



Licence 2 Informatique

Rapport du projet IF04

Jeu de Taquin

Réalisé par :

**ARNOULT Simon, MEKHILEF Wissame, OUSSAD Jihad,
RETY Martin**

13 mai 2015

Résumé

Nous avons le plaisir de vous présenter notre travail sur ce projet. Durant la deuxième année de la l2 Informatique à l'Université d'Orléans, nous avons travaillé en groupe de quatre sur un projet de résolution de Taquin.

Table des matières

1	Introduction	5
1.1	Cahier des charges	5
2	Etude	6
2.1	Analyse de faisabilité	6
2.2	Conception UML	7
3	La gestion de projet	8
3.1	Le travail de groupe	8
3.2	Le diagramme de Gantt	8
4	Phase de développement	10
4.1	L'architecture	10
4.2	Le codage	10
4.3	Junit et Benchmark	10
4.3.1	Utilisation des benchmarks	10

5	Analyse et Conclusion	12
5.1	Analyse des Benchmark	12
5.1.1	Jeu simple	12
5.1.2	Jeu moyen	12
5.1.3	Jeu compliqué	12
5.2	Conclusion	12
6	Resources utilisées	14

Table des figures

3.1	Diagramme de GANTT	9
4.1	Exemple	11

Chapitre 1

Introduction

Nous allons ensemble, aborder quatres points principaux dans ce rapport :

- L'étude du projet
- L'organisation de travail au sein du groupe
- Le développement du code
- L'analyse des algorithmes.

1.1 Cahier des charges

Chapitre 2

Etude

L'étude du projet fut une première étape importante pour se mettre dans une bonne dynamique de groupe. Dès la première semaine nous avons réalisé une version jouable du Taquin, puis nous avons travaillé sur cette version durant toute la première phase d'essai.

Rapidement il nous a fallu faire des essais sur ce jeu pour qu'il puisse se résoudre algorithmiquement. Nous avons chacun travaillé de notre côté sur nos idées pour pouvoir en tester un plus grand nombre, mais tout en restant en contact régulièrement pour avancer ensemble.

Cette manière de répartir les tâches nous a permis de se rendre compte des difficultés que l'on allait rencontrer plus rapidement.

2.1 Analyse de faisabilité

L'analyse de faisabilité fut un moment qui a duré du début du projet jusqu'aux vacances de février, durant cette phase chacun travaillant sur des fonctionnalités différentes, nous avons pu voir ou étaient les problèmes dans notre architecture de départ.

Ces tests ont été très divers. Nous avons travaillé sur les fonctionnalités VT100 du terminal et la récupération des touches tapées par l'utilisateur. Mais nous avons aussi créé des algorithmes pour essayer de résoudre le Taquin.

Tous ces tests nous ont permis au mois de février d'avoir une idée claire de l'architecture du projet.

2.2 Conception UML

Cette analyse de février a permis d'aboutir au diagramme UML suivant, diagramme qui n'a pas beaucoup évolué jusqu'à la version finale.

Chapitre 3

La gestion de projet

La gestion de projet est à la base de tout projet, et encore plus quand il se fait avec une équipe de quatre personnes.

3.1 Le travail de groupe

La gestion de projet fut au coeur de nos préoccupations avant même que le projet ne démarre, nous avons cherché à créer une équipe dynamique. Nous nous sommes donc vus régulièrement dans les salles de l'Université. Malgré cela il nous a fallu mettre en place des moyens dédiés pour faciliter le travail et éviter une dégradation de l'entente.

Dès la première semaine, nous avons mis en place un dépôt git sur le site de l'hébergeur GitHub, ce dépôt nous a permis d'avoir le réflexe de l'utiliser même si nous avons rencontré quelques problèmes, il nous a permis à chacun de voir l'avancement du projet.

Cependant un tel dépôt ne répond pas à la question de la communication, chacun habitant une ville différente, nous avons donc communiquer via Skype pour remédier à ce problème.

3.2 Le diagramme de Gantt

Nous avons utilisé la technique du diagramme de GANTT pour vérifier en interne que le projet avançait convenablement. Ci-dessous vous pouvez voir ce diagramme.

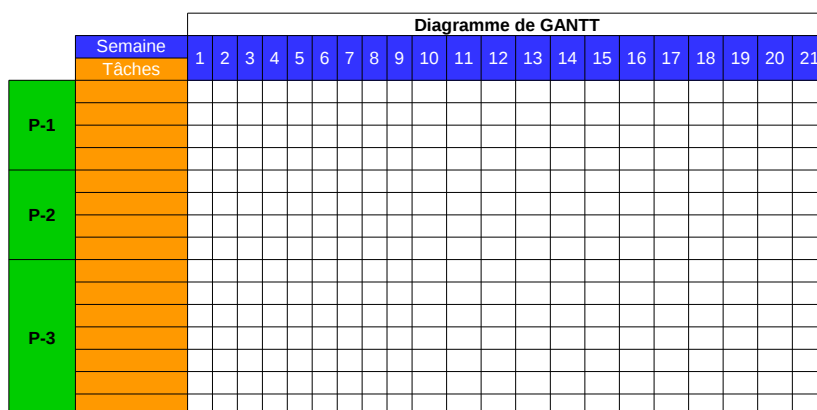


FIGURE 3.1 – Diagramme de GANTT

Chapitre 4

Phase de développement

La phase de développement...

4.1 L'architecture

On explique les choix qu'on a fais

4.2 Le codage

4.3 Junit et Benchmark

4.3.1 Utilisation des benchmarks

Nous avons utilisé les benchmarks pour améliorer l'efficacité de l'algorithme de solution. En effet, nous avons remarqué entre autres que la taille de l'ensemble incomplet avait une importance dans le temps de résolution. Nous avons donc créé un test avec plusieurs tailles d'ensemble pour voir laquelle permettait une resolution rapide.

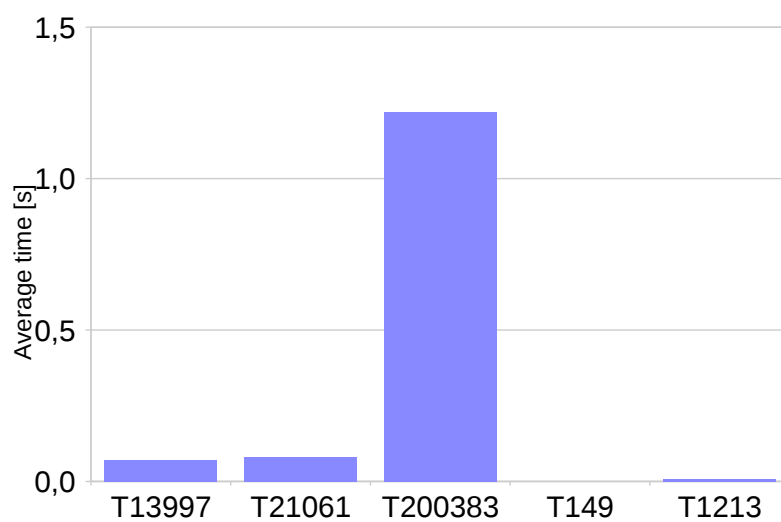


FIGURE 4.1 – Exemple

Chapitre 5

Analyse et Conclusion

5.1 Analyse des Benchmark

5.1.1 Jeu simple

5.1.2 Jeu moyen

5.1.3 Jeu compliqué

5.2 Conclusion

Bibliographie

Chapitre 6

Resources utilisées

- JUNIT
- JAVA
- Dia
- Eclipse
- Argouml
- L^AT_EX(sous kile)
- Google Drive
- Kate
- Git, GitHub