Rapport projet HASHIWOKAKERO

DELAR Emmanoe

19 avril 2017

Table des matières

1 Introduction

Lors de ce projet, en petit groupe, 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 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 éfficaces et nécessessaires afin d'obtenir un programme optimal et un code de qualité. L'objectif étant de maitriser 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 Présentation du jeu

2.1 Règle du jeu HASHIWOKAKERO

Hashiwokakero est un jeu de logique qui se joue sur une grille rectangulaire. Dans cette grille, on retrouve des cercles entourant des chiffres allant de 1 8. Ces cercles sont appelés îles (nodes ou noeuds dans notre cas) et les chiffres sont les degrés de chaque nodes. Le but du jeu est de relier toutes les îles en un seul groupe en créant une srie de ponts (simples ou doubles) entre les îles. Le jeu étant terminé lorsque chaque node est relié aux autres avec un nombre de ponts égal son degré. Ci-dessous un exemple du jeu résolu :

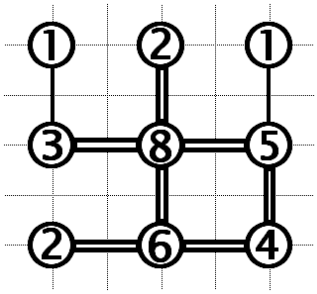

Figure 1 – Jeu rsolu.

2.2 Caprice du client

Dans le jeu Hashiwokakero "officiel", il y a au plus 2 ponts entre 2 îles et les ponts sont soit verticaux, soit horizontaux. Dans la "v2" le nombre de ponts variera et ont pourra avoir des ponts obliques.

3 Implémentation de la bibliothéque Game

Une bibliothèque en informatique est un regroupement de fonctions pouvant être utilisé par différrents 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.

3.1 Structures et énumérations

Le language C nous permet de créer nos propre 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 varialbles de différents types. Chaque variable stocke les informations nécessaires afin de generer une grille de jeu. On y retrouve : le nombre de noeuds, un tableau de taille dynamique qui stocke ces noeuds, un tableau à deux dimensions (x et y) qui gère les ponts reliant les différents noeuds, un entier représentant le nombre maximal de ponts et un autre le nombre de directions autorisées pour la partie.

3.2 Programmation du jeu en version texte

3.3 Extension de la lib "caprice du client"

3.4 Solveur

4 Version graphique : SDL2

5 Conclusion

Sur une période de deux semestres, les différents programmes que j'ai développé, répondent bien au cachier 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 ultilisé est de type naif (Bruteforce). 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 éfficace 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 supplmentaire. J'ai consolidé mes connaissances générales sur le language C et appris à coder à un niveau supérieur.