

Algorithmique et programmation objet

Projet N°1 : Le mastermind

Daugreilh Bastien et Gardiès Samuel

1) Les choix de construction

Tout au long de la conception de ce projet, nous avons dû prendre des décisions de construction, que ce soit au niveau du programme en lui-même ou de « l'interface » que vas utiliser le joueur. Nos choix ont été motivés par deux objectifs principaux : créer un programme complexe et complet dans les possibilités offertes au joueur, tout en utilisant un code le plus simple et compréhensible possible.

Ainsi, au niveau du jeu en lui-même, nous avons choisi d'utiliser des scanner et de créer un menu dans le sous-programme principal permettant de choisir le mode de jeux pour qu'une fois le programme exécuté, le joueur n'ai qu'à sélectionner le mode de jeux voulu et entrer les informations demandées par le programme. Cela évite de demander au joueur d'appeler plusieurs programmes pour pouvoir jouer. De plus, le programme nécessitant donc que le joueur entre des informations pour fonctionner, nous avons fait en sorte qu'il soit robuste. Enfin, pour que le joueur comprenne facilement son erreur de saisie, nous avons implémenté différents messages en fonction de la nature celle-ci.

Toujours dans un souci de réaliser un jeu le plus clair possible pour l'utilisateur, nous avons décidé d'afficher l'historique des combinaisons proposées par le joueur (et leurs résultats) à chaque nouvelle proposition ainsi qu'à la fin de la partie. Cela permet au joueur de suivre sa progression et d'avoir un récapitulatif dès la fin de la partie mais également de voir par quelles propositions l'ordinateur est passé lorsque c'est lui qui doit trouver la combinaison.

En plus d'être facilement compréhensible par l'utilisateur, nous avons aussi décidé de proposer un jeu pouvant s'adapter à celui-ci. Ainsi, nous avons mis en place trois niveaux de difficultés différents que le joueur peu choisir lui-même en début de partie : débutant, intermédiaire, expert et libre. Ces niveaux de difficultés restreignent le nombre de propositions possibles à respectivement cinquante, vingt et dix propositions. Une fois ce nombre de propositions atteint, la partie est considérée perdue et prend fin immédiatement. Le mode de difficulté libre permet à l'utilisateur de choisir combien de tour sont possibles.

Au niveau du code, nous avons privilégié de nombreux sous-programmes aillant chacun un rôle bien précis plutôt qu'un programme contenant tous leurs codes. Cela permet d'éviter d'avoir des blocs de code compacts mais également de bien comprendre par quelles étapes passent chaque programme.

2) Les heuristiques

Pour la deuxième partie du projet, nous avons commencé par la stratégie la plus simple possible pour avoir la structure générale du programme, puis nous l'avons fait évoluer pour minimiser le nombre de propositions nécessaires avant de trouver la solution. Ainsi, notre première heuristique a simplement été de parcourir l'ensemble des combinaisons possibles et de les proposer une à une. Le nombre de combinaisons avec 4 pions de 8 couleurs différentes étant de 4 096, nous avons ensuite réfléchi à une façon d'améliorer notre programme pour réduire ce nombre.

Notre première idée fut de déterminer les couleurs présentes dans la combinaison. Pour ce faire, notre programme commencerait par proposer 8 combinaisons composées d'une seule couleur (exemple : rouge, rouge, rouge, rouge). Pour chacune de ces propositions, le nombre de pions bien placés correspond alors au nombre de fois où la couleur de la proposition est présente dans la solution. Ainsi, en une à huit propositions, notre programme détermine les couleurs présentes dans la solution.

En utilisant ce principe, notre deuxième heuristique consiste à trouver les couleurs de la solution, puis à proposer toutes les combinaisons possibles avec ces couleurs une à une, portant ainsi le nombre de propositions nécessaire à un maximum de 24, 87, ou 264, selon que la solution comporte deux, trois ou quatre couleurs différentes (1-8 propositions pour trouver les bonnes couleurs + 16, 81 ou 256 combinaisons possibles avec ces couleurs) . Avec cette deuxième heuristique, il nous restait encore un point à améliorer : trouver une façon plus efficace que le test de toutes les combinaisons possibles une fois les couleurs présentent dans la solution trouvée.

Ainsi, notre troisième et dernière heuristique commence comme la deuxième par trouver toutes les couleurs présentes dans la solution. Ensuite, elle consiste à proposer ces couleurs combinées d'une certaine façon, qui sera notre combinaison de référence. Si cette combinaison n'est pas la bonne, on intervertit deux couleurs de la combinaison (en partant de la fin) et on propose cette nouvelle combinaison. Deux cas sont alors possibles : soit le nombre de pions bien placés a augmenté et cette combinaison devient notre combinaison de référence, soit il a diminué et on repart à la combinaison de référence actuelle. On répète ensuite l'opération à partir de notre nouvelle combinaison de référence jusqu'à trouver la solution. À noter que si les deux couleurs à intervertir sont les mêmes, on passe alors bien entendu à la couleur suivante.

Grâce à cette heuristique finale, le nombre de propositions nécessaire pour trouver la bonne solution est maintenant compris entre 1 et 20 environ.

2) Le 3^e mode de jeu

Le deuxième mode de jeu demandé par le sujet s'apparentant plus à un test de l'algorithme qu'à une réelle partie de mastermind, nous avons décidé d'implémenter un 3^e mode de jeu permettant à l'utilisateur de réellement faire deviner sa combinaison à l'ordinateur.

Ainsi, ce mode de jeu interactif ne demande pas au joueur de donner sa combinaison, mais plutôt de la garder en tête et de la lui faire deviner à partir du nombre de pions bien et mal placés, comme lors d'une vraie partie de mastermind.

En effet, l'ordinateur va alors donner au joueur chacune de ses propositions pour qu'il détermine lui-même le nombre de pions bien et mal placés.

Concernant la stratégie, ce 3^e mode de jeux reprend le même fonctionnement que le 2^e pour trouver la bonne réponse, mais en demandant donc directement le nombre de pions bien et mal placés au joueur au lieu de le déterminer lui-même.