

# Conception Logiciel

Projet SOKOBAN

**LEDOUX Eloi,**  
**TAUPIN Erwann,**  
**KITSOUKOU Manne Emile,**  
*L1 informatique, Groupe 2B,*

Université Caen-Normandie

19 avril 2021

# Table des matières

- 1 Travaux préliminaires
- 2 Structure du projet
- 3 Aspect graphique
  - Approche
  - Les couches
- 4 L'algorithme de recherche
- 5 Les fonctionnalités ajoutées
- 6 Conclusion

# Travaux préliminaires

Ils se sont articulés autour de trois(3) points :

- Les renseignements sur le projet - Qu'est-ce qu'un Sokoban ?
- La répartition des fichiers
- La répartition des tâches au sein du groupe

Les étapes pour aboutir à une version fonctionnelle en console :

- Création d'un fichier grille
- Lecture d'un niveau de sokoban (fichier .sok)
- Adaptation des symboles du programme

# Version jouable en Console

Implémentations des déplacements :

- Déplacements du personnages
- Déplacements des caisses
- Conditions de victoire

# Aspect graphique

Pour gérer l'aspect graphique nous nous appuyons sur une programmation événementielle à l'aide du module Pygame.



# Aspect graphique

Les événements utilisateurs :

- JOUER
- GO\_MENU
- RUN\_SOLUTION
- ECRAN\_FIN
- NIVEAUX
- AUTOMATIQUE

# Les couches

Qu'appelons nous par couche ?

Une couche est un objet possédant des données, des informations pouvant être manipulées et modifiées à travers deux(2) méthodes :

- **react\_to**
- **draw**

Les couches intermédiaires :

- Le menu
- La sélection des niveaux
- La couche de fin



# La couche JEU

La couche JEU permet de gérer la partie de jeu.



# La couche JEU

Pour afficher le plateau de jeu d'un niveau de SOKOBAN, il est nécessaire d'utiliser l'objet **Image**.

L'objet **Image** permet, à partir d'un tableau 2D (liste de liste ou niveau SOKOBAN), de dessiner ces éléments.

Cela s'effectue grâce à la méthode **dessine\_grille** ou **dessine\_level**.

# Fonctionnement des méthodes d'affichage d'un tableau 2D

*Début de la méthode :*

*itération sur les éléments du tableau 2D (par ligne et colonne ou par numéro de case) :*

*tester si le caractère est dans le dictionnaire des caractères prédéfinis*

*afficher le caractère sur l'écran à des coordonnées calculées selon sa position dans le tableau*

*Fin de la méthode*

## Recherche d'un solveur compatible

Comme nous n'avions plus beaucoup de temps à consacrer à la création d'un solveur, nous avons fait le choix de procéder, non pas à l'écriture d'un algorithme mais, à l'intégration d'un algorithme déjà existant.

Le solveur choisi a été écrit par *KnightofLuna* (lien GitHub du projet).

Ce solveur peut utiliser quatre(4) algorithmes différents : A\*, BFS, DFS, UCS.

Cependant, pour notre projet, nous avons choisi de ne conserver que l'algorithme A\*.

# L'algorithme de recherche

Comment fonctionne un algorithme de recherche ?

L'algorithme  $A^*$  est un algorithme de recherche de chemin dans un graphe, de plus c' est un algorithme itératif.

Le  $A^*$  donne rapidement une assez bonne solution, cependant cette solution n'est pas toujours optimale (cf. annexe C du rapport).

À chaque coup, il va essayer de se rapprocher de la destination par le chemin le plus direct possible.

# Intégration de l'algorithme

Modifications apportées :

- Remplacement des caractères utilisés par l'algorithme
- Ajout de la possibilité pour le personnage d'être sur un objectif
- Remplacement des lettres qui définissent les actions possibles

# Intégration de l'algorithme

Comment l'algorithme fonctionne dans le programme ?

La résolution s'effectue par changement de couche. Cette couche fonctionne selon ces points :

- Recherche des solutions lors de l'initialisation
- Lance la résolution dans la méthode **react\_to**
- Affiche le plateau grâce à la méthode **draw**

# Les fonctionnalités ajoutées

Les divers ajouts que nous avons pu implémenter :

- Le son
- Les ajouts textuelles
- Les fonctionnalités de "sauvetages" dans une partie :
  - Le Undo : retour en arrière
  - Le Reset : Réinitialiser la partie en cours



# Conclusion

Un travail formateur, tant pour la programmation que le travail en équipe.

Un travail qui reste néanmoins améliorable sur de nombreux points :

- Avec l'ajout d'un intégrateur de niveaux directement en jeux.
- Avec une possibilité de résolution par l'algorithme plus poussée et simultanée.
- Avec la possibilité de jouer à plusieurs en comparant ses meilleurs scores sauvegardés par le jeu.