

## SAÉ S1.01 : implémentation d'un besoin client

### Le jeu du Sokoban : consignes pour la version 2

#### Présentation générale

Vous êtes maintenant chargés de compléter votre programme du jeu Sokoban, c'est-à-dire d'ajouter à votre première version ces nouvelles fonctionnalités :

1. proposer la possibilité de zoomer (touche '+' ) ou de dézoomer (touche '-' ) ,
2. mémoriser la suite des déplacements effectués,
3. proposer d'annuler le dernier déplacement (touche 'u' pour undo),
4. proposer de sauvegarder la suite des déplacements effectués quand la partie est terminée par abandon ou par victoire.

Les nouvelles touches gérées ( '+' , '-' et 'u' ) seront mentionnées dans l'entête affiché au-dessus du plateau.

#### Détail des nouvelles fonctionnalités

##### 1. Possibilité de zoom

L'utilisateur pourra agrandir ou rétrécir la taille du plateau affiché à l'écran. L'échelle (ou niveau de zoom) ira de 1 à 3 ; au début du jeu, l'échelle sera de 1 et sera incrémentée à chaque appui sur la touche '+' . À l'inverse elle sera décrémentée à chaque appui sur la touche '-' . Aggrandir le plateau c'est aggrandir chaque symbole qui le constitue dans les deux dimensions, en répétant le caractère autant que nécessaire.

Par exemple, à l'échelle 1 Sokoban est représenté par le caractère @. À l'échelle 2 il sera représenté par :

@@  
@@

et à l'échelle 3, il sera représenté comme ceci

@@@  
@@@  
@@@

Autre exemple, avec un extrait de plateau :

échelle 1 :

```
#  #  
#$
```

échelle 2 :

```
##  ##  
##  ##  
####  
####
```

échelle 3 :

```
###  ###  
###  ###  
###  ###  
####  
####  
####
```

## 2. Mémoriser les déplacements

En cours de partie, tout déplacement de Sokoban, poussant une caisse ou non, sera mémorisé sous la forme d'un caractère parmi les suivants :

g : pour un déplacement de Sokoban seul vers la gauche,  
h : pour un déplacement de Sokoban seul vers le haut,  
b : pour un déplacement de Sokoban seul vers le bas,  
d : pour un déplacement de Sokoban seul vers la droite,  
G : pour un déplacement de Sokoban et d'une caisse vers la gauche,  
H : pour un déplacement de Sokoban et d'une caisse vers le haut,  
B : pour un déplacement de Sokoban et d'une caisse vers le bas,  
D : pour un déplacement de Sokoban et d'une caisse vers la droite.

## 3. Annuler le dernier déplacement

En cours de jeu, l'appui sur la touche 'u' annulera le dernier déplacement effectué (donc le dernier mémorisé). Le plateau reviendra à son état avant ce déplacement.

## 4. Enregistrer les déplacements

En fin de partie, qu'elle soit gagnée ou abandonnée, le programme proposera d'enregistrer les déplacements dans un fichier. Si l'utilisateur confirme, il devra saisir le nom d'un fichier .dep dans lequel la liste des déplacements effectués au cours de la partie seront sauvegardés.

## Détails techniques

### Mémorisation des déplacements

La mémorisation des déplacements se fera dans un tableau 1D de caractères. Vous créerez un type `t_tabDeplacement` pour ce tableau de déplacements.

Il peut y avoir plusieurs centaines de déplacements dans une partie.

### Annulation

L'annulation du dernier déplacement devra faire l'objet d'une procédure ou d'une fonction.

### Sauvegarde des déplacements

La procédure d'écriture dans un fichier de déplacements est fournie sur Moodle, [vous l'ajouterez à votre code source](#) :

```
void enregistrerDeplacements(t_tabDeplacement t, int nb, char fic[]);
```

Cette procédure enregistre dans le fichier `fic` les `nb` caractères du tableau `t`.

**En aucun cas vous ne devez modifier cette procédure.**

## Important

Vous produirez un code source **bien structuré**. Pour cela, vous pourrez créer les fonctions et procédures que vous jugerez nécessaires.

Enfin, une attention particulière sera portée au respect des **conventions de codage**, et notamment :

- pas de variables courtes (une lettre, sauf `i`, `j`, `x`, `y`),
- que du camelCase pour les noms de variable,
- que du snake\_case en majuscule pour les constantes,
- que du snake\_case pour les fonctions,
- espace partout sauf après une parenthèse ouvrante et avant une parenthèse fermante,
- maximum de 80 caractère par ligne,
- pas plus de 5 indentations,
- pas plus de 60 lignes par procédure ou fonction.