

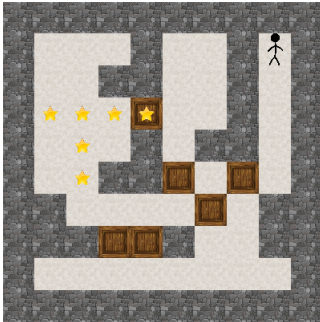
# Défier l'ordinateur au Sokoban

Aymeric BEAUCHAMP  
Dimitri CHAGNEUX  
Valentin LEBLOND  
Baptiste MORI

Avril 2018

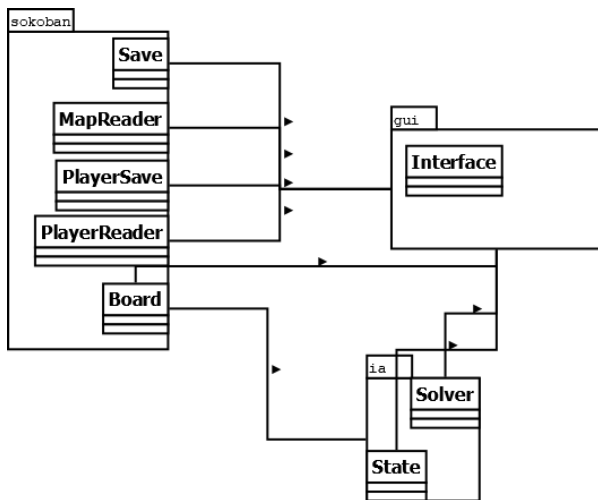
- 1 Notre projet
  - Objectifs
  - Présentation du sokoban
  - Organisation du projet
- 2 Architecture du projet
  - Packages utilisés
- 3 Éléments techniques
  - Deadlocks
  - Résolution automatique
  - Fonctionnement anytime
- 4 Expérimentations et usages
  - Performance du solveur
- 5 Conclusion
  - Réalisation des objectifs
  - Améliorations possibles

- Programmer le jeu Sokoban
- Ajout d'une interface graphique
- Résolution automatique de niveau
- Jouer contre l'ordinateur en temps réel



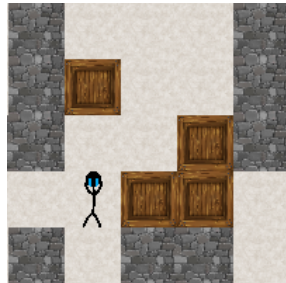
- Jeu de réflexion (puzzle)
- Poussée de caisses
- Objectif : ranger toutes les caisses

- 1ère séance : conception structure projet + début
- Puis, 3 groupes :
  - ① Version console, gestion de sauvegarde/chargement de fichiers
  - ② Solveur
  - ③ Interface graphique
- Décalage entre version console et version graphique
- Rassemblement pour la finalisation



## Définition

Une caisse en deadlock, est une caisse qu'on ne peut plus déplacer directement ou indirectement. Le jeu est donc bloqué.







## Déplacement automatisé

- Algorithme A\*

## Déplacement automatisé

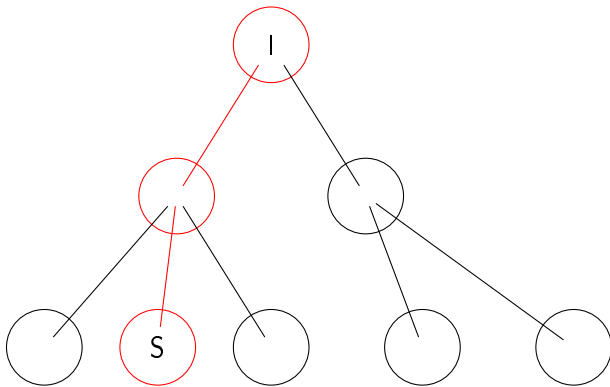
- Algorithme A\*

## Solveurs

- 1er solveur inspiré de minmax (peu efficace)
- 2nd solveur basé sur A\*

## Heuristiques :

- distances de Manhattan
- distance de Hamming



## Démonstration



Deux threads :

- Thread principal en charge de l'interface graphique et de l'écoute des entrées joueur
- Thread auxiliaire pour la recherche de chemin

Interruption du thread auxiliaire lors du déplacement du joueur et mise à jour du canvas de l'ordinateur

### Série de niveaux Sokoban junior par Laura Wheeler (58 niveaux)

Heuristique	Hamming	Manhattan
Limite de temps	5 minutes	10 minutes
Limite d'états en mémoire	175 000	180 000
Niveaux résolus	34	12
Niveaux considérés comme insolubles	0	20

- Globalement les objectifs ont été réalisés
- Première expérience de conception logicielle
- Travail en groupe

- Solveur
  - Réduire la mémoire utilisée
  - Trouver de meilleurs chemins
- Meilleur comportement de l'ordinateur en mode *anytime*
- Ajouter des fonctionnalités de terrain (par exemple des téléporteurs, etc.)
- Statistiques (temps de résolution, nombre de coups)