

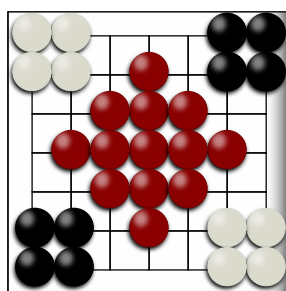
Bases de la Programmation

Projet – Kuba

V. Padovani, PPS - IRIF

1 Kuba

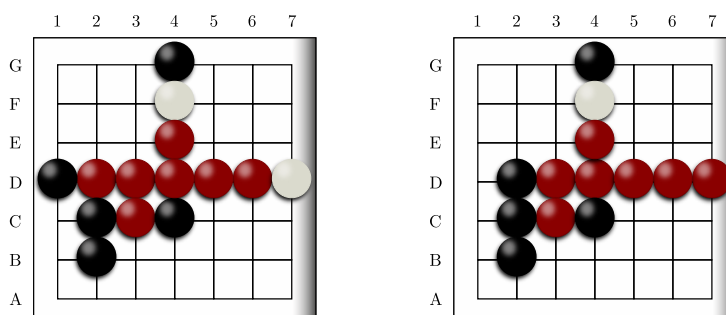
Kuba est un jeu de plateau à deux joueurs, Blanc et Noir, de règles très simples. Le matériel du jeu consiste en un plateau quadrillé de taille 7×7 , pour chaque joueur, de 8 billes de sa couleur, et de 13 billes rouges. Les billes sont posées sur les intersections de ce quadrillage et non sur ses cases. Au début du jeu, elles sont disposées ainsi :



Les joueurs effectuent à tour de rôle l'action suivante : décaler d'une position toutes les billes d'une rangée de billes adjacentes en poussant l'une de ses extrémités. Cette action, qui peut bien sûr éjecter une bille du plateau, est soumise aux règles suivantes :

- Un joueur ne peut pousser que l'une de ses billes.
- Si une bille est éjectée, elle doit être soit une bille adverse, soit une bille rouge. Elle est dans ce cas capturée par le joueur, qui peut alors, seulement s'il le souhaite, immédiatement rejouer un nouveau tour.
- Un joueur ne peut pousser une bille si la configuration résultante est la même que celle à son tour précédent.

Par exemple, dans la situation à gauche ci-dessous, en supposant que c'est le tour de Noir :



La bille en C4 ne peut être poussée vers le Nord, car cela ferait sortir une bille noire du plateau. Elle ne peut être poussée vers le Sud, qui n'est pas la direction des autres billes de la rangée verticale C4-G5. Noir peut pousser la bille en D1 vers le Nord, le Sud ou mieux encore, vers l'Est, ce qui lui fera gagner la bille blanche en D7, lui permettra de rejouer et produira la configuration à droite.

Un joueur est déclaré gagnant s'il parvient soit à capturer sept billes rouges, soit toutes les billes adverses.

2 Projet minimal

Le travail qu'il vous est demandé de réaliser est une implémentation du jeu de Kuba en langage C. Votre programme devra être raisonnablement bien conçu (*cf.* le Chapitre 6 du cours sur les structures) et bien écrit : pas de répétitions dans le code, pas de lignes de plus de 80 caractères ou de fonctions trop longues, choix pertinents de noms de variables et de fonctions, des commentaires sans en faire trop. Il devra au minimum implémenter les fonctionnalités suivantes :

1. Il devra permettre de jouer à deux joueurs, chaque joueur proposant à tour de rôle un déplacement à la machine. Le plateau de jeu devra être affiché (en ASCII) au début du jeu et réaffiché après chaque coup joué.

Le programme devra bien sûr vérifier que chaque entrée d'un joueur est correcte, et conforme aux règles du jeu. En cas d'entrée incorrecte, il devra préciser au joueur la nature de son erreur, et lui redemander de spécifier son coup. Il devra également détecter la fin de la partie, et annoncer quel est le joueur gagnant.

2. Il devra permettre de jouer contre l'ordinateur, ce dernier jouant au moins avec une stratégie simpliste (*eg.* choix aléatoire parmi tous les coups possibles), voire plus élaboré (*cf.* les extensions), mais toujours conformément aux règles du jeu.

3 Extensions

Afin de consolider votre note, de nombreuses extensions et variantes du jeu sont envisageables. Toute extension approuvée par votre enseignant sera la bienvenue. Si vous êtes à court d'idées, voici quelques propositions.

Variations de la taille du plateau. Le jeu est jouable avec les mêmes règles avec une taille de plateau différente, des billes en moins ou d'avantage de billes, et des configurations initiales différentes. À vous de trouver quelles sont celles intéressantes.

Variantes du jeu. Libre à vous d'ajouter de nouvelles règles, de nouvelles sortes de billes, etc. Par exemple, chaque joueur pourrait disposer d'une unique super-bille, capable de rouler sur le plateau jusqu'à atteindre une rangée de billes qu'elles décalera d'une position. Le plateau pourrait aussi disposer d'emplacements au comportement non-standard : position absorbant les billes, les téléportant, etc. Soyez créatifs.

Chargement et sauvegarde, historique. Votre programme sera capable de charger et sauvegarder une partie. Il pourra aussi garder en mémoire la suite de coups d'une partie en cours (complète ou non), annuler un ou plusieurs coups parmi les derniers joués, ou de rétablir une suite de coups annulés.

Interface graphique. Le jeu se prête idéalement à la conception et la réalisation d'une interface graphique. Les outils nécessaires pour cette extension (peu difficile conceptuellement, mais assez coûteuse en temps d'écriture) vous seront fournis ultérieurement.

Intelligence artificielle avancée. Il est possible pour l'ordinateur de faire beaucoup mieux qu'un choix de coup totalement aléatoire, en explorant l'arborescence des coups possibles jusqu'à une certaine profondeur¹.

4 Organisation du projet

Le projet est à faire de préférence en binôme, éventuellement en monôme – aucun projet à plus de deux ne sera accepté. Veillez à équilibrer les tâches : la simple écriture de la communication avec l'utilisateur (`printf/scanf`) ne vous apprendra rien ; la gestion des règles du jeu est la seule partie du projet qui puisse vous faire avancer en programmation.

Votre code doit évidemment être strictement personnel : il ne doit s'inspirer ni de code trouvé sur le web, ni de celui d'un autre groupe, ni d'une quelconque “aide” trouvée sur un forum. Il doit être compilable tel quel sous Linux. Votre rendu consistera en :

- un fichier `READ_ME`, décrivant brièvement comment compiler votre code,
- le code-source de votre programme, en un ou plusieurs fichiers,
- tout autre fichier utile pour la compilation ou l'exécution, (fichiers d'images, fichiers de configuration du plateau ...).
- un rapport de quelques pages en pdf décrivant vos choix de conception et d'implémentation, les extensions réalisées, les algorithmes non triviaux utilisés, les problèmes non résolus, tout autre élément non immédiat, etc.

Ces éléments, répartis dans un ou plusieurs répertoires, devront être envoyés par email sous la forme d'une archive `tar.gz` portant obligatoirement le nom `nom.tar.gz` si vous êtes en monôme ou `nom_1-nom_2.tar.gz` si vous êtes en binôme. Cette archive devra se décompresser en un répertoire portant le nom `nom` ou `nom_1-nom_2`. Les dates de rendu et de soutenance seront précisées ultérieurement sur la page du cours.

1. Voir par exemple https://fr.wikipedia.org/wiki/Algorithme_minimax