

Énoncé du projet « Hexmo »

Présentation

Cette année, l'activité intégrative consistera en l'implémentation d'une variante du jeu « [Hex](#) » que nous appellerons « Hexmo ». « Hex » est un jeu de réflexion sur lequel tu vas pouvoir défier tes amis et ta famille. Des parties endiablées sont à la clé de cette AI. Cerise sur le gâteau, tu pourras jouer sur ta propre version du jeu !

Ce document décrit les règles générales du jeu, ainsi que l'organisation de l'activité intégrative.

Règles du jeu « Hexmo»

Dans le jeu « Hexmo », deux joueurs tentent de relier deux bords d'un plateau de jeu hexagonal tout en empêchant leur adversaire de faire de même. La Figure 1 présente un exemple de plateau :

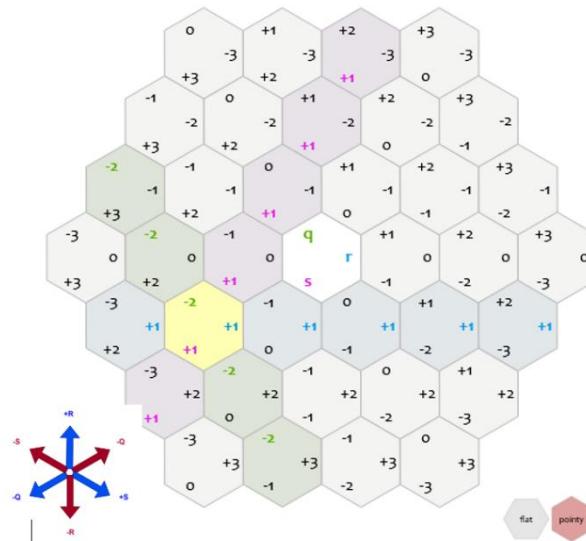


Figure 1 : un plateau de jeu Hexmo

Comme vous pouvez le voir, le plateau est composé de cases hexagonales. Le système d'axes en bas à gauche de la Figure 1 indique les bords qu'un joueur doit connecter. Un joueur devra connecter deux bords positifs ($+q$, $+r$ et $+s$ sur la figure) et son adversaire les bords négatifs ($-q$, $-r$ et $-s$). Plus généralement, la présence de trois axes introduit un nouveau système de coordonnées : le système de coordonnées axiales.

Activité Intégrative – Hexmo – Énoncé général

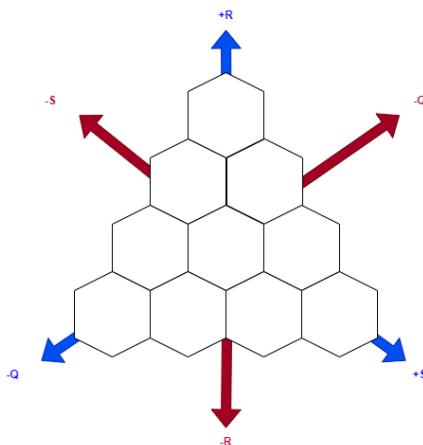


FIGURE 2 CHAQUE AXE ORIENTE DEUX ARÈTES D'UN HEXAGONE

Éléments de coordonnées axiales

Chaque case occupe une position déterminée par un système de coordonnées axiales. Dans un système de coordonnées axiales, une coordonnée est un triplet d'entiers $\langle q, r, s \rangle$ tels que $s = -q-r$. Vu que la composante s est calculable, on se contente de représenter une coordonnée axiale par la paire $\langle q, r \rangle$.

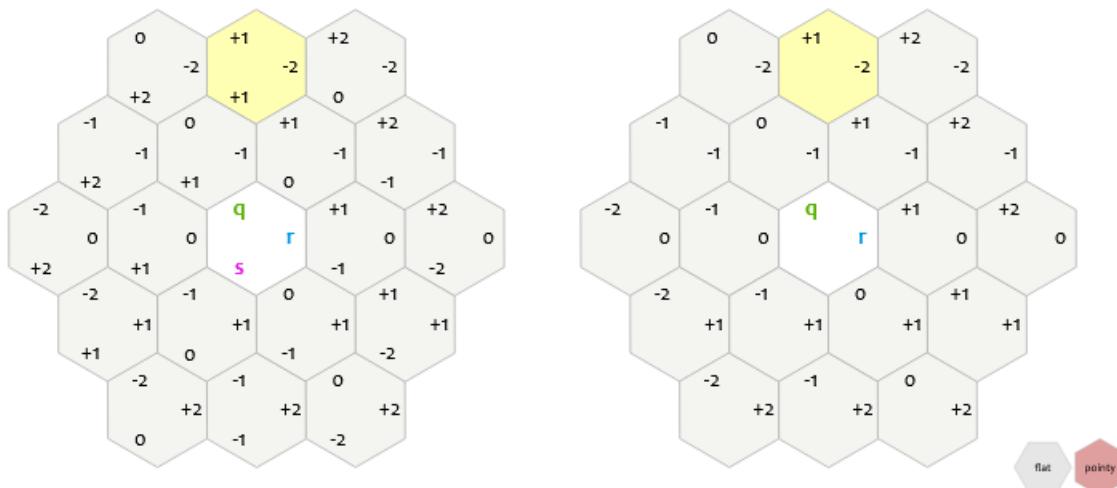


FIGURE 3 COORDONNÉES AXIALES COMPLÈTES (À GAUCHE) ET SIMPLIFIÉES (À DROITE)

Le système de coordonnées axiales permet d'introduire des concepts utiles tels que la notion de distance entre les cases, de voisins d'une case, etc. Par exemple, la distance entre deux cases correspond à la formule $d(c_1, c_2) = \frac{|c_{1q}-c_{2q}|+|c_{1r}-c_{2r}|+|c_{1s}-c_{2s}|}{2}$. En particulier, la distance entre la case du centre et celle d'une case du bord est constante. Sur la Figure 1, elle vaut 3. Cette valeur correspond au rayon du plateau. Outre la notion de rayon, nous dirons que le diamètre du plateau, exprimé en nombre de cases, correspond à 2 fois son rayon + la case centrale. Le diamètre du plateau de la Figure 1 vaut 7.

Les cases voisines d'une autre case seront celles dont la distance à cette autre case vaut 1. Elles correspondent au plus à six cases présentées par la figure 4. On constate qu'il y a moyen de calculer les coordonnées voisines d'une case sans faire intervenir la composante s . Cependant, sa valeur changera. En effet, la valeur de s est déduite de celles de q et r .

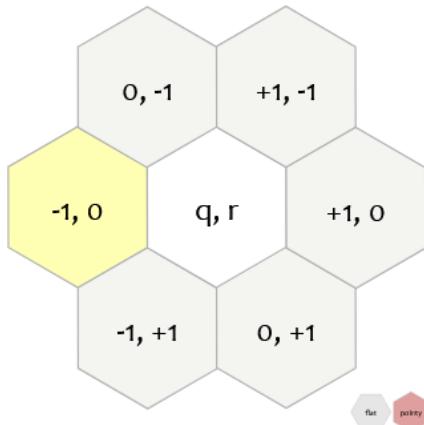


FIGURE 4 CASES VOISINES

Déroulement d'une partie

Une partie entre deux joueurs commence par l'attribution d'une couleur. Dans la suite du document, nous parlerons de **joueur rouge** et de **joueur bleu**, mais toute autre paire de couleurs convient.

Ensuite, le **joueur rouge** place un de ses pions sur une case de son choix¹. Le **joueur bleu** peut ensuite décider de changer de couleur : c'est l'étape de « swapping ». Le « swapping » incite le premier joueur à placer son premier pion sur une case « neutre », qui ne favorisera aucun joueur pour la suite de la partie.

Remarque

Le « swapping » n'a lieu qu'une seule fois pendant la partie, après le placement du premier pion par le joueur rouge.

Après le « swapping », c'est au tour du joueur bleu de poser un pion. Les joueurs posent alternativement un pion jusqu'à ce que l'un d'eux connecte deux de ses bords, deux bords négatifs pour le joueur rouge et deux bords positifs pour le joueur bleu. Les joueurs placent leurs pions sur n'importe quelle case, sauf pour le cas particulier des cases du bord :

- Le **joueur rouge** peut placer un pion sur une case du bord si le signe de son plus grand composant est négatif.
- Le **joueur bleu** peut placer un pion sur une case du bord si le signe de son plus grand composant est positif.

Attention, il existe des cases du bord qui peuvent être affectées par le joueur rouge ou le joueur bleu. Elles correspondent aux six coins du plateau de jeu.

Fin de partie

Dans la plupart des cas, le jeu se termine quand il existe un chemin composé de cases de la même couleur qui lie deux bords de cette couleur. Le joueur qui possède cette couleur est déclaré vainqueur.

Après chaque partie, l'application affichera le nombre de parties remportées par chaque joueur.

¹ On considère que les pions sont disponibles en quantité illimitée.

Contraintes

Vous devez programmer une application Java pour jouer des parties de « Hexmo ». Elle utilisera la bibliothèque standard Swing pour interagir avec l'utilisateur (le code de l'interface utilisateur vous sera fourni).

Votre projet devra répondre aux contraintes suivantes :

1. Il est **personnel**. Toute tentative de plagiat mènera à une sanction disciplinaire (cf. [Fraude et plagiarism](#))
2. Il cible la version 17 de Java et intègre l'interface graphique Swing (fournie).
3. Il est compatible avec la version d'Eclipse de référence (voir « ressources informatiques »).
4. Les tests unitaires du projet seront écrits avec la bibliothèque JUnit 5.
5. Il utilise le *plug-in PMD en version 7*, en intégrant un fichier de règles fourni (*ruleset*) qui valide plusieurs critères.

Itérations

L'activité intégrative est divisée, pour la première session, en trois itérations :



Chaque itération vise un objectif différent :

- Itération 1 : Créer le projet, démarrer et afficher la partie.
- Itération 2 : Jouer une partie, sans détecter la fin, et assister les joueurs.
- Itération 3 : Déetecter la fin de partie en analysant l'historique des coups joués et afficher le nombre de parties remportées par chaque joueur.

Une itération dure l'équivalent de 10 heures de laboratoires. Il vous faudra très probablement compléter ces heures par du travail à domicile. À la fin de chaque itération, vous devez soumettre votre projet et vous recevrez un retour de la part de votre responsable de laboratoire.

Remarque

Entre l'itération 2 et l'itération 3, vous profiterez des vacances de printemps. L'énoncé de l'itération 3 sera disponible après ces vacances.

Chaque itération sera évaluée par votre responsable afin d'observer un ou plusieurs des acquis d'apprentissage suivant :

1. Programmer des énoncés de conception en Java selon les principes de la POO.
2. Valider les comportements des objets programmés par des tests unitaires.
3. Documenter son code et rédiger un rapport technique spécifiant un problème donné, incluant un plan de test, une évaluation de complexité et une solution correspondante.

4. Justifier le choix d'une structure de données particulière (tableaux, listes, files, piles, ensembles, maps), et d'une implémentation spécifique

Conditions de réussite

Le cours est réussi lorsque vous avez **validé** les cinq acquis d'apprentissages (voir Fiche UE). Les acquis 1 à 4 sont considérés comme validés dès qu'ils ont été **observés deux fois**, donc sur 2 itérations différentes. Le 5^e acquis (*présenter et défendre le fruit de son travail de réflexion, de comparaison, de création oralement selon un schéma/canevas spécifique*) peut être validé lors d'une défense orale, organisée en juin. Cette défense orale permet en outre de valider jusqu'à deux autres acquis manquants.

Dès lors, l'accès à l'oral est restreint par les règles suivantes :

- Avoir au moins deux acquis validés parmi les 4 ;
- Si deux acquis doivent encore être validés, l'un des deux doit avoir été observé préalablement ;
- Si un seul acquis doit encore être validé, il n'est pas nécessaire qu'il ait déjà été observé.

Puisqu'il faut valider les 5 acquis pour réussir, ne pas avoir accès à l'oral signifie que vous recevez une note d'échec en première session et que vous devez présenter la seconde session en aout.

Pour être clair, voici un tableau récapitulatif :

Nombre d'acquis validés	Nombre d'acquis observés mais non validés	Accès à l'oral ?
3 ou plus	1 ou moins	OUI
2	1 ou plus	OUI
2	0	NON
1 ou moins	Peu importe	NON

Il va de soi que l'objectif est de viser la situation correspondant à la première ligne du tableau 😊

Fraude et sanction

Rappelons que ce travail est personnel : tout partage de code, toute utilisation de code que vous n'avez pas écrit sans autorisation explicite de votre responsable est une **fraude**.

Dans ce cas, nous entamerons directement une **procédure disciplinaire**.

Le résultat de la procédure n'est pas de notre responsabilité. Sachez cependant qu'elle *peut conduire à la suspension de toute votre session* si la fraude est avérée. Soyez également conscient que **le copieur et le copié** sont susceptibles d'être reconnus coupables de fraude.

Bref, ne partagez pas votre code, sous quelque forme que ce soit !