École Internationale des Sciences du Traitement de l'Information

Projet Semestre 3 : MasterMind

Programmation Fonctionnelle

J. Dingley, M. Perie, V. Maillot

Année Universitaire 2018-2019

1 Introduction

FIGURE 1 – Mastermind

Le projet Mastermind a pour but de coder à l'aide du langage Ocaml le jeu du même nom. Pour ce faire, nous disposions de cinq semaines réparties du mois de Décembre 2018 au mois de Janvier 2019. Chaque groupe étant composé de trois personnes, le nôtre comprend : Mathieu Périé, Jesse Dingley et Victor Maillot.

Ce rapport présentera sous divers points notre projet informatique du troisième semestre consacré au Mastermind.

2 Structuration du code

Notre code se décompose sur quatre fichiers différents :

1. Le module Code : ce module définit tous les paramètres et les caractéristiques liées aux codes du Mastermind.
2. Le module IA : ce module définit la stratégie de recherche de codes que va adopter l'intelligence artificielle.
3. Le module mastermind : ce module permet de gérer les différents paramètres d'entrée de la fonction mastermind qui est liée aux règles de la partie.
4. Le module UI : ce module permet la représentation du Mastermind sous la forme d'une interface graphique, ainsi que l'éxecution de l'ensemble des parties en appelant les fonctions des autres modules.

De manière générale nous avons essayé de découper nos fonctions en fonctions élémentaires afin de simplifier la compréhension, l'implémentation, la correction, ainsi que l'utilisation de celles-ci. Cette structuration nous a permis à de maintes reprises de corriger ou améliorer notre code, ce qui a été plus compliqué à mettre en œuvre dans le module UI en raison de l'utilisation superposée des fonctions traitant du fonctionnement du jeu avec les fonctions graphiques affichant le déroulement de ce dernier.

3 Choix algorithmiques

3.1 Intelligences artificielles

Débutons par le module Code et IA dont la structure nous étaient imposées, limitant ainsi la liberté de nos choix algorithmiques. Dans ces deux modules nous avons seulement pu personnaliser notre code dans l'implémentation des intelligences artificielles utilisées. La différence notable entre les deux intelligences artificielles se trouve dans la fonction choix qui permet de déterminer le prochain code que l'IA va proposer : l'IA naïve prend le premier code de la liste des codes encore jouables tandis que Knuth propose un choix optimal.

Nous avions envisagé une première version de l'algorithme "Five guess" de Donald Knuth. Elle consistait en l'élaboration d'une unique liste composée de codes, d'une liste de couple associé, comportant eux-mêmes une réponse ainsi que le nombre d'itérations de cette réponse en fonction du code et de la liste des codes possibles. Au vu de la complexité spatiale et temporelle, nous avons décidé de créer une seconde version beaucoup plus efficace se basant sur l'utilisation de la fonction "filtre".

Néanmoins notre implementation du "Five guess" ne permet de jouer seulement avec des codes composés de deux à six pions. En effet nous n'avons pas gérer le cas où le nombre de pion serait égal à un, car l'intérêt de Knuth sur un code de un pion est limité. Pour ce qui est de jouer avec plus de six pions, un Stack Overflow est obtenu mais le temps d'exécution nécessaire serait de toute manière trop long pour être jouable sur les machines de l'EISTI.

3.2 Interface graphique

Également, nous avons choisi de représenter le Mastermind sous la forme d'une interface graphique plutôt que d'une interface textuelle. Ce choix a été déterminé par la volonté de créer un jeu intuitif et compréhensible. De plus, cela nous a permis de découvrir l'utilisation du module Graphics. Malgré les quelques difficultés notamment liées à l'injection des IA au sein de l'interface graphique, nous ne regrettons pas notre choix.

Par un soucis esthétique, il est préférable de jouer au maximum avec quinze tentatives à chaque tour afin d'éviter un affichage de mauvaise qualité.

3.3 Difficultés

Pour ce qui est de l'utilisation de formes au lieu des couleurs pour représenter les pions du Mastermind. Nous n'avons pas ajouter cette option car l'implémentation original de l'interface graphique était penser via les fonctions de gestion de couleur du module Graphics, plus intuitive à manipuler. L'ajout des formes revenait par conséquent à redéfinir l'intégralité du module UI. De plus, nous avons imposé une limite de dix-sept tentatives à chaque tour afin d'éviter un disfonctionnement de l'interface graphique.

4 Répartition du travail

La répartition de notre travail au sein de notre trinôme s'est effectué comme suit :

- Jesse Dingley : Responsable du module UI, il a également travaillé sur le module Code notamment sur le conversion des codes ainsi que la génération de ces derniers.
- Mathieu Périé : Responsable du module IA, il a aussi réalisé l'injection des IA dans l'interface graphique.
- Victor Maillot : Responsable du module Code, il a activement participé à l'élaboration de l'algorithme de Knuth.

Dans l'ensemble, notre groupe reste avant tout solidaire et soudé. Il faut noter que chaque personne a participé à chaque étape de l'élaboration du jeu et que personne n'a été laissé de côté. Il n'était pas rare de nous retrouver tous les trois ensemble dans le but de réfléchir à un problème pour débloquer une situation handicapante.

5 Utilisation de l'interface

Pour démarrer le jeu, il faut tapper dans le terminal les commandes (dans cet ordre) :

- rlwrap ocaml
- #use "mastermind.ml";;
- Mastermind.mastermind NomDuJoueur NbParties NbTentatives RepAuto;;
  exemple : Mastermind.mastermind "moi" 10 3 true;;

Pour la séléction des couleurs, il suffit de cliquer sur les couleurs à droite de l'interface.

Après avoir deviné ou pas le code, le jeu passera à la partie suivante, soit la partie où l'IA doit jouer contre vous. Vous devez donc définir le type d'IA (naïf ou Knuth) et puis choisir votre code secret.

Lorsqu'on doit rentrer un code caché, il faut faire attention à ne pas taper plus de fois que le nombre de pions par code, cela peut entrainer un "Unix.Unixerror (unix.EINTR, "select", "").

6 Bilan

Durant ce projet, nous avons rencontré divers obstacles, dont le plus dur à dépasser était le développement des intelligences artificielles et l'insertion dans l'interface de celles-ci. De plus, c'était assez compliqué pour trouver un moyen de stocker les données rentrées par l'utilisateur au niveau de l'interface.

De manière générale, nous sommes satisfaits de notre travail et nous avons pu rendre un travail presque complet. Nous avons fourni un travail régulier tout au long du projet, cependant plus intensif vers la fin du temps imparti.

Nous avons également pu élargir nos connaissances en OCaml, notamment dans la gestion des listes et le développement d'une interface graphique.