

```

1
2 import Milieu
3 import threading
4 import time
5
6 import tkinter as tk
7 from tkinter import Tk
8
9 import pyscreenshot as ImageGrab
10
11 from Milieu import listescores
12
13 import marshal
14
15 def init_var(compteur): #initialise toutes les variables
16     global gamma,actions,etats,Q,cpteur
17     cpteur=compteur
18     gamma = 0.9
19     actions = Milieu.actions
20     etats = []
21     Q = {}
22     #Q = Milieu.lecture('Q.txt') #On lit une matrice Q précédemment enregistrée
23     cpteur=0
24
25
26 init_var(0)
27
28 def init_mat():
29     #on initialise tous les états possibles , soit tous les tuples (a,b) où a,b
    #appartiennent à [|0;x|]*[|0;y|]
30     for i in range(Milieu.x):
31         for j in range(Milieu.y):
32             etats.append((i, j))
33
34     #ce passage est à commenter si la matrice Q est chargée depuis un fichier
35     for etat in etats:
36         temp = {}
37         for action in actions:
38             temp[action] = 0.1
39         Q[etat] = temp
40
41     for (i, j, c, w) in Milieu.special:
42         for action in actions:
43             Q[(i, j)][action] = w
44
45 def signal_arret(liste,stop):
46     #cette fonction nous permet d'envoyer un signal d'arret au programme principal
    #quand le score a convergé
47     #on considère qu'il y a convergence lorsqu'un même score se répète "stop" fois
48     n=len(liste)
49     if n<stop+1:
50         return False
51     aux= liste[n-1]
52
53     for k in range(n-2,n-stop,-1):
54         if liste[k] != aux:
55             return False
56         aux = liste[k]
57     return True
58
59
60 def faire(action):
61     s = Milieu.coordonnees
62     r = -Milieu.score
63
64     if action == actions[0]:
65         Milieu.mouvement(0, -1)
66     elif action == actions[1]:
67         Milieu.mouvement(0, 1)

```

```

68     elif action == actions[2]:
69         Milieu.mouvement(-1, 0)
70     elif action == actions[3]:
71         Milieu.mouvement(1, 0)
72     else:
73         return
74     s2 = Milieu.coordonnees
75     r += Milieu.score
76     return s, action, r, s2
77
78 def max_Q(s):#retourne la plus grande valeur de Q[s] et l'action associée
79     val ,act= 0,0
80     for a, q in Q[s].items():
81         if val == 0 or (q > val):
82             val , act = q,a
83     return act, val
84
85 def inc_Q(s, a, alpha, inc):
86     Q[s][a] = round((1-alpha)*Q[s][a] + alpha*inc , 4) #incrémente la valeur de
87     la matrice Q
88
89 def lancer():
90
91     global cpteur,listescores,gamma,alpha
92     #time.sleep(1)
93     alpha = 1
94     t = 1
95     while not signal_arret(listescores,100):
96         # Choix de l'action menant à la meilleure récompense
97         s = Milieu.coordonnees
98         max_act, max_val = max_Q(s)
99         (s, a, r, s2) = faire(max_act)
100
101         #print("Q("+str(s)+") =" +str(Q[s]))
102
103         # Modification de la matrice Q
104         max_act, max_val = max_Q(s2)
105
106         inc_Q(s, a, alpha, r + gamma * max_val)
107
108         # on verifie si le monde doit être réinitialisé
109         t += 1.0
110         if Milieu.etat_reinit() or abs(Milieu.score) > 10000: #limite nécessaire
111             Milieu.reinitialisation()
112             cpteur += 1
113             #time.sleep(0.001)
114             t = 1.0
115
116         # vitesse de rafraichissement de l'interface graphique
117         #time.sleep(0.5)
118
119         # lignes de capture d'écran
120         x = Milieu.master.wininfo_rootx()
121         y = Milieu.master.wininfo_rooty()
122         w = Milieu.master.wininfo_width()
123         h = Milieu.master.wininfo_height()
124
125         titre= "matrice "+str(Milieu.x)+"*"+str(Milieu.y)+" gamma:"+str(gamma)+ "
126         alpha:" + str(alpha)
127         img= ImageGrab.grab((x+2, y+2, x+w-2, y+h-2)).save(titre+"_grille.png")
128         # lignes de capture d'écran
129
130 init_mat()
131 #lancer()
132 #avec lancer: l'interface graphique n'est affichée qu'à la fin
133 #utilisation de threading: permet une execution en temps reel
134 t = threading.Thread(target=lancer)

```

```
134 t.daemon = True
135 t.start()
136
137
138 Milieu.gui()
139
140 import matplotlib.pyplot as plt
141 plt.close()
142 plt.plot(range(cpteur), listescores)
143 titre= "matrice "+str(Milieu.x)+"*"+str(Milieu.y)+" gamma:"+str(gamma)+ " alpha:"
144 + str(alpha) +"(récompense immediate:mur)"
145 plt.title(titre)
146 plt.xlabel('Nombre de générations ( matrice '+str(Milieu.x)+'*'+str(Milieu.y)+' )')
147 plt.ylabel('Score')
148 plt.savefig(titre+'.png')
149 plt.show()
150 marshal.dump(Q, open("Q.txt", 'wb'))
151
152
153
154
155
156
157
```