

Rapport du Projet GL-QL

PUISSANCE 4

PUISSANCE 4

PRÉPARÉ PAR

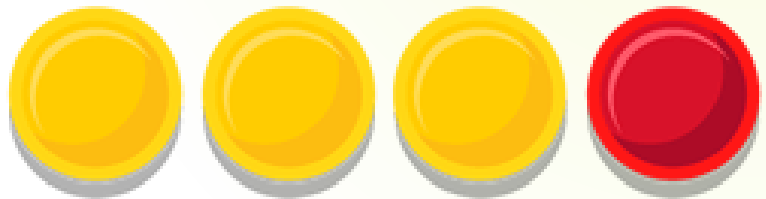
YANIS SEGHAU

YASSER IBOUDA

AMIN MARHERAOUI

ILYES ABBASSI

DHIA EDDINE SOUSSI



Sommaire

I. INTRODUCTION

II. ORGANISATION DU PROJET

III. FORCES ET FAIBLESSES

IV. BILAN DU PROJET

V. REMERCIEMENTS

Introduction

Durant les deux derniers mois, notre équipe composée de Ilyes ABBASSI, Yasser IBOUDA, Amin MARHERAOUI, Yanis SEGHDAU, et Dhia Eddine SOUISSI, a travaillé avec enthousiasme sur le développement d'une application de Puissance 4 en Java, jouable sur console. Le choix de ce projet s'est fait naturellement, étant donné notre attachement personnel à ce jeu stratégique classique auquel nous avons tous joué durant notre jeunesse.

Ce projet nous a offert une excellente opportunité de consolider et d'appliquer les compétences acquises au cours de notre formation, notamment en algorithmique, et en architecture applicative. En optant pour une application console, nous avons décidé de nous concentrer sur la logique, la qualité de code et les mécanismes du jeu, évitant ainsi les complexités associées au développement d'une interface web.

Ce rapport retrace notre parcours, depuis l'organisation initiale jusqu'à la réalisation finale, en passant par les défis rencontrés et les aspects un peu mieux réussis. Nous partagerons les outils et méthodes utilisés pour faciliter la collaboration, les obstacles surmontés et les leçons apprises tout au long de cette aventure.



Lien vers le repo GitHub

Organisation du projet

Au cours du développement de ce projet, nous avons utilisé plusieurs outils et méthodes pour faciliter la collaboration et la gestion du code :

- Trello : Nous avons utilisé Trello pour le suivi de l'avancement du projet. Chaque fonctionnalité ou problème à résoudre était représenté par un ticket, assigné à un membre de l'équipe. Une fois terminé, le ticket était déplacé dans la colonne "Fait".
- Discord : Nous avons utilisé Discord pour communiquer en temps réel. Nous avons créé un serveur Discord pour notre équipe, avec des canaux textuels pour discuter de l'avancement du projet et des problèmes rencontrés, et des canaux vocaux pour organiser des réunions et discuter de questions plus complexes.
- Github : Le versioning a été effectué sur Github. Pour l'intégration du nouveau code, des Pull Requests ont été effectuées et vérifiées par chacun des membres avant d'être fusionnées dans la branche Master. Un Workflow GitHub Actions a été mis en place pour exécuter automatiquement les tests à la création de la PR mais aussi lors de la fusion des Pull Requests (Merge).
- IntelliJ : L'ensemble de l'équipe a travaillé sur IntelliJ, utilisant parfois la fonctionnalité "Code With Me" pour du pair programming. De plus, nous avons utilisé le plugin SonarLint pour analyser la qualité du code et détecter les mauvaises pratiques.

Faiblesses

Au cours du développement de ce projet, nous avons rencontré plusieurs défis :

- **Injections de dépendances** : Nous avons eu des difficultés à comprendre quels composants devaient être injectés et comment le faire correctement. Nous avons utilisé notre diagramme de classes pour nous guider dans cette tâche.
- **Utilisation de Git** : La gestion des conflits entre plusieurs fichiers a été un défi, notamment lorsque plusieurs membres modifiaient simultanément des fichiers similaires. Nous avons utilisé des outils de gestion des conflits intégrés à Git et IntelliJ pour résoudre ces problèmes plus efficacement.
- **Gestion des entrées utilisateurs et des sorties console** : Nous avons eu du mal à gérer les entrées et sorties en respectant les bonnes pratiques Java. Les erreurs de saisie de l'utilisateur et la gestion de l'internationalisation ont été particulièrement complexes.
- **Algorithme de joueur IA** : L'implémentation d'un algorithme de la famille Minimax pour l'IA a été difficile. Nous avons finalement opté pour une IA simple qui choisit une colonne semi-aléatoirement, bloque les coups gagnants de l'adversaire et essaie de gagner si elle en a l'occasion.
- **Tests (mock)** : La création des tests a posé des défis, notamment pour déterminer ce qu'il fallait mocker ou non. Nous avons dû apprendre à isoler les composants à tester correctement pour garantir la fiabilité et l'efficacité de nos tests. Certaines classes ont été plus compliquées à tester car leurs actions consistaient à gérer l'affichage, nous avons donc dû tester l'appel de certaines méthodes dans les fonctions testées.

- Fonctionnalité jeu en pause : Nous avons eu des difficultés lors de l'ajout de cette fonctionnalités en raison des mauvaises dépendances que le nouveau code allait impliquer. Nous avons donc effectué un grand travail de refactoring pour intégrer la fonctionnalité et garder un code robuste.

Forces

- Gitflow : L'utilisation de Gitflow a structuré notre processus de développement.
- Organisation : L'utilisation d'outils comme Trello, Discord, et Github a facilité la collaboration.
- Qualité du code et de l'architecture de l'application : Nous avons développé une application robuste aux changements de dépendances (exemple : passer de la console à un affichage web sera plus simple à implémenter). La bonne gestion des exceptions est aussi un des points qui a contribué à la robustesse de l'application.
- L'internationalisation : nous avons fait une application qui peut être utilisée dans plusieurs langues (anglais, français), cela a été possible grâce à l'intégrations des messages de jeu dans des fichiers de paramétrage et à la modification d'une seule variable pour switcher.
- L'utilisation de Docker : Nous avons pu mettre en place une image Docker afin que les utilisateurs de notre application puissent l'exécuter n'importe où et dans un environnement isolé.

Bilan du projet

Ce projet nous a permis de mettre en pratique les concepts que nous avons appris en cours. Nous avons pu travailler en équipe et utiliser des outils de gestion de projet pour faciliter la collaboration. Nous avons également pu améliorer nos compétences en conception et architecture pour créer une application bien structurée et facile à maintenir.

Des outils comme ChatGPT et Copilot nous ont également été d'une grande aide lorsque nous avons besoin d'assistance. Ces outils nous ont permis de surmonter rapidement les obstacles techniques et de trouver des solutions efficaces à nos problèmes de programmation.

Nous les avons utilisés pour générer de la documentation technique, l'implémentation d'algorithmes complexes, et pour tester des cas d'usage compliqués. Le point sur lequel ces outils nous ont moins aidé a été sur l'organisation et l'architecture du code. En effet, en tant que développeurs, nous avons une vision plus précise et plus orientée sur la qualité générale de notre projet.

Pour ce qui est du cours, nos avis sont assez convergents. Nous avons trouvé le cours très parlant du fait que vous avez donné beaucoup d'exemples concrets à partir de votre propre expérience professionnelle. Cela nous a permis de comparer avec notre propre expérience en tant qu'alternants. Nous avons aussi trouvé que vous étiez très franc et direct dans vos propos lorsqu'on avait des questions par exemple et c'est une chose qu'on a beaucoup apprécié. Nous n'avons pas vraiment trouvé de points d'amélioration si ce n'est le fait d'avoir un peu plus de démos en direct comme vous l'aviez fait pour des tests unitaires. Des TP pratiques de 30min par exemple auraient peut-être aussi aidé à intégrer les notions du cours.

Remerciements

Nous tenons à vous exprimer notre sincère gratitude pour avoir pris le temps de lire notre rapport. Votre attention et considération à notre travail sont très appréciées. Nous espérons que le rapport vous a fourni des informations utiles et éclairantes.

Nous restons à votre disposition pour toute question ou discussion supplémentaire concernant le contenu du rapport. Votre feedback est non seulement bienvenu, mais aussi essentiel pour nous permettre d'améliorer nos travaux futurs.

Merci encore pour votre temps et votre intérêt

Coordonnées :

Yanis SEGHD AU : yanis.seghdau@etu.u-paris.fr

Amin MARHERAOUI : amin.marheraoui-berrahhou@etu.u-paris.fr

Yasser IBOUDA: yasser.ibouda@etu.u-paris.fr

Ilyes ABBASSI : ilyes.abbassi@etu.u-paris.fr

Dhia Eddine SOUISSI : dhia-eddine.souissi@etu.u-paris.fr