Rapport projet HASHIWOKAKERO

DELAR Emmanoe, Amaury Tressieres

16 avril 2017

Table des matières

1 Introduction

Lors de ce projet, réalisé en petit groupe normalement, nous avions 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 avant le temps imparti, dès le début 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 propre 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 enseigants nous ont présentés plusieurs outils éfficaces et nécessessaires afin d'obtenir un programme optimal et un code de qualité, l'objectif étant de maitriser le langage C.
Dans ce rapport je vais vous présenter les differentes étapes qui m'ont permis de réaliser ce projet.
Ci-dessous l'exemple d'un jeu résolu :

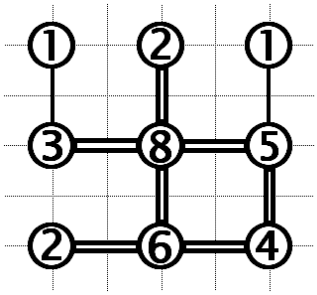

FIGURE 1 – Jeu résolu.

2 Programmation du jeu en version texte

2.1 Implémentation de la librairie game

2.2 Structure

Il tait une fois dans une lointaine galaxie...

2.3 Extension de la lib "caprice du client"

2.4 Solveur

3 Version graphique : SDL2

4 Conclusion

Sur une période de deux semestres, le programme que j'ai développé, répond bien au cachier des charges imposé par le client. À l'aide de différents outils (gdb, valgrind, git, gcov, CMake) et aussi à l'organisation sous forme de plusieurs petit module, j'ai pu mener à bien ce projet. Toutefois, le programme proposé n'est pas le plus sophistiqué faute de temps. Par exemple pour le solveur, l'algorithme ultilisé est de type naif (Bruteforce). En effet, pour résoudre une instance, ce dernier teste tous les cas 1 par 1 jusqu'à ce qu'il trouve une bonne combinaison cependant il peut s'avérer éfficace sur les petites instances de jeu. Une des solutions face à ce problème est d'écrire un programme qui s'éxcute juste avant le solveur afin de jouer un maximum de coup obligatoire.