

# Rapport projet HASHIWOKAKERO

DELAR Emmanoe

18 avril 2017

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Implémentation de la bibliothèque Game</b>	<b>2</b>
2.1	Structures et énumérations . . . . .	2
2.2	Programmation du jeu en version texte . . . . .	2
2.3	Extension de la lib "caprice du client" . . . . .	2
2.4	Solveur . . . . .	2
<b>3</b>	<b>Version graphique : SDL2</b>	<b>2</b>
<b>4</b>	<b>Conclusion</b>	<b>3</b>

# 1 Introduction

Lors de ce projet, en petit groupe, nous avons pour objectif de développer un programme en langage C se basant sur le principe du jeu Hashiwokakero. Au bout de la deuxième séance, mon groupe s'est décomposé et je me suis retrouvé seul.

Afin de finir dans les délais impartis, j'ai dû organiser le travail en plusieurs petits modules (implémentation de la librairie Game, version 1, version 2, tests, solveur, version graphique SDL2, portage Android), dans cet ordre de développement. L'une de mes priorités était l'application à produire un code optimisé et simple à lire, j'ai aussi porté une attention particulière à ce que mon archive soit propre et bien organisée.

Tout au long de l'année, nos enseignants nous ont présentés plusieurs outils efficaces et nécessaires afin d'obtenir un programme optimal et un code de qualité. L'objectif étant de maîtriser la base de la programmation en langage C. Dans ce rapport, je vais vous présenter ces outils et les différentes étapes qui m'ont permis de réaliser ce projet.

## 2 Implémentation de la bibliothèque Game

Une bibliothèque en informatique est un regroupement de fonctions pouvant être utilisé par différents programmes. Dans notre cas, un fichier "header" (game.h) nous présentait ces fonctions. Ce fichier avait la valeur d'un cahier des charges qui formulait les besoins d'un client. On avait pas à le modifier.

### 2.1 Structures et énumérations

Le langage C nous permet de créer nos propres types de variables. Pour la bibliothèque Game, on en a utilisé deux sortes : les structures et les énumérations. La structure "game.s" de ma bibliothèque contient des variables de différents types. Chaque variable stocke les informations nécessaires afin de générer un jeu.

### 2.2 Programmation du jeu en version texte

### 2.3 Extension de la lib "caprice du client"

### 2.4 Solveur

## 3 Version graphique : SDL2

## 4 Conclusion

Sur une période de deux semestres, les différents programmes que j'ai développé, répondent bien au cahier des charges imposé par le client. À l'aide de différents outils (gdb, valgrind, git, gcov, CMake) et à l'organisation sous forme de plusieurs petits modules, j'ai pu mener à bien ce projet.

Toutefois, certains programmes proposés ne sont pas les plus sophistiqués faute de temps. Par exemple pour le solveur, l'algorithme utilisé est de type naïf (Brute-force). En effet, pour résoudre une instance, ce dernier teste tous les cas un à un jusqu'à ce qu'il trouve une bonne combinaison. Cependant il peut s'avérer efficace sur les petites instances de jeu. Une des solutions face à ce problème est d'écrire un programme qui joue un maximum de coup obligatoire avant le solveur. Il manque donc un travail d'optimisation.

En prenant du recul, je peux dire que le projet m'a apporté un bagage supplémentaire. J'ai consolidé mes connaissances générales sur le langage C et appris à coder à un niveau supérieur.