Xavier MOYON
Sifaks SITAYEB

TG08
NSI

Projet Jeu De L'oie

Types De Données :

_Trois Classes («***plateau***» et «***joueur» et «écran»***) (Pourquoi des Classes : Parce que cela permet de stocker plusieurs valeurs pour un joueur ou un plateau ou un écran sans avoir a faire de listes de listes)

_Deux Matrices («***plateauvisu***» pour afficher le plateau avec le nom des cases spéciales et «***plateau***» pour vérifier les position avec les index des positions)(Les matrices nous permettent de visualiser le plateau et de faciliter l'affichage en pygame)

```
plateauvisu=[["Depart","1","2","3","4","Oie","Pont","7","8","Oie"],
["","28","29","30","Puit","Oie","33","34","35","10"],
["","Oie","48","49","Oie","51","Prison","53","Oie","11"],
["","26","47","60","61","62","63","Oie","37","12"],
["","25","46","Oie","LA MORT","57","56","55","38","13"],
["","24","Oie","44","43","Labyrinthe","Oie","40","39","Oie"],
["","Oie","22","21","20","Hotel","Oie","17","16","15"]]
plateau=[["0","1","2","3","4","5","6","7","8","9"],
["","28","29","30","31","32","33","34","35","10"],
["","27","48","49","50","51","52","53","36","11"],
["","26","47","60","61","62","63","54","37","12"],
["","25","46","59","58","57","56","55","38","13"],
["","24","45","44","43","42","41","40","39","14"],
["","23","22","21","20","19","18","17","16","15"]]
```

DESCRIPTION DU CODE :

Code Général

```
 8 """importation des modules.....
 9 ..................................
10 import pygame_jeu_de_oie
11 from random import *
12 import time
```

_Ligne 10 a 12 :

_on importe le fichier « pygame_jeu_de_oie » afin d'avoir accès aux fonctions du fichiers
_On importe le module random pour le lancée de des

_Import de la fonction time pour marquer des temps d'arrêt entre les temps de jeu des Joueurs (Devenu inutile lorsque que l'on a améliorer le fichier pygame)

```
19 class plateau():
20     def __init__(self,liste_jr):
21         self.liste_case=crea_pla(liste_jr)
22         self.case_spe=[]
23         self.liste_jr=liste_jr
24         for i in range(len(self.liste_case[1:])):
25             if type(self.liste_case[i][0])!=int:
26                 self.case_spe.append(self.liste_case[i])
27         print(self.case_spe)
28 
29 
30     def joueurpos(self,jr):
31         self.liste_case[jr.position].append(jr)
32     def cases_prises(self,pos):
33         return self.liste_case[pos][1] != ""
34 
```

Ligne 19 a 178 : Création de la classe plateau qui va créer le plateau et positionner les joueurs et gérer toutes les exceptions (prison , hotel , 2 joueurs sur une meme case) et gerer les échanges et gère en partie son affichage en pygame (Matrices).

Ligne 20 a 27 : On donne créer le constructeur :

Ligne 21: On appelle la fonction crea_pla avec pour paramètre la liste des joueurs afin de créer le plateau

Ligne 24 a 26: On cherche les cases spéciales pour les stocker ce qui va etre utile pour la suite

Ligne 30 a 31 : il s'agit d'une méthode (joueurpos) permettant d'ajouter les objets joueurs a la bonne position dans la liste liste_case

Ligne 32 a 33 : Cette methode permet de vérifier si une case est prises les case sont sous ce format [«index ou nom» , objjoueur ou «»] si l'element 1 est «» alors la case est déjà prise si il y a un objet joueur alors la case est prise

```
36     def echangespos(self,jr,pos1,pos2):#pos1 est la position du joueur qui joue actuellement pos 2 poshypot
37         print("self.liste_case[pos1]:",self.liste_case[pos1],"self.liste_case[pos2]",self.liste_case[pos2])
38 
39         #for i in self.liste_case[pos1][1:]:
40             #print("pos1 : ",i.nom)
41 
42         #for i in self.liste_case[pos2][1:]:
43             #print("pos2 : ",i.nom)
44         if self.liste_case[pos2] in self.case_spe and pos2!=pos1:# on ajoute une condition pour permettre aux joueur d'etre sur une meme case et de debloquer d'autres joueur(puits
45             #print("spe")
46             if self.liste_case[pos2][0]=="hotel":
47                 self.deplacement(pos2 , jr)#on fait avancer normalement le joueur
48 
49             elif self.liste_case[pos2][0]=="prison":
50                 #print("eh o ")
51                 #on fait avancer normalement le joueur
52 
53                 #self.liste_case[pos1][1].prisontour=None#On se libere soi meme et l'autre joueur
54                 #if self.cases_prises(pos2):
55 
56                 if self.liste_case[pos2][-1].prisontour!=None:
57                     self.liste_case[pos2][-1].prisontour=None
58 
59                 self.deplacement(pos2 , jr)
60 
61 
62             elif self.liste_case[pos2][0]=="puit":
63                 txt=("Bravo  ", self.liste_case[pos2][-1].nom ," Tu as fait preuve de malice et tu as reussi a berner ",self.liste_case[pos1][1].nom,"Qui prend desormais ta place
64                 pygame_jeu_de_oie.aff_casepos(self,txt)
65 
66                 self.liste_case[pos2][-1].puittour=None
67                 self.deplacement(pos2 , jr)#on fait avancer normalement le joueur
68 
69             else:
70 
71                 jr.position=pos2#on inverse les position des joueurs de la class Joueur
72                 self.liste_case[pos2][1].position=pos1#
73                 self.liste_case[pos1][self.liste_case[pos1].index(jr)],self.liste_case[pos2][1]=self.liste_case[pos2][1],self.liste_case[pos1][self.liste_case[pos1].index(jr)]
74 
75                 print("Gare a toi voyageur du Temps , tu ne peux pas etre present sur la case de ton adversaire , Ainsi Commence un combat acharner entre ", self.liste_case[pos1][
76                 time.sleep(1)
77                 print("Le Gagnant est evidement ",self.liste_case[pos1][1].nom ," Bravo A lui , il echange alors sa place avec ",self.liste_case[pos2][1].nom)
78 
79         elif pos1!=pos2:#Cas de Base
80             self.liste_case[pos2][1].position=pos1#on inverse les position des joueurs de la class Joueur
81             self.liste_case[pos1][self.liste_case[pos1].index(jr)].position=pos2#
82 
83             self.liste_case[pos1][self.liste_case[pos1].index(jr)],self.liste_case[pos2][1]=self.liste_case[pos2][1],self.liste_case[pos1][self.liste_case[pos1].index(jr)]
84 
85             print("Gare a toi voyageur du Temps , tu ne peux pas etre present sur la case de ton adversaire , Ainsi Commence un combat acharner entre ", self.liste_case[pos1][1].n
86             time.sleep(1)
87             print("Le Gagnant est evidement ",self.liste_case[pos1][1].nom ," Bravo A lui , il echange alors sa place avec ",self.liste_case[pos2][1].nom)
```

Ligne 36 a 87 : Méthode permettant d'organiser les échanges des joueurs

Ligne 44 a 77 : on organise les échanges lorsque les joueurs sont sur des cases spéciales . Cependant il faut que pos1 et pos2 soit différents car si ce n'est pas le cas cela veut dire que le joueur est revenu sur sa case de depart (du tour)

Ligne 46 a 47: Sur la case hotel on accepte qu'il y ait plusieurs joueurs en même temps on appelle donc seulement la méthode déplacement

Ligne 49 a 61 : On accepte aussi que plusieurs joueurs soient sur la case prison cependant lorsqu'un joueur arrive sur la case prison il libère le dernier joueur arrivé en prison (c'est pour cela que l'on vérifie si le dernier joueur arriver est bloquée(l 56-57) afin de le libèrer),on appelle donc seulement la méthode déplacement

ligne 62 a 67: On libère le dernier joueur du puits et on affiche le message txt sur le plateau grâce a la fonction «affcaspos» du fichier pygame, et on accepte aussi plusieurs joueur sur le puits donc on appelle seulement la méthode

ligne 69 a 77:Si 2 joueurs s'apprête a être sur une même case spéciale (différentes du puits , de l'hotel , de la prison)(ce peut être la case depart), alors on effectue un echange de position simple . Le joueur qui joue prend la place de l'autre joueur (jr.position = pos2) et le joueur qui ne joue pas prend la position de depart du joueur qui joue (self.liste_case[pos2][1].position=pos1(on dit que l'objet en position pos2 aura pour position pos 1))

Ensuite on echange les objet sur la liste des cases **(self.liste_case[pos1][self.liste_case[pos1].index(jr)],self.liste_case[pos2][1]=self.liste_case[pos2][1],self.liste_case[pos1][self.liste_case[pos1].index(jr)])**

(self.liste_case[pos1].index(jr) (on cherche le a trouver l'index du joueur qui joue sur la liste des cases au cas ou il y a plusieurs jours sur la case du joueur qui joue (afin de ne pas échanger le mauvais joueur)))

Ligne 79 a 87 : On exécute le même code qu'aux ligne 62 a 67 mais dans ce cas si 2 joueurs s'apprête a être sur une même case qui n'est pas spécial

```
91
92      def deplacement(self,pos_hypot,jr):
93          if len(self.liste_case[jr.position])<=2:#self.cases_prises a remplacer par liste_case[0]==["depart",jr]
94              self.liste_case[jr.position][1]=""
95              if self.liste_case[pos_hypot][1]=="":
96                  self.liste_case[pos_hypot][1]=jr
97              else:
98                  self.liste_case[pos_hypot].append(jr)
99          else:
100             self.liste_case[jr.position].pop(self.liste_case[jr.position].index(jr))
101             if self.liste_case[pos_hypot][1]=="":
102                 self.liste_case[pos_hypot][1]=jr
103             else:
104                 self.liste_case[pos_hypot].append(jr)
```

Ligne 92 a 104 : Méthode permettant de faire deplacer un joueur

Ligne 93 a 98:On vérifie si les il n'y a qu'un joueur sur la case de départ du joueur si c'est pas le cas on remplace l'objet par des «» pour montrer que la case est désormais vide

Ligne 95 a 98 : si la case sur laquelle le joueur va est vide alors on remplace les «» par l'objet du joueur(l.96) .Sinon si le joueur va sur une case déjà prises alors on ajoute a la liste des joueurs de cette case l'objet du joueur(l 98)

Ligne 99 a 104 : Si il y a plusieurs joueurs sur la case de départ du joueur alors on supprime l'objet du joueur de sa case de départ et ensuite on fait la

même action qu'aux ligne 95 a 98

```
105
106     def est_jrsurcasespe(self,pos_hypot,jr):
107
108         if type(self.liste_case[pos_hypot][0])==str and pos_hypot!=0:#si il s'agit d'une case spe != de de depart
109             if jr.hoteltour==None and jr.prisontour==None:
110                 if jr.puittour==None:
111                     if self.liste_case[pos_hypot][0]=="prison":
112
113
114                         return (True,3)#si le joueur va sur la case prison pour la premiere fois
115                     else:
116                         return (True,1)#si le joueur est sur une autre case speciale
117
118         if type(self.liste_case[jr.position][0])==str and jr.position!=0:
119             if (jr.hoteltour!=None or  jr.prisontour!=None) or jr.puittour!=None:#si c'est egale a None alors il le jopueur veint de sortir
120                 return (True,2)#si le joueur vient d'une case speciale de type hotel prison et puits
121
122         return (False,1)#si le joueur ne va pas sur une case speciale
123
```

Ligne 106 a 122 : Méthode permettant de vérifier si un joueur est sur une case spécial ou va être sur une case spécial et de quel type est cette case

Ligne 108 a 114 : On vérifie si le joueur va être sur la case prison si oui on retourne (True , 3) pour éviter une boucle infini lors de l'exécution de la méthode **jrsurcasespe**(Car lorsque que l'on [eut aller en prison il se peut que le compteur de tour ne se lance pas (si quelqu'un est déjà sur la case prison) et donc que l'on relance True 1 en continu , Le TRUE ,3 permet de sortir de la boucle)

Sinon Ligne 115 a 116 : le joueur est déjà sur une case spéciale et a déjà commencée et emprisonnes sur cette case (Hôtel , prison , puits)

Sinon le Joueur ne vas pas aller sur une case spéciale donc on n'applique pas d'exception

```
123
124     def jrsurcasespe(self,de,poshypot,jr):
125         if self.est_jrsurcasespe(poshypot,jr)==(True,1) or self.est_jrsurcasespe(poshypot,jr)==(True,3):
126             if self.liste_case[poshypot][0]=="oie":
127                 return cases_oie(de ,jr , poshypot,self)
128
129             if self.liste_case[poshypot][0]=="pont":
130                 return cases_pont(de , jr , poshypot,self)
131
132             if self.liste_case[poshypot][0]=="hotel":
133                 return cases_hotelarri(de, jr ,poshypot,self)
134
135             if self.liste_case[poshypot][0]=="puit":
136                 return cases_puitarri(de, jr ,poshypot,self)
137
138             if self.liste_case[poshypot][0]=="labyrinthe":
139                 return cases_labyrinthe(de,jr,poshypot,self)
140
141             if self.liste_case[poshypot][0]=="prison":
142                 return cases_prisonarri(jr,poshypot,self)
143
144             if self.liste_case[poshypot][0]=="mort":
145                 return cases_mort(de,jr,poshypot,self)
146
147         if self.est_jrsurcasespe(poshypot,jr)==(True,2):
148
149
150             if self.liste_case[jr.position][0]=="hotel":
151                 return cases_hotelfin(de, jr ,poshypot,self)
152
153             if self.liste_case[jr.position][0]=="puit":
154                 return cases_puitfin(de, jr ,poshypot,self)
155
156             if self.liste_case[jr.position][0]=="prison":
157                 return cases_prisonfin(de,jr,poshypot,self)
158         else:
159             return poshypot
```

Ligne 124 a 159 :Méthode permettant de déterminer quelles exceptions doit exécuter le programme vis a vis du joueur

ligne 125 a 145 :*On cherche a savoir sur quelle sorte de case spéciale va aller le joueur a condition qu'il ne soit pas bloquer sur une autre case (True 1 et True 3 permettent de savoir que le joueur n'est pas bloquer sur une case spéciale (prisontour=None , hoteltour=None , puittour=None)) Si les conditions sont remplis alors on va appeler des fonctions qui vont permettre de déterminer quel est la nouvelle position du joueur et des conséquences que*

cela aura sur le joueur(prison ,...).On retourne ensuite cette position .
Ligne 147 a 157:*On cherche a savoir si le joueur doit rester sur sa case spéciale (Hôtel ,...) et si le joueur peut sortir alors on retourne sa nouvelle position sinon il reste sur la case spéciale*
Sinon Ligne 158 a 159 : *On retourne la nouvelle position du joueur au cas ou la situation du joueur ne corresponde pas a une de prévu*

```
162     def affichage(self):
163         '''for i in range(len(self.liste_case)):
164             print("|| ",self.liste_case[i][0],end="")
165             for z in range(1,len(self.liste_case[i])):
166                 if self.liste_case[i][z]=="":
167                     print(" *",end="    ")
168                 else:
169 
170                     print("  ",self.liste_case[i][z].nom)
171             print("|| ")'''
172         plateauvisu=[["Depart","1","2","3","4","Oie","Pont","7","8","Oie"],
173         ["","28","29","30","Puit","Oie","33","34","35","10"],
174         ["","Oie","48","49","Oie","51","Prison","53","Oie","11"],
175         ["","26","47","60","61","62","63","Oie","37","12"],
176         ["","25","46","Oie","LA MORT","57","56","55","38","13"],
177         ["","24","Oie","44","43","Labyrinthe","Oie","40","39","Oie"],
178         ["","Oie","22","21","20","Hotel","Oie","17","16","15"]]
179         plateau=[["0","1","2","3","4","5","6","7","8","9"],
180         ["","28","29","30","31","32","33","34","35","10"],
181         ["","27","48","49","50","51","52","53","36","11"],
182         ["","26","47","60","61","62","63","54","37","12"],
183         ["","25","46","59","58","57","56","55","38","13"],
184         ["","24","45","44","43","42","41","40","39","14"],
185         ["","23","22","21","20","19","18","17","16","15"]]
186 
187         self.ecran_accueil=pygame_jeu_de_oie.Ecran((139,69,19),plateauvisu,plateau)
188         pygame_jeu_de_oie.affichagepl(self.ecran_accueil,self.liste_jr)
189 
190 
```

Ligne 162 a 188 : Méthode permettant de créer deux matrices representant le plateau et en parti de l'afficher avec Pygame en utilisant la fonction affichagepl du fichier pygame_jeu_de_oie

Ligne 172 a 178 : on créer un matrices plateauvisu qui contient les noms des cases du plateau

Ligne 179 a 185 : on créer un matrices plateau qui contient les numeros des cases du plateau

Ligne 187 : on créer un objet de la classe écran(sur le fichier pygame_jeu_de_oie) qui contient la couleur de l'écran ainsi que les cases du plateau

Ligne 188 : on appelle la fonction affichagepl du fichier pygame_jeu_de_oie avec pour paramètres l'objet ecran_acceuil et la liste des joueurs afin d'afficher le plateau

```
191
192 def cases_oie(de, jr ,poshypot,plateaudejeu):
193     newpos=sum(de)+poshypot
194     txt=("Interdiction de restee sur les cases 'oie' vas t-en aussi vite que tu peux va a la case ",newpos  )
195     pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
196     return newpos
197
198
199 def cases_pont(de, jr ,poshypot,plateaudejeu):
200      txt=("Bonjour a toi jeune Voyageur Bienvenue Sur le petit pont de bois qui te premettra d'aller jusqu'a la case 12")
201      pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
202      return 12
203
204 def cases_hotelarri(de, jr ,poshypot,plateaudejeu):
205     txt=("Bienvenue a l'hotel 3 tours tu devras attendre Afin de pouvoir sortir")
206     pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
207     jr.hoteltour=0
208     return poshypot
209
210 def cases_puitarri(de, jr ,poshypot,plateaudejeu):
211     if not plateaudejeu.cases_prises(poshypot):
212         txt=("Bienvenue dans le puit , fait preuve de patiente et de ruse afin que quelqu'un prenne ta place")
213         pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
214     jr.puittour=0
215     return poshypot
216
217 def cases_prisonarri(jr,poshypot,plateaudejeu):
218     if plateaudejeu.cases_prises(poshypot):
219         if plateaudejeu.liste_case[poshypot][-1].prisontour!=None:
220             txt=("Sonner l'alarme ", jr.nom , "prepare une evasion dans la prison")
221             pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
222             return poshypot
223         else:
224             txt=("Bienvenue en prison tu devras attendre que quelqu'un t'aide a t'evader afin de sortir")
225             pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
226             jr.prisontour=0
227             return poshypot
228     else:
229         txt=("Bienvenue en prison tu devras attendre que quelqu'un t'aide a t'evader afin de sortir")
230         pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
231         jr.prisontour=0
232         return poshypot
```

Ligne 191 a 272 : Il s'agit de fonctions permettant de gérer les exceptions liées aux cases spéciales

Ligne 192 a 244 : Il s'agit de fonctions permettant de gérer les exceptions lorsque l'on arrive sur une case spéciale

ligne 192 a 196 : La fonction cases_oie ajoute a la position hypothétique du joueur la somme des des et affiche un texte (txt) sur l'ecran jeu de l'oie avec la fonction aff_casepos du fichier pygame_jeu_de_oie on retourne ensuite la nouvelle position du joueur

Ligne 199 a 202 : La fonction cases_pont affiche un texte (txt) sur l'ecran jeu de l'oie avec la fonction aff_casepos du fichier pygame_jeu_de_oie et modifie la position hypothétique (6 –> 12) , on retourne ensuite la nouvelle position du joueur

Ligne 204 a 208 : La fonction cases_hotelarri affiche un texte (txt) sur l'ecran jeu de l'oie avec la fonction aff_casepos du fichier pygame_jeu_de_oie et initialise la variable hoteltour de l'objet du joueur (Ceci marque le début pour le joueur de son enfermement a l'hotel)on retourne ensuite la nouvelle position du joueur

Ligne 210 a 215 : La fonction cases_puitarri affiche un texte (txt) sur l'ecran jeu de l'oie avec la fonction aff_casepos du fichier si et seulement si personne d'autres n'est sur cette case , sinon on affichera un autre message avec la méthode echangepos (cf . Ligne:62 a 67)pygame_jeu_de_oie et i nitialise la variable puitstour de l'objet du joueur (Ceci marque le début pour le joueur de son enfermement dans le puits)on retourne ensuite la nouvelle position du joueur

Ligne 217 a 232: La fonction cases_prisonarri va vérifier si le joueur va aller sur la case prison et rester bloqué dessus ou simplement y aller en libérant le joueur bloquée dessus

Ligne 218 a 227 : On vérifie si la case est déjà occuper et si oui on vérifie que la dernière personnes dessus est enfermé

Si c'est le cas alors le joueur libère la dernière personne en prison (on sait que la dernière personnes sur cette case est celle

a l'index -1 (suite au .append dans la méthode deplacement))on affiche alors un texte expliquant que le joueur prépare une évasion et on remet a None le compteur de tour de prison du joueur libéré afin qu'il puisse sortir au tour suivant(suita a la méthode est_jrsurcasespe qui retournera False,1)
et on retourne la nouvelle position du joueur
Sinon le joueur est enfermé en prison() on initialise son compteur prisontour et on affiche un message expliquant ceci et on retourne la nouvelle position du joueur

```
235
236 def cases_labyrinthe(de,jr,poshypot,plateaudejeu):
237     txt=("Tu T'aventure dans des Bien trop complexes tu retourne a la case 30")
238     pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
239     return 30
240 def cases_mort(de,jr,poshypot,plateaudejeu):
241     txt=("Tu echoue si proche du but , la vie t'offre une nouvelle chance mais tu repart au point de depart")
242     pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
243     return 0
```

Ligne 236 a 239 : La fonction Case_labyrinthe affiche un message expliquant les conséquences de cette cases sur le joueur et retourne la nouvelle position du joueur (30)
Ligne 240 a 243 : La fonction Case_mort affiche un message expliquant les conséquences de cette cases sur le joueur et retourne la nouvelle position du joueur (0)

```
248 def cases_hotelfin(de,jr,poshypot,plateaudejeu):
249     jr.hoteltour+=1
250     txt=("Tu es a l'hotel depuis ",jr.hoteltour," Tour ; Plus Que",4-jr.hoteltour, " a l'hotel")
251     if jr.hoteltour>3:
252         txt="Tu sors de l'hotel (et oui toute les bonnes choses ont une fin )"
253         jr.hoteltour=None
254         pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
255         return poshypot
256     else:
257         pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
258         return jr.position
259
260 def cases_prisonfin(de,jr,poshypot,plateaudejeu):
261     jr.prisontour+=1
262     txt=("Tu es enfermee en prison Depuis ", jr.prisontour," tour(s)")
263     pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
264     print("prison",jr.position)
265     return jr.position
266
267 def cases_puitfin(de,jr,poshypot,plateaudejeu):
268     jr.puittour+=1
269     txt=("Tu es Bloquee Dans Le Puit Depuis ", jr.puittour," tour(s)" )
270     pygame_jeu_de_oie.aff_casepos(plateaudejeu,txt)
271     print("puit",jr.position)
272     return jr.position
```

Ligne 248 a 272: _Il s'agit de fonctions permettant de gérer les exceptions lorsqu'un joueur est deja sur une case spéciale

Ligne 248 a 258:La Fonction cases_hotelfin incrémente de 1 le compteur de tour a l'hotel du joueur
on vérifie si le joueur est a l'hotel depuis plus de 3 tours

Si c'est le cas : on remet a None le compteur de tour a l'hotel du joueur et on affiche un message expliquant que le joueur est libéré et on retourne la nouvelle position du joueur
Sinon :On affiche que le joueur reste bloqué et on retourne la position actuelle du joueur étant donne que le joueur est bloquee

Ligne 260 a 265: La fonction cases_prisonfin incrémente de 1 le compteur de tour de prison du joueur et affiche que le joueur reste bloqué et retourne la position actuelle du joueur étant donne que le joueur est toujours bloquee

Ligne 267 a 272: La fonction cases_puitfin fonctionne de la meme maniere que la fonction cases_prisonfin mais on incremente le compteur de tour de puits du joueur

```
277 def crea_pla(liste_jr):
278     case_spe=[[6,"pont"],[19,"hotel"],[31,"puit"],[42,"labyrinthe"],[52,"prison"],[58,"mort"]]
279 
280     plateau=[[i,""] for i in range(1,64)]
281     plateau=[["Depart"]] +plateau
282 
283     for i in range(5,63,9):
284         plateau[i][0]="oie"
285     for i in range(9,63,9):
286         plateau[i][0]="oie"
287 
288 
289     for i in range(len(case_spe)):
290         plateau[case_spe[i][0]][0]=case_spe[i][1]
291 
292 
293     for i in liste_jr:
294         plateau[0].append(i)
295 
296     return plateau
297 
298 
```

Ligne 277 a 296 : La fonction crea_pla va generer un plateau automatiquement

Ligne 278 : on definit les case speciale avec leurs nom et leurs index

Ligne 280: on creer le tableau par comprehension

Ligne 281:on ajoute une case depart par concatenation

Ligne 283 a 287: on ajoute au plateau des cases oies en remplacant les numeros des cases oie par des oie

Ligne 289 a 290: on remplace le numero des cases speciale par leurs noms

Ligne 293 a 294 : On ajoute a la case depart les objets joueurs

Ligne 296: On retourne le plateau

```
303 class joueur():
304     def __init__(self,nom,colorrg):
305         self.nom=nom
306         self.position=0
307         self.tour=0
308         self.hoteltour=None
309         self.prisontour=None
310         self.puittour=None
311         self.color=colorrg
312     def gagnant(self):
313         return self.position==63
314 
```

Ligne 303 a 313: On créer la classe Joueur

Ligne 304 a 311: Dans le constructeur on initialise toute les variable de l'objet qui seront utile tel que la couleur du joueur sous forme rgb , son nom sa position , le nombre de tour qu'il a passé a l'hôtel en prison dans le puits

Ligne 312 a 313 : Méthode permettant de déterminer si le joueur a gagné

```
318 def lancement_jeu():
319     liste_jr=[]
320     nbr_jr=int(input("combien de joueurs vont jouer( il doit y a avoir -= 4 jr : \n"))
321     while nbr_jr>4:
322         nbr_jr=int(input("combien de joueurs vont jouer( il doit y a avoir -= 4 jr : \n"))
323 
324     lcolor=[["vert",(0,255,0)],["bleu",(0,0,255)],["rouge",(255,0,0)],["jaune",(255,255,0)]]
325     for i in range(nbr_jr):
326         nom=str(input("quel est le nom du joueur :"))
327         print("liste des couleurs : ")
328         for i in range (len(lcolor)):
329             print(lcolor[i][0]," = ",i)
330 
331         color=int(input("Quelle couleur souhaite-il : "))
332 
333         while color > len(lcolor)-1 or color<0:
334             print("liste des couleurs : ")
335             for i in range (len(lcolor)):
336                 print(lcolor[i][0]," = ",i)
337 
338                 color=int(input("Quelle couleur souhaite-il : "))
339 
340         nom=joueur(nom,lcolor[color][1])
341         liste_jr.append(nom)
342         lcolor.pop(color)
343     plateaudejeu=plateau(liste_jr)
344     print(liste_jr)
345     fin=False
346     tour=0
347     plateaudejeu.affichage()
348     pygame_jeu_de_oie.jeu_plateau(plateaudejeu,liste_jr,tour,fin)
349     '''while not fin:
350         tour+=1
351 
352         for i in range(len(liste_jr)):
353 
354             print("Tour",tour,"\n a " ,liste_jr[i].nom,"de jouer")
355             jeu(plateaudejeu,liste_jr[i])
356 
357 
358             fin=liste_jr[i].gagnant()'''
359 
```

Problème d'indentation corrigé dans le code

Ligne 318 a 348 : La fonction lancement_jeu va permettre d'initialiser le jeu en créant les joueur et le plateau et en lançant le jeu

Ligne 319 a 342: On demande des information pour la creation des joueur tel que le nombre de joueur leurs couleurs et leurs noms ensuite on ajoute l'objet joueur a la liste des joueur(liste_jr) et ensuite on retire du choix des couleurs la couleur choisit

Ligne 343 et 347: on creer l'objet plateaudejeu de la classe plateau et on lance la methode affichage afin de creer un objet ecran pour la partie pygame

Ligne 345 a 347 : on appelle la fonction jeu_plateau du fichier pygame_jeu_de_oie en initialisant la variable fin a False pour rentrer dans la boucle while de la fonction jeu_plateau

```
361 def jeu(plateaudejeu,jr,tour):
362
363     jr.tour+=1
364     #print(plateaudejeu.affichage())
365     print("Voici la position ",jr.nom," joueur au debut du tour  ; Case = ",jr.position)
366     position=jr.position
367     pygame_jeu_de_oie.lancedeveri()
368
369     de=dee_lancement(plateaudejeu)
370     print("de1", de[0] , "de2" , de[1])
371     pos_hypot=position+sum(de)
372     pos_hypot=premier_tour_exception(jr,de,pos_hypot)
373     pos_hypot=supa63(pos_hypot)
374     if plateaudejeu.est_jrsurcasespe(pos_hypot,jr)[0]==True:#ici on met une condition pour la
375         #print("pl",plateaudejeu.est_jrsurcasespe(pos_hypot,jr))
376
377         #print("posi1",pos_hypot)
378         pos_hypot=plateaudejeu.jrsurcasespe(de,pos_hypot,jr)
379
380         #print("posi",pos_hypot)
381         pos_hypot=supa63(pos_hypot)
382
383         while plateaudejeu.est_jrsurcasespe(pos_hypot,jr)==(True,1):
384             pos_hypot=plateaudejeu.jrsurcasespe(de,pos_hypot,jr)
385             pos_hypot=supa63(pos_hypot)
386
387     avance(jr,pos_hypot,plateaudejeu)
388
389     print(jr.position)
390     #time.sleep(4)
```

Ligne 361 a 390: La fonction jeu permet de faire tourner le jeu

Ligne 363: on incremente de 1 le compteur de tour du joueur

Ligne 366: on affecte a la variable position la position du joueur au debut du tour

Ligne 367 a 373: on appelle la fonction lancedeveri du fichier pygame_jeu_de_oie qui va verifier si le joueur a appuyer sur les cases permettant de lancer les des , si c'est le cas alors on appelle la fonction dee_lancement qui retourne une liste qui contient les resultats des des et on va ensuite en deduire une premiere position hypothétique , on va ensuite verifier ligne 372 si une 1ere exception s'applique (dans ce cas c'est lorsque c'est le premier tour et que le joueur fait 9) , on verifie ensuite si cette position ne depasse la case 63 (ligne 372 avec la fonction supa63)

Ligne 374 a 385: On verifie qu'il n'y ait pas d'exceptions (si le joueur n'est pas sur une case speciale ou ne va pas sur une case speciale)On applique pour cela la methode jrsurcasespe pour determiner quelles exceptions , cela va permettre de determiner une nouvelle position hypothétique et tant que cette position hypothétique ne correspond pas on rappelle la methode tout en verifiant que l'on ne depasse pas 63

Ligne 387: Maintenant que l'on a trouver une position hypothétique sur on fait avance le joueur a cette position

```
392 def dee_lancement(plateaudejeu):
393     de1=randint(1,6)
394     de2=randint(1,6)
395     #de1=int(input("valde1"))
396     #de2=int(input("valde2"))
397     pygame_jeu_de_oie.affichede(de1,de2,plateaudejeu)
398
399     return [de1,de2]
```

Ligne 392: La fonction dee_lancement permet de determiner la valeur des des et de les afficher

Ligne 393 et 394: on determine la valeur des 2 des

Ligne 395 a 396 : ces lignes de codes mises en commentaires nous ont permis de tester notre programme (exception , …) en choisissant son les valeurs des des

ligne 397: on affiche la valeur des des sur l'ecran en uttilisant la fonction affichede du fichier pygame_jeu_de_oie

Ligne 399: On retourne sous forme d'une liste la valeur des des

```
402 def premier_tour_exception(jr,de,poshypot):
403     if jr.tour==1:
404         if sum(de)==9:
405             if de==[6,3] or de==[3,6]:
406                 return 26
407             else:
408                 return 53
409         else:
410             return poshypot
411     else:
412         return poshypot
```

Ligne 402: Fonction premier_tour_exception permet de determiner la position hypothétique d'un joueur lorsqu'il fait 9 au premier tour

Ligne 403 a 408 : On verifie qu'il s'agit du premier tour , et que la somme des des est 9 et de quelle façon la somme des des fait 9 Ensuite on retourne une nouvelle position hypothétique 26 ou 53

Ligne 409 a 412: Si les conditions ne sont pas reunis alors ont retourne la valeur hypothétiqued'entree

```
414 def avance(jr,pos_hypot,plateaudejeu):
415     pos_hypot=supa63(pos_hypot)#
416     if plateaudejeu.cases_prises(pos_hypot):
417         plateaudejeu.echangespos(jr,jr.position,pos_hypot)#jr.position = pos1
418     else:
419         plateaudejeu.deplacement(pos_hypot,jr)
420     jr.position=pos_hypot
421
```

Ligne 414 : Fonction avance permet de faire avance le joueur jusqu'a sa nouvelle position

Ligne 415: On determine une nouvelle fois quelle est la position hypothétique du joueur en verifiant qu'il ne depasse pas 63

Ligne 416 a 417: On verifie si la case suivante du joueur n'est pas prises et si elle est prises alors on uttilise la methode echangespos pour proceder a l'echange

Sinon : On uttilise la methode deplacement pour faire se deplacer le joueur

Ligne 420 : on donne comme nouvelle position au joueur sa position hypothétique

```
422 def supa63(pos_hypo):
423
424     if pos_hypo>63:
425
426         return 63-(pos_hypo-63)
427     return pos_hypo
428
```

Ligne 422: Fonction supa63 elle permet de determinee la position hypothétique du joueur si il depasse ou non la case 63

Ligne 424 a 426: On verifie que la position hypothétique du joueur ne depasse pas 63

et si ce n'est pas le cas alors on fait reculer le joueur du nombre de case qu'il depasse de la case 63 et on retourne sa nouvelle position hypothétique

SINON Ligne 437 : On retourne la position hypothétique sans changement

```
441 def classement_partie(liste_jr):
442     for i in range(0,len(liste_jr)-1):
443         imax=i
444         for z in range(i+1,len(liste_jr)):
445             if liste_jr[z].position>liste_jr[imax].position:
446                 imax=z
447         liste_jr[i],liste_jr[imax]=liste_jr[imax],liste_jr[i]
448     for t in range(len(liste_jr)):
449         print(liste_jr[t].position)
450     return liste_jr
451
452 liste_jr=lancement_jeu()
```

Ligne 441:La fonction classement_partie permet de trier la liste des objets joueurs par rapport a leurs position dans le jeu a l'aide d'une methode de tri par selection

ligne 442 a 447 : on parcours la liste en comparant la position de l'objet de l'index imax aux autres et lorsque un objet est superieur a l'objet a l'index imax alors imax prend pour valeur l'index de l'objet superieur et ensuite on echanges l'objet a l'index i avec l'objet a l'index imax

Ligne 450 : On retourne le tableau triee

CODE PYGAME

```
 8 import projet_jeu_de_loie_beta_part1
 9 import pygame,sys,time
10 from pygame.locals import*
11 from random import randint
12 import time
13
14
15 pygame.init()
```

Ligne 8 a 15: On Importe toute les modules nescessaire et le fichier projet_jeu_de_loie_beta_part1

```
20 class Ecran:
21     def __init__(self,couleur,case,casenumber):#proportion pour donner qu
22         self.couleur=couleur
23         self.fen_axex=900#longueur ecran
24         self.fen_axey=900#hauteur ecran
25         self.case=case#on stocke les nom des cases
26         self.casenumber=casenumber#on stocke les numeros associes aux cas
27         self.pointdedepartx=0#point de placement de base pour la premiere
28         self.pointdedeparty=100
29         self.la=50#largeur des cases du plateau
30         self.lon=60#longueur des cases du plateau
31         self.taillejr=20#taille des images des joueurs
32
33         self.fen=pygame.display.set_mode((self.fen_axex,self.fen_axey))
```

Ligne 20 a 33: on creer une classe Ecran qui va stocker des donnees sur un ecran tel que la couleur de fond(self.couleur) la taille des cases du plateau(self.la , self.lon) la taille de l'ecran (self.fen_axex , self.fen_axey), la taille des joueurs(self.taillejr)

```
35 def affichetourjr(plateaudejeu,jr,tour):
36     pygame.draw.rect(plateaudejeu.ecran_accueil.fen,jr.color,(500,20,40,40))#on affiche la couleur du joueur a
37     font= pygame.font.Font(None ,20)
38 
39     '''txt = "TOUR", tour , "C'EST A ",jr.nom,"DE JOUER"# on ecrit le texte txt sur le plateau pygame
40     image =font.render(str(txt), 1 , (255,255,255) )
41 
42     plateaudejeu.ecran_accueil.fen.blit(image,(40,20))
43     pygame.display.update()'''
44 
45     txt = "TOUR", tour , "C'EST A ",jr.nom,"DE JOUER"
46     image =font.render(str(txt), 1 , (255,255,255) )
47 
48     plateaudejeu.ecran_accueil.fen.blit(image,(40,20))
49     pygame.display.update()
```

Ligne 35 a 49 : la fonction affichetourjr affiche qui est le joueur sur l'ecran

Ligne 36: on Fait un carre de la couleur du joueur afin de savoir a quelle couleur est associe le joueur

Ligne 37 a 49: on ecrit qui est le joueur actuel et le numero du tour

ligne 37: on determine le format d'ecriture et la taille d'ecriture

Ligne 45 : on definit le texte a ecrire

ligne 46 : on affecte une couleur aux texte

Ligne 48 : On determine une position pour le texte

Ligne 49 : on actualise l'ecran

```
51 def aff_casepos(plateaudejeu,txt):
52     font= pygame.font.Font(None ,20)
53     image =font.render(str(txt), 1 , (255,255,255) )
54     plateaudejeu.ecran_accueil.fen.blit(image,(40,80))
55     pygame.display.update()
```

Ligne 51 a 55: La fonction aff_casepos permet d'afficher un texte en dessous du texte disant quel joueur joue , le texte afficher correspond au texte renvoyer par les methodes et fonction gerant les exceptions et permet de connaitre la position du joueurs et la positions dans laquelle il est (Hotel ,prison ,...)

```
58 def affichage_temps_pause(plateaudejeu):
59     coord=(725,300,112,40)
60     font= pygame.font.Font(None ,20)
61 
62     image =font.render("Joueur Suivant", 1 , (255,255,255) )
63     plateaudejeu.ecran_accueil.fen.blit(image,(coord[0],coord[1]+(coord[3]/2)))
64 
65 
66     pygame.display.update()
67     pygame.draw.rect(plateaudejeu.ecran_accueil.fen,(0,0,0),coord,5)
68     pygame.display.update()
69     boutsui=False
70     while boutsui==False:
71         for event in pygame.event.get():
72                         #print(pygame.event.get)
73                         if event.type==QUIT:
74                             pygame.quit()
75                             sys.exit()
76                         elif event.type==KEYDOWN:
77                             print(event.key)
78                         elif event.type == MOUSEBUTTONDOWN :
79                             print("Event pos",event.pos)
80                             if event.pos[0]>coord[0] and event.pos[0]<coord[0]+coord[2]:
81                                 print("x ok")
82                                 if event.pos[1]<coord[1]+coord[3] and event.pos[1]>coord[1]:
83                                     print("Y ok")
84 
85                                     #print("reussi ev",event.pos)
86                                     return True
87 
```

Ligne 58 a 87: La fonction affichage_temps_pause permet de laisser le temps au joueur de regarder le score de ses des , sa position ,ou fermer la page ou alors passer au tour suivant

Ligne 59 : on donne la taille et la position du boutton (rectangle) sur lequel il faut appuyer pour passer au joueur suivant

Ligne 60 a 66 : on ecrit le texte “Joueur suivant“ sur l’ecran

Ligne 67 a 68: On dessine un rectangle transparant (aux contours noirs) qui servira de boutton et qui aura comme coordonees et taille , le tuple coord

Ligne 69 a 86: On collecte tout les evenements et ensuite on verifie si l’uttilisateur a souhaitee quitter la partie (ligne 73 a 75) ou si il a decide de passer au tour suivant c’est a dire d’appuyer sur le boutton “joueur suivant”(boutton qui se situe sur l’axe x entre coord[0] et coord[0] +coord[2] et sur l’axe x entre coord[1] et coord[1] +coord[3])

```
90 def affichagepl(obj,liste_jr):#obj vient de la case ecran
91 ##    pygame.display.update()
92     #obj.affichage()
93     obj.fen.fill(obj.couleur)
94     #pygame.display.update()
95     font= pygame.font.Font(None ,20)
96     listecasecolor=[]
97     for i in range(len(liste_jr)):
98         listecasecolor.append(liste_jr[i].position)
99 
100    for i in range(len(obj.case)):
101            #if i%2==0:
102                for z in range(len(obj.case[i])):
103                    if obj.case[i][z]!="":
104 
105                        #print("z=",z,"i=",i)
106                        #print("lo=",z*obj.lon,"la=",z*obj.la)
107                        #print("pointcase x",obj.pointdedepartx+(z*(obj.lon+0.1*obj.lon)),"pointcase y ",obj.pointdedeparty+(i*(obj.la+0.4*ob
108                        pygame.draw.rect(obj.fen,(0,0,0),(obj.pointdedepartx+(z*(obj.lon+0.1*obj.lon)),obj.pointdedeparty+(i*(obj.la+0.4*obj.
109                        image =font.render( obj.case[i][z], 1 , (255,255,255) )
110                        obj.fen.blit(image,(obj.pointdedepartx+(z*(obj.lon+0.1*obj.lon)),obj.pointdedeparty+(i*(obj.la+0.4*obj.la))))
111                        #print("obj",obj.casenumber[i][z] , "lst",listecasecolor)
112                        #time.sleep(0.1)
113                        #if obj.casenumber[i][z] in listecasecolor:
114                         #   print("bien,1")
115                        for d in range(len(listecasecolor)):
116                                #print("lst",int(listecasecolor[d]),"obj",int(obj.casenumber[i][z]))
117                               # print(int(obj.casenumber[i][z])==int(listecasecolor[d]))
118                                if int(obj.casenumber[i][z])==int(listecasecolor[d]):
119                                    #print("bien")
120                                    positionjroncase=obj.pointdedepartx+z*(obj.lon+0.1*obj.lon)+10+d*10
121                                    while positionjroncase+obj.taillejr>obj.pointdedepartx+(z*(obj.lon+0.1*obj.lon))+obj.lon:
122 
123                                        obj.taillejr=obj.taillejr//2
124                                        #print("o")
125                                    pygame.draw.rect(obj.fen,liste_jr[d].color,(obj.pointdedepartx+z*(obj.lon+0.1*obj.lon)+10+d*10,obj.pointd
126                                    pygame.display.update()
127                        #time.sleep(0.1)
128                        pygame.draw.rect(obj.fen,(0,0,0),(725,129,40,40),5)
129                        pygame.draw.rect(obj.fen,(0,0,0),(800,129,40,40),5)
130 
131                        pygame.display.flip()
132                        pygame.display.update()
```

Ligne 90 a 132: la fonction affichagepl permet le plateau ainsi que les joueurs

ligne 93 : on remplace l’ecran par ecran de couleur obj.couleur

ligne 95: on definit une taille et une police d’ecriture

ligne 96 a 98 : on creer une liste (listecasecolor) qui va stocker la position des joueurs

Ligne 100 : on rentre dans une boucle qui se repete autant de fois qu’il y a d’element dans la matrice obj.case

Ligne 102 a 110: On creer les cases du plateau

Ligne 103 : on verifie que l’element a l’index z du tableau i n’est pas “ ” car sinon il ne s’agit pas d’une case

```
plateauvisu=[["Depart","1","2","3","4","Oie","Pont","7","8","Oie"],
["","28","29","30","Puit","Oie","33","34","35","10"],
["","Oie","48","49","Oie","51","Prison","53","Oie","11"],
["","26","47","60","61","62","63","Oie","37","12"],
["","25","46","Oie","LA MORT","57","56","55","38","13"],
["","24","Oie","44","43","Labyrinthe","Oie","40","39","Oie"],
["","Oie","22","21","20","Hotel","Oie","17","16","15"]]
```

Les parties entourées en bleu ne sont pas des cases

Ligne 108 a 110 : on dessine les rectangles (representant les cases) les un par rapport aux autres en y ajoutant les noms des cases associes

Ligne 115 a 126 : On ajoute les joueurs sur les cases

Ligne 115 a 118: on verifie si parmis les joueurs un d’eux est a le meme numero pour sa position que l’element en positionz de la liste i de la matrices casenumber(differente de la matrice case , elle permet de connaitre les index

de toutes les cases (speciales comprises))

Ligne 121 a 123 : on modifie la taille des carres des joueurs afin qu'ils puissent rentrees dans une case (plus ou moins inneficace (***POINT a Ameliorer***))

Ligne 125 a 126: On dessine les carres(joueurs) dans les cases (les carres ont la couleur choisis par le joueur au lancement du jeu)

Ligne 128 a 132: On ajoute des cases qui nous serviron de de a l'avenir

```
143 def lancedeveri():
144     delan=False
145     while delan==False:
146         for event in pygame.event.get():
147                         #print(pygame.event.get)
148                         if event.type==QUIT:
149                             pygame.quit()
150                             sys.exit()
151                         #elif event.type==KEYDOWN:
152                          #   print(event.key)
153                         elif event.type == MOUSEBUTTONDOWN :
154                             #print("Event pos",event.pos)
155                             if event.pos[0]>600 and event.pos[0]<900:
156                                 if event.pos[1]>124 and event.pos[1]<172:
157
158                                     #print("reussi ev",event.pos)
159                                     return True
```

Ligne 143 a 159: Fonction lancedeveri qui permet de la lance les des lorsque l'on appui sur une des cases en haut a droite

ligne 144 a 145: On rentre dans une boucle infini qui ne s'arretera que si on souhaite fermer la fenetre (event.type==QUIT) ou si on lance les des c'est a dire si on appui sur l'axe des x entre 600 et 900 et sur l'axe des y entre 124 et 172

Dans le cas ou on appui sur les des alors on repasse sur la fonction jeu du fichier Projet_jeu_de_loie_beta_part1 et lancera ensuite la fonction de lancement de des

/!\ Ceci permet donc de marquer une pause entre le moment ou le joueur commence a jouer et le moment ou il jette les des /!

```
161 def affichede(de1,de2,plateaudejeu):
162     font= pygame.font.Font(None ,20)
163     image1 =font.render( str(de1), 1 , (255,255,255) )
164     plateaudejeu.ecran_accueil.fen.blit(image1,(735,135))
165     image2 =font.render( str(de2), 1 , (255,255,255) )
166     plateaudejeu.ecran_accueil.fen.blit(image2,(810,135))
167     font= pygame.font.Font(None ,30)
168     image3 =font.render( str(de1+de2), 1 , (255,255,255) )
169
170
171     plateaudejeu.ecran_accueil.fen.blit(image3,(772,200))
172     pygame.display.flip()
173     pygame.display.update()
```

Ligne 161 a 173 : Fonction affichede qui va afficher les résultats de des et leurs sommes

Ligne 162 a 171 : on définit la taille et la police d'écriture ainsi que les différents textes contenant les résultats des des et leurs positions sur l'ecran .

Ligne 172 a 173 : on actualise l'ecran

```
177 def affga(ecran,listejr):
178     font= pygame.font.Font(None ,20)
179     pos=(300,100)
180     ecran.fen.fill(ecran.couleur)
181     txt=str("LE GRAND GAGNANT EST "+ str(listejr[0].nom))
182     texte =font.render( txt.upper(), 1 , (255,255,255) )
183     ecran.fen.blit(texte,pos)
184     pygame.display.flip()
185     pygame.display.update()
186     for i in range(1,len(listejr)):
187         txt=str("LE Joueur en position " + str(i) +" est "+(str(listejr[i].nom)).upper() +" et est arriver en positi
188         texte =font.render( txt, 1 , (255,255,255) )
189         ecran.fen.blit(texte,(pos[0],pos[1]+(40*i)))
190         pygame.display.flip()
191         pygame.display.update()
192     pygame.draw.rect(ecran.fen,(0,0,0),(300,pos[1]+(i+5)*40,300,40),2)
193     pygame.display.update()
194     texte =font.render( "__FIN__", 1 , (255,255,255) )
195     ecran.fen.blit(texte,(450,pos[1]+(i+5.5)*40))
196     pygame.display.flip()
197     pygame.display.update()
198     over=False
199     while not over:
200         for event in pygame.event.get():
201                         if event.type==QUIT:
202                             pygame.quit()
203                             sys.exit()
204                         elif event.type == MOUSEBUTTONDOWN :
205                             #print("Event pos",event.pos)
206                             if event.pos[0]>300 and event.pos[0]<600:
207                                 if event.pos[1]>pos[1]+(i+5)*40 and event.pos[1]<pos[1]+(i+6)*40:
208
209                                     #print("reussi ev",event.pos)
210                                     return True
```

Ligne 177 a 210 : la fonction affga permet d'afficher le classement final(il faut que la liste des joueurs soit au préalable triée)

ligne 178 : on définit la taille et la police d'ecriture

ligne 179 : on définit la position de départ de l'affichage

ligne 180 : on recouvre l'écran précédent pour afficher le score

ligne 181a 185 : on affiche le nom du gagnant

ligne 186 a 191 : on affiche le classement des joueurs

ligne 192 a 197 : on créer un bouton avec ecrit dessus « ___FIN__ »

Ligne 199 a 210 : On rentre dans une boucle infini qui ne s'arretera que si on souhaite fermer la fenetre (event.type==QUIT) ou si on appuie sur le bouton __FIN__ c'est a dire si on appui sur l'axe des x entre 300 et 600 et sur l'axe des y entre pos[1]+(i+5)*40 et pos[1]+(i+6)*40

```
211
212 def jeu_plateau(plateaudejeu,liste_jr,tour,fin):
213
214     #plateaudejeu.affichage()
215     affichagepl(plateaudejeu.ecran_accueil,liste_jr)
216     while not fin:
217
218         for event in pygame.event.get():
219                     #print(pygame.event.get)
220                     if event.type==QUIT:
221                         pygame.quit()
222                         sys.exit()
223
224         tour+=1
225
226
227         for i in range(len(liste_jr)):
228             if not fin:
229                 #plateaudejeu.affichage()
230                 affichagepl(plateaudejeu.ecran_accueil,liste_jr)
231                 affichetourjr(plateaudejeu,liste_jr[i],tour)
232                 projet_jeu_de_loie_beta_part1.jeu(plateaudejeu,liste_jr[i],tour)
233
234
235                 fin=liste_jr[i].gagnant()
236             if not fin:
237                 affichage_temps_pause(plateaudejeu)
238
239     liste_jr=projet_jeu_de_loie_beta_part1.classement_partie(liste_jr)
240     affga(plateaudejeu.ecran_accueil,liste_jr)
241
242     pygame.quit()
243     sys.exit()
```

Ligne 212 a 243 : Fonction jeu_plateau est une sorte de centre car elle permet de coordonnées la partie pygame et la partie fonctionnement en affichant le plateau et en lançant la fonction jeu du fichier projet_jeu_de_loie_beta_part1

Ligne 215 : on appelle la fonction affichagepl qui va afficher le plateau

ligne 216 a 237 : On rentre dans une boucle infini qui ne s'arretera que si on souhaite fermer la fenetre (event.type==QUIT) ou si un joueur gagne avec while not fin(fin est définit dans les paramètre de la fonction), cette boucle permet l'affichage du plateau et le fonctionnement du jeu

Ligne 218 a 222 : on vérifie que le joueur ne souhaite pas quitter le jeu

Ligne 227 a 237 : on lance le jeu pour tout les joueurs

Ligne 228 : on ne lance le jeu que si personne n'as encore gagne (fin==False) (car si on enlève cette ligne , si un le joueur 1 gagne ,le joueur 2 va tout de même jouer et va annuler la victoire du joueur 1 sans avoir gagner (le joueur 1 ne peut pas gagner)

Ligne 130 a 132 :On affiche la position des joueurs et le plateau (***l.130***)et le nom du joueur actuel(***l.131***) et on lance la partie jeu(***l.132***)

Ligne 135 : On actualise la variable gagnant afin de vérifier si il y a un gagnant

Ligne 236 -237 : On vérifie si il y a un gagnant et si non alors on appelle la fonction affichage_temps_pause qui va permettre de laisser le temps dont a besoin le joueur pour voir son avancé

Ligne 239 : on appelle la fonction classement_partie du fichier projet_jeu_de_loie_beta_part1 pour trier la liste des joueur afin d'en faire en classement

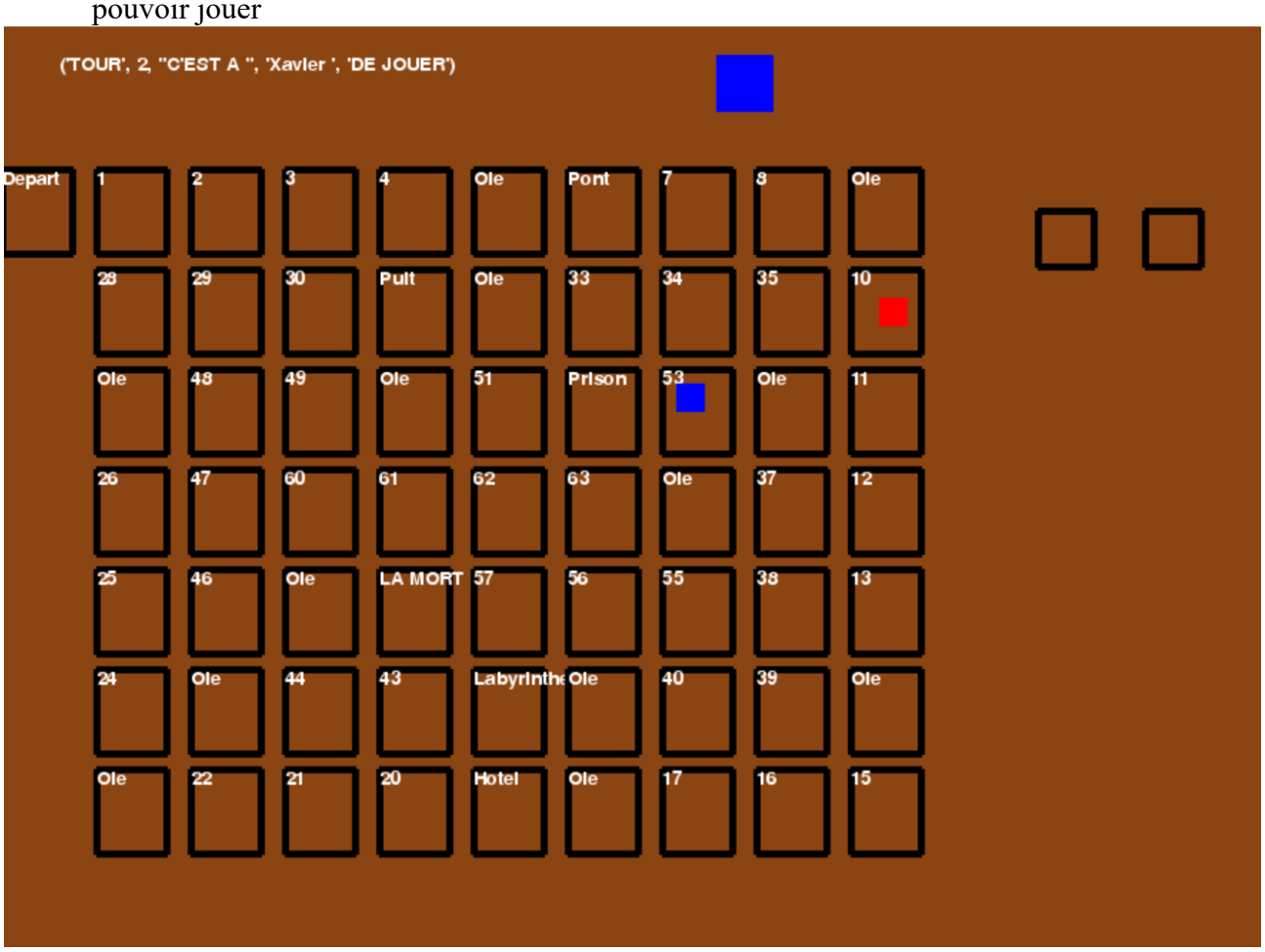
Ligne240 : On appelle la fonction affga qui va afficher le classement et cela va permettre de mettre en temps de pause pour que les joueurs aient le temps de regarder le classement avant la fin du jeu

Ligne 142 a 143 : On ferme la fenetre pygame et le programme ce qui met fin au jeu

---

DESIGN :

_Le design du jeu est plutôt épurée pour l'instant , avec uniquement le nécessaire pour pouvoir jouer



Idées d'amélioration :

_Créer plusieurs écran pygame (lancement de jeu , plateau de jeu)

_faire des pions plus ressemblant que des carres (ex : un rectangle avec un rond dessus cf image)

_Faire que les pions se déplace case par case

_ rendre l'affichage plus générale (que les tailles et les positions des rectangles soit proportionnelle a n'importe qu'elle taille d'écrans)

_Diminuer la complexité sur la partie affichage du pygame

_Optimiser le code générale du Jeu de l'oie pour le rendre plus lisibles

_améliorer le design du jeu en rendant plus lisible le sens du plateau et en rajoutant des photos sur les cases