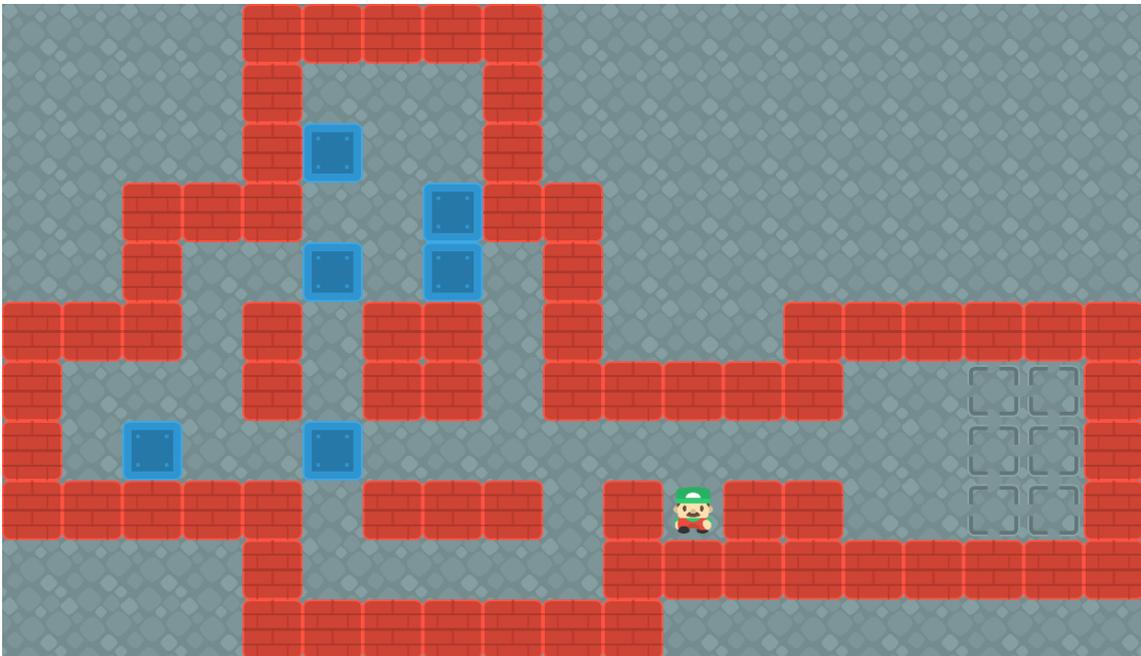


# APP : Sokoban

Florent Bouchez Tichadou — Maxime Calka

Mars 2024



*Objectifs d'apprentissage* : Graphes et cheminement, exploration intelligente, résolution automatique de problème, recherche de plus court chemin.

# I. APP : Sokoban

## I.1. Objectifs de la situation-problème

À l'issue de l'APP, vous serez capables de :

- Modéliser un problème avec des graphes ;
- Utiliser un graphe implicite plutôt que d'en construire un explicite ;
- Parcourir un graphe en utilisant un algorithme de plus court chemin ;
- Modéliser un problème de recherche de solution dans un espace combinatoire en une recherche de chemin dans un graphe ;
- Connaitre et utiliser des heuristiques pour accélérer une recherche de solution.

## I.2. Organisation des séances

Cet APP comportera quatre séances de groupe (encadrées, groupes de 4 à 6 personnes, sur les créneaux des TDs) et trois séances de mise en pratique (en autonomie et par **binômes**, sur les créneaux des Apnées).

**Entre chaque séance de groupe, du TRAvail Personnel (TRAP) est nécessaire et attendu !**

- **Séance Groupe** d'ouverture (1h30) : découverte du problème et première analyse.
- **Séance Pratique** le vendredi en binômes (minimum 3h00) : Premiers pas. Mise en place des fonctions importantes à la résolution du problème. (Le travail se base sur l'étape 09 (IA) du cours Prog6.)
- **Séance Groupe** de mise en commun (1h30) : réflexion commune sur la résolution du problème à une caisse.
- **Séance Pratique** le mercredi en binômes (minimum 3h00) : implantation de votre algorithme de résolution automatique. NOTE : pour les étudiants du groupe 3, qui ont TD le mercredi matin, la séance pratique aura lieu après le TD, **il est nécessaire d'avoir avancé en autonomie sur le code avant le TD pour avoir de quoi travailler en TD.**
- **Séance Groupe** de mise en commun (1h30) : Terminez la réflexion sur la façon de résoudre tous les problèmes proposés.
- **Séance Pratique** en binômes (minimum 3h00) : Mise en pratique de vos réflexions.
- **Séance Groupe** de mise en commun (1h30) : Terminez de résoudre vos problèmes ou entamez une réflexion sur les problèmes ouverts restants.
- **Séance Pratique** en binômes (minimum 3h00) : Mise en pratique de vos réflexions.
- **Démo** (15 minutes) : « mini-soutenance » présentation de votre programme et réponses aux questions.

## I.3. Ressources pour la recherche d'informations

Cours : Cruches, graphes et chemins ; plus courts chemins (Dijkstra et A\*) ; exploration.

TD2 : au pays des amphipodes.

## I.4. Rôles

Si vous le souhaitez, vous pouvez répartir quelques rôles au sein de votre groupe pour améliorer votre fonctionnement. Inscrivez les personnes volontaires pour chacun des trois rôles ci-dessous.

Modérateur \_\_\_\_\_  
Gardien du temps \_\_\_\_\_  
Secrétaire \_\_\_\_\_

## Situation — problème

« Une célèbre entreprise de jeux vidéos vous a recruté·e afin de finir le développement de son tout nouveau jeu vidéo de réflexion nommé *Sokoban*. Vos nouveaux collègues ont déjà réussi à programmer l'interface et les principales fonctionnalités permettant au joueur de jouer au jeu. Malheureusement, il ne reste que 4 semaines avant la sortie de ce jeu et les développeurs, ne connaissant pas bien l'algorithmique, n'ont pas été capables de développer une fonctionnalité de résolution automatique permettant aux joueurs et joueuses de pouvoir obtenir la solution du niveau dans le cas où ils/elles ne seraient pas en mesure de le finir ou afin d'obtenir la meilleure solution possible.

C'est pourquoi vous avez été recruté·e pour vos connaissances d'algorithmicien·ne. Connaissances indispensables au développement de cette fonctionnalité.

**Règles du jeu** Vous incarnez un personnage de gardien d'entrepôt, lequel est composé d'espaces de sol (cases grises) ou de murs (cases rouges). Dans cet entrepôt se trouvent également des caisses qui sont mal rangées. Vous devez les déplacer sur des cases de sol cibles. Le personnage peut se déplacer dans les quatre directions orthogonales (mais non en diagonale), 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 vous passez au niveau suivant, plus difficile. L'idéal est de réussir avec le moins de coups possibles (déplacements et poussées).

## Mes premier pas

Résoudre un problème complexe tel que le Sokoban ne peut pas se faire d'un coup. Il faut réfléchir à une manière progressive d'obtenir tous les outils pour résoudre automatiquement un niveau. Par exemple en règle générale, il est intéressant de se concentrer sur une version simplifiée du problème, ce qui souvent donne des pistes pour la suite. De plus, vous devez faire vos preuves dans l'entreprise pour gagner votre place. Pour vous aider, nous vous conseillons une première étape qui semble atteignable : Il s'agit de trouver les cases accessibles en un mouvement depuis la position du personnage pousseur.

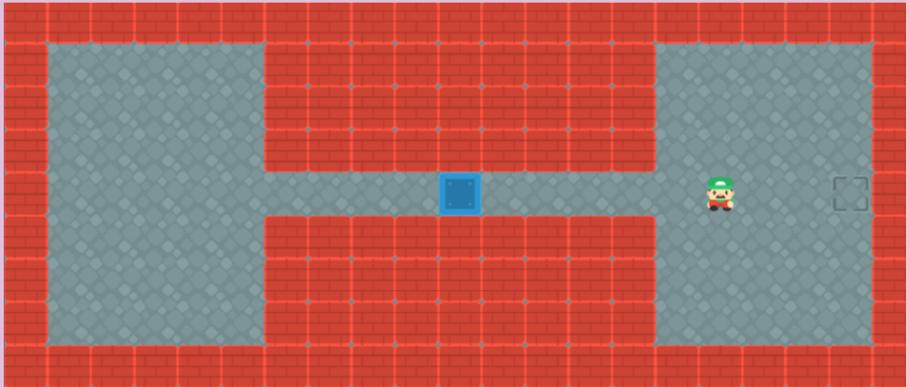


Une fois cette étape terminée et bien mise au clair, essayer de penser par vous-même à ce qui pourrait être une prochaine étape nécessaire à la mise en place d'une résolution automatique du problème. Prenez le temps de faire ce travail par vous-même avant de consulter la suite de ce document. Posez-vous par exemple les questions suivantes : « qu'est-ce qui va maintenant être difficile ? », « dans quels cas le problème sera-t-il facile ? », « ... et au contraire difficile ? »

## Résolution automatique pour une caisse

Maintenant que vous vous êtes mis en jambe et que vous avez réussi à faire vos preuves au sein de l'équipe de développement, vous devriez maintenant être capables de déplacer le pousseur et la caisse, et potentiellement de résoudre certains niveaux simples.

Mais êtes vous capables de résoudre automatiquement le niveau ci-dessous ?.



Si votre programme est capable de résoudre ce niveau, tant mieux. Sinon il va falloir commencer à réfléchir à ce qu'il manque à votre programme pour en être capable.

**Indice :** La structure de donnée que vous utilisez actuellement prend-elle en compte la caisse et le pousseur ?

**Important :** Pour vous assurer que votre programme fonctionne bien, un ensemble de niveaux vous ont été fournis. Ils permettent de s'assurer que le programme fonctionne dans des cas difficiles.

## Passage à la vitesse supérieure

Si vous arrivez avec succès à résoudre les niveaux ne contenant qu'une caisse, vous allez devoir réfléchir à comment gérer plusieurs caisses.

Cependant, attention car l'espace des possibilités risque de devenir très (**très**) gros. Il faudra donc réfléchir à comment représenter un état du jeu et comment réduire l'espace de recherche, par exemple :

- éviter l'exploration des branches qui ne peuvent mener à la solution ;
- privilégier l'exploration des branches les plus prometteuses.

## Complément d'informations

Soyez attentifs lors de votre travail personnel (implantation et réflexion sur le problème) aux difficultés que vous pouvez avoir. Notez-les sous forme de questions que vous pouvez poser sur durant le cours ou la prochaine séance de groupe.

À l'issue de cet APP, chaque **binôme** devra préparer une démo dans laquelle il devra présenter le travail réalisé ainsi que justifier ses choix de modélisation.

Pour préparer votre démonstration vous pouvez, par exemple, prévoir :

- la description de l'algorithme de résolution utilisé (notamment la métrique minimisée) ;
- de présenter les niveaux de test utilisés et expliquer pourquoi ils sont pertinents ;
- une discussion rapide des limites de votre programme et des problèmes rencontrés.