

# Sokoban 2

---

## Le jeu



Nous avons vu dans la première partie comment lire le fichier décrivant les niveaux et comment construire la scène du jeu à partir de cette description.

Il s'agit maintenant de programmer la dynamique du jeu.

Nous déplaçons le personnage à l'aide des flèches du clavier. Si devant le personnage, il y a une étoile, alors le personnage pousse l'étoile.

Ici le personnage ne peut pas se déplacer vers la gauche, car il y a deux étoiles qui se suivent.

Le but du jeu est donc de déplacer les étoiles de façon à ce que chaque étoile aboutisse sur une case marquée d'une étoile rose : objectif.

Les étoiles arrivant dans un coin, ne peuvent plus bouger.

Le personnage peut marcher sur toutes les zones grises même si elles ne sont pas complètes. Il peut également marcher sur les cases marquées d'une étoile et il peut déplacer une étoile qui était déjà positionnée sur un objectif : case marquée d'une étoile rose.

Voici comment résoudre ce niveau :

Partant de sa position de départ, le personnage suit les points rouges et pousse la deuxième étoile vers le haut.



Le personnage revient ensuite sur ses pas (points verts) pour pousser la première étoile à la position montrée sur la deuxième capture d'écran en suivant les points rouges.



Le personnage va ensuite chercher l'étoile en haut et la pousse sur le côté à droite.

Il va ensuite chercher l'étoile à gauche et la monte afin de dégager la case où elle se trouve actuellement.



Le personnage va chercher l'étoile à gauche, la pousse d'un cran vers la gauche, puis la fait descendre sur la ligne du dessous.



Il ne reste plus qu'à la pousser vers la droite pour la mettre sur le premier objectif.  
Ensuite il suffit de faire redescendre l'étoile en haut à gauche vers le deuxième objectif.

## Les déplacements du personnage

Les flèches permettent de déplacer le personnage sur les différentes cases.

A chaque appui sur une de ces touches, il faudra :

- Vérifier que le déplacement est possible
- Enregistrer la nouvelle position du personnage
- Enregistrer s'il y a lieu, la nouvelle position d'une étoile, si dans son mouvement le personnage l'a poussée.

def faireMouvement(carte, jeuEtat, directionJoueur):

```
    """Etant donné une carte, un état du jeu cette fonction regarde
    si le joueur peut faire le mouvement demandé. Si oui, elle change la
    position du joueur ainsi que d'une éventuelle étoile poussée.
    Sinon, elle ne fait rien.
    Retourne True si le joueur bouge, sinon False."""
    # Position actuelle du joueur
    joueurx, jouey = jeuEtat['joueur']
    #positions actuelles des étoiles
    etoiles = jeuEtat['etoiles']
    # La direction vers laquelle le joueur veut aller est indiquée
    #par les valeurs : HAUT, BAS, DROITE, GAUCHE placées dans
    directionJoueur
    if directionJoueur == 'HAUT':
        xOffset = 0
        yOffset = -1
    elif directionJoueur == 'DROITE':
        xOffset = 1
        yOffset = 0
    elif directionJoueur == 'BAS':
        xOffset = 0
        yOffset = 1
    elif directionJoueur == 'GAUCHE':
        xOffset = -1
```

```
yOffset = 0
```

```
# Le joueur peut-il se déplacer sans entrer dans un mur?
```

```
if estMur(carte, joueurs + xOffset, jouey + yOffset):
```

```
    return False
```

```
else:
```

```
#le joueur pousse t'il une étoile?
```

```
if (joueurs + xOffset, jouey + yOffset) in etoiles:
```

```
# Vérifier que l'étoile peut être poussée.
```

```
if not estBloque(carte, jeuEtat, joueurs + (xOffset*2), jouey + (yOffset*2)):
```

```
# Déplacer l'étoile.
```

```
    ind = etoiles.index((joueurs + xOffset, jouey + yOffset))
```

```
    etoiles[ind] = (etoiles[ind][0] + xOffset, etoiles[ind][1] + yOffset)
```

```
else:
```

```
    return False
```

```
# Déplacer le joueur.
```

```
jeuEtat['joueur'] = (joueurs + xOffset, jouey + yOffset)
```

```
return True
```

Vérifier si une étoile est bloquée ou non

```
def estBloque(carte, jeuEtat, x, y):
```

```
    """Renvoie True si la position (x,y) sur la carte correspond à un mur, ou  
    si cette position est en dehors de la carte, ou si cette position est  
    celle d'une étoile'"""
```

```
    """
```

```
#position sur un mur
```

```
if estMur(carte, x, y):
```

```
    return True
```

```
#position en dehors de la carte
```

```
elif x < 0 or x >= len(carte) or y < 0 or y >= len(carte[x]):
```

```
    return True
```

### #position sur une étoile

```
elif (x, y) in jeuEtat['etoiles']:  
    return True  
return False
```

## Vérifier si le niveau est terminé

Il suffit de vérifier pour chacun des objectifs du niveau, si une étoile a les mêmes coordonnées.

```
def testNiveauTermine(niveau, jeuEtat):  
    """Renvoie vrai si tous les objectifs sont recouverts par une étoile."""  
    for objectif in niveau['objectifs']:  
        if objectif not in jeuEtat['etoiles']:  
            # les coordonnées de l'objectif ne correspondent à aucune des  
            #coordonnées des étoiles.  
            return False  
    return True
```

## Programme d'essai

Par rapport à la première version Sokoban1,

- On prend maintenant en compte l'appui sur les touches de direction,
- On essaie d'effectuer le mouvement demandé
- Si ce mouvement a été possible, on redessine la scène car des images ont changé de place.
- On vérifie si le niveau est terminé c'est-à-dire si tous les objectifs ont été atteints.

Vous vérifierez que lorsque le niveau est terminé, il n'est plus possible de déplacer le personnage.

Vous pouvez aussi tester le programme en utilisant le fichier Microban1.txt.