

Rapport Projet : SOKOBAN

Julien JACQUET 21400579

8 janvier 2017

Note : L'intégralité des fichiers nécessaires a la compilation de ce pdf se trouvent dans le dossier Rapport.

Note : La partie creation de niveau est non fonctionnelle.

Note : En cas de tests avec valgrind il faut entrer :
valgrind make test
Sinon erreur : valgrind : mmap(0x315000, 1600327680) failed in UME with error 22 (Invalid argument).
(cependant cette methode entraine une perte de 10 octets a cause de la fonction make)
L'erreur semble venir de valgrind.

1 Explications des structures de données :

```
1 struct une_case{
2     int val;
3 };
4
5 struct sokoban{
6     int mode;
7     CASE une_case[100][100]; //ne devrais jamais dépasser
8     100*100 cases
9     int niveau;
10 };
```

La structure *sokoban* est composée d'un entier mode qui sera uniquement utilisé pour l'affichage. La partie importante de cette structure est la *CASE*, c'est un double tableau de 100 par 100, la forme de tableau semble convenir puisqu'il est aisé de représenter mentalement le sokoban sous forme matricielle. En dernier element de cette structure on a un entier niveau qui sert a retenir le niveau auquel on se trouve. La structure *une_case* n'est pas vraiment nécessaire mais elle permet de clarifier la construction de la structure *sokoban*. Une chose a noter cependant : La structure *sokoban* ne stockera jamais autre chose que des entiers, alors que le fichier *sasquatch1.xsb* contient des *char*. La raison a cela est une grande simplification de la lecture du fichier, la lecture et le stockage des *char* pouvant s'avérer laborieux, uniquement le code ascii des caracteres est retenu.

```
1 struct action{
2     int mode;
3     int fleche;
4 };
5
6 struct coord{
7     int x;
8     int y;
9 };
```

La structure *action* contient deux entiers un entier *mode* qui définira les actions a effectuer (UNDO,REDO,QUIT,SUIVANT,etc) et un entier *fleche* qui sert a renvoyer la valeur de la flèche quand l'utilisateur appuie dessus.

La structure *coord* contient deux entiers qui sont en fait les coordonnées de l'emplacement du joueur dans le *sokoban*.

```
1 struct level{
2     int largeur_max_level;
3     int hauteur_max_level;
4     int debut; //debut S.niveau
5     int fin;   //fin S.niveau
6 };
```

La structure *level* a deux usages, le premier est de determiner la plus grande largeur et hauteur possible d'un niveau contenu dans un fichier. Ces donnees servent dans la fonction *preprocess()* et servent a determiner la taille de la fenetre graphique et les données qui en dépendent. Le second usage est de retenir la ligne de debut d'un niveau ainsi que la ligne de fin.

```
1 struct stackRedo{ //pour redo
2     SOKOBAN STredo[STACK_MAX];
3     int top;
4 }stRedo;
5
6 struct stackUndo{ //pour undo
7     SOKOBAN STprec[STACK_MAX];
8     int top;
9 }stUndo;
```

Les structures *stackRedo* et *stackUndo* sont identiques, ce sont de piles, elles contiennent un tableau de type *sokoban* pour stocker l'état du sokoban, et un *int top* qui est necessaire pour créer une pile.

```
1 struct STRUCTURE_CREATION_NIVEAU{
2     SOKOBAN C;
3     int mode_creation;
4 };
```

Bien que le module de creation ne soit pas fonctionnel, la strucutre de creation de niveau existe, elle contient un *sokoban* et un mode qui devait correspondre au type de caractere(mur , caisse), choisi par l'utilisateur, à ajouter dans le fichier.

2 Critiques :

Certaines fonctions ont demandées un travail extremement chronophage, principalement la lecture prenant plusieurs jours pour ecrire une 150 lignes alors que d'autres (la partie des mouvements d'*action*), après 20 minutes de reflexion ont pris seulement une journée a écrire (400 lignes). Il est difficile d'évaluer a l'avance la difficulté que poseront certains modules du projet. D'autre part

certaines fonctions du projet relevent du bricolage mais elles fonctionnent. Le module de creation ne fonctionne pas, quelques jours supplémentaires auraient étés nécessaires malgré un mise au travail debut decembre.

3 Références :

Pour la lecture : <http://stackoverflow.com/questions/4179671/read-in-text-file-1-character-at-a-time-using-c>

Pour les piles : l'intégralité des resultats de la premiere page google de la recherche "stack in C".

Pour les converions ascii : <https://www.branah.com/ascii-converter>