



I] Le jeu du MASTERMIND : les règles

Voici l'un des jeux les plus célèbres de déduction : le MASTERMIND. Il s'agit d'un jeu alliant logique et déduction reposant sur un principe simple : retrouver une combinaison de 4 couleurs (parmi 8 possibles) en moins de 12 essais. Ce jeu de société se joue à 2 joueurs :

- Dans un premier temps, le joueur 1 choisit une combinaison de 4 couleurs (que l'on va appeler « combinaison secrète ») sans la montrer au joueur 2. Le joueur 1 peut choisir d'utiliser plusieurs fois la même couleur pour la définition de sa combinaison secrète.

Par exemple, le joueur 1 choisit la combinaison suivante : **ROUGE VERT NOIR BLEU**.

- le joueur 2 doit deviner cette combinaison secrète en proposant au joueur 1 une combinaison qui lui semble être la bonne ! Pour être aiguillé dans le choix des couleurs de sa combinaison, le joueur 2 reçoit à la fin de chacun de ses essais des « indices », donnés par le joueur 1. Ces indices correspondent aux nombres de couleurs bien placées, et aux nombres de couleurs correctes mais pas à la bonne place.

Si on reprend l'exemple ci-dessus, si à l'essai n°1, le joueur 2 propose la combinaison **VERT ORANGE MAUVE BLEU**, le joueur 1 doit lui indiquer qu'il y a une couleur bien placée (en l'occurrence ici, c'est le BLEU) et une couleur correcte mais pas à sa place (le VERT). Le reste des couleurs n'intervient donc pas dans la combinaison secrète.

La stratégie repose sur le fait de bien choisir les couleurs et leur emplacement en fonction des coups précédents déjà joués. Ce choix doit être bien judicieux d'autant plus que le nombre d'essais possibles par le joueur 2 est limité !

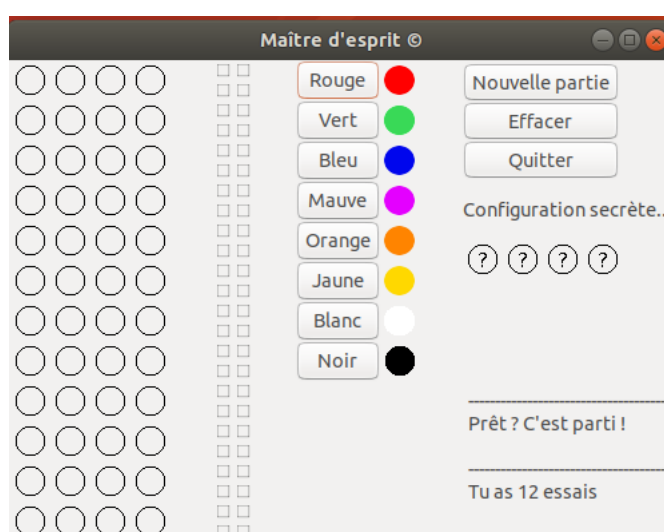
Une partie de Mastermind se termine si le joueur 2 a réussi à découvrir la combinaison secrète en 12 ou moins essais, et dans ce cas là, il remporte la partie. Sinon, c'est le joueur 1 qui l'emporte.

II] Présentation de l'interface graphique

Nous avons donc programmé ce jeu à l'aide de la bibliothèque GTK, ce qui nous a permis de réaliser une interface graphique. Voici une photo représentant le jeu dans sa version jeu de société :



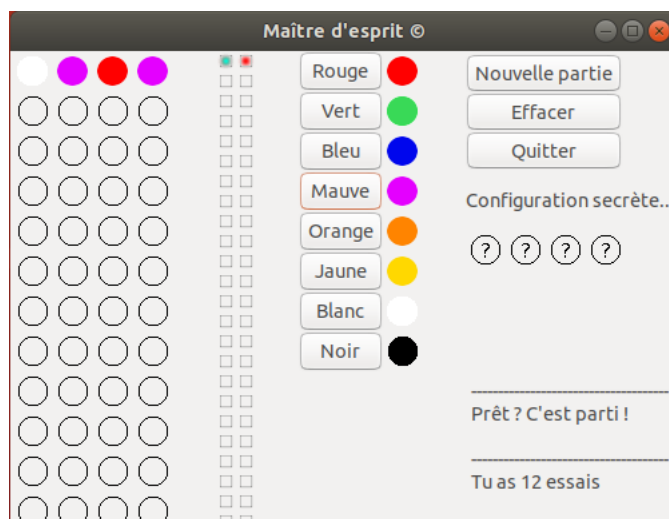
Et voici ce que ça donne sur notre interface graphique :



1) Les grilles de jeu et de vérification

Nous avons dans un premier temps représenté la grille de jeu du Mastermind par une grille (GtkGrid) d'images 12 x 4, qui correspond (sur la photo ci-dessus) à la grande grille composée de cercles vides. De plus, la grille de vérification, représentée sur l'interface par une « colonne » de carrés vides, est également une grille d'images.

Dans notre programme, une couleur bien placée est symbolisée par un carré qui s'illumine en rouge, et une couleur correcte mais pas à la bonne place est symbolisée par un carré qui s'illumine en bleu ciel :



2) Les boutons de couleurs

Nous avons représenté le choix des couleurs par 8 boutons (GtkButton). Chaque bouton est associé à une couleur. La grille de jeu se remplit de gauche à droite lorsque l'on clique sur ces boutons. Ainsi, si on reprend la photo ci-dessus, on a d'abord cliqué sur le bouton Blanc, puis Mauve, puis Rouge, et enfin Mauve.

3) Les boutons de menu

Nous avons mis à disposition 3 boutons, que l'on appellera les boutons du menu. Il s'agit des boutons Nouvelle partie, Effacer, et Quitter.

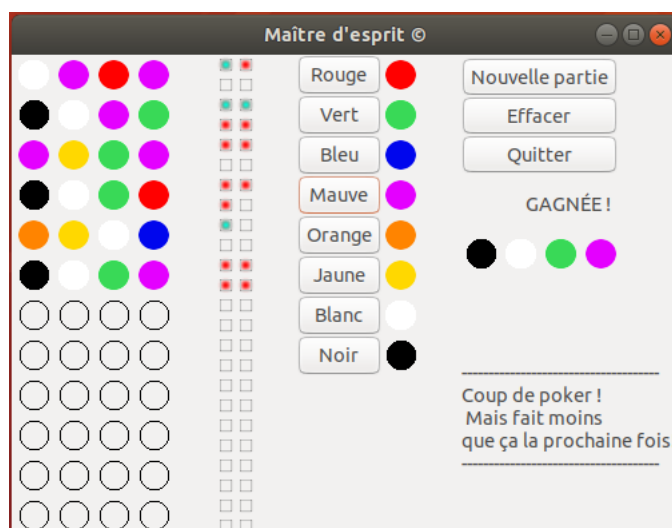
- Le bouton Nouvelle partie, comme son nom l'indique, permet au joueur de relancer une nouvelle partie de Mastermind en cliquant dessus. Ainsi, si en pleine partie le joueur décide de recommencer, alors il peut cliquer sur le bouton : les grilles de jeu et de vérification se vident et **une nouvelle combinaison secrète est élaborée.**

- Le bouton Effacer permet au joueur de modifier sa combinaison en effaçant la/les couleur(s) qu'il veut changer. Cela peut être utile si le joueur a cliqué sans faire exprès sur un bouton de couleur, ou bien si le joueur a changé d'avis en cours de route sur le choix de sa combinaison. Cependant, si le joueur a déjà choisi ses 4 couleurs, il ne peut pas faire de retour en arrière, car le programme fait passer le joueur directement à l'essai suivant !

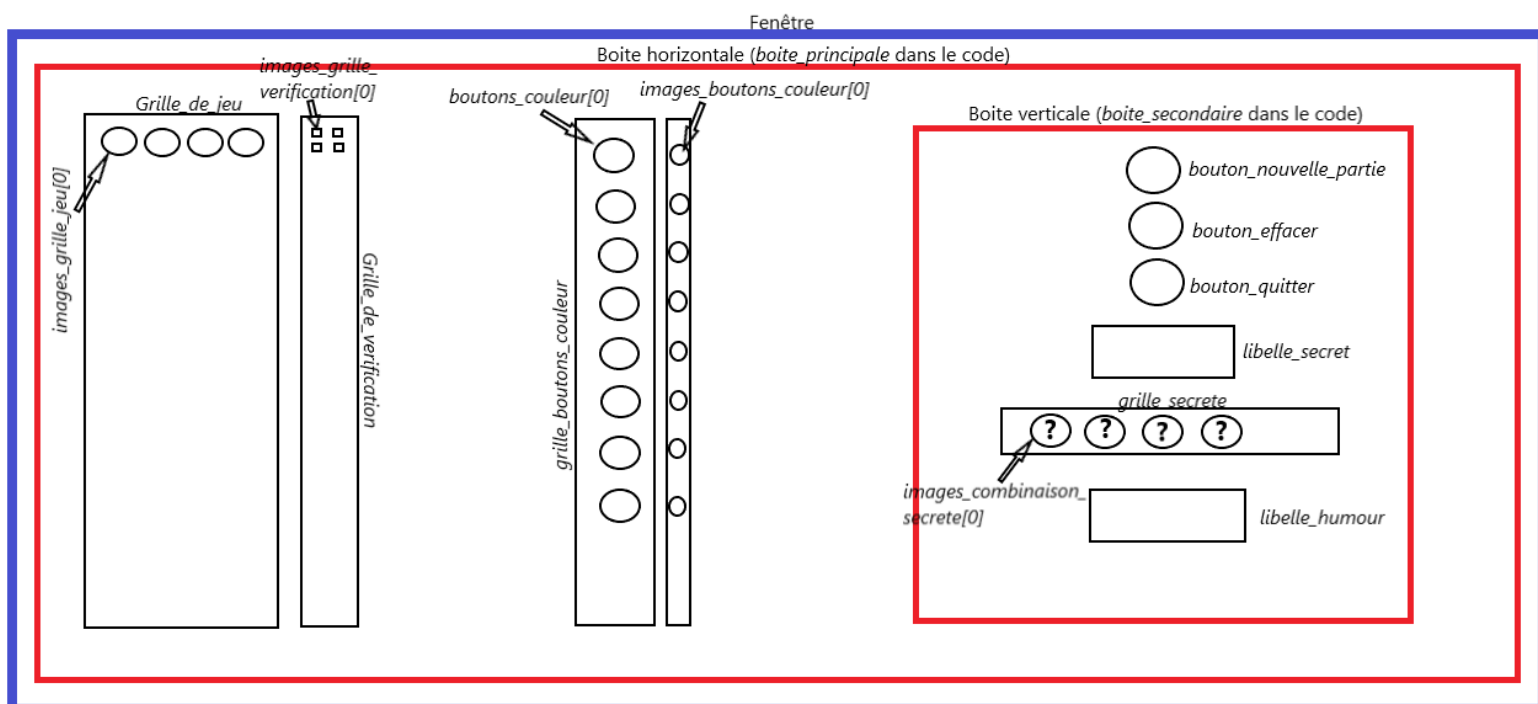
- Le bouton Quitter permet simplement de quitter le jeu en fermant la fenêtre de l'interface graphique.

4) La combinaison secrète

Enfin, juste en dessous de ces boutons, on peut voir le texte « Combinaison secrète... » suivi de 4 cercles avec un point d'interrogation au centre : il s'agit de la combinaison secrète qu'il faut trouver ! La « combinaison secrète », qui est initialement choisie par le joueur 1 dans le jeu « classique », est ici définie aléatoirement par notre programme. En cas de défaite ou de victoire du joueur, le programme affiche la combinaison secrète à la place des cercles avec un point d'interrogation au centre :



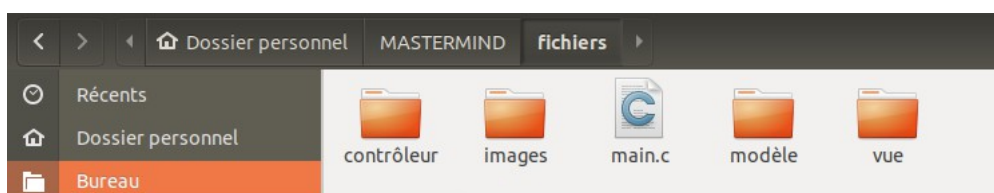
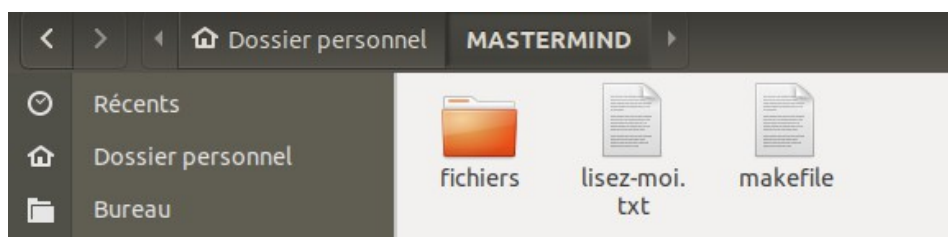
Schématiquement, voici à quoi ressemble l'organisation des widgets de notre interface :



III] Structure du programme

Nous avons essayé, pour rendre notre programme plus « agréable », de le structurer au maximum. Ainsi, nous avons créé un dossier « fichiers » dans lequel se trouve un sous-dossier « modèle », un sous-dossier « vue », un sous-dossier « contrôleur », un sous-dossier « images » et enfin notre fichier main.c !

Ainsi cela donne :



Comme son nom l'indique, le sous-dossier « vue » contient les fichiers *vue_mastermind.c* et *vue_mastermind.h*. Dans *vue_mastermind.h*, on peut notamment trouver la structure de la vue ainsi que la déclaration des fonctions de manipulation de la vue, tandis que dans *vue_mastermind.c*, on y trouve la définition. Ces fonctions sont purement « visuelles », en effet, elles se chargent uniquement de construire la vue (création des boutons, positionnement des boutons, mise en boîte des différents widgets, etc...)

Le sous-dossier « modèle » contient quant à lui les fonctions du modèle fourni par l'enseignante.

Le sous-dossier « images » contient toutes les images nécessaire à la création de l'interface.

Enfin, le sous-dossier « contrôleur » contient les fichiers *controleur_mastermind.c* et *controleur_mastermind.h*. Dans *controleur_mastermind.h*, on peut notamment trouver la structure du contrôleur ainsi que la déclaration des fonctions de manipulation du contrôleur, tandis que dans *controleur_mastermind.c*, on y trouve la définition. Ces fonctions assurent le lien entre la vue et le modèle.

IV] Quelques mots sur le contrôleur...

Le contrôleur est composé de plusieurs fonctions de manipulation (leur utilité est détaillée dans les commentaires du fichier *controleur_mastermind.h* :

- *void vue_mastermind_controler()*
- *void initialiser_partie()*
- *void jouer()*
- *void effacer()*
- *void afficher_combinaison_secrete()*

La structure de notre contrôleur est quant à elle relativement simple. On dispose dans cette structure :

- d'un pointeur *vue* qui peut contenir l'adresse d'une variable de type *vue_mastermind* (qui correspond à la structure de la vue), ce qui permet dans un premier temps de lier le contrôleur et la vue.
- d'un pointeur *mm* qui peut contenir l'adresse d'une variable de type *mastermind* (qui correspond à la structure du modèle), ce qui permet dans un second temps de lier le contrôleur et le modèle.
- d'un entier *rg* qui donne la position du joueur à chaque essai.
- d'un entier *nb_case* qui donne la position du joueur dans la grille dans son ensemble.

Voici un exemple pour mieux comprendre *rg* et *nb_case* :



Si on se place au niveau de la petite croix, alors dans ce cas là, *rg* = 3 et *nb_case* = 23 !

V] Les difficultés rencontrées et les possibles évolutions

1) Les difficultés rencontrées

Concernant la phase de conception de la vue, nous n'avons pas rencontré de problèmes ou de difficultés en tant que telle, il s'agissait d'ailleurs de la partie la plus simple du projet selon nous. En revanche, lorsqu'il fallait concevoir le contrôleur, les choses se sont un peu plus compliquées au début. En effet, nous étions un peu perdus face à toutes les fonctions du modèle, on ne savait pas trop par quoi commencer, bref, nous étions au point mort. Mais finalement tout est rentré dans l'ordre lorsque l'on a totalement compris le fonctionnement des fonctions du modèle, et on a pu finaliser le projet sans trop de soucis. Heureusement que Mme TOULOUSE était là pour nous aider

sur quelques petits soucis (explication plus détaillée de certaines fonctions, etc..) !

7/7

2) L'évolution du programme

Voici les éléments qu'on pourrait rajouter à l'avenir ;) ...

- Rajouter un compteur qui affiche le nombre de parties remportées par le joueur.
- Rajouter une interface de menu (avec un bouton pour changer la langue, un bouton « Play », etc...)
- Rajouter une musique/des sons (je ne sais pas si c'est possible avec GTK?)
- Rajouter un système de manches, qui fait en sorte de réduire le nombre d'essais autorisés à la partie suivante (pour pimenter un peu le jeu!) à chaque partie de Mastermind remportée.

On espère que notre projet vous plaira ! Remerciements particuliers à Mme TOULOUSE. Amusez-vous bien !

