Département d'Informatique
DUT Informatique S1
Année 2014/2015
Algorithmique et programmation

Projet d'algorithmique et programmation

« Reversi hexagonal »

décembre 2014

1 Objectif

L'objectif du projet est d'écrire un programme permettant de jouer à une variante du jeu de Reversi. La différence avec le jeu de Reversi classique est ici que la grille est hexagonale. Le programme devra afficher le plateau du jeu et permettre à 2 joueurs de jouer en s'assurant que les règles sont respectées. À la fin de la partie, le programme devra afficher la couleur du joueur qui a gagné, ou bien s'il y a eu match nul.

2 Règles du jeu

Le jeu se joue à deux, avec des pions noirs et blancs sur une grille hexagonale. La grille fait onze cases de diamètre. Chaque joueur a une couleur. Au départ, quelques pions sont déjà posés (cf. figure 1). Les noirs commencent. Les joueurs placent un pion de leur couleur chacun son tour, jusqu'à ce que la grille soit pleine, ou qu'aucun joueur ne puisse jouer. Le gagnant est alors celui qui a le plus de pions de sa couleur.

Un pion peut être posé uniquement s'il permet de capturer des pions adverses. Si un joueur ne peut pas placer de pion, il passe son tour. Les pions capturés sont les pions adverses qui se trouvent contigus sur une ligne, entre le pion qui est joué et un autre pion du même joueur. Les pions sont capturés dans les six directions possibles. Ainsi, la figure 2 indique les positions où les noirs peuvent jouer pour le premier tour. Lorsque des pions sont capturés, ils changent de couleur. La figure 3 donne un exemple du jeu après que les noirs ont joué sur la case en haut à gauche.

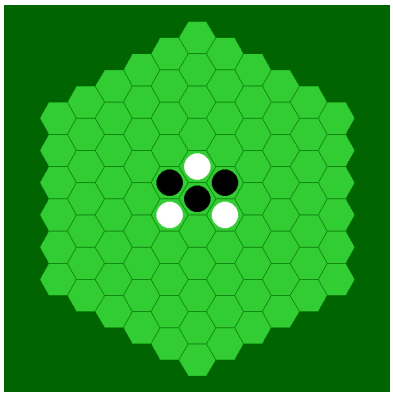

FIGURE 1 – Configuration de départ. Les noirs commencent.

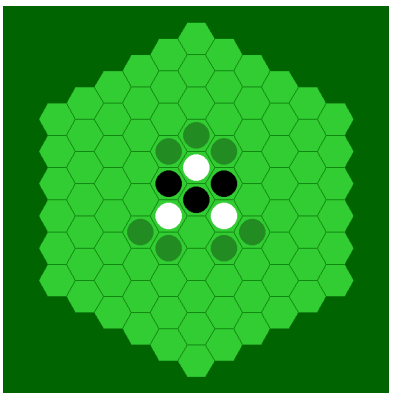

FIGURE 2 – En vert, les positions où les noirs peuvent jouer.

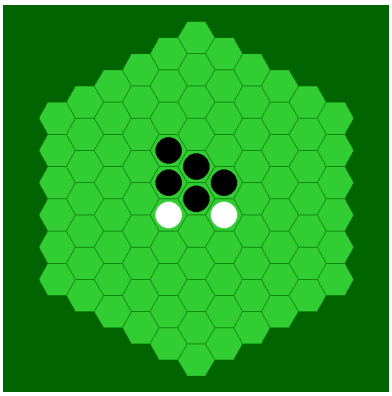

FIGURE 3 – Configuration après que les noirs ont joué.

3 Travail demandé

Un squelette de programme est fourni : `Reversi.java`. Il est demandé de compléter ce programme pour apporter les fonctionnalités listées ci-dessous.

Liste des fonctionnalités demandées Voici la liste des fonctionnalités demandées. On notera qu'il n'est pas forcément obligatoire de suivre l'ordre donné pour réaliser le travail.

1. Corriger la taille du plateau de jeu pour respecter les règles du jeu.
2. Compléter la fonction initialiserPlateau(), et placer correctement les pions de la configuration initiale. Compléter également la fonction dessinerPlateau() pour dessiner ces pions.
3. À chaque tour, afficher la couleur du joueur qui s'apprête à jouer.
4. À tout instant, afficher le nombre de pions noirs et blancs sur le plateau.
5. Terminer automatiquement la partie en cours dès qu'il n'y a plus de cases vides.
6. Ajouter la possibilité pour un joueur de passer son tour. Il n'est pas demandé de vérifier si le joueur est effectivement bloqué, ou s'il pourrait jouer un coup valide.
7. Ajouter la possibilité d'abandonner une partie en cours.
8. Vérifier la validité d'un coup, et n'autoriser que les coups valides.
9. Lors d'un coup valide, déterminer les pions qui sont capturés, et changer automatiquement leur couleur.
10. À la fin d'une partie, afficher le gagnant. Corriger également la valeur de retour de la fonction jouerPartie(). S'assurer que le message affiché sur la console lorsqu'on quitte le programme est cohérent.
11. Ajouter la possibilité de recommencer une nouvelle partie si les joueurs le souhaitent.
12. Ajouter la possibilité de changer la taille du plateau en l'indiquant sur la ligne commande. La taille sera définie par le rayon de la grille de jeu (variable rayonPlateau).

Les points 6, 7 et 11 peuvent être réalisés en dessinant un bouton (un rectangle) et en vérifiant si le joueur a cliqué à l'intérieur ou non.

Programme à rendre Le programme à rendre doit être une évolution du code fourni. En particulier, le nom du fichier (`Reversi.java`), et le nom de la classe (Reversi) doivent être conservés. Comme dans le code fourni, le programme doit afficher le jeu en utilisant la bibliothèque graphique DrawingWindow utilisée en TP. Les paramètres de rendu (couleurs, etc.) peuvent cependant être librement adaptés.

Le programme doit être composé d'un seul fichier source Java. Ce fichier doit pouvoir être **compilé sans erreur**. Un programme qui ne se compile pas sera considéré comme nul. Il est impératif de compléter le commentaire en début de fichier pour rappeler le nom, le prénom et le groupe de l'étudiant.

Le code source doit être correctement indenté. Un retrait de 4 espaces doit être utilisé, et les lignes ne doivent pas dépasser 80 caractères [1]. Lors de la correction, il sera de plus porté une attention particulière :

- à l'usage pertinent des commentaires ;
- à l'usage pertinent des fonctions ;
- au choix des noms de variables et de fonctions ;
- à la lisibilité du code en général.

1. Cf. `http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-136091.html`

Rapport Un rapport doit également être rendu. Ce rapport doit être écrit en français, sans faute d'orthographe ou de grammaire. Il doit expliquer ce qui a été réalisé pour chacune des fonctionnalités requises. Éventuellement, il peut comporter des algorithmes en langage algorithmique, mais pas de copier-coller du code Java. Si des fonctionnalités supplémentaires ont été programmées, elle doivent également être décrites dans le rapport.

Enfin le rapport doit être rendu sous forme de **fichier au format PDF**. Un rapport rendu dans tout autre format sera considéré comme nul.

4 Commentaires sur le programme fourni

Le programme `Reversi.java` implémente les fonctions essentielles pour la gestion d'une grille hexagonale. Il fournit également un squelette pour le jeu de Reversi. Les différentes fonctions sont commentées dans le code source, mais voici néanmoins un résumé de leur rôle dans le programme.

Le programme commence par la définition d'un certain nombre de constantes et de variables globales. On trouvera par exemple des constantes pour définir les couleurs utilisées pour le rendu graphique. Les constantes PION_NOIR et PION_BLANC servent à désigner la couleur

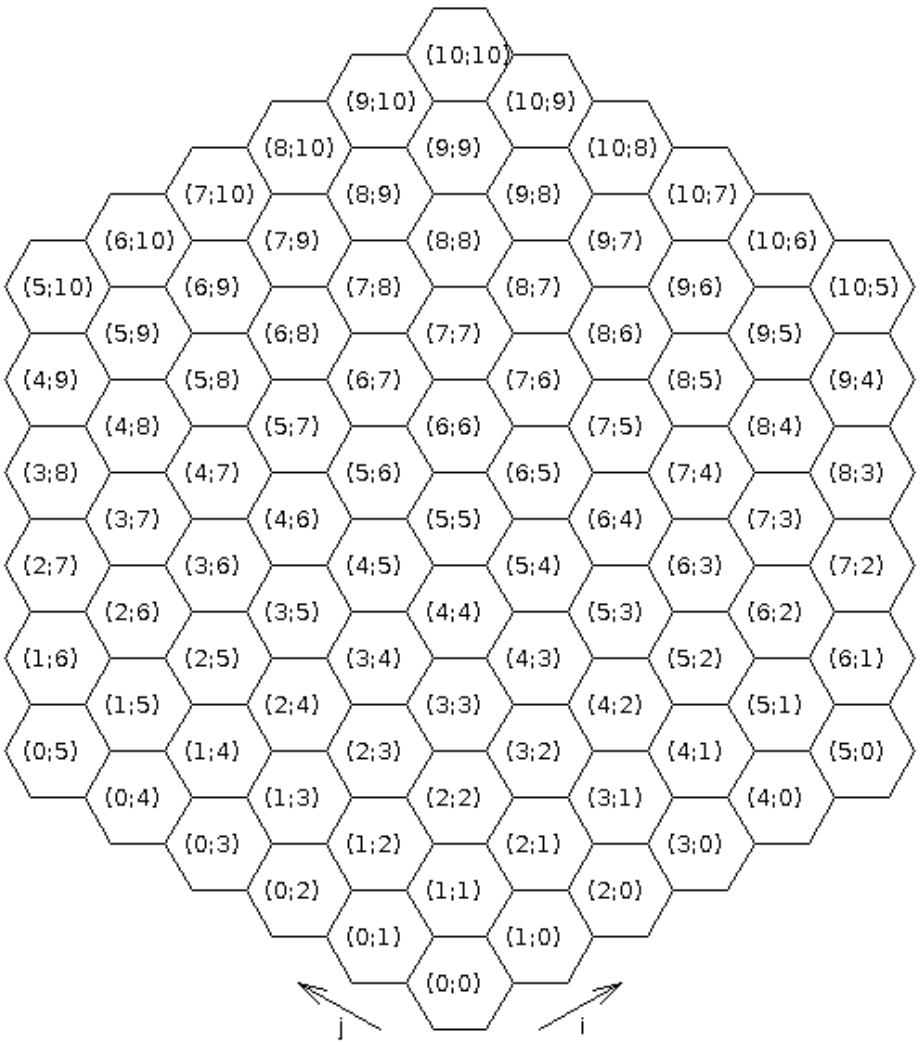

FIGURE 4 – Coordonnées $(i; j)$ des cases hexagonales.

d'un joueur. Associées à PION_VIDE, elles servent également à coder le contenu d'une case de la grille de jeu. Les variables globales sont celle qui servent à l'ensemble du programme. On trouvera par exemple la taille de la grille (taille), le tableau représentant l'état du jeu (plateau) ou la fenêtre graphique (w). Ces variables sont initialisées une fois pour toutes au début du programme par la fonction initialiserParametres().

Les cases de la grille hexagonale sont repérées grâce à un système à deux coordonnées $(i; j)$ illustré par la figure 4. La case de coordonnées $(0; 0)$ est placées tout en bas. Ces coordonnées servent également à retrouver le contenu d'une case dans le tableau plateau. La fonction caseValide() sert à vérifier si une paire de coordonnées désigne une case valide de la grille. Les fonctions calculerCoord() et calculerCase() servent à effectuer les conversions (dans les deux sens) entre les coordonnées d'une case et les coordonnées d'un pixel de la fenêtre.

Les fonctions dessinerCase(), dessinerPion() et dessinerPlateau() servent, comme leurs noms l'indiquent, à dessiner une case, un pion ou la grille du jeu.

Au début d'une partie, le plateau est initialisé par la fonction initialiserPlateau(). C'est ensuite la fonction jouerPartie() qui gère la partie. Elle utilise pour cela les fonctions jouerCoup() pour jouer le coup d'un joueur, et inverserJoueur() pour changer de joueur après chaque coup.

Les fonctions initialiserPlateau(), jouerPartie() et jouerCoup() font partie des fonctions qui doivent être complétées et/ou modifiées, de même que la fonction recommencerPartie() qui doit permettre de demander aux joueurs s'ils veulent recommencer une partie.

Enfin, il est tout à fait autorisé (voire même recommandé) de modifier le code fourni et de définir de nouvelles fonctions s'il y en a besoin.

5 Contraintes diverses

Date importante. Le programme devra être rendu pour le **mardi 6 janvier 2015, 10h00** au plus tard.

Combien de personnes par projet ? Les projets sont à faire **individuellement**. Des outils automatiques pourront être utilisés pour comparer les codes sources et détecter les tentatives de copie.

Comment rendre le projet ? Le projet devra être envoyé par mail à l'adresse suivante :

Arnaud Giersch <arnaud.giersch@univ-fcomte.fr>.

Un accusé de réception vous sera retourné. Si vous n'avez pas reçu d'accusé de réception le mardi 6 janvier à 14h00, il faudra renvoyer votre projet.

Que faut-il rendre exactement ? Il est demandé de rendre exactement deux fichiers : le code source du programme, et le rapport au format PDF.

Il est inutile de rendre le fichier `DrawingWindow.java` ou le programme compilé (fichiers .class).