L2 informatique, L2 maths-info, M1 linguisique info, EIDD
POOIG **2016-2017**

Projet: Jeux d'alignement

1 Modalités

Le projet est à faire en binôme au plus. Les soutenances se feront à deux, mais la note pourra être individualisée si le travail a été trop inégalement réparti. Naturellement, chacun doit être capable de répondre à toutes les questions. Si vous faites le projet seul, vous serez noté sur le même barème que si vous étiez à deux.

La modélisation est à rendre sur Didel (diagramme de classes plus brève explication sur ce que fait chaque classe) sous format *PDF* d'ici le 1er Décembre, le nom des participants au binôme sera indiqué dans le *PDF* et dans le nom du fichier.

Vous aurez le droit de modifier cette modélisation.

Le projet est à soumettre, sous forme d'une archive (tar ou zip) contenant le nom des participants au binôme. (Pas d'archive rar). Elle devra contenir :

- les sources de votre programme
- un fichier nommé `README` qui indique comment on se sert de votre programme ;
- un rapport au format *PDF* de quelques pages expliquant concisément les parties traitées, les problèmes connus, et les pistes d'extensions que vous n'auriez pas encore implémentées. Naturellement, il contiendra un diagramme de classes. Vous imprimerez ce rapport pour le jour de la soutenance. La date de rendu du projet vous sera communiquée ultérieurement.

Votre code devra être soigneusement commenté. La notation prendra fortement en compte la façon dont le jeu est codé (architecture, factorisation, propreté du code, etc) Nous vous invitons à ne pas perdre votre temps dans des graphismes sophistiqués, mais plutôt à soigner la qualité de votre code, sans quoi vous risquez d'être extrêmement déçus de votre note.

1.1 Soutenances

Les soutenances se feront à deux, mais la note pourra être individualisée si le travail a été trop inégalement réparti, naturellement chacun doit être capable de répondre à toutes les questions.

Il nous faudra pouvoir tester le projet durant les soutenances, il est donc préférable qu'il fonctionne sur les machines du script. Ceci même si vous avez prévu d'apporter un portable pour l'occasion car il peut tomber en panne au mauvais moment.

2 Sujet

Le but de ce projet est d'implanter deux jeux ayant des similarités. Dans les deux jeux, la taille du plateau peut être choisi avant le début du jeu.

2.1 Travail à réaliser

vous devez modéliser les deux jeux et au moins une variante de Color Lines.

Pour la réalisation, vous devez faire les deux jeux mais la variante de Color Lines n'est pas obligatoire.

De plus, pour tous les jeux: vous devez programmer, en plus d'un joueur humain, un joueur "robot stupide" qui à chaque tour, jouera un coup autorisé, le premier que le programme trouve, si bien sûr on peut encore jouer. Il ne s'agit donc pas d'intelligence artificielle!

Par ailleurs, pour tous ces jeux, la taille du plateau doit être paramétrable.

2.2 Gomuku

Le Gomuku se joue à deux. Il se joue sur un plateau avec des pions noirs et blancs. L'un des jouueurs pose les pions blancs, l'autre les noirs. À chaque tour, un joueur pose un seul pion sur une case vide.

Le gagnant est celui qui a réussi à faire le plus d'alignement de 5 pions de sa couleur. L'alignement se fait horizontalement, verticalement ou en diagonale. Les pions qui ont servi à un alignement peuvent resservir pour un autre.

Le jeu s'arrête lorsque le plateau est plein. (ou variante, lorsqu'un des joueurs a totalisé un certain nombre d'alignements, ce nombre étant décidé avant le début du jeu).

2.3 Color lines

Ce jeu est une version modifiée du jeu "kolorlines" appelé aussi "fiveormore", etc.

On y joue à un contre l'ordinateur. On utilise des pions colorés sur un plateau. Le nombre de couleurs utilisées est fixé à l'avance. Il existe de plus une catégorie de pions spéciaux "arc-en-ciel" dont nous reparlerons. Le but du joueur est de faire là aussi le maximum d'alignements (horizontaux, verticaux et diagonaux) d'au moins 5 pions de même couleur.

À chaque tour, l'ordinateur place au hasard 3 pions colorés sur le plateau. Ensuite le joueur peut déplacer un des pions. Si un alignement d'au moins 5 pions de même couleur a lieu (avant ou après le déplacement) les pions concernés sont éliminés du jeu. Les pions arc-en-ciel se combinent avec toutes les couleurs: on peut même imaginer qu'un pion arc-en-ciel rentre, en même temps, dans un alignement vertical avec 4 pions rouges et un alignement diagonal avec 4 pions bleus.

On joue jusqu'à ce que le plateau soit plein. Le score final est alors affiché.

A chaque alignement correspond un certain nombre de points: un alignement de 6 rapporte plus qu'un alignement de 5, de même deux alignements croisés simultanés rapportent plus que les même alignements faits séparément. Nous vous laissons libre du barème exact.

Variantes: Vous devez modéliser, et éventuellemnt programmer, en plus du jeu de base, une variante de ce jeu. Voici quelques idées:

- Le joueur peut convertir une partie des points gagnés en le droit de choisir, pour un tour, la couleur des pions qui vont être posés, mais pas leur place. Il n'a pas le droit de choisir la couleur arc-en-ciel.
- Même chose, mais là, le joueur peut choisir la place mais pas la couleur.
- Les cases du plateau sont colorées en damier noir et blanc. Les pions ne peuvent maintenant plus être déplacés que vers une case de même couleur. Si par le plus grand des hasards toutes les cases noires sont occupées et aucune des blanches ne l'est (ou vice-versa), le joueur ne plus déplacer de pion: le jeu est alors fini.

3 Interface graphique

Elle est obligatoire. L'interface minimum devra afficher l'état du jeu, par contre il est possible que le joueur donne des instructions au clavier plutôt qu'à la souris.

Nous attirons votre attention sur le fait qu'il faut privilégier une architecture MVC (modèle-vue-contrôleur) et en particulier séparer le modéle de la vue.

3.1 Conseils pour faire l'interface graphique

cette section sera complétée ultérieurement

4 Conseils pour faire le projet

Il est chaudement recommandé de réfléchir à l'architecture du jeu avant de coder. Nous attirons votre attention, qu'il ne s'agit pas de faire un jeu + un jeu, les deux jeux n'ayant rien en commun.

Si vous n'avez pas le temps de faire tous les jeux nous vous recommandons de faire au moins Color lines et de nous montrez, schéma à l'appui, comment l'autre jeu et la variante choisie de colorlines, rentrerait dans votre architecture.

1. Prenez le temps de réfléchir à la conception du projet avant de coder.
2. Réfléchissez aussi à la manière de répartir le travail entre les deux personnes du binôme. Faites des points réguliers entre vous.
3. Compilez et testez régulièrement ce que vous faites.
4. Vous pouvez éventuellement dans un premier temps vous contenter d'une interface texte pour pouvoir tester votre jeu et ajouter l'interface graphique après. Si jamais vous n'arriviez pas à obtenir une interface graphique qui fonctionne, vous pourrez présenter le projet juste avec une interface texte. Vous ne pourrez pas avoir tous les points, mais vous en aurez plus que si vous présentez un projet non testable.
5. Enfin, pensez à faire des sauvegardes fréquentes, sur au moins deux supports différents et en particulier sauvez les versions testables de votre projet même s'il reste des corrections à y apporter.

6. Si vous n'avez pas le temps de tout faire:

    - vous pouvez supprimer l'interface graphique en proposant une interface texte.
    - dans tous les cas, nous vous recommandons de faire au moins Color lines et de nous montrer, schéma à l'appui, comment l'autre jeu et la variante choisie de colorlines, rentreraient dans votre architecture