



# DOSSIER DE SYNTHÈSE

19/01/2016

Développement d'un jeu MasterMind en console – rédaction des spécifications fonctionnelles d'un jeu Tic-Tac-Toe.

# TABLE DES MATIÈRES

## Contenu

Introduction	1
Organisation du travail	2
Problèmes rencontrés & solutions trouvées	4
Ce qui est perfectible	5
Retours personnels	6

## Introduction

Ce dossier de synthèse permet de rendre compte du travail effectué pendant les séances de Génie Logiciel. Il nous a tout d'abord été demandé de rédiger les besoins fonctionnels et scénario de test sur le jeu Tic-Tac-Toe, pour une autre équipe de développeurs. Ensuite, en se basant sur les besoins fonctionnels de cette autre équipe, nous avons dû écrire les spécifications fonctionnelles et développer le jeu MasterMind, sous la forme d'une application console.

Pour cela, quelques contraintes nous ont été imposées : l'utilisation de git, la portabilité de notre application, l'utilisation de markdown pour la rédaction des besoins et spécifications fonctionnelles concernant le Tic-Tac-Toe, l'utilisation de java pour implémenter notre jeu MasterMind et une documentation complète.

Ce dossier s'articule autour du lien entre la théorie vue en cours et son application au travers du projet. Nous présenterons également l'organisation du travail au sein de l'équipe, les problèmes rencontrés et leurs solutions, ainsi que ce qui est perfectible sur le projet, et enfin un retour personnel.

# ORGANISATION DU TRAVAIL

## Organisation du travail

### PRESENTATION DE L'EQUIPE

WHITNEY BURNER DEVELOPPEUR	MAXIME DELERIN DEVELOPPEUR	TED RAMONI DEVELOPPEUR/ TESTEUR
		
ANTOINE GALLAIRE DEVELOPPEUR/ TESTEUR	ALEXIS COLONNA MOA/ DOCUMENTATI ON	ALEX VEYNAND MOA/ DOCUMENTATI ON
		

# ORGANISATION DU TRAVAIL

## REPARTITION DES TACHES

Whitney et Maxime se sont occupés de la définition de l'architecture générale du projet et de l'implémentation de son squelette. Ils ont également travaillé sur l'algorithme de comparaison.

Ted et Antoine se sont quant à eux occupé de l'implémentation générale du code, la configuration du git pour le projet. Ils ont également mis en place l'affichage console, l'environnement de test et ont réalisé des tests fonctionnels.

Alexis et Alex ont rédigé les besoins fonctionnels et le cahier de recette de l'application console Tic-Tac-Toe pour une autre équipe de développeur.

Ils ont également rédigé les spécifications fonctionnelles pour le MasterMind, ainsi que la documentation du projet. Ensuite, ils se sont occupés de la rédaction de la Jadavoc, du dossier de synthèse et ont effectué des tests fonctionnels.

## Problèmes rencontrés & solutions trouvées

L'équipe a rencontré peu de difficultés dans l'ensemble. Il y a eu seulement quelques obstacles à surmonter au niveau organisation comme au niveau développement.

### ORGANISATION & FONCTIONNEL

Quelques lacunes dans la communication entre les développeurs a entraîné un besoin de changer de manière importante l'architecture, entraînant une perte de temps.

En effet, cela a nécessité un gros refactoring.

Fonctionnellement, le plus gros problème qu'on ai eu a été d'être capable de rédiger un cahier de recettes permettant d'effectuer un test complet de l'application. Il est en effet assez laborieux de penser à tout les cas possibles, d'autant plus quand l'application n'est pas encore créée.

### TECHNIQUE

Au niveau programmation, le problème qui s'est posé a été l'algorithme de comparaison des deux combinaisons. En effet, nous voulions trouver un algorithme qui soit performant et qui ne nécessite pas un grand nombre de boucles imbriquées.

Nous avons eu aussi quelques difficultés quant à la gestion des conflits sur Git ce qui a entraîné quelques pertes de données (de la javadoc).

Par ailleurs, nous avons eu une coupure de réseau à l'IUT, nous faisant perdre une heure de travail.

## Ce qui est perfectible

Dans les différentes améliorations que pourrait subir notre projet nous pouvons citer une amélioration, au niveau de la vitesse de comparaison, de l'algorithme, mais ce n'est pas une amélioration majeure car nous n'effectuons pas beaucoup de comparaisons lors d'une partie.

Il aurait été possible d'améliorer le système de résolution des parties en employant des expressions rationnelles, mais le manque de temps et la nécessité de produire rapidement cette fonctionnalité afin de poursuivre le développement nous a contraints à adopter une solution plus classique.

Nous aurions pu utiliser Jira ainsi qu'une plateforme de télémétrie afin d'assurer le suivi du code mais l'ampleur du projet ne nécessitait clairement pas un tel déploiement.

Si nous avions eu plus de temps, nous aurions également pu implémenter les couleurs dans la console système.

## Retours personnels

Organisation du travail => 2 jours présents à l'IUT seulement...

Antoine :

*« Le projet fut intéressant d'un point de vue organisation. Le fait de découper le projet en deux équipes, une MOA et une MOE nous ont permis de nous mettre en situation réelle et d'appliquer les notions inculquées pendant les cours. En m'occupant plus de la partie test et codage des fonctionnalités simples, cela m'a permis de les coder d'une manière simple et performante. Les tests étant ma faiblesse dans la programmation, j'ai pu réapprendre la méthode et l'appliquer dans ce projet. »*

Alex :

*« Pour la première fois depuis le début de l'année, nous réalisons un projet à 6, ce qui implique une bonne organisation interne de l'équipe et une répartition des tâches. Sachant peu coder, j'ai pu prendre part avec Alexis à la maîtrise d'ouvrage et m'attarder sur les besoins rédactionnels du projet. J'ai trouvé commode de diviser les équipes de travail selon les compétences, mais nous avons été limité en terme d'organisation par le manque de présence sur le projet : le rythme d'alternance a constitué un frein sur ce type de projet où 6 membres doivent coordonner leur travail. »*

Alexis :

*« Le rôle de MOA m'a particulièrement plu car il m'a permis d'apprendre à correctement définir un besoin, de manière à ce qu'il soit compréhensible pour l'équipe de MOE. Je me suis également occupé de la Javadoc, qui m'a permis de mieux comprendre comment fonctionne le code. Et je me suis occupé de faire les tests fonctionnels. L'organisation du travail était un peu compliquée, étant donné le nombre de développeur et le temps que nous avions pour nous voir. La communication a été plutôt bonne grâce à notre groupe Facebook. »*

Ted :

*« Le problème de communication sur git fut particulièrement contraignant, et j'aurais pris plaisir à utiliser un framework pour apprendre à utiliser Spring par exemple, mais cela ne s'appliquait pas à la taille de notre projet, c'est tuer une mouche avec un tank ! Par ailleurs, nous avons rencontré un problème avec le clear de la console, qui s'est résolu rapidement en utilisant une invite de console plutôt que la console Eclipse/IntelliJ. Sinon, bon point pour le git, car nous avons su gérer les gitignore, d'autant plus que peu de membres de l'équipe était familiarisé avec cet outil. A part une fois où nous avons rencontré des conflits, tout s'est bien passé. »*



## RETOURS PERSONNELS

Whitney :

*« La communication s'est faite via Facebook majoritairement, car c'est un réseau simple d'utilisation. Le partage des sources et des différentes versions du projet s'est fait lui sur git. Le plus grand problème rencontré est l'algorithme de résolution des combinaisons. L'implémentation de notre première version de l'algorithme ne réalisait pas correctement la comparaison dans certains cas particuliers. Nous avons néanmoins résolu ce problème assez rapidement, après quelques redéfinitions de l'algorithme. Il marche désormais parfaitement. »*

Maxime :

*« Le projet s'est bien déroulé dans son ensemble. Tout le monde a contribué au code grâce à la bonne configuration du dépôt, et les versions quasi hebdomadaires nous ont permis de tout tester au fur et à mesure afin de ne pas être acculé par les bugs au dernier moment. Chaque mise à jour majeure du code fut commentée par son développeur au reste de l'équipe. La communication et la synergie de groupe était donc très bonne, quoiqu'un peu compliquée lorsqu'à distance. Le projet en lui-même était clairement établi par les documents de conception. Il ne nécessitait aucune utilisation de bibliothèques externes mais notre architecture nous permettrait d'en implanter facilement (Un framework d'interfaces graphiques par exemple). Je regrette personnellement l'utilisation de Java : l'application s'exécute dans une console système, et n'exploite absolument pas l'environnement virtuel java. De plus, d'autres langages (comme le c/c++) nous auraient permis de mieux optimiser le logiciel, ou de l'intégrer aisément sur un site web (JavaScript) sans rajouter de charge de travail conséquente. »*