

# COMPTE RENDU PROJET INFORMATIQUE

## SEMESTRE 1 CPI 2

CYRIL ANANDHAN, ANTHONY JACQUELIN, KIMAN OULEBSIR

18 Janvier 2019

# Table des matières

1	Introduction . . . . .	3
2	Cahier des charges . . . . .	4
	2.1 Introduction au problème posé . . . . .	4
	2.2 Expression fonctionnelle du besoin . . . . .	4
3	Structuration du code . . . . .	5
4	Répartition du travail . . . . .	6
5	Critiques du projet . . . . .	7
6	Conclusion . . . . .	8

# **1 Introduction**

Dans le cadre du semestre 1 de CPI 2 à l'EISTI, nous avons dû réaliser un projet en Informatique. Ce premier projet de l'année en CPI 2 consiste en la programmation d'un jeu. Ce projet a pour but de faire coder les élèves sur des programmes longs, complexes, et riches en informations, pour nous obliger à analyser un problème et rendre la solution viable en code. Le jeu qui nous était demandé de recréer est le "Mastermind".

## 2 Cahier des charges

### 2.1 Introduction au problème posé

Le projet consiste en la réalisation du jeu "Mastermind", un jeu de société pour 2 joueurs, parut en 1976 en France dont le concepteur est Mordecai Meirowitz.

Ce projet a été réalisé par les élèves de CPI 2, Groupe 3 :

- Anthony Jacquelin
- Cyril Anandhan
- Kiman Oulebsir

### 2.2 Expression fonctionnelle du besoin

Voici les 2 possibilités possibles de déroulement d'une partie que nous devons récréer :

Le joueur humain doit deviner le code :

Le joueur non-humain crée un code parmi tous les codes possibles. Ensuite, pour chaque proposition de code faite par l'humain, il doit avoir les retours suivants :

- Le nombre de pions bien placés et le nombre de pions mal placés.
- Le rappel du plateau de jeu contenant toutes les propositions faites par le joueur humain avec les réponses associées.

Le joueur humain doit créer le code :

Le joueur non-humain doit faire des propositions de code, alors :

- Soit la machine calcule les réponses et tout s'enchaîne sans intervention de l'utilisateur
- Soit le joueur humain entre lui-même la réponse, laquelle nous devons vérifier pour détecter si une erreur a été faite ou non.

Afin de permettre au joueur non-humain de deviner le code, nous devons réaliser au moins un des deux algorithmes suivants :

- Un algorithme dit « naïf » qui consiste à choisir une combinaison aléatoire parmi la liste des couleurs possibles.
- Un algorithme proposé par Donald E. KNUTH en 1976 qui trouve le code en cinq coups au plus.

### 3 Structuration du code

Au delà du code qui nous a été imposé nous avons dû nous même penser à ajouter des fonctions pour compléter le code. En voici les principales ainsi que leur utilité :

- `liste_code_possible` : Cette fonction permet de donner la liste de tous les codes possibles.
- `createListCoul` : Cette fonction crée une liste de différentes couleurs.
- `createListEnt` : Permet de créer une liste d'entiers.
- `tuple_to_string` : Celle-ci donne le nombre de couleurs bien placées et couleurs non bien placées.
- `saisie_code` : Permet au joueur d'entrer une proposition de code au début de la partie.
- `decision_finale` : Donne le résultat à la fin de la manche.
- `generation_code_secret` : Génère le code de l'intelligence artificielle.
- `joueur_cherche` : Permet au joueur de chercher le code de l'intelligence artificielle.
- `reponse_manuel_joueur` : le joueur entre sa réponse
- `qui_commence_auto` : Détermine qui commence de manière automatique.
- `qui_commence_manu` : Détermine qui commence de manière manuelle.
- `qui_commence` : Permet d'appeler les 2 fonctions ci-dessus.
- `alternance` : alterne entre joueurs et/ou IA.
- `victoire` : Fonction qui indique qui a gagné la partie.
- `mastermind` : fonction pour lancer le jeu.

## 4 Répartition du travail

Concernant la répartition du travail au sein du groupe, nous avons choisi d'avancer ensemble, nous avons travaillé sur les mêmes parties en même temps. Nous avons choisi une certaine méthode pour l'utilisation de GitLab. Toutes les 3 fonctions environ que chacun faisait nous devions "push" sur GitLab notre travail. Ensuite après avoir "push" sur les branches module code et module IA nous avons tout "merge" sur la branche Master du projet.

## 5 Critiques du projet

A l'unanimité du groupe, ce projet dans son ensemble nous aura beaucoup fait progresser en Ocaml. Certes, il fut long et compliqué, mais c'est la recherche des solutions et les nombreuses erreurs que nous avons eu dans notre code qui nous ont fait progresser.

Afin de réaliser le projet, nous avons distingué 3 grandes parties dans notre projet.

La première partie étant le code imposé pour le projet, elle nous a été assez pénible à réaliser, car cela était complexe et nous devions rester dans ce qui a été demandé. De plus nous avons passé pas mal de temps dessus au début du projet.

Ensuite dans un second temps nous nous sommes occupé de toutes les fonctions que nous pouvions créer afin de compléter et d'améliorer les fonctions de celles imposées. Il s'agit par exemple des fonctions d'affichage, de calcul et de vérification des erreurs... Cette partie a été assez rapide à réaliser, cependant, c'était la première fois que nous devions faire une interface graphique en Oaml, nous avons donc dû apprendre sur Internet comment faire cela.

Enfin, et la plus complexe de ces 3 parties de réalisation du code a été la programmation des IA. L'intelligence Artificielle dite "naïve" a été assez naturelle et rapide à réaliser contrairement à l'algorithme proposé par Knuth. Cette dernière a été assez complexe et réaliser et compiler, cette dernière partie du code est celle qui aura donc pris le plus de temps.

Ce projet nous a permis de mettre en place un vrai travail de groupe, dans lequel les tâches étaient réparties. Le fait que le projet à réaliser soit un jeu que l'on connaissait nous a cependant aidé pour savoir dans quelles directions aller. L'utilisation de Git pour la réalisation de ce projet nous ayant été imposée nous avons eu du mal à tout configurer pour que tous les membres puissent déposer du code sur la plateforme. Avec le temps, nous avons réussi à bien comprendre le principe et l'utilité de la plateforme.

## 6 Conclusion

Ce projet Informatique pour le semestre 1 de CPI 2 nous aura permis de fortement consolider nos connaissances sur la programmation en OCaml. De plus, le fait d'avoir un but bien défini à ce projet lui donne tout son sens. Nous sommes globalement satisfait du Travail réalisé, même si avec une meilleure gestion du temps nous aurions pu être plus efficaces. Cependant, les tâches étaient bien réparties et l'ambiance de groupe était très bonne.

Une bonne expérience à renouveler !