

# RAPPORT DE PROJET:SOKOBAN

SOKOBAN

PLAY

EXIT

PRÉSENTÉ PAR:MOUNGUENGUI YANNICK

# PLAN

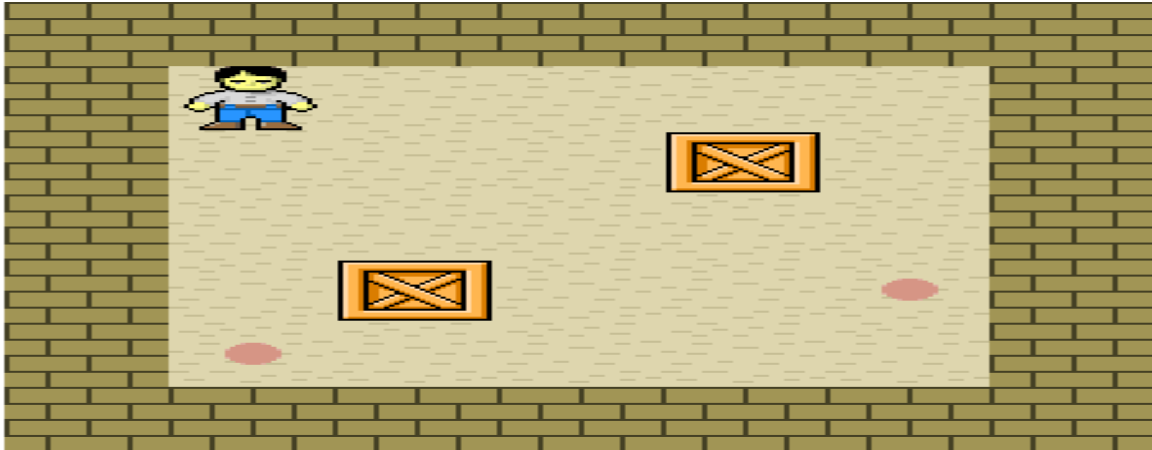
I/ PRÉSENTATION

II/ CONCEPTION

III/ DIFFICULTES

IV/ AMÉLIORATION

# I/ PRÉSENTATION

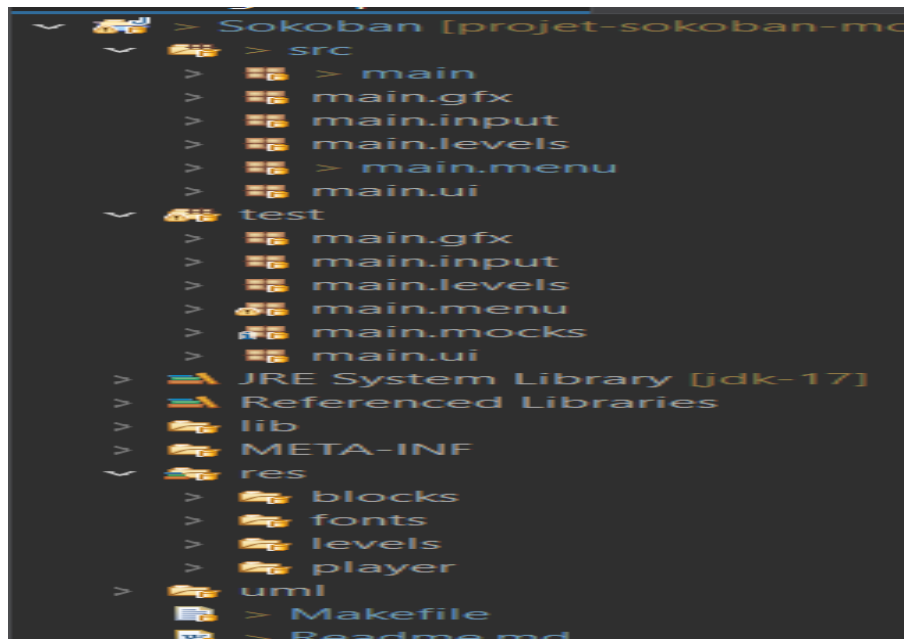


Sokoban est un jeu vidéo de réflexion inventé au Japon. Le jeu original a été écrit par Hiroyuki Imabayashi et comportait 50 niveaux. Il remporte en 1980 un concours de jeu vidéo pour ordinateur grâce à ce jeu.

Le but du jeu est simple, le joueur doit ranger des caisses sur des cases cibles. Il peut se déplacer dans les quatre directions, et pousser (mais pas tirer) une seule caisse à la fois. Une fois toutes les caisses rangées (c'est parfois un vrai casse-tête), le niveau est réussi et le joueur passe au niveau suivant, plus difficile en général. L'idéal est de réussir avec le moins de coups possibles (déplacements et poussées).

Le choix de ce projet est parce que j'aime l'univers du jeu vidéo ainsi j'ai décidé de faire ce jeu comme projet. Aussi cela a été due au fait que les idées précédentes qui étaient de faire du Fire emblem ou Dragon quest s'avèrent compliquées en 3 mois de conception et un peu plus complexe à faire. Ainsi je me suis penché sur un jeu simple à faire et que personne n'aurait pensé. Concernant le choix de langage, java semblait le plus aisé pour moi car je comprends mieux à travers ce langage et qu'il offre de meilleurs bibliothèques que le langage python en terme d'aspects visuels pour un jeu.

## II/ CONCEPTION



Concernant la conception du projet, il a été préférable de le faire en langage de programmation java pour l'utilisation de bibliothèques qui ajoutent de meilleurs graphismes. Le projet a été structuré en différents dossiers.

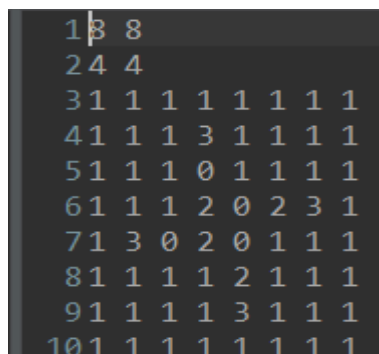
Un dossier src pour les classes sources réparties en différents paquetages qui sont: paquetage main pour l'exécution du jeu, paquetage gfx pour l'ensemble des graphismes, des dessins, des images et textes. Le paquetage input quant à lui sert au déplacement de la souris sur la fenêtre de jeu et le déplacement au clavier, le paquetage ui sert à l'interface de l'utilisateur en permettant l'ajout de boutons sur la fenêtre de jeu et d'utiliser les clics du clavier ou souris. Les deux derniers paquetages sont les paquetages levels et menu. Le premier paquetage cité permet la création d'un niveau du jeu, le déplacement du joueur et sa résolution quand au dernier lui sert à la création de la fenêtre de jeu, le lancement du jeu via la classe Window et à la création des différentes étapes du jeu(menu principal, menu de sélection de niveau, chargement du niveau et déroulement d'un niveau).

Le deuxième dossier est le dossier test, permettant de tester les différentes méthodes possibles des classes du projet car il n'est pas nécessaire de tester toutes les classes du projet.

Le troisième dossier utile est le dossier "res" à ajouter au classpath du projet, permettant d'ajouter des images et des fichiers de niveau à la construction du jeu.

Il y a aussi la présence d'un Makefile permettant une exécution plus rapide des commandes pour la javadoc, compilation des classes, exécution des tests ,création de jar et exécution du jar pour le jeu.

Dans les différents paquetages on utilise une bibliothèque java qui est la bibliothèque AWT. AWT de son vrai nom Abstract Window Toolkit est une bibliothèque graphique pour Java, faisant partie de Java Foundation Classes (JFC). Cette bibliothèque a été introduite dès les premières versions de Java depuis Java 2. Elle permet notamment la création d'objets graphiques préconçus tels que des boutons, des sliders, des menus déroulants. Étant une bibliothèque beaucoup utilisée, elle apporte son plus dans la construction de ce projet.



```
10 8
2 4 4
3 1 1 1 1 1 1 1
4 1 1 1 3 1 1 1
5 1 1 1 0 1 1 1
6 1 1 1 2 0 2 3 1
7 1 3 0 2 0 1 1 1
8 1 1 1 1 2 1 1 1
9 1 1 1 1 3 1 1 1
10 1 1 1 1 1 1 1 1
```

Pour la création des niveaux on utilise un fichier texte (comme l'image ci-dessus) qu'on convertit en chaîne pour pouvoir lire et utiliser le fichier correctement. Chaque chiffre a une interprétation ainsi que les lignes. À la ligne 1, on a le nombre de ligne et de colonne du plateau. La ligne 2 représente les coordonnées où sera positionné le joueur au lancement du niveau. À partir de la 3e ligne jusqu'à la dernière cela représente le plateau du niveau où le chiffre 1 représente les murs, le chiffre 3 la cible où doit se positionner la boîte, le chiffre 0 les emplacements libres et le chiffre 2 les blocs à déplacer.

Enfin une autre manière de conception dans ce projet est l'utilisation de thread dans la Classe Window du paquetage menu. Un thread est une unité d'exécution faisant partie d'un programme. Cette unité fonctionne de façon autonome et parallèlement à d'autres threads. Le principal avantage des threads est de pouvoir répartir différents traitements d'un même programme en plusieurs unités distinctes pour permettre leurs exécutions "simultanées". Ainsi dans la classe Window cela permet de répartir le traitement de la création d'un niveau, son lancement, sa résolution et le passage à un niveau supérieur pour tout lancer simultanément dans une fenêtre de jeu.

### III/ DIFFICULTES

Il y a eu différentes difficultés rencontrées dans le cheminement de ce projet. Une des difficultés est l'accessibilité des ressources possibles à ajouter au projet. Il a été difficile de les ajouter car les différentes fonctions utilisées avaient du mal à trouver les fichiers ce qui entraîna par exemple le fait que la fenêtre du jeu ne s'affichait pas ou juste une fenêtre vide. Ainsi il a fallu ajouter le dossier "res" au classpath du projet sous ide Eclipse pour l'exécuter correctement. Lors de la création du jar exécutable le même problème est survenu, pour y remédier, il y a eu création d'un dossier "META-INF" contenant un manifest, ce manifest a été utilisé à la création du jar pour charger correctement les différentes ressources, ainsi à l'exécution du jar le jeu se lance sans problème.

Une autre difficulté rencontrée est le fait de pouvoir tester des méthodes en utilisant les ressources attribuées dans le projet ou en comparant. Dans un terminal de commande en utilisant la commande pour exécuter les tests (via makefile possible), ces tests renvoyaient tous une erreur car les ressources ne sont pas trouvées. Il n'a pas eu de solution possible pour corriger cela via ligne de commande néanmoins comme alternative, sur un ide tel qu'Eclipse et ayant intégré le dossier "res" au classpath du projet, les tests fonctionnent tous.

Enfin, le fait d'utiliser certaines méthodes et attributs de la bibliothèque AWT de java, certains tests de classes n'ont pas pu être faits et il y a eu des problèmes lors de l'utilisation de certains attributs pour les classes telles que Window pour créer la fenêtre graphique.

## IV/ AMÉLIORATION

Pour conclure nous pouvons évoquer les différentes améliorations possibles à ajouter à ce projet.

D'abord, on peut ajouter un score qui se fait par un temps effectué à chaque niveau du jeu et avoir un score cumulé à la fin du jeu avec un classement actualisable.

Ensuite, on peut ajouter des obstacles au jeu qui feront en sorte que, dès que le joueur touche un avec une boîte celle-ci se brise et l'empêche d'accéder au niveau suivant, ainsi il devra mieux réfléchir pour résoudre le niveau.

En outre, on peut ajouter des images d'autres personnages pour faire en sorte que le joueur a le choix parmi 3 personnages pour le jeu.

Enfin, on peut ajouter une classe permettant d'utiliser une intelligence artificielle qui utilisera un algorithme de recherche pour effectuer les déplacements les plus optimaux pour résoudre chaque niveau.