

Projet Sokoban – Trophées de la NSI

Cahier des charges

L'objectif du projet est obtenir une application qui utilise :

- Une fenêtre.
- Un canevas.
- Des listes.
- Créer des graphiques, du texte, des composants graphiques.
- Des fonctions.
- Des événements.

Le travail **doit se faire par groupe de 3 à 5 personnes**.

Cette application va se faire en plusieurs étapes

Étape 1 : Il s'agira de recopier l'ensemble des codes fournis. C'est une démarche importante qui nécessite de coder sans erreurs, en comprenant ce que vous écrivez.

Étape 2 : Il s'agira de faire une version personnelle de l'application qui devra être une création numérique originale (→ Trophées de la NSI 2024).

```
1  from tkinter import *
2
3  # Création de la fenêtre principale
4  Mafenetre = Tk()
5  Mafenetre.title('Sokoban')
6
7  #variables globales
8  numero_niveau=0
9  fini=False
10
11  # Création du plateau
12  #plateau vide au départ
13  plateau = []
14  for i in range(12):
15      plateau.append([])
16      for j in range(16):
17          plateau[i].append([])
18          for k in range(4):
19              (plateau[i][j]).append(0)
```

```
20
21  #niveau 1
22  #murs
23  for i in range(12):
24      plateau[i][0][0]=1
25      plateau[i][1][0]=1
26      plateau[i][14][0]=1
27      plateau[i][15][0]=1
28
29  for j in range(16):
30      plateau[0][j][0]=1
31      plateau[1][j][0]=1
32      plateau[10][j][0]=1
33      plateau[11][j][0]=1
34
35  #joueur
36  plateau[3][3][1]=1
37  #caisses
38  plateau[5][4][2]=1
39  plateau[5][6][2]=1
40  #interrupteurs
41  plateau[3][8][3]=1
42  plateau[5][5][3]=1
43
```

```

44 #Fonctions, appelées au bon moment, pour les niveaux suivants
45 def genere_niveau_2():
46     #on efface le plateau
47     for i in range(2,10):
48         for j in range(2,14):
49             plateau[i][j][0]=0
50             plateau[i][j][1]=0
51             plateau[i][j][2]=0
52             plateau[i][j][3]=0
53     #on crée un nouveau plateau
54     #murs
55     plateau[8][8][0]=1
56     plateau[7][3][0]=1
57     #joueur
58     plateau[9][9][1]=1
59     #caisses
60     plateau[5][4][2]=1
61     plateau[5][6][2]=1
62     #interrupteurs
63     plateau[3][8][3]=1
64     plateau[5][5][3]=1
65
66 def genere_niveau_3():
67     for i in range(2,10):
68         for j in range(2,14):
69             plateau[i][j][0]=0
70             plateau[i][j][1]=0
71             plateau[i][j][2]=0
72             plateau[i][j][3]=0
73
74 #Fonction testant si un niveau est fini
75 def test_victoire():
76     for i in range(12):
77         for j in range(16):
78             if plateau[i][j][3]==1 and plateau[i][j][2]==0:
79                 return False
80             #s'il y a au moins un interrupteur sans caisse on n'a pas fini
81     return True
82
83 # Création d'un widget Canvas (zone graphique)
84 Largeur = 800
85 Hauteur = 600
86 Canevas = Canvas(Mafenetre, width = Largeur, height =Hauteur, bg ='grey')

```

```

87  #Titre
88  Canvas.create_text(400,100,fill="darkblue",font="Times 60 italic bold",
89  |               |               |               |
90  |               |               |               |
91  Canvas.create_text(400,250,fill="darkblue",font="Times 20",
92  |               |               |               |
93  |               |               |               |
94  Canvas.create_text(400,300,fill="darkblue",font="Times 20",
95  |               |               |               |
96  |               |               |               |
97  def affiche_plateau_canvas():
98  |   for i in range(12):
99  |       Canvas.create_line(0,50*i,800,50*i,width=0.5)
100 |   for j in range(16):
101 |       Canvas.create_line(50*j,0,50*j,600,width=0.5)
102 |   for i in range(12):
103 |       print()
104 |       for j in range(16):
105 |           if (plateau[i][j][0]==1):
106 |               #affichage mur
107 |               Canvas.create_rectangle(50*j,50*i,50*j+50,50*i+50,fill='blue')
108 |
109 |           elif (plateau[i][j][1]==1):
110 |               #affichage joueur
111 |               Canvas.create_oval(50*j,50*i,50*j+50,50*i+50,fill='yellow')
112 |           elif (plateau[i][j][2]==1):
113 |               #affichage caisse
114 |               Canvas.create_rectangle(50*j,50*i,50*j+50,50*i+50,fill='red')
115 |           elif (plateau[i][j][3]==1):
116 |               #affichage interrupteur
117 |               Canvas.create_oval(50*j+10,50*i+10,50*j+40,50*i+40,fill='red')
118

```

```

119 def Clavier(event):
120     global numero_niveau
121     global fini
122     """ Gestion de l'événement Appui sur une touche du clavier """
123     if fini==False: #quand le jeu est fini on ne peut plus se déplacer
124         #on efface le canevas
125         Canevas.delete("all")
126         mvt_poss=True
127         touche = event.keysym
128         for i in range(12):
129             for j in range(16):
130                 if (plateau[i][j][1]==1 and mvt_poss ==True ):
131                     # déplacement vers le haut
132                     #possible si pas de mur dans la case destination ni de caisse suivie d'une caisse ou d'un mur
133                     if touche == 'Up' and plateau[i-1][j][0]!=1 and not(plateau [i-1][j][2]==1 and (plateau[i-2][j][2]==1 or plateau[i-2][j][0]==1)):
134                         if plateau[i-1][j][2]==1:
135                             plateau[i-2][j][2]=1
136                             plateau[i-1][j][2]=0
137                         plateau[i][j][1]=0
138                         plateau[i-1][j][1]=1
139                     elif touche == 'Left' and plateau[i][j-1][0]!=1 and not(plateau [i][j-1][2]==1 and (plateau[i][j-2][2]==1 or plateau[i][j-2][0]==1)):
140                         if plateau[i][j-1][2]==1:
141                             plateau[i][j-2][2]=1
142                             plateau[i][j-1][2]=0
143                         plateau[i][j][1]=0
144                         plateau[i][j-1][1]=1
145                     elif touche == 'Right' and plateau[i][j+1][0]!=1 and not(plateau [i][j+1][2]==1 and (plateau[i][j+2][2]==1 or plateau[i][j+2][0]==1)):
146                         if plateau[i][j+1][2]==1:
147                             plateau[i][j+2][2]=1
148                             plateau[i][j+1][2]=0
149                         plateau[i][j][1]=0
150                         plateau[i][j+1][1]=1
151                     elif touche == 'Down' and plateau[i+1][j][0]!=1 and not(plateau [i+1][j][2]==1 and (plateau[i+2][j][2]==1 or plateau[i+2][j][0]==1)):
152                         if plateau[i+1][j][2]==1:
153                             plateau[i+2][j][2]=1
154                             plateau[i+1][j][2]=0
155                         plateau[i][j][1]=0
156                         plateau[i+1][j][1]=1
157                 mvt_poss=False #pour ne pas se déplacer de plusieurs cases à la fois
158     #le cas échéant on change de niveau :
159     if (test_victoire()==True):
160         numero_niveau=numero_niveau+1
161         if numero_niveau==1:
162             genere_niveau_2()
163         if numero_niveau==2:
164             genere_niveau_3()
165         Canevas.create_text(400,300,fill="darkblue",font="Times 60 italic bold",text="BRAVO !!!")
166         fini=True #bloque les commandes
167     #on raffiche le canevas
168     affiche_plateau_canvas()

```

```

169
170 Canevas.focus_set()
171 Canevas.bind('<Key>',Clavier)
172 Canevas.grid(row=0,column=0)
173
174 # Création d'un widget Button (bouton Quitter)
175 BoutonQuitter=Button(Mafenetre, text ='Quitter', command = Mafenetre.destroy)
176 BoutonQuitter.grid(row=1,column=0)
177
178 #boucle principale
179 Mafenetre.mainloop()

```

Des idées (liste ni impérative, ni exhaustive évidemment) :

- Afficher un compteur de déplacement (*)
- Afficher un score (*)
- Permettre de recommencer un niveau (*)
- Agrandir le plateau de jeu (*)
- Faire des niveaux variés, de difficulté progressive. (**)
- Faire des niveaux avec des étages et des escaliers pour faire tomber les caisses. (***)
- Proposer 2 modes de jeux : 1 mode qui limite le nombre de déplacements, l'autre sans limitation. (**)
- Proposer un éditeur qui permet de créer soit même ses propres niveaux. (****)
- Modifier (= améliorer) l'aspect graphique (**)
- Ajouter une musique (**)
- Permettre de choisir les touches (*)
- Remplacer le carré pousseur par un personnage (**)

Le livrable

Vous rendrez votre travail sous forme d'un dossier zippé nommé **Nom1_Nom2.zip** via la **messagerie** ou le **casier** de l'ENT à Louis-Marie Madrias.

Le dossier zippé doit contenir :

- le code source commenté (**10 points**),
- une vidéo de 2 min max de présentation de l'équipe et du projet (**5 points**),
- une documentation de 4 pages (**5 points**).

Exemples de niveaux supplémentaires :

