

# *Mini-Projet n°1 :* *Mastermind*

**Table des matières**

Introduction .....3

Hypothèses de travail ou choix de programmation.....3

Difficultés rencontrées ou de problème(s) non résolu(s).....3

Jeu d'essais.....4

Conclusion.....6

## Introduction

Le Mini-Projet n°1 avait pour but de créer une version simplifiée du jeu Mastermind.

Dans le Mastermind le but du jeu est de découvrir le code mystère passé par un autre joueur, le code mystère est composé de quatre couleurs parmi les six couleurs disponibles : Rouge, Jaune, Vert, Bleu, Orange, Blanc, Violet, Fuchsia. Il existe aussi des pions blancs et rouges (ou noirs) utilisés pour donner des indications sur le code à trouver.

Une fois le code mystère dissimulé, l'autre joueur propose une séquence, le joueur qui a choisi le code mystère indique :

- Le nombre de pions de la bonne couleur bien placés en utilisant le même nombre de pions rouges.
- Le nombre de pions de la bonne couleur, mais mal placés, avec les pions blancs.

Le joueur gagne une manche s'il donne la bonne combinaison de pions avant la fin du nombre d'essais disponibles.

Pour ce mini-projet, nous devons programmer une variante simplifiée dans laquelle l'information concernant le nombre de pions de la bonne couleur, mais mal placés, ne sera pas donnée. Dans notre variante c'est l'ordinateur qui choisira une séquence mystère aléatoirement, de 4 pions parmi les 6 couleurs à notre dispositions : Rouge, Vert, Jaune, Magenta, Bleu et Cyan.

## Hypothèses de travail ou choix de programmation

Au début nous étions parti sur deux « switch » afin que l'ordinateur puisse faire un code mystère aléatoirement, nous avons réussi mais cela nous faisait beaucoup de variables, nous avons décidé de faire un tableau de 4 valeur qu'il choisirai au hasard. Nous nous sommes mis d'accord afin d'utiliser plusieurs variables booléennes pour ne pas nous perdre dans nos conditions. Nous avons décidé de travailler chacun notre tour afin d'avancer le plus vite possible, et nous nous consultations pour les problèmes ou les choix de programmations. Nous avons proposé tous les deux des hypothèses de programmation, et nous avons choisi la meilleur solution de programmation, d'après nous. Nous avons décidé de faire l'ensemble du programme en réglant les gros problèmes (problème de boucle infinie, de caractères en trop) faisant tout planter d'abord, puis, comme nous avons commencé assez tôt, de régler les petits détails par la suite.

## Difficultés rencontrées ou de problème(s) non résolu(s)

Lors de l'élaboration du programme nous avons été confrontés à divers problèmes, nous n'en avons pas rencontrés de majeurs, mais il était tout de même important de les corriger pour un fonctionnement parfait.

Parmi ces difficultés se trouvaient :

- La mise en couleur des « pions » dans le jeu, qui tout d'abord testé sur Windows ne fonctionnait pas, car le `#include` était placé au mauvais endroit (avant `using namespace std`), nous avons ensuite essayé sur linux où nous obtenions des astérisque (\*) en couleur à la place des caractères attendus. Nous sommes ensuite repassé sur Windows en plaçant correctement cette fois-ci le `#include` et il n'y eut aucun problème depuis.
- Lors de la demande de saisie du nombre d'essais souhaité par le joueur, si par inadvertance un ou plusieurs caractères étaient saisis à la place d'un nombre alors on se retrouvait dans une boucle infinie, suite à des recherches et plusieurs tests de commande nous avons finalement compris comment gérer ce cas particulier sans aucune incidence sur le programme.
- Lors de la demande de saisie de la séquence selon l'utilisateur, nous désirions détecter si l'utilisateur avait rentré une séquence valide ou non, et si elle n'était pas valide (c'est-à-dire qu'elle comportait une couleur non reconnue par le programme) nous devons faire en sorte que le nombre d'essai ne soit pas incrémenté, nous avons dû nous y reprendre à plusieurs fois avant que notre objectif soit atteint.
- Lors de la demande de saisie de la séquence encore une fois, là encore nous étions confrontés à un léger souci, car une séquence comporte 4 caractères, et si le joueur en entrait par exemple 8, alors il « entrait » sa séquence pour son prochain essai, bien sûr étant donné que cela peut ne pas être voulu, par exemple si l'on appuie sur une touche sans faire exprès avant de valider, il faut alors ignorer les caractères situés après les 4 premiers, là aussi, des recherches et différents tests furent nécessaires avant d'arriver à nos fins.

A côté de cela, nous n'avons pas rencontré de réelles difficultés, nous avons consacré énormément de temps à optimiser, aérer, organiser, rendre plus clair, et expliquer le programme.

### Jeu d'essais

Afin de tester correctement le programme, la séquence à trouver sera connue dès le départ pour ainsi traiter plus aisément chaque cas.

BUT DE L'ESSAI	VALEURS ENTRÉES	RÉSULTATS PRÉVUS
<u>Cas général</u> : Le joueur entre le nombre d'essai qu'il souhaite, puis tente de trouver la séquence aléatoire, et réussit à la trouver avant d'écouler son nombre d'essai, puis le joueur décide de quitter le jeu.	10 RJBV JBVR MCBJ BJJR Oui	- Résultat trouvé après 4 essais, sur un total de 10 possibles, le programme félicite le joueur et affiche la séquence à trouver. - Le programme se termine normalement

BUT DE L'ESSAI	VALEURS ENTRÉES	RÉSULTATS PRÉVUS
<u>Cas particulier #1</u> : Le joueur entre un nombre d'essai invalide, puis en entre valide, suite à ses essais il n'arrive pas à trouver la séquence aléatoire, puis décide de quitter le jeu normalement.	1 5 VBCM BBBB CBJM BVCM MCJB Oui	- Le programme annonce au joueur qu'il n'a pas trouvé la séquence après les essais accordés, et affiche la séquence qui était à trouver. - Le programme se termine normalement.
<u>Cas particulier #2</u> : Le joueur entre un nombre d'essai, puis dans la séquence qu'il propose il entre une couleur qui n'est pas dans le jeu, puis retente sa chance jusqu'à trouver la séquence aléatoire, puis quitte le jeu.	20 RRRA RRBB RBJV JVMC JRMC Oui	- Lorsque le joueur entre RRRA, le programme lui annonce que cette séquence ne peut être traitée, il lui demande donc de réessayer sans lui décompter un essai. - Le joueur trouve la bonne séquence et le programme se termine normalement.
<u>Cas particulier #3</u> : Le joueur joue une partie, puis souhaite en rejouer une autre sans quitter le programme.	5 BJVC MCBJ RRBJ BRBJ Non 5 BJMR BCMR BJCR BVCR Oui	- Le joueur joue une première partie qu'il gagne, indique qu'il ne veut pas quitter le jeu. - Le jeu se relance et génère à nouveau une séquence, le joueur la trouve au bout de 4 essais et quitte le programme normalement.
<u>Cas particulier #4</u> : Le joueur entre une séquence avec plus de couleurs que demandées.	5 RRRRBBBB RRJMVV Oui	- Le programme ignore tous les caractères après le 4e caractère, affiche en couleur la partie sélectionnée et la traite normalement. - Le joueur trouve la séquence qui était donc RRJM et quitte le programme normalement.

BUT DE L'ESSAI	VALEURS ENTRÉES	RÉSULTATS PRÉVUS
<u>Cas particulier #5</u> : Le joueur entre moins de 4 valeurs lorsque le programme lui demande la séquence qu'il a à proposer.	5 RRR R RRR VVV	<ul style="list-style-type: none"> <li>- Le programme attend la saisie du 4e caractère.</li> <li>- Si nécessaire, comme dans le cas précédent, le programme ignore les caractères en trop.</li> <li>- La séquence à trouver était ici RRRV et le joueur quitte le programme normalement.</li> </ul>
<u>Cas particulier #6</u> : Le joueur entre un ou plusieurs caractères lorsque le programme lui demande le nombre d'essais qu'il souhaite.	j jjj 5 RRRR RJVC Oui	<ul style="list-style-type: none"> <li>- Le programme indique une erreur, même si plusieurs caractères sont entrés, puis demande à nouveau à l'utilisateur d'entrer le nombre d'essai souhaité.</li> <li>- Le joueur trouve la séquence et quitte le programme normalement.</li> </ul>
<u>Cas particulier #7</u> : Le joueur entre un nombre d'essais inférieur à 5 ou supérieur à 20.	3 42	<ul style="list-style-type: none"> <li>- Le programme redemande au joueur de taper un nombre entre 5 et 20.</li> <li>- Le programme redemande au joueur de taper un nombre entre 5 et 20.</li> </ul>

## Conclusion

Ce mini-projet nous a montré le travail de groupe au niveau du codage en C++, jusqu'à présent nous avons travaillé sur des programmes chacun de notre côté, cela nous permet de nous rendre compte de notre futur métier lorsque nous travaillerons en groupe. Pour nous deux, nous sommes fiers de notre programme, nous avons atteint les objectifs que nous nous étions fixés. C'est l'un des programmes le plus complet que nous avons fait jusque-là.