

// Nom : Slimani  
// Prénom : Arezki  
// n° étudiant : 21502933

# Rapport De Projet

## Description des structures de données utilisées :

1) En premier lieu, nous trouverons la première structure squelette du jeu <SOKOBAN> .

La structure SOKOBAN est composé de plusieurs variable , ainsi que deux autres structures:

-> Variable niveau : qui enregistre le numéro du niveau actuel.

-> Variable coupsjouer : qui enregistre le nombre de coups jouer par le joueur lors de la résolution du niveau.

-> Variable modifiable : variable qui gère le passage du mode création au mode jouer création (jouer a l'envers).

La structure PERSO sauvegarde les coordonner du personnage :

La structure PERSO personnage est composé de :

-> Variable personnage.x : coordonner des abscisses du personnage.

-> Variable personnage.y: coordonner des ordonnées du personnage.

La structure une\_case sauvegarde la valeur d'une case du jeu, ainsi que le nombre de fois qu'on clic sur la case en mode création.

La structure une\_case est composé de :

-> Variable val : enregistre la valeur de la case.

-> Variable nbrcllic : qui enregistre le nombre de fois qu'on clic sur la case pour changer la valeur de la case en mode création.

Dans la structure SOKOBAN on introduit la structure une\_case sous forme d'un tableau a deux dimensions qui couvre toute la map de jeu.

2) En deuxième lieu, nous trouverons la structure PileElement ou \*Pile qui est une Pile comme son nom l'indique .

La Structure Pile est une Pile d'éléments de type SOKOBAN qui constituerons par la suite l'historique du jeu.

**Mode de fonctionnement de l'historique :** L'historique est constituer de deux Piles nommées P et P2, la Pile principale est P car elle enregistre tout les mouvement du personnage, la pile P2 est utiliser pour stocker les éléments de P lors du dépile , on dépile P lors d'un UNDO et on empile sur P lors d'un REDO ( on empile les valeurs se trouvant dans P2).

## Description des algorithmes un peu compliqués :

1) Fichier : creation.c : Fonction "int verifier\_murs\_fermer(SOKOBAN S, int x, int y);"  
Cette fonction permet d'explorer toute les cases du jeu en faisant un appelle récursive sur les quatre case a coté de chaque case ,on fait cela pour vérifier si les murs sont bien fermer. L'exploration commence a partir de la case ou se trouve le personnage.

2) Fichier : creation.c et action.c : Les fonctions "modifier\_map\_trois\_position" ou "modifier\_map\_deux\_position" servent a changer le contenu des case a coté du personnage pour faire bouger celui si selon des conditions précis.

### **Améliorations possibles :**

La fonction d'écriture "void ecrire\_niveau(SOKOBAN S)" est améliorable car elle ne permet pas d'écrire en mode ajout , c'est a dire on ne peut pas crée plusieurs niveau , on peut seulement en crée un dans un fichier et celui si sera écraser s'il on veut en crée un autre.

### **Critiques sur le code :**

Quelques fonctions comportent des répétitions d'affectations sur des variables qu'on pourrait éviter.

La fonction "int verifier\_murs\_fermer(SOKOBAN S, int x, int y);" est non optimal pour les grande maps de jeu contenant peut de murs car elle prend énormément de temps a explorer toute les combinaisons de déplacement possible pour déterminer si les murs sont fermer. celle si fonctionne bien pour les maps de taille moyenne contenant beaucoup de murs. Le problème ne se pose pas pour les petites maps.

### **Eventuelles références à des livres ou site web :**

1) Une chaine Youtube "FormationVidéo" : une formation de 21 épisode qui vous enseigne toutes les facettes de la programmation en c.

2) le site Openclassroom : vous connaissez sans doute.

**A bientôt !**