

```

1 ##### Ceci est une variante du fichier Milieu sans interface graphique (GUI =
  Graphic User Interface)
2
3 import random
4 import marshal , copy
5
6
7 taille = 10 #taille graphique, ici inutile
8 (x, y) = (20, 20) #taille de la grille
9 actions = ["up", "down", "left", "right"]
10
11 coordonnees = (0, y-1) #l'agent commence tout en bas à gauche
12 score = 0
13 restart = False
14 walk_reward = -.05
15 obstaclage=.15
16
17 poubelle=[]
18 coffre={}
19 objets={}
20
21 #position des murs
22
23
24 def ecriture_obstacles(chemin,x,y,obstaclage): #permet d'enregistrer la liste
  d'obstacles dans un fichier pour une réutilisation future
25     obstacles= [(random.randint(0,x-1),random.randint(0,y-1)) for mur in
  range(int(x*y*obstaclage))]
26     fichier=open(chemin,"wb")
27     marshal.dump(obstacles,fichier) #marshal.dump sert à enregistrer la variable
  dans un fichier
28     fichier.close()
29
30 def lecture (chemin): #permet de retourner une variable contenue dans un fichier
31
32     fichier=open(chemin,"rb")
33
34     variable= marshal.load(fichier)
35     return(variable)
36
37
38 #ecriture_obstacles("stockage.txt",x,y,obstaclage)
39 obstacles= copy.copy(lecture("stockage.txt"))
40
41 #cases bonus/malus
42 special = [(4, 1, "red", -1),(x-1,0 , "yellow", 1)]
43
44 """bonus=10
45 for k in range (3):
46     special.append((random.randint(0,x-1),random.randint(0,y-1),"blue",bonus))
47 """
48
49 print(special)
50
51 def mouvement(dx, dy):
52     global coordonnees, x, y, score, walk_reward, restart , coffre
53     if restart == True:
54         reinitialisation()
55         new_x = coordonnees[0] + dx
56         new_y = coordonnees[1] + dy
57         score += walk_reward
58
59         if (new_x >= 0) and (new_x < x) and (new_y >= 0) and (new_y < y) and not
  ((new_x, new_y) in obstacles): #on vérifie que le mouvement est possible
60             coordonnees = (new_x , new_y)
61         else:
62             score -= .1 #sinon on décrémente tout de meme le score de l'agent pour
  avoir tenté une action inutile
63

```

```
64     for (i, j, c, w) in special:
65         if new_x == i and new_y == j:
66             score -= walk_reward
67             score += w
68             if c == "yellow" or c == "red":
69                 #for bonus in coffre:
70                     #score+=10*int(coffre[bonus])
71                 restart = True
72                 #print("score=", score)
73
74             elif c == "blue":
75                 coffre[i, j] = True
76                 obstacles.append((i, j))
77             else:
78                 poubelle.append((i, j, c, w))
79                 obstacles.append(i, j)
80                 special.remove((i, j, c, w))
81         return
82
83 def reinitialisation(liste):
84     global coordonnees, score, restart, coffre
85     coordonnees = (0, y-1)
86     liste.append(score)
87     score = 0
88     #####
89     coffre = {}
90     #####
91     restart = False
92     for tupls in poubelle:
93         special.append(tupls)
94         poubelle.remove(tupls)
95
96 def etat_reinit():
97     return restart
98
99
100
```