Thomas BROGGI

Maud CHOLEZ

Sophie DA COSTA

RAPPORT DE PROJET 2048

*UFD61 Mathématiques et informatique*

*Ingénierie Logicielle*

*Enseignant : M CASTAGNOS Sylvain*

*L3 MIASHS – Parcours Sciences Cognitives*

*2016 – 2017*

Sommaire

[INTRODUCTION 3](#_Toc480379476)

[RECETTE INFORMATIQUE 4](#_Toc480379477)

[DESCRIPTION DE L’APPLICATION 5](#_Toc480379478)

[DIAGRAMMES 6](#_Toc480379479)

[REPOSITORY 7](#_Toc480379480)

[DOCUMENTATION 8](#_Toc480379481)

[(TESTS REALISES) 9](#_Toc480379482)

[RETROSPECTIVE ET CONCLUSION 10](#_Toc480379483)

[(ANNEXES) 11](#_Toc480379484)

# INTRODUCTION

Dans le cadre de notre 3ème année de licence en Mathématiques et Informatique Appliqués aux Sciences Humaines et Sociales (MIASHS), parcours Sciences Cognitives, nous avons eu pour projet de créer le jeu 2048. Ce projet devait être réalisé dans le cadre de l’UFD41 Mathématiques et Informatique, avec comme enseignant M. CASTAGNOS Sylvain.

Tout d’abord, nous allons nous intéresser à ce qu’est le jeu 2048. Il a été conçu en mars 2014, il est disponible sur ordinateur, tablette et smartphone. Il s’agit d’un jeu représenté par une grille de 16 cases. Le but du jeu est de réunir des tuiles de même valeur, afin d’atteindre la tuile portant la valeur de 2048. Les chiffres disponibles sont des puissances de 2 : 2, 4, 8, 16, 32, …, 2048, … Après chaque déplacement réalisé, une nouvelle tuile est générée, avec comme chiffre 2 ou 4. Une fois l’objectif atteint, il est possible de stopper la partie ou alors de continuer.

Le projet s’est déroulé en deux principales étapes :

* La mise en place du projet : Création de différents diagrammes pour gérer les différentes classes et élaboration du planning pour gérer les étapes de la conception.
* La conception des différentes classes, méthodes, fonctions nécessaires pour atteindre les achievements.

Dans ce rapport, nous allons décrire les étapes qui nous ont permis de mener à bien ce projet. Ce qui permettra de comprendre nos choix et de faciliter leur expérience sur notre application.

# RECETTE INFORMATIQUE

Liste achievements atteints, explication, planning

## Achievements à atteindre

Dans un premier temps, nous allons revoir les achievements que nous avions à réaliser, dans l’ordre de notre choix.

1. *Jeu en solo dans la console Netbeans, qui a déjà été réalisé en TP noté de L2,*
2. Jeu en solo avec interface graphique au clavier,
3. Affichage fluide dans l’interface graphique avec des Threads,
4. Possibilité de demander de l’aide à une I.A. pour le prochain coup,
5. Possibilité de laisser l’I.A. jouer toute seule jusqu’à 2048 (ou interruption à l’aide d’un bouton stop),
6. Possibilité de recharger la partie qui était en cours lors du précédent lancement du jeu,
7. Possibilité de voir votre classement parmi les joueurs de l’application,
8. Jeu multi-joueurs.

## Achievements atteints ou en cours

### Jeu en solo dans la console Netbeans

En ce qui concerne le premier achievement de ce projet, nous avons décidé de garder l’intégralité du code fourni dans la correction du TP noté de l’année passée. En effet, nous avons analysé l’ensemble des classes données, et nous avons bien compris les méthodes utilisées, alors nous n’avions aucune raison de ne pas l’utiliser.

### Jeu en solo avec interface graphique au clavier

Cet achievement est le premier que nous avons décidé de faire. Il nous paraissait important de le terminer. Pour jouer à 2048, c’est plus simple d’avoir une interface graphique nous permettant de savoir où l’on se trouve dans la partie. Cet achivement a été réalisé par Thomas.

### Affichage fluide dans l’interface graphique avec des Threads

La seconde étape que nous voulions accomplir était celle de l’intégration des Threads dans l’interface graphique. Ainsi, ce sont Sophie et Thomas qui se sont penchés dessus. Mais après plus essais infructueux, nous avons décidé de le laisser de côté, afin de pouvoir avancer sur d’autres points. Il semblerait que le fait que nous n’avons pas réussi cette étape serait lié à un manque de compétences pour le graphisme en JAVA, notamment avec JavaFX.

### Possibilité de demander de l’aide à une I.A. pour le prochain coup

Pour cette partie, c’est Maud qui avait décidé, en premier, de s’y attarder. Il fallait tout d’abord commencer à réfléchir à quelle heuristique nous voulions utiliser pour que l’I.A.conseille le prochain coup. De ce fait, nous sommes arrivés à la méthode suivante :

* L’I.A. cherche deux tuiles voisines ayant la même valeur pour les fusionner. Dans le cas où il existe plusieurs couples de tuiles voisines identiques, l’I.A. choisit le couple pour lequel la valeur est la plus élevée. De ce fait, le coup effectué par l’I.A. permet de regrouper les tuiles les plus intéressantes.
* Dans le cas où aucune fusion n’est possible, nous décidons de favoriser une direction plutôt qu’une autre : par exemple l’I.A. décide de faire un coup vers la gauche, mais dans le cas où ce n’est pas possible, elle déplace les tuiles vers le haut, ou vers la droite, ou vers le bas.

Cet achievement a été repris par Thomas.

### Possibilité de laisser l’I.A. jouer toute seule jusqu’à 2048

Dès le départ, nous ne pensions pas atteindre cet objectif. En effet, nous savons que nous avons quelques lacunes avec la mise en place d’une intelligence artificielle, donc qu’il serait difficile d’atteindre le niveau précédent, ainsi que celui-ci. Nous avons tout de même pris le temps de nous poser quelques questions à ce sujet. C’est pour cela que nous savons que si l’I.A. pour le prochain coup venait à fonctionner, il n’y aurait pas beaucoup de choses à ajouter pour réaliser cette étape.

En effet, une fois que l’I.A. est en mesure de donner un conseil pour le prochain déplacement ; pour pouvoir jouer une partie entière, il suffirait d’appeler, tant que la partie n’est pas terminée, le programme pour prévoir le coup suivant, et une fois ce coup connu, il faudrait établir le déplacement associé. Il resterait ensuite à mettre en place le bouton permettant l’interruption de la partie jouée par l’intelligence artificielle.

### Possibilité de recharger la partie en cours

Cet achievement est atteint grâce à Sophie et Thomas. Après avoir rencontré quelques problèmes dans la mise en place de la sérialisation, nous avions laissé le code de côté. Pour finalement y revenir est réussir à la boucler plus facilement.

### Possibilité de voir le classement des joueurs de l’application

Objectif atteint grâce à Thomas.

### Jeu multi-joueurs

Nous avons décidé dès le début du projet que nous ne ferions pas cet objectif en priorité, car nous savions que nos compétences en programmation Java n’étaient pas suffisamment grandes pour y parvenir.

## Planning de notre projet

# DESCRIPTION DE L’APPLICATION

Installation, fonctionnalités, capture d’écran de l’interface graphique pour expliquer le fonctionnement

# DIAGRAMMES

Diagramme de cas d’utilisation

Diagramme de classes

Diagramme de séquence

# REPOSITORY

GitHub et Trello

# DOCUMENTATION

JavaDoc

# (TESTS REALISES)

# RETROSPECTIVE ET CONCLUSION

# (ANNEXES)