

Module 1106 : PROJET

ALGORITHMIQUE-PROGRAMMATION

HALMA

Samir Loudni, IUT CAEN – Département Informatique (Version 1.1)

20 octobre 2014

1 Description du projet

Le jeu Halma est un jeu de stratégie qui se joue sur un plateau fermé de 16 par 16 cellules avec 19 pions par joueur si le jeu se déroule à 2, ou 13 pions par joueur si le jeu se déroule à 4. On se propose, dans un premier temps, de réaliser la variante à deux joueurs.

Préparation : Chaque joueur choisit sa couleur. Les pions sont disposés aux deux angles opposés du plateau (voir le schéma à l'URL : http://jeuxstrategieter.free.fr/Halma_complet.php). Les joueurs jouent à tour de rôle.

Objectif : Le but du jeu consiste à occuper le camp adverse avec ses propres pions. Le premier qui y parvient a gagné la partie.

Déroulement du jeu : Un pion peut se déplacer sur une case voisine libre, dans les 8 directions. Si des cases autour du pion à déplacer sont occupées par des pions, il peut sauter par-dessus un autre pion, adverse ou non. Il doit venir se placer sur la case libre située immédiatement derrière. Il avance ainsi de deux cases en un seul coup. Il est permis de sauter un pion dans toutes les directions. Les pions sautés ne sont pas retirés du jeu. Le saut n'est pas obligatoire. Un pion peut enchaîner plusieurs sauts, même en changeant de direction (voir le schéma à l'URL : http://jeuxstrategieter.free.fr/Halma_complet.php).

Il est interdit de cumuler dans un même coup un saut et un déplacement simple. Il est interdit de bloquer volontairement l'adversaire en laissant ses propres pions dans son camp pour l'empêcher de venir s'y installer. Lorsqu'un joueur laisse ses pions dans son propre camp, il est déclaré perdant si son adversaire parvient à les immobiliser.

version à 4 joueurs : Une fois la version à deux joueurs fonctionnelle, on réalisera une version à quatre joueurs. Les règles sont identiques. Les joueurs peuvent former deux équipes de deux joueurs. Dans ce cas, les coéquipiers se font face sur la diagonale du plateau. Lorsqu'un joueur a placé tous ses pions, il utilise son tour de jeu pour avancer les pions de son coéquipier.

Votre programme devra permettre de visualiser une séquence de jeu en affichant à chaque tour : le plateau de jeu, le nombre de déplacements et de sauts courant du joueur (ou des joueurs). Pour ceux qui veulent aller plus loin, vous pouvez réaliser une version graphique avec animation.

2 Notes de mise en œuvre

Dans un premier temps, définissez les principales structures de données dont vous allez avoir besoin pour représenter les éléments du jeu : plateau de jeu, informations sur les joueurs et gestionnaire de jeu (contenant les informations globales concernant l'état du jeu courant). Pour chaque structure de données, définissez les données correspondantes et les fonctions associées.

Dans un deuxième temps, définissez la boucle de contrôle globale du jeu qui assurent les différentes phases du jeu. Il est conseillé d'avoir en permanence un programme qui compile et qui définit une version partielle du jeu. Seul devra être réalisée une version textuelle de ce jeu. Le programme devra permettre de visualiser la partie.

Si vous souhaitez développer une version graphique de votre programme, il est préférable dans un premier temps de définir une version non graphique afin de bien séparer les fonctions associées au jeu et l’affichage graphique du jeu.

3 Travail à rendre

Le projet est à réaliser en binômes (ou, avec l’accord de votre chargé de TP, en monôme). Le travail à rendre est un projet sous forme d’une archive zip à **déposer sur la plateforme Filex à l’adresse suivante : <http://ent.unr-runn.fr>**.

La procédure pour déposer votre projet est la suivante :

- sélectionner Université de Caen Basse Normandie puis renseigner les identifiant et mot de passe de votre messagerie universitaire.
- cliquer sur le lien Dépôt de fichiers. Télécharger depuis votre ordinateur le fichier à transmettre dans le champ Parcourir puis Envoyer.
- l’envoi du document génère une fenêtre avec un lien direct vers celui-ci. Copier ce lien pour l’insérer dans un mail adressé à votre enseignant chargé de TP.

Le nom de l’archive doit avoir la forme suivante : `Nom1Nom2.grTP.zip` ou `Nom1.grTP.zip` où `Nom1` et `Nom2` sont les noms de famille des membres des polynômes et `grTP` est le nom du groupe de TP auquel ils appartiennent (1.1, 1.2, etc.).

Note : Lors des séances de TP, les enseignants chargés de TP suivront l’avancement de votre projet et pourront vous aider sur certains points difficiles.

Ce qu’il faut rendre :

- Le code source complet de votre application en C largement commenté.
- Un exécutable testé **sous Linux** et opérationnel avec sa documentation d’installation et d’utilisation. **ATTENTION** : si votre programme fait appel à d’autres bibliothèques externes (comme par exemple la bibliothèque SDL), il est impératif de les inclure pour les besoins de test.
- Un court rapport d’une longueur comprise entre 3 et 10 pages présentant :
 - les fonctionnalités implémentées (très brièvement).
 - organisation du programme : découpage en fonctions, rôle de ces fonctions, explications du programme.
 - l’organisation et la répartition des tâches au sein du binôme durant la durée du projet (brièvement).
 - bilan qualitatif du travail, difficultés rencontrées, points qui vous ont paru intéressants.
 - un mode d’emploi avec quelques illustrations (p. ex. capture d’écrans, scénario d’exécution...), destinées à montrer l’opérationnalité de votre application.
 - Une conclusion sur l’apport (ou non) du projet en termes technique, scientifique, humain.

Le code source ne doit pas faire partie du rapport d’une dizaine de pages (sinon en annexe).

4 Évaluation du projet et calendrier

Le projet est à rendre le **le vendredi 2 janvier 2015 avant minuit** (tout retard conduira à des pénalités). L’évaluation sera réalisée en fonction des critères ci-dessous :

- *qualité technique du code* : découpage en fonctions, modularité et réutilisabilité, instructions, algorithmes, efficacité, gestion des erreurs lors de saisies, ou lorsque les données fournies en paramètres des fonctions sont incorrectes.
- *lisibilité du code* : présentation du programme (indentation), usage de variables et de fonctions ayant des noms explicites, commentaires pour préciser les points difficiles dans les algorithmes, paramètres des fonctions . . .
- *documentation fournie* : organisation du programme et son mode d’emploi, bilan.
- *présentation orale* : démonstration du programme et questions sur le travail réalisée.