



ECOLE INTERNATIONALE DES SCIENCES
ET DU TRAITEMENT DE L'INFORMATION

Projet informatique : MASTERMIND

Auteurs :

Gauthier Petitjean

Tom Soulage

Etienne Tournier

Professeur :

Awa Diattara



Table des matières

| | | |
|----------|-------------------------------|----------|
| 1 | Introduction | 2 |
| 2 | Structuration du code | 3 |
| 2.1 | Partie Code | 3 |
| 2.1.1 | Le code en détails | 3 |
| 2.2 | Partie IA | 4 |
| 2.2.1 | Le code en détails | 4 |
| 3 | Choix algorithmique | 5 |
| 3.1 | Les points positifs | 5 |
| 3.2 | Les points négatifs | 5 |
| 4 | Répartition du travail | 6 |
| 5 | Bilan du projet | 7 |

Projet informatique: Mastermind

18 janvier 2019

1 Introduction

Le mastermind est un jeu de société, créé dans les années 70. Le jeu se joue à deux joueurs, l'un des joueurs doit choisir un code de quatre couleurs, l'autre doit tenter de le trouver en 10 coups. Pour aider ce dernier dans sa tâche, le joueur qui a composé le code doit donner le nombre de couleurs bien placées et le nombre de couleur mal placée à chaque essai de l'autre joueur.

Ainsi, le joueur qui cherche le code doit faire appel à son esprit de réflexion pour trouver le code. Il existe des algorithmes qui peuvent résoudre ce jeu en un certain nombre de coups. L'un des meilleurs le réussit en 5 coups.

Notre projet consiste à coder ce jeu avec deux modes : joueur contre joueur (PVP) et joueur contre l'ordinateur (PVE). Pour ce dernier, il nous faut donc rechercher et coder les algorithmes les plus puissants pour trouver les codes proposés par les joueurs, en moins de tours possibles.

2 Structuration du code

Notre code est composé de 4 grandes parties :

- Code : traitement des codes proposés afin de donner une réponse adéquate, et création de listes utiles (tous,toutes-reponses) à la recherche du code dans la partie IA
- IA : algorithmes de recherche d'un code particulier (il en existe 2 : naïf et Knuth) ainsi que des algorithmes dans le quel le joueur humain doit trouvé le code secret généré aléatoirement par l'ordinateur
- Pvp : mode de jeu humain contre humain
- Interface : aspect interface/affichage
- Run : fichier exécutable, lancement du jeu

2.1 Partie Code

La partie code rassemble toutes les données du jeu. Elle comprend tous les codes possibles, les couleurs possibles, puis différentes fonctions tels que des fonctions de contrôle (contrôle de saisie valide, des pions à la bonne ou mauvaise place etc...).

2.1.1 Le code en détails

Dans cette partie, nous avons défini les différentes structures et fonctions utilisées dans ce projet :

- les pions, les codes, les couleurs possibles, tous les codes mais aussi les réponses possibles
- la fonction "compare" qui compare deux codes
- les fonctions de codage "string-of-code" et "code-of-string" pour changer le code en chaîne de caractère pour le traitement et sa fonction inverse. En effet, nous avons attribué à chaque couleur un chiffre afin de faciliter le traitement dans la suite du code.
- la fonction qui génère la réponse à donner à l'utilisateur par rapport à sa proposition.

2.2 Partie IA

La partie IA comporte différents modes, certains génèrent des codes afin que le joueur humain essaie de trouver le code secret (humainDevine) alors que d'autres essayent de déchiffrer le code secret proposé par le joueur humain (l'algorithme naïf ou de Knuth).

2.2.1 Le code en détails

Les différents modes de jeu :

- Humain devine : l'utilisateur doit deviner le code généré par l'ordinateur de façon aléatoire, l'ordinateur suit les règles du mastermind.
- Knuth : l'ordinateur tente de trouver le code en suivant l'algorithme de Knuth. Ce dernier consiste à choisir le code dont le score est le plus approprié. Le score suit la méthode connue minMax. Ainsi, l'algorithme résout le code en 5 coups maximum .
- Naïf : cet algorithme dit "naïf" suit le même raisonnement que l'algorithme de Knuth, avec une vision plus simpliste. En effet, la différence entre les deux algorithmes est le choix du prochain code. Ici, le prochain code est le premier code de la liste issue d'un filtrage. Il trouve le code en 6-7 coups en moyenne. Il est donc moins performant que celui de Knuth, bien qu'il soit plus rapide.

3 Choix algorithmique

3.1 Les points positifs

Nous avons fait plusieurs choix que nous qualifierons de positifs dans notre projet tels que :

- Nous avons créé le mode humain contre humain (facultatif), où deux joueurs peuvent s'affronter. Nous voulions que la fonction première du jeux mastermind (jouer contre un ami) reste disponible dans notre projet afin que celui-ci soit complet.
- Nous avons maximisé l'emploi des "List.* ", pour éviter la récursivité comme il est recommandé dans le sujet.
- Nous avons créé un mode de jeu dans lequel le joueur donne les réponses attendues (nombre de pion bien placé et mal placé) à l'ordinateur en fonction de ses essais. En cas d'erreur de la part du joueur, celui-ci perd la manche et l'ordinateur gagne un point.
- A propos du système de comptage de points, celui-ci favorise la rapidité pour trouver le code secret. Plus le joueur gagne en un nombre de tentatives faibles, plus il gagnera de points.

3.2 Les points négatifs

Nous avons choisi de ne pas réaliser d'interface graphique car cela nous aurait pris beaucoup trop de temps, alors qu'une interface terminale suffit largement à ce type de projet, selon nous. C'est pourquoi nous avons opté pour une interface plus simple, mais tout autant esthétique.

De plus, nous aurions pu réaliser un dernier type d'algorithme de recherche. Malheureusement, dans le temps imparti, cela nous aurait été beaucoup trop dur.

4 Répartition du travail

Gauthier PETITJEAN :

- Programmation de la partie Code
- Programmation des IA
- Programmation de la partie Run

Tom SOULAGE :

- Programmation de la partie Interface
- Programmation de la partie pvp
- Création et entretien du Git

Etienne TOURNIER RIGAUDY :

- Programmation de la partie IA
- Programmation de la partie Run
- Rédaction du rapport

5 Bilan du projet

— Gauthier PETITJEAN :

Ce projet a été très enrichissant d'un point de vue algorithmique car, comme l'année dernière, le projet est ce qui me fait comprendre comment fonctionne le langage. Grâce à ce projet, je comprends réellement mieux le fonctionnement et la manipulation des listes. De plus, ce projet m'a permis de comprendre comment fonctionne la récursivité (terminale/non terminale). Cependant, au début du projet, avoir des fonctions imposées (entre le sig...end) m'a perturbé car j'étais obligé de suivre une ligne conductrice. Maintenant, j'ai compris pourquoi il a fallu suivre ce chemin : tout était plus facile lors de la conception et de la réalisation du module IA, avec ce que j'avais déjà effectué.

— Tom SOULAGE :

Ce projet m'a permis d'améliorer mon apprentissage du langage OCAML, notamment pour la manipulation des listes, très utilisées dans les algorithmes de l'IA. L'avantage du mastermind est qu'il possède des règles simples, en revanche les algorithmes de l'IA, notamment l'algorithme de Knuth a été une partie du projet assez longue et complexe à réaliser.

— Etienne TOURNIER RIGAUDY :

Ce projet était intéressant d'une part pour le fait que le jeu du mastermind est connue, cela m'as permis de prendre rapidement le projet en main car je connaissais déjà les règles. Je pense cependant que le guidage durant le projet est difficile a prendre en main , on ne voyait pas toujours où il voulait en venir.