STEAKELLO

Rapport de projet

GROUPE 8

Projet de Java

Pierre Dambois, Jean-Baptiste Bonhomme, Steffi Baugnies

Développement informatique avancé : application

Projet Java 2016 - Steakello
Développement informatique avancé : application

Descriptif / Cahier des charges

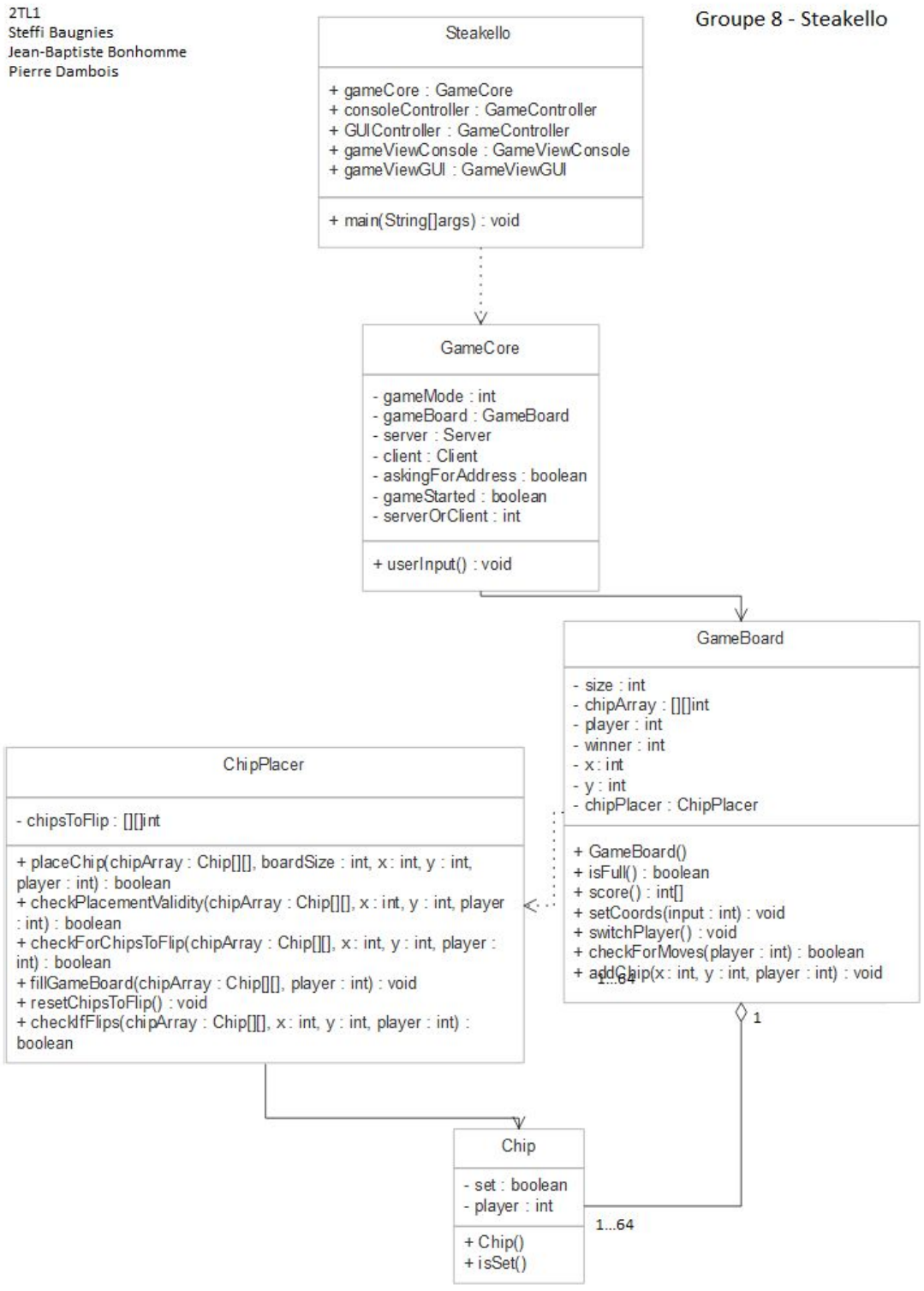
Notre jeu sera de type plateau qui s'affichera sous forme de grille de huit cases sur huit. Le but de ce jeu sera pour les joueurs de prendre le contrôle d'un maximum de cases grâce à des pièces de couleur. Dans le cas présent, ces pièces seront représentées par des steaks cuits ou crus en fonction du joueur. Le jeu commence avec quatre pièces placées au centre du plateau, deux de chaque couleur. Les pièces de même couleur sont placées en diagonales. Tour par tour, chaque joueur placera une pièce sur le plateau de manière à englober une ou une ligne des pièces de l'adversaire que cette ligne soit en horizontale, verticale ou diagonale. Si aucun coup n'est possible, le tour du joueur sera passé automatiquement. Si deux pièces d'un joueur se trouvent de part et d'autre d'une pièce ou d'une ligne de pièces du second joueur, ces dernières seront converties en pièces de la couleur du premier joueur. Le jeu se termine lorsqu'il n'y a plus de cases vides sur le plateau et le gagnant est le joueur ayant le plus de pièces de sa couleur. Pendant la partie, les deux joueurs pourront voir le nombre de pièces qu'ils possèdent sur la grille. Le jeu comportera deux modes : les deux joueurs sur une même machine et deux joueurs sur des machines différentes.

Groupe

- Steffi Baugnies
- Pierre Dambois
- Jean-Baptiste Bonhomme

GitHub https://github.com/DamboisP/steakello

Diagramme UML



Choix d'implémentation

Threads

Nous avons décidé d'implémenter des threads quand nous nous sommes retrouvés bloqués entre l'interface graphique et la vue console. Nous avions besoin que les deux fonctionnent indépendamment et l'utilisation des threads était le meilleur moyen que nous avons trouvé. Grâce à ceux-ci, certaines fonctionnalités dont nous avions besoin pouvaient tourner comme tâches de fond.

Sockets

Nous avons opté pour une communication par le biais de sockets pour faire interagir un serveur et un client. Nous ne voyions pas vraiment ce que pouvait apporter une base de données dans notre projet alors que les sockets nous ont permis d'avoir deux modes de jeu différents. Malgré leurs implémentation complexe, ils apportent tout de même un élément intéressant en plus à notre application.

Difficultés

Choix du sujet

Le premier problème que nous avons rencontré était le choix du sujet. Nous étions partis sur une idée qui n'aurait pas pu répondre à tous les critères d'évaluation. Nous avons donc dû changer d'idée ce qui n'a pas été facile. Beaucoup de nos idées auraient été beaucoup trop difficiles à réaliser mais nous n'avions pas non plus envie de choisir un sujet trop simple. Nous voulions également un élément d'originalité dans notre projet donc il nous a fallu trouver une idée qui se prêtait à cela. Nous nous sommes finalement mis d'accord sur une version d'Othello que nous pouvions revisiter à souhait.

Console et GUI

Nous avons aussi rencontré un problème pour rendre la console et la GUI indépendante. Nous avons cherché assez longtemps pour trouver quelque chose qui aurait pu fonctionner sans devoir changer trop de notre code. Nous avons finalement décidé d'utiliser des threads séparés comme susmentionné.

MVC

Notre problème avec le MVC était la compréhension. Nous avons voulu l'implémenter assez rapidement dans le développement de l'application car c'était cela qu'il nous manquait dans notre première idée. Nous voulions donc être sûr de ne pas nous manquer là-dessus. Nous avons donc essayé d'implémenter un modèle MVC avant d'avoir vu la théorie dessus. Le plus gros problème était que la plupart de nos recherches renvoyaient quelque chose de légèrement différent voire même complètement contradictoire. Nous avons donc commencé notre première version de MVC sans trop savoir si nous partions dans la bonne direction. Après avoir vu la théorie, nous étions rassurés ce que nous avions compris par nous-même n'était pas trop loin de ce qui était voulu. Avec quelques changements, le modèle MVC était implémenté d'une manière que nous pensons être correcte.

Bug

Nous avons, vers la fin du projet, rencontré plusieurs problèmes que nous n'avions pas remarqué auparavant dans le jeu. Ceux-ci nous ont pris beaucoup plus de temps à trouver et réparer que prévu. Comme les problèmes avaient l'air de se présenter de manière assez aléatoire, nous avons eu beaucoup de mal à trouver leur origine. Cela nous a empêcher de peaufiner le jeu comme nous le désirions.

Sockets

Le problème que nous avons rencontré pendant l'implémentation des sockets était la récupération des adresses IP. Étant donné que cela est une des premières étapes de l'implémentation, ça a été assez bloquant. Après beaucoup de tentatives, nous avons réussi à faire fonctionner cela puis avons été bloqué à l'étape suivante, l'attribution d'un port au serveur. Les sockets ont probablement été la partie la plus difficile du projet pour nous.

Améliorations possibles

Refactoring

Certaines de nos méthodes (en particulier checkForChipsToFlip) auraient pu être séparée en plusieurs méthodes plus simples. Les performances auraient également être plus optimisée. Sur la fin, nous avons préféré nous concentrer sur le fonctionnement.

GUI

Nous aurions aimé améliorer la GUI. Nous étions partis sur une GUI fonctionnelle et rapide pour pouvoir travailler sur le côté interface graphique de notre jeu le plus rapidement possible. Le problème principal de cette GUI est qu'il faut cliquer dans le haut des cases pour la sélectionner. Compte tenu des problèmes que nous avons rencontré lors de l'implémentation des sockets, la GUI est devenue une tâche de seconde importance et nous n'avons pas eu le temps de l'améliorer comme nous aurions aimé le faire.

Restart

Nous avions dans l'intention d'implémenter un bouton restart qui permettrait de recommencer une partie ou de rejouer avec la même personne. Par faute de temps, cela n'a pas été fait.

Vérifications

Comme spécifié dans la Javadoc, une de nos méthodes (checkIfFlips dans ChipPlacer.java) a été rajoutée en dernière minute pour traiter des cas particulier que nous n'avons remarqué que très tard dans l'avancement du projet. De ce fait, nous n'avons pas eu le temps d'adapter les tests unitaires pour cette méthode.

IA

Avec la méthode checkForMoves, nous vérifions déjà tous les coups possibles pour un joueur à chaque tour. En faisant une liste de ces coups et en en choisissant un au hasard, nous aurions pu faire une intelligence artificielle rudimentaire.

Conclusions personnelles

Steffi

Ce projet s'est avéré être une expérience assez enrichissante. C'était le premier grand projet qu'il nous était amené de réaliser depuis le début de nos études. Nous n'avons pas rencontré de problèmes pour ce qui est de l'entente dans le groupe ce qui nous a permis d'avancer dans notre travail assez sérieusement. Dans la plupart des cas, nous n'avons pas eu de difficultés pour nous mettre d'accord. Les idées de tous ont été entendue. Nous avons souvent travaillé ensemble au lieu de chacun de notre côté ce qui a rendu l'expérience chouette en plus d'être instructive. Nous avons appliqué les principes du développement agile dont le pair programming. Plus personnellement, le développement de ce projet du début à la fin m'a permis de voir les choses d'une manière encore différente des travaux pratiques ou des cours théoriques. Ce projet était vraiment une bonne manière d'enrichir et d'approfondir mes connaissances en Java. Le plus gros problème que j'ai rencontré était l'implémentation des sockets. C'est la partie de l'application que j'ai le moins bien compris. C'est grâce à mon groupe que cette partie a été terminée et que j'ai une meilleure compréhension de cette matière.

Pierre

Ce projet s'est globalement bien déroulé pour un premier projet de programmation sérieux. Nous avons rarement manqué d'idée que ce soit lors du processus créatif ou bien lorsque nous devions penser de manière logique. En effet, la bonne entente dans le groupe a permit à tous d'avoir confiance en ses idées, personne n'a donc été laissé en retrait et tout le monde a pu être écouté. Les principaux freins que l'on a pu rencontrer étaient l'implémentation de MVC, le fait de pouvoir utiliser au choix la console et l'interface graphique ainsi que les sockets. La mise en place de Github a elle aussi été complexe au début mais il s'est avéré être un outils pratique pour travailler à plusieurs sur un même projet, lorsqu'on s'y est finalement habitué. Globalement ce projet était une bonne manière d'approfondir nos connaissances en Java mais cela nous a aussi appris à travailler correctement en groupe, ce que nous n'avions jusque là jamais fait.

Jean-Baptiste

Ce projet s'est bien passé dans sa globalité, pour un premier projet important de programmation. La bonne entente présente dans notre groupe nous a permis d'avancer sereinement dans le projet sans laisser de membres à la traîne. Nous n'avons pas eu de problèmes pour nous mettre d'accord dans la majorité des cas et l'avis de chacun a été entendu. Pour ce qui est des apports personnels, nous avons préféré travailler la majeur partie du temps ensemble que dans notre coin, ce qui nous as permis d'avancer à un bon rythme. Nous avons pu expérimenter plusieurs principes du développement Agile dans notre projet dont le pair programming qui s'est avéré être très efficace dans notre cas. Pour ma part, ce projet m'a permis d'appliquer la théorie vue aux cours d'une manière un plus poussée qu'au tp. Le projet m'a permis d'approfondir la matière et d'améliorer ma compréhension de certains points en Java. Ce projet aura été dans son ensemble une expérience enrichissante.