 14))
835
836     hpmax=str(self.Pteam[0][6])
837     if len(hpmax)==1: #affiche les hpmax du pokémon allié
838         self.win.blit(self.hpbox, (458,
839             195), (self.numbers(hpmax[0]), 166, 16, 14))
840     elif len(hpmax)==2:
841         self.win.blit(self.hpbox, (458,
842             195), (self.numbers(hpmax[0]), 166, 16, 14))
843         self.win.blit(self.hpbox, (475,
844             195), (self.numbers(hpmax[1]), 166, 16, 14))
845     else:
846         self.win.blit(self.hpbox, (458,
847             195), (self.numbers(hpmax[0]), 166, 16, 14))
848         self.win.blit(self.hpbox, (475,
849             195), (self.numbers(hpmax[1]), 166, 16, 14))
850         self.win.blit(self.hpbox, (492,
851             195), (self.numbers(hpmax[2]), 166, 16, 14))
852
853     if self.Pteam[0][5]>self.Pteam[0][6]/2:
854         self.win.blit(self.hpbox, (394,
855             178), (0, 166, 96*self.Pteam[0][5]/self.Pteam[0][6], 14))
856     elif self.Pteam[0][5]>self.Pteam[0][6]/4:
857         self.win.blit(self.hpbox, (394,
858             178), (0, 182, 96*self.Pteam[0][5]/self.Pteam[0][6], 14))
859     else:
860         self.win.blit(self.hpbox, (394,

```

```

178), (0, 198, 96 * self.Pteam[0][5] / self.Pteam[0][6], 14))

849
850
851
852 self.win.blit(self.textbox, (0, 736 - 406 - 96))
853
854 if self.Pteam[0][5] == 0 and "pokemon" not in self.wmenu:
855     self.wmenu = "pokemenu"
856
857 if replace: # En cas de capture et d'équipe complète
858     self.win.blit(self.textbox, (0, 736 - 406 - 96))
859     self.win.blit(pygame.font.SysFont("monospace",
15).render("Choisissez ou non quel pokémon voulez vous remplacer",
1, (0, 0, 0)), (20, 736 - 490))
860     if "pokemon" in self.wmenu:
861         self.Pteam[int(self.wmenu[7])] = self.Bteam[0]
862         keepfight = False
863         self.wmenu = False
864     elif self.wmenu == "menu":
865         keepfight = False
866         self.wmenu = "pokemenu"
867     played = False
868     attack = False
869     if self.wmenu == "menu": # si le menu qui doit être affiché est le principal
870         self.bmenu()
871     elif self.wmenu == "fight": # si le menu qui doit être affiché est le menu
des attaques
872         self.bfight()
873     elif self.wmenu == "pokemenu": # si le menu qui doit être affiché est le
menu des pokémons
874         self.bpokemenu()
875     elif self.wmenu == "bag": # si le menu qui doit être affiché est le menu du
sac
876         self.bbag()
877     elif "pokemon" in self.wmenu: # si un pokémon a été sélectionné
878         if self.tsoin == False:
879             if self.Pteam[int(self.wmenu[7])][5] == 0:
880                 self.wmessage("ce pokemon est KO, il ne peut plus combattre")
881                 self.wmenu = "pokemenu"
882             elif int(self.wmenu[7]) == 0:
883                 self.wmessage("impossible, ce pokémon est déjà sur le
terrain")
884                 self.wmenu = "pokemenu"
885             else:
886                 if self.Pteam[0][5] != 0:
887                     played = True
888                     temp = self.Pteam[int(self.wmenu[7])]
889                     self.Pteam[int(self.wmenu[7])] = self.Pteam[0]
890                     self.Pteam[0] = temp
891                     self.wmenu = "menu"
892         else:
893             menu = self.wmenu
894             if self.tsoin[4] in ["0", "1", "2", "3"]:
895                 if self.Pteam[int(self.wmenu[7])][5] > 0:
896                     if
897                         self.Pteam[int(self.wmenu[7])][5] != self.Pteam[int(self.wme
nu[7])][6]:
898                         if self.tsoin[4] == "0":
899                             self.Pteam[int(self.wmenu[7])][5] += 25
900                             played = True
901                         elif self.tsoin[4] == "1":
902                             self.Pteam[int(self.wmenu[7])][5] += 50
903                             played = True
904                         elif self.tsoin[4] == "2":
905                             self.Pteam[int(self.wmenu[7])][5] += 200
906                             played = True
907                         elif self.tsoin[4] == "3":
908
909                             self.Pteam[int(self.wmenu[7])][5] += self.Pteam[int(
self.wmenu[7])][6]
910                             played = True
911                     if

```



910

911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
  
923  
924  
925  
  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970

```
self.Pteam[int(self.wmenu[7])][5]>self.Pteam[int(self.wmenu[7])][6]:

    self.Pteam[int(self.wmenu[7])][5]=self.Pteam[int(self.wmenu[7])][6]
    self.bag[int(self.tsoin[4])+4][1]-=1
    self.tsoin=False
else:
    self.wmessage("Ce pokémon est déjà en pleine santé")
    menu="pokemenu"
else:
    self.wmessage("Ce pokémon est KO")
    menu="pokemenu"
else:
    if self.Pteam[int(self.wmenu[7])][5]==0:
        if self.tsoin[4]=="4":

            self.Pteam[int(self.wmenu[7])][5]=int(self.Pteam[int(self.wmenu[7])][6]/2)
            played=True
            elif self.tsoin[4]=="5":

                self.Pteam[int(self.wmenu[7])][5]=self.Pteam[int(self.wmenu[7])][6]
                played=True
                self.bag[int(self.tsoin[4])+4][1]-=1
                self.tsoin=False
            else:
                self.wmessage("Ce pokémon n'est pas KO")
                menu="pokemenu"
        self.wmenu=menu

elif "attaque" in self.wmenu:#si une attaque à été sélectionnée
    played=True
    attack=True
elif "soin" in self.wmenu:#si un soin à été sélectionné
    if self.bag[int(self.wmenu[4])+4][1]>0:
        self.tsoin=self.wmenu
        self.wmenu="pokemenu"
    else:
        self.wmessage("Vous ne possédez pas de "+self.bag[int(self.wmenu[4])+4][0])
        self.wmenu="bag"
elif "ball" in self.wmenu:#si une ball à été sélectionnée
    if self.bag[int(self.wmenu[4])][1]>0:
        if self.pnj!=False:
            self.wmessage("vous ne pouvez pas capturer un pokémon qui appartient à un dresseur")
            self.wmenu="bag"
        else:
            self.bag[int(self.wmenu[4])][1]-=1
            if self.wmenu[4]=="0":
                luck=random.random()<0.25
            elif self.wmenu[4]=="1":
                luck=random.random()<0.50
            elif self.wmenu[4]=="2":
                luck=random.random()<0.75
            else:
                luck=True
            if luck:
                self.wmessage("vous avez capturé le pokémon")#ajouter à l'équipe si place sinon proposer d'enlever un pokémon
                if len(self.Pteam)<6:
                    self.Pteam.append(self.Bteam[0])
                    keepfight=False
                else:
                    replace=True
            else:
                self.wmessage("le pokémon s'est échapé!")
                self.wmenu="menu"
                played=True
        else:
```



```

971         self.wmessage("Vous ne possédez pas de
972         "+self.bag[int(self.wmenu[4])][0])
973         self.wmenu="bag"
974     elif self.wmenu=="run":#si le joueur tente de fuir le combat
975         if self.pnj==False:
976             if random.random()>0.8:
977                 keepfight=False
978             else:
979                 self.played=True
980         else:
981             self.wmessage("vous ne pouvez fuir le combat contre un dresseur")
982             self.wmenu="menu"
983     else:
984         self.wmenu="menu"
985
986 if played==True:#quand le joueur à joué
987     if attack==True:#si le joueur à choisi d'attaquer
988         if self.Pteam[0][7]>=self.Bteam[0][7]:
989             self.wmessage("Vous infligez
990             "+str(self.Pteam[0][12][int(self.wmenu[6])-1][2])+" points
991             de dégats")
992             self.Bteam[0][5]-=self.Pteam[0][12][int(self.wmenu[6])-1][2]
993             if self.Bteam[0][5]>0:
994                 self.wmessage("Le pokemon ennemi vous inflige
995                 "+str(self.Bteam[0][12][0][2])+" points de dégats")
996                 self.Pteam[0][5]-=self.Bteam[0][12][0][2]
997             else:
998                 self.Bteam[0][5]=0
999         else:
1000             self.wmessage("Le pokemon ennemi vous inflige
1001             "+str(self.Bteam[0][12][0][2])+" points de dégats")
1002             self.Pteam[0][5]-=self.Bteam[0][12][0][2]
1003             if self.Pteam[0][5]>0:
1004                 self.wmessage("Vous infligez
1005                 "+str(self.Pteam[0][12][int(self.wmenu[6])-1][2])+"
1006                 points de dégats")
1007                 self.Bteam[0][5]-=self.Pteam[0][12][int(self.wmenu[6])-1][
1008                 2]
1009             else:
1010                 self.Pteam[0][5]=0
1011         else:
1012             self.wmessage("Le pokemon ennemi vous inflige
1013             "+str(self.Bteam[0][12][0][2])+" points de dégats")
1014             self.Pteam[0][5]-=self.Bteam[0][12][0][2]
1015             if self.Pteam[0][5]<0:
1016                 self.Pteam[0][5]=0
1017             if self.Bteam[0][5]<0:
1018                 self.Bteam[0][5]=0
1019
1020 up=0
1021 for i in range(0,len(self.Pteam)):
1022     up+=self.Pteam[i][5]
1023 if up==0:
1024     lost=True
1025     while lost:
1026         for event in pygame.event.get():
1027             if event.type == pygame.QUIT:
1028                 keepfight = False
1029                 run=False
1030                 lost=False
1031         self.win.fill((255,255,255))
1032         self.win.blit(pygame.font.SysFont("monospace",
1033         100).render("Perdu", 1, (0,0,0)), (100, 100))
1034         pygame.display.update()
1035         #####
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301

```



```

1094         if event.type == pygame.QUIT:
1095             keepfight = False
1096             run=False
1097         self.win.blit(self.textbox, (0, 736-406-96))
1098         self.win.blit(pygame.font.SysFont("monospace",
1099             15).render("Voulez vous remplacer une attaque par
1100             "+row[0]+" ?", 1, (0,0,0)), (20,736-490))
1101         self.bfight()
1102         if "attack" in self.wmenu:
1103             self.Pteam[0][12][int(self.wmenu[6])-1]=[]
1104             for row in c.execute("SELECT * FROM attaques WHERE
1105                 idAttaques = "+str(att)):
1106                 for k in range(0,len(row)):
1107                     self.Pteam[0][12][int(self.wmenu[6])-1].append
1108                         (row[k])
1109                 selattack=False
1110             elif self.wmenu=="menu":
1111                 selattack=False
1112             pygame.display.update()
1113
1114
1115         c.execute("DELETE FROM pokemon WHERE
1116             appartientequipe="+str(joueur.numteam))
1117         for i in range(0,len(self.Pteam)):
1118             npokemon=1
1119             for row in c.execute("SELECT * FROM pokemon"):
1120                 npokemon+=1
1121             npokemon=str(npokemon)
1122             c.execute("INSERT INTO pokemon VALUES
1123                 (" +npokemon+", "+str(self.Pteam[i][1])+", '"+self.Pteam[i][2]+'', '"+self
1124                 .Pteam[i][3]+'', "+str(self.Pteam[i][4])+", "+str(self.Pteam[i][5])+", "+
1125                 str(self.Pteam[i][6])+", "+str(self.Pteam[i][7])+", "+str(self.Pteam[i][
1126                 8])+", "+str(self.Pteam[i][9])+", "+str(self.Pteam[i][10])+", "+str(joueu
1127                 r.numteam)+")")
1128             c.execute("DELETE FROM attaquespokemon WHERE
1129                 appartientpokemon="+npokemon)
1130
1131             for j in range(0,len(self.Pteam[i][12])):
1132                 c.execute("INSERT INTO attaquespokemon VALUES
1133                     (NULL, "+str(self.Pteam[i][12][j][0])+", "+npokemon+")")#attribue
1134                     les attaques aux pokemons
1135
1136
1137         c.execute("DELETE FROM objets WHERE refsac="+str(joueur.nsac))
1138         for i in self.bag:
1139             c.execute("INSERT INTO objets VALUES
1140                 (NULL, '"+i[0]+'', "+str(i[1])+", "+str(joueur.nsac)+")")
1141
1142
1143         conn.commit()
1144         conn.close()
1145
1146
1147 def bmenu(self):
1148     """
1149     methode permettant de construire et dinteragir avec le menu principal du
1150     combat
1151     """
1152     global joueur
1153     global login
1154     self.win.blit(self.menu, (0,736-406))
1155     self.win.blit(self.Pteam[0][len(self.Pteam[0])-2], (160,455))
1156     if 736-406<pygame.mouse.get_pos() [1]<736-110:

```

```

1150         if int((time.time()*1.5)%2)==0:
1151             self.win.blit(self.fightselect, (40, 736-406+80))
1152             self.select="fight"
1153     elif pygame.mouse.get_pos()[1]>736-110:
1154         if pygame.mouse.get_pos()[0]<512/3:
1155             if int((time.time()*1.5)%2)==0:
1156                 self.win.blit(self.optionselect, (0, 736-100))
1157                 self.select="bag"
1158             elif pygame.mouse.get_pos()[0]<512*2/3:
1159                 if int((time.time()*1.5)%2)==0:
1160                     self.win.blit(self.optionselect, (512/3, 736-100))
1161                     self.select="run"
1162             else:
1163                 if int((time.time()*1.5)%2)==0:
1164                     self.win.blit(self.optionselect, (512*2/3, 736-100))
1165                     self.select="pokemenu"
1166         else:
1167             self.select="menu"
1168     if pygame.mouse.get_pressed()[0]==1:
1169         if self.cc:
1170             self.cc=False
1171             self.wmenu=self.select
1172         else:
1173             self.cc=True
1174
1175     def bfight(self):
1176         """
1177         methode permettant de construire et dinteragir avec le menu des attaques
1178         """
1179         global joueur
1180         global login
1181         nbattack=len(self.Pteam[0][len(self.Pteam[0])-3])
1182         self.win.blit(self.attackmenu,(0,736-406))
1183
1184         #première attaque
1185         loc=self.bgattacks(self.Pteam[0][12][0][4])
1186         self.win.blit(self.attacksbg, (4, 736-406+48),(loc[0],loc[1],248,110))
1187         self.win.blit(pygame.font.SysFont("monospace",
1188         20).render(self.Pteam[0][12][0][1], 1, (0,0,0)), (80, 736-406+70))#a remplacer
1189         if nbattack>1:
1190             loc=self.bgattacks(self.Pteam[0][12][1][4])
1191             self.win.blit(self.attacksbg, (512/2+6,
1192             736-406+48),(loc[0],loc[1],248,110))
1193             self.win.blit(pygame.font.SysFont("monospace",
1194             20).render(self.Pteam[0][12][1][1], 1, (0,0,0)), (350, 736-406+70))#a
1195             remplacer
1196         else:
1197             self.win.blit(self.emptyattackslot, (512/2+6, 736-406+48))
1198         if nbattack>2:
1199             loc=self.bgattacks(self.Pteam[0][12][2][4])
1200             self.win.blit(self.attacksbg, (4, 736-406+172),(loc[0],loc[1],248,110))
1201             self.win.blit(pygame.font.SysFont("monospace",
1202             20).render(self.Pteam[0][12][2][1], 1, (0,0,0)), (80, 736-406+200))#a
1203             remplacer
1204         else:
1205             self.win.blit(self.emptyattackslot, (4, 736-406+172))
1206         if nbattack>3:
1207             loc=self.bgattacks(self.Pteam[0][12][3][4])
1208             self.win.blit(self.attacksbg, (512/2+6,
1209             736-406+172),(loc[0],loc[1],248,110))
1210             self.win.blit(pygame.font.SysFont("monospace",
1211             20).render(self.Pteam[0][12][3][1], 1, (0,0,0)), (350, 736-406+200))#a
1212             remplacer
1213         else:
1214             self.win.blit(self.emptyattackslot, (512/2+6, 736-406+172))
1215
1216     if 736-406<pygame.mouse.get_pos()[1]<736-240:
1217         if pygame.mouse.get_pos()[0]<512/2:
1218             if int((time.time()*1.5)%2)==0:
1219                 self.win.blit(self.attackselect, (2, 736-360))
1220                 self.select="attack1"
1221             else:

```

```

1213         if int((time.time()*1.5)%2)==0:
1214             self.win.blit(self.attackselect, (512/2+2, 736-360))
1215             self.select="attack2"
1216     elif 736-240<pygame.mouse.get_pos()[1]<736-110:
1217         if pygame.mouse.get_pos()[0]<512/2:
1218             if int((time.time()*1.5)%2)==0:
1219                 self.win.blit(self.attackselect, (2, 736-236))
1220                 self.select="attack3"
1221         else:
1222             if int((time.time()*1.5)%2)==0:
1223                 self.win.blit(self.attackselect, (512/2+2, 736-236))
1224                 self.select="attack4"
1225     elif pygame.mouse.get_pos()[1]>736-110:
1226         if int((time.time()*1.5)%2)==0:
1227             self.win.blit(self.cancelselect, (10, 736-100))
1228             self.select="menu"
1229     else:
1230         self.select="fight"
1231
1232     if "attack" in self.select:
1233         if int(self.select[6])>len(self.Pteam[0][12]):
1234             self.select="fight"
1235
1236
1237
1238     if pygame.mouse.get_pressed()[0]==1:
1239         if self.cc:
1240             self.cc=False
1241             self.wmenu=self.select
1242         else:
1243             self.cc=True
1244
1245     def bgattacks(self,typ):
1246         """
1247         methode permettant de changer l'affichage de l'attaque en fonction du type
1248         de cette dernière
1249         """
1250         global joueur
1251         global login
1252         if typ=="ground":
1253             return [0,0]
1254         elif typ=="fight":
1255             return [0,112]
1256         elif typ=="normal":
1257             return [0,224]
1258         elif typ=="steel":
1259             return [0,336]
1260         elif typ=="flying":
1261             return [0,448]
1262         elif typ=="water":
1263             return [250,0]
1264         elif typ=="psycho":
1265             return [250,112]
1266         elif typ=="poison":
1267             return [250,224]
1268         elif typ=="rock":
1269             return [250,336]
1270         elif typ=="fire":
1271             return [250,448]
1272         elif typ=="ghost":
1273             return [250,0]
1274         elif typ=="grass":
1275             return [500,112]
1276         elif typ=="electric":
1277             return [500,224]
1278         elif typ=="dragon":
1279             return [500,336]
1280         elif typ=="ice":
1281             return [500,448]
1282         elif typ=="bug":
1283             return [750,0]
1284         elif typ=="dark":

```

```

1284         return [750,112]
1285     else:
1286         return [750,224]
1287
1288
1289 def bpokemenu(self):
1290     """
1291     methode permettant de construire et dinteragir avec le menu des pokémons
1292     """
1293     global joueur
1294     global login
1295     nbpoke=len(self.Pteam)
1296     self.win.blit(self.pokemenu,(0,736-406))
1297     self.win.blit(pygame.font.SysFont("monospace", 20).render(self.Pteam[0][2],
1298         1, (0,0,0)), (90, 350))#a remplacer
1299     if nbpoke>1:
1300         self.win.blit(self.pokeframe, (512/2+2, 736-384))
1301         self.win.blit(pygame.font.SysFont("monospace",
1302             20).render(self.Pteam[1][2], 1, (0,0,0)), (340, 366))#a remplacer
1303     if nbpoke>2:
1304         self.win.blit(self.pokeframe, (2, 736-304))
1305         self.win.blit(pygame.font.SysFont("monospace",
1306             20).render(self.Pteam[2][2], 1, (0,0,0)), (90, 446))#a remplacer
1307     if nbpoke>3:
1308         self.win.blit(self.pokeframe, (512/2+2, 736-288))
1309         self.win.blit(pygame.font.SysFont("monospace",
1310             20).render(self.Pteam[1][2], 1, (0,0,0)), (340, 462))#a remplacer
1311     if nbpoke>4:
1312         self.win.blit(self.pokeframe, (2, 736-208))
1313         self.win.blit(pygame.font.SysFont("monospace",
1314             20).render(self.Pteam[2][2], 1, (0,0,0)), (90, 542))#a remplacer
1315     if nbpoke>5:
1316         self.win.blit(self.pokeframe, (512/2+2, 736-192))
1317         self.win.blit(pygame.font.SysFont("monospace",
1318             20).render(self.Pteam[1][2], 1, (0,0,0)), (340, 558))#a remplacer
1319
1320     if 736-406<pygame.mouse.get_pos()[1]<736-300:
1321         if pygame.mouse.get_pos()[0]<512/2:
1322             self.win.blit(self.activepokeframeselect, (2, 736-406))
1323             self.win.blit(pygame.font.SysFont("monospace",
1324                 20).render(self.Pteam[0][2], 1, (0,0,0)), (90, 350))#a remplacer
1325             self.select="pokemon0"
1326         elif nbpoke>1:
1327             self.win.blit(self.pokeframeselect, (512/2, 736-390))
1328             self.win.blit(pygame.font.SysFont("monospace",
1329                 20).render(self.Pteam[1][2], 1, (0,0,0)), (340, 366))#a remplacer
1330             self.select="pokemon1"
1331         elif 736-300<pygame.mouse.get_pos()[1]<736-210:
1332             if pygame.mouse.get_pos()[0]<512/2 and nbpoke>2:
1333                 self.win.blit(self.pokeframeselect, (0, 736-310))
1334                 self.win.blit(pygame.font.SysFont("monospace",
1335                     20).render(self.Pteam[2][2], 1, (0,0,0)), (90, 446))#a remplacer
1336                 self.select="pokemon2"
1337             elif nbpoke>3:
1338                 self.win.blit(self.pokeframeselect, (512/2, 736-294))
1339                 self.win.blit(pygame.font.SysFont("monospace",
1340                     20).render(self.Pteam[1][2], 1, (0,0,0)), (340, 462))#a remplacer
1341                 self.select="pokemon3"
1342             elif 736-210<pygame.mouse.get_pos()[1]<736-100:
1343                 if pygame.mouse.get_pos()[0]<512/2 and nbpoke>4:
1344                     self.win.blit(self.pokeframeselect, (0, 736-214))
1345                     self.win.blit(pygame.font.SysFont("monospace",
1346                         20).render(self.Pteam[2][2], 1, (0,0,0)), (90, 542))#a remplacer
1347                     self.select="pokemon4"
1348                 elif nbpoke>5:
1349                     self.win.blit(self.pokeframeselect, (512/2, 736-198))
1350                     self.win.blit(pygame.font.SysFont("monospace",
1351                         20).render(self.Pteam[1][2], 1, (0,0,0)), (340, 558))#a remplacer
1352                     self.select="pokemon5"
1353             elif pygame.mouse.get_pos()[1]>736-100 and pygame.mouse.get_pos()[0]>512-130:
1354                 if int((time.time()*2)%2)==0:
1355                     self.win.blit(self.cancelselect, (10, 736-100))

```

```

1344         if self.tsoin==False:
1345             self.select="menu"
1346         else:
1347             self.select="bag"
1348     else:
1349         self.select="pokemenu"
1350
1351
1352     #corriger avec la vraie police
d'écriture#####
#####
###
1353     self.win.blit(self.Pteam[0][len(self.Pteam[0])-2],(30,320))
1354     self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[0][4]), 1, (0,0,0)), (40, 395))
1355     self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[0][5]), 1, (0,0,0)), (130, 395))
1356     self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[0][6]), 1, (0,0,0)), (190, 395))
1357     if nbpoke>1:
1358         self.win.blit(self.Pteam[1][len(self.Pteam[1])-2],(286,336))
1359         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[1][4]), 1, (0,0,0)), (296, 412))
1360         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[1][5]), 1, (0,0,0)), (130+256, 412))
1361         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[1][6]), 1, (0,0,0)), (190+256, 412))
1362     if nbpoke>2:
1363         self.win.blit(self.Pteam[2][len(self.Pteam[2])-2],(30,416))
1364         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[2][4]), 1, (0,0,0)), (40, 495))
1365         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[2][5]), 1, (0,0,0)), (130, 495))
1366         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[2][6]), 1, (0,0,0)), (190, 495))
1367     if nbpoke>3:
1368         self.win.blit(self.Pteam[3][len(self.Pteam[3])-2],(286,432))
1369         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[3][4]), 1, (0,0,0)), (296, 512))
1370         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[3][5]), 1, (0,0,0)), (130+256, 512))
1371         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[3][6]), 1, (0,0,0)), (190+256, 512))
1372     if nbpoke>4:
1373         self.win.blit(self.Pteam[4][len(self.Pteam[4])-2],(30,512))
1374         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[4][4]), 1, (0,0,0)), (40, 595))
1375         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[4][5]), 1, (0,0,0)), (130, 595))
1376         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[4][6]), 1, (0,0,0)), (190, 595))
1377     if nbpoke>5:
1378         self.win.blit(self.Pteam[5][len(self.Pteam[5])-2],(286,528))
1379         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[5][4]), 1, (0,0,0)), (296, 612))
1380         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[5][5]), 1, (0,0,0)), (130+256, 612))
1381         self.win.blit(pygame.font.SysFont("monospace",
20).render(str(self.Pteam[5][6]), 1, (0,0,0)), (190+256, 612))
1382
1383
1384
1385
1386
1387
1388
1389     if pygame.mouse.get_pressed()[0]==1:
1390         if self.cc:
1391             self.cc=False
1392             self.wmenu=self.select
1393         else:
1394             self.cc=True

```

```

1395
1396 def bbag(self):
1397     """
1398     methode permettant de construire et dinteragir avec le sac
1399     """
1400     global joueur
1401     global login
1402     self.tsoin=False
1403     self.bsel=self.sel
1404     self.win.blit(self.menubag,(0,736-406),(0,0,512,390))
1405     self.win.blit(self.menubag,(388,736-71),(87,411,122,48))
1406     if 736-406<pygame.mouse.get_pos()[1]<736-340:
1407         if 230<pygame.mouse.get_pos()[0]<286:
1408             self.win.blit(self.menubag,(231,736-406+7),(821,277,56,56))
1409             self.bsel="soin"
1410         if 389<pygame.mouse.get_pos()[0]<445:
1411             self.win.blit(self.menubag,(390,736-406+7),(821,277,56,56))
1412             self.bsel="ball"
1413
1414     if self.sel=="soin":
1415         self.win.blit(self.menubag,(234,736-406+12),(528,12,47,47))
1416
1417         self.win.blit(self.menubag,(10,736-406+87),(526,126,244,75))
1418         self.win.blit(pygame.font.SysFont("monospace",
1419             20).render(self.bag[4][0]+":"+str(self.bag[4][1]),1,(0,0,0)),
1420             (70,736-406+112))
1421         self.win.blit(self.menubag,(266,736-406+87),(526,126,244,75))
1422         self.win.blit(pygame.font.SysFont("monospace",
1423             20).render(self.bag[5][0]+":"+str(self.bag[5][1]),1,(0,0,0)),
1424             (326,736-406+112))
1425         self.win.blit(self.menubag,(10,736-406+164),(526,126,244,75))
1426         self.win.blit(pygame.font.SysFont("monospace",
1427             20).render(self.bag[6][0]+":"+str(self.bag[6][1]),1,(0,0,0)),
1428             (70,736-406+189))
1429         self.win.blit(self.menubag,(266,736-406+164),(526,126,244,75))
1430         self.win.blit(pygame.font.SysFont("monospace",
1431             20).render(self.bag[7][0]+":"+str(self.bag[7][1]),1,(0,0,0)),
1432             (326,736-406+189))
1433         self.win.blit(self.menubag,(10,736-406+241),(526,126,244,75))
1434         self.win.blit(pygame.font.SysFont("monospace",
1435             20).render(self.bag[8][0]+":"+str(self.bag[8][1]),1,(0,0,0)),
1436             (70,736-406+266))
1437         self.win.blit(self.menubag,(266,736-406+241),(526,126,244,75))
1438         self.win.blit(pygame.font.SysFont("monospace",
1439             20).render(self.bag[9][0]+":"+str(self.bag[9][1]),1,(0,0,0)),
1440             (326,736-406+266))
1441
1442     if 736-406+88<pygame.mouse.get_pos()[1]<736-406+165:
1443         if pygame.mouse.get_pos()[0]<512/2:
1444             self.win.blit(self.menubag,(12,736-406+89),(528,207,244,75))
1445             self.select="soin0"
1446         else:
1447             self.win.blit(self.menubag,(268,736-406+89),(528,207,244,75))
1448             self.select="soin1"
1449     elif 736-406+163<pygame.mouse.get_pos()[1]<736-406+241:
1450         if pygame.mouse.get_pos()[0]<512/2:
1451             self.win.blit(self.menubag,(12,736-406+166),(528,207,244,75))
1452             self.select="soin2"
1453         else:
1454             self.win.blit(self.menubag,(268,736-406+166),(528,207,244,75))
1455             self.select="soin3"
1456     elif 736-406+240<pygame.mouse.get_pos()[1]<736-406+318:
1457         if pygame.mouse.get_pos()[0]<512/2:
1458             self.win.blit(self.menubag,(12,736-406+243),(528,207,244,75))
1459             self.select="soin4"
1460         else:
1461             self.win.blit(self.menubag,(268,736-406+243),(528,207,244,75))
1462             self.select="soin5"
1463     elif pygame.mouse.get_pos()[1]>736-80 and
1464         pygame.mouse.get_pos()[0]>512-130:
1465         self.win.blit(self.menubag,(390,736-71),(215,413,118,48))
1466         self.select="menu"

```



```

1454         else:
1455             self.select="bag"
1456
1457
1458
1459
1460         self.win.blit(self.items,(20,736-406+100),(0,0,30,44))
1461         self.win.blit(self.items,(276,736-406+100),(32,0,30,44))
1462         self.win.blit(self.items,(20,736-406+178),(64,0,30,44))
1463         self.win.blit(self.items,(276,736-406+178),(98,0,34,44))
1464         self.win.blit(self.items,(20,736-406+256),(134,0,20,44))
1465         self.win.blit(self.items,(276,736-406+256),(158,0,40,44))
1466
1467     else:
1468         self.win.blit(self.menubag,(394,736-406+12),(579,12,47,47))
1469         self.win.blit(self.menubag,(10,736-406+87),(526,126,244,75))
1470         self.win.blit(pygame.font.SysFont("monospace",
1471         20).render(self.bag[0][0]+":"+str(self.bag[0][1]), 1, (0,0,0)),
1472         (70,736-406+112))
1473         self.win.blit(self.menubag,(266,736-406+87),(526,126,244,75))
1474         self.win.blit(pygame.font.SysFont("monospace",
1475         20).render(self.bag[1][0]+":"+str(self.bag[1][1]), 1, (0,0,0)),
1476         (326,736-406+112))
1477         self.win.blit(self.menubag,(10,736-406+164),(526,126,244,75))
1478         self.win.blit(pygame.font.SysFont("monospace",
1479         20).render(self.bag[2][0]+":"+str(self.bag[2][1]), 1, (0,0,0)),
1480         (70,736-406+189))
1481         self.win.blit(self.menubag,(266,736-406+164),(526,126,244,75))
1482         self.win.blit(pygame.font.SysFont("monospace",
1483         20).render(self.bag[3][0]+":"+str(self.bag[3][1]), 1, (0,0,0)),
1484         (326,736-406+189))
1485
1486     if 736-406+88<pygame.mouse.get_pos()[1]<736-406+165:
1487         if pygame.mouse.get_pos()[0]<512/2:
1488             self.win.blit(self.menubag,(12,736-406+89),(528,207,244,75))
1489             self.select="ball0"
1490         else:
1491             self.win.blit(self.menubag,(268,736-406+89),(528,207,244,75))
1492             self.select="ball1"
1493     elif 736-406+163<pygame.mouse.get_pos()[1]<736-406+241:
1494         if pygame.mouse.get_pos()[0]<512/2:
1495             self.win.blit(self.menubag,(12,736-406+166),(528,207,244,75))
1496             self.select="ball2"
1497         else:
1498             self.win.blit(self.menubag,(268,736-406+166),(528,207,244,75))
1499             self.select="ball3"
1500     elif pygame.mouse.get_pos()[1]>736-80 and
1501     pygame.mouse.get_pos()[0]>512-130:
1502         self.win.blit(self.menubag,(390,736-71),(215,413,118,48))
1503         self.select="menu"
1504     else:
1505         self.select="bag"
1506
1507
1508         self.win.blit(self.items,(20,736-406+100),(202,0,36,44))
1509         self.win.blit(self.items,(276,736-406+100),(240,0,36,44))
1510         self.win.blit(self.items,(20,736-406+178),(278,0,36,44))
1511         self.win.blit(self.items,(276,736-406+178),(316,0,36,44))
1512
1513
1514
1515
1516     if pygame.mouse.get_pressed()[0]==1:
1517         if self.cc:
1518             self.cc=False
1519             self.wmenu=self.select
1520             self.sel=self.bsel
1521         else:
1522             self.cc=True

```

```

1517
1518 def numbers(self,num):#police d'écriture des nombres
1519     """
1520     methode permettant d'afficher chaque chiffre
1521     """
1522     global joueur
1523     global login
1524     if num=="0":
1525         return 100
1526     elif num=="1":
1527         return 118
1528     elif num=="2":
1529         return 134
1530     elif num=="3":
1531         return 152
1532     elif num=="4":
1533         return 170
1534     elif num=="5":
1535         return 188
1536     elif num=="6":
1537         return 206
1538     elif num=="7":
1539         return 224
1540     elif num=="8":
1541         return 242
1542     elif num=="9":
1543         return 260
1544
1545 def wmessage(self,mes):
1546
1547     """
1548     methode permettant d'afficher du texte
1549     prenant mes en argument qui est le texte à afficher
1550     """
1551     global joueur
1552     self.win.blit(self.textbox, (0, 736-406-96))
1553     self.win.blit(pygame.font.SysFont("monospace", 15).render(mes, 1, (0,0,0)),
1554                  (20,736-490))
1555     self.win.blit(self.blankbottomscreen,(0,736-406))
1556     pygame.display.update()
1557     wmes=True
1558     while wmes:
1559         for event in pygame.event.get():
1560             if event.type == pygame.QUIT:
1561                 global run
1562                 keepfight = False
1563                 run=False
1564             if pygame.mouse.get_pressed()[0]==1:
1565                 if self.cc:
1566                     self.cc=False
1567                     wmes=False
1568             else:
1569                 self.cc=True
1570
1571
1572
1573 def jouer(logon):
1574     global run
1575     global joueur
1576     global login
1577     login=logon
1578     run = True
1579     pygame.init()
1580     pygame.display.set_caption("Pokeroyal")
1581     joueur=play()
1582     while run:
1583         pygame.time.delay(10)
1584         for event in pygame.event.get():
1585             if event.type == pygame.QUIT:
1586                 run = False
1587     joueur.move()

```

```
1588         pygame.display.update()
1589     pygame.quit()
1590     jouer('123456')
1591
```