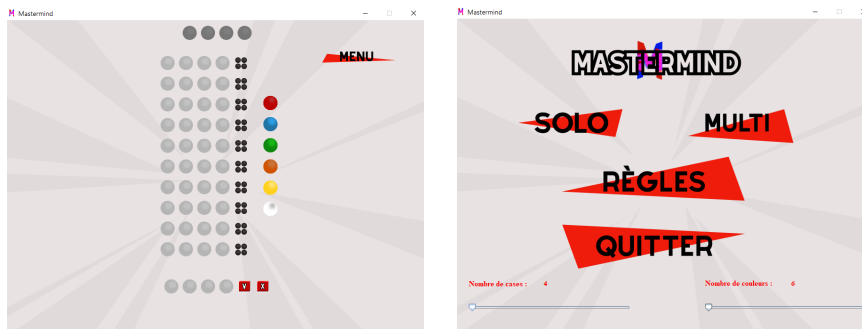


EPHEC

2ème Technologie de l'informatique

Dév. informatique avancé : application (Pratique)

Mastermind



Groupe 23 :

Clément BRANCART
Nassim BRAHMI
Florence SALPIETRO

Encadrant :

Mr VAN DORMAEL

Lien du projet github : <https://github.com/flosalpietro/Projet-Java>
21 décembre 2018

Table des matières

1	Cahier des charges	2
1.1	Dans le mode solo :	2
1.2	Dans le mode multijoueur :	2
2	Ce qui a été mis en place	3
3	Diagramme UML	4
4	Choix des implémentations et difficultés rencontrées	6
5	Améliorations possibles	6
6	Mode d'emploi	6
7	Conclusions personnelles	7
7.1	Clément Brancart	7
7.2	Nassim Brahmi	7
7.3	Florence Salpietro	7
8	Conclusion	8

1 Cahier des charges

Ceci correspond au cahier des charges rendu le 26 octobre.

Le jeu possède deux vues : une vue console et une vue graphique.

Le jeu doit pouvoir être joué en solitaire contre l'ordinateur ou en mode multijoueur avec une personne distante.

Le menu propose 4 options :

- Lancer une partie contre l'ordinateur
- Lancer une partie multijoueur avec une personne distante
- Accéder aux règles du jeu
- Fermer l'application

1.1 Dans le mode solo :

L'ordinateur crée aléatoirement une combinaison secrète que l'utilisateur doit deviner. Une combinaison secrète est composée de 4 pions. Pour créer une combinaison l'ordinateur dispose de plusieurs couleurs. Le nombre de couleur varie en fonction de la difficulté du niveau.

Le joueur dispose de plusieurs tentatives pour trouver la bonne combinaison. Le nombre varie selon le niveau de jeu.

Au premier niveau, l'ordinateur utilisera 5 couleurs différentes et l'utilisateur aura 8 tentatives pour retrouver la bonne combinaison de 4 pions.

Après chaque tentative, l'ordinateur signal au joueur si des pions sont de bonne couleur et se trouvent à la bonne place sur la ligne ou s'ils sont uniquement de la bonne couleur. La combinaison secrète est révélée si le joueur la trouve ou si le nombre de tentative est épuisé.

A la fin de chaque partie, le jeu propose au joueur s'il désire relancer une partie ou retourner au menu principal.

1.2 Dans le mode multijoueur :

Le mode multijoueur utilisera des sockets pour que les deux joueurs puissent jouer une partie. Le joueur qui initie la partie sera le serveur et le deuxième joueur le client, ils pourront de cette manière échanger les données durant la partie.

Le joueur qui initie la partie crée la combinaison secrète, il a le choix entre les différents niveaux (différents nombres de couleur et de tentative).

Il doit guider l'autre joueur à la fin de chaque tentative en lui signalant s'il a trouvé une couleur ou un emplacement correct.

2 Ce qui a été mis en place

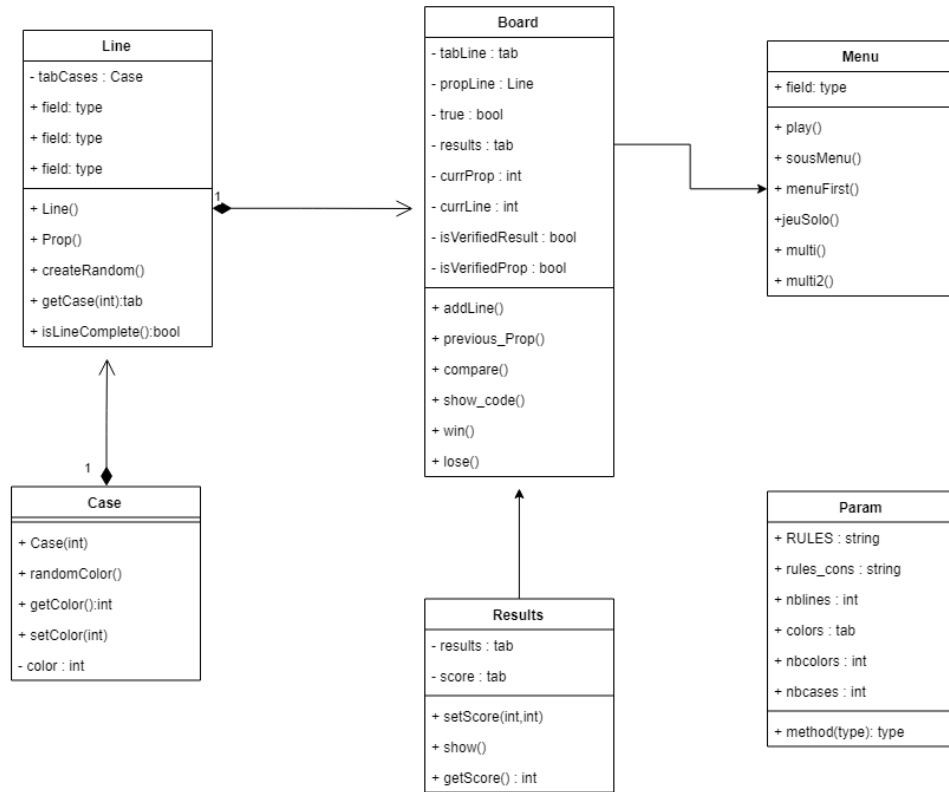
Au fil de l'implémentation de notre jeu nous avons choisi de modifier quelques règles par rapport à notre cahier de charges.

Le jeu que ce soit en mode solo ou multijoueur, il vous propose de jouer avec une combinaison de 4 pions choisis parmi 6 couleurs différentes. En dessous du menu général sur l'interface graphique, on retrouve deux sliders qui permettent de faire varier ces valeurs. La combinaison peut être composée jusqu'à 6 pions parmi 10 couleurs différentes. Le nombre de tentative pour retrouver la combinaison lui ne varie pas, il reste fixé à 10 tentatives. Le jeu affiche un pion rouge si une couleur à la bonne place a été trouvée et un pion blanc si la bonne couleur est présente mais pas au bon endroit. Quand la partie est terminée, le jeu affiche la combinaison secrète qu'il fallait trouver en haut du plateau et vous invite à revenir au menu. De là on choisit de rejouer une partie ou de quitter le jeu.

Concernant le mode multijoueur, le jeu vous demandera si vous voulez être serveur ou client. L'ordinateur créera une combinaison secrète que les deux joueurs vont devoir essayer de trouver. Ils jouent chacun à leur tour et le premier qui a trouvé la bonne combinaison gagne la partie.

3 Diagramme UML

UML Console
Mastermind
Groupe 23



4 Choix des implémentations et difficultés rencontrées

La partie qui concernait la conversion de tout notre code en architecture MVC fut difficile. Nous avons pris beaucoup de temps à essayer de comprendre comment répartir notre code dans les différentes couches MVC. En effet les vues du MVC ne devraient pas effectuer de traitements sur les données mais juste afficher les données qui se trouvent dans le modèle. Mais redistribuer les classes entre le modèle, le contrôleur et les vues est compliqué à faire quand les deux interfaces fonctionnent déjà indépendamment l'une de l'autre avec les traitements de données dedans.

Nous n'avons pas réussi à utiliser l'architecture et les méthodes MVC (setChanged(), notifyObservers(), Update(),...) qui allaient avec. Vous pouvez retrouver nos essais sur le github dans le dossier MVC.

L'autre partie qui nous a pris beaucoup de temps fût l'implémentation de l'interface graphique.

Rappel, lien du projet github : <https://github.com/flosalpietro/Projet-Java>

5 Améliorations possibles

- Premièrement améliorer notre jeu pour qu'il puisse correspondre à l'architecture MVC permettrait certainement qu'il soit plus performant. Cela permettrait également de pouvoir lancer les deux interfaces en même temps.
- En mode multijoueur, dans l'idéal les deux joueurs pourraient communiquer à l'aide d'un petit Chat.
- Le joueur pourrait également décider du nombre de tentative avant de commencer une partie.

6 Mode d'emploi

Télécharger la dernière version jeuMastermind sur notre github, l'ouvrir dans eclipse et lancer Main dans le package viewConsole pour la vue console ou MainFrame dans le package viewGraphique pour l'interface graphique.

// executer le .jar

7 Conclusions personnelles

7.1 Clément Brancart

Ce projet m'a été bénéfique car j'ai pu revoir pas mal de choses qui étaient connues pour moi mais dont je ne savais plus comment les utiliser. Mais j'ai également vu de nouvelles choses comme les sockets. Pour ce qui est du groupe, nous avons un bon groupe qui n'a pas cessé de travailler. Nous nous étions bien réparti les tâches et mise à part quelques fois où l'on a dû se presser les uns les autres nous avons pu finir le projet dans l'ensemble plus ou moins.

7.2 Nassim Brahmi

De mon point de vue, la réalisation du projet ne s'est pas déroulée dans les meilleures conditions. Chaque membre du groupe ayant un horaire différent cela a rendu moins aisé la mise en place d'un planning de travail de groupe. Dès le départ, il y'a eu un manque de leadership afin de coordonner les différentes tâches ainsi que leur réalisation. Ce qui a donné lieu à un manque de synchronisation et de communication entre les différents membres du groupe. Le résultat a été que certaines parties du code ont parfois été faites de manière redondante et non concertée. Je suis également conscient que le projet a été commencé sur de mauvaises bases. Il aurait dû être démarré par une première phase d'analyse devant tenir compte du pattern MVC. Les concepts d'orienté objet ont tant bien que mal été suivis afin que l'apport d'amélioration à l'application soit aisé. S'il fallait recommencer ce projet, il serait réalisé dès le départ avec une implémentation en MVC. Hormis tout cela, le projet a été très bénéfique dans la compréhension de la programmation O.O Apports : - Mise en place du code de l'application

7.3 Florence Salpietro

Le projet a été instructif dans son ensemble mais je trouve le choix du thème du projet compliqué. C'est difficile de trouver un sujet pas trop ambitieux et qui passionne tout le groupe. Même si le projet n'est pas totalement abouti, j'ai l'impression que nous étions proche du but, notre inter-

face graphique est plutôt complète et fonctionne bien. L'obligation d'utiliser l'architecture MVC complique certainement les choses et nous aurions dû commencer directement le projet en "pensant" MVC. Mais nous avons d'abord réalisé presque tout le projet de manière classique et après nous avons essayé de repasser en MVC. Ce qui se compliquait de plus en plus au fur et à mesure où on avançait. Je suis tout à fait d'accord avec la conclusion de Nassim. Si on devait recommencer ce projet, je m'impliquerais plus dès le départ pour faire en sorte qu'on parte tous sur de bonnes bases : avoir une meilleure coordination, une meilleure communication, un plan bien détaillé, etc. C'est ce qu'il nous a manqué pour avoir un projet bien abouti.

8 Conclusion

Le bilan du projet par rapport à ses objectifs est en demi-teinte. Nous n'avons certes pas rempli l'ensemble des fonctions attendues au cahier des charges, en particulier l'architecture MVC (model vue contrôleur). Nous sommes quand même satisfaits d'avoir su produire un jeu qui fonctionne avec une interface graphique. Et retiendrons qu'il faut parfois prendre un peu plus de temps de réflexion et d'analyse avant de se lancer !