

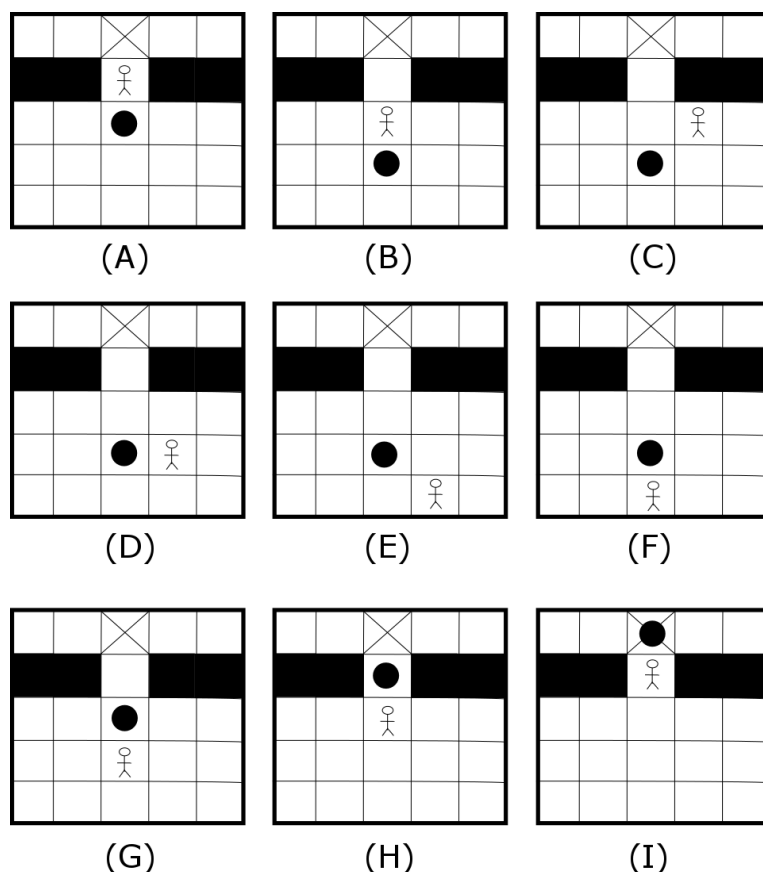
MODULE PROJET 2I013 – GROUPE 2

08 Mars 2017 – 2h

Sokoban – Plateforme de jeu et stratégies A*

On s'intéresse au jeu Sokoban, qui est un jeu de plateau à 1 joueur où l'on est amené à chaque niveau à déplacer un bloc d'un point de départ à un point d'arrivée, par déplacement successifs sur le plateau de jeu. Les mouvements du bloc sont déclenchés uniquement par poussée par un personnage dont on contrôle les déplacements. Le personnage peut se déplacer uniquement selon 4 directions haut, bas, gauche ou droite, de la case dans laquelle il se trouve à une case libre adjacente. Si la case dans laquelle il souhaite se déplacer est occupée par un mur, le déplacement est interdit. Si cette case est occupée par le bloc, le bloc est déplacé d'une case suivant la même direction que le mouvement du personnage (le personnage pousse le bloc). Si le glissement du bloc ne peut se faire, alors le déplacement est interdit. L'objectif est de trouver la meilleure séquence d'actions pour passer de la configuration initiale à la configuration visée.

Dans la figure suivante, on donne une solution optimale permettant de passer de la configuration de départ (A) à la configuration visée (I) où le bloc, représenté par un rond noir, est placé sur la position cible symbolisée par une croix :



On note que le personnage ne peut pas tirer le bloc et qu'il doit donc en faire le tour avant de pouvoir le remonter. Remarquons que si l'on se déplace vers le bas à partir de la configuration (B), aucune action ultérieure ne pourrait alors jamais permettre de remonter le bloc au niveau de la croix. Le bloc serait alors coincé sur la ligne du bas.

En réutilisant au maximum les structures développées au cours du projet pour les jeux Awélé et Othello, on cherche à définir un programme permettant de résoudre différents niveaux du jeu Sokoban. Contrairement aux jeux à deux joueurs où l'on demande au joueur de saisir tour à tour le coup qu'ils souhaitent jouer, ici le programme demande au joueur de saisir une séquence de coups (menant à une solution ou un blocage) que l'on pourra éventuellement rejouer à l'écran.

1. Donner l'idée générale du code avant de passer à l'implémentation avec un joueur humain, en explicitant l'architecture modulaire à développer et les structures utilisées. Différents plateaux de jeu pourront être considérés, comme autant de fonctions d'initialisation différentes.
2. Rappeler ce qu'est une heuristique acceptable et en proposer plusieurs (au moins 2) pour la résolution du jeu par l'algorithme A*. Développer un joueur par heuristique. Chaque joueur donne une solution qui doit pouvoir être visualisée à l'écran.
3. Comparer les heuristiques sur différents niveaux, en étudiant le nombre de nœuds de l'arbre de recherche considérés pour trouver la solution. Discuter des résultats obtenus.