Justifications techniques

Architecture globale

Le dossier MasterMind est constitué d'une solution C# qui contient le projet. Les fichiers .txt contenant les paramètres et les enregistrements des meilleurs scores sont stockés à la racine du projet.

Structuration du code

Notre code source comporte une fonction principale Main qui appelle différentes fonctions et procédures qui vont veiller au bon fonctionnement du Mastermind.

La fonction Main s'articule de la manière suivante :

Premier temps : accueil

- **menu()** : Affiche un menu permettant 3 actions (aide, ràz des scores, jouer).

Deuxième temps : initialisation des paramètres de jeu

- **chargerParametres()** : Charge les paramètres de la partie (thématiques et niveaux) depuis le fichier "mastermind_param.txt".
- **choixPseudonyme()**, **choixNiveau()** et **choixThematique()** : Demande au joueur quels paramètres il veut utiliser pour cette partie.
- **defElements()** : Remplit un tableau contenant les éléments qui peuvent former la combinaison.
- **defCombinaison()** : Génère la combinaison secrète.

Troisième temps : jeu

- **jouer()** : Fonction de jeu dans laquelle le joueur va essayer de deviner la combinaison.
- **gagner()** : Détermine si le joueur a remporté la partie.
- **rejouer()** : Avec les mêmes paramètres (hormis la combinaison secrète).

Quatrième temps : gestion des scores

- **affichageScore()** : Affiche le score du joueur sur la partie jouée, ainsi que son meilleur score pour la thématique et le niveau choisi.
- **enregistrerScore()** : Enregistre le score du joueur s'il réalise son meilleur score.
- **afficherMeilleursScores()** : Affiche les 10 meilleurs scores pour la thématique et le niveau choisi par le joueur au début de la partie.

Choix techniques

Types de données :
Nous avons choisi d'utiliser des tableaux de strings pour la grille de jeu et la combinaison car ils permettent de regrouper, trier et classer facilement un même type de données.
Le stockage en type int de "thematique" et "difficulteNiveau" permet de les manipuler comme des indices dans les tableaux.
L'affichage de la grille de jeu à la console permet de visualiser l'ensemble des propositions des tours précédents et il se fait de bas en haut pour respecter la disposition classique du jeu.

Génération de la combinaison secrète :
Les éléments qui peuvent être utilisés lors de la partie de MasterMind sont tirés au hasard parmi ceux proposés dans le fichier de paramétrage .txt.
Une combinaison peut contenir plusieurs fois le même élément.

Gestion du score :
Le score correspond au tour auquel la combinaison est découverte par le joueur. On aurait pu proposer un système de score dépendant des pions rouges / blancs obtenus pondéré par le niveau de difficulté utilisé, mais une absence de pions donne aussi des informations nécessaires pour résoudre la combinaison et peut être stratégiquement correct.

Améliorations envisagées

Gestion des couleurs et des formes :
Nous n'avons pas intégré de couleurs sur les éléments de couleur, ni proposé d'afficher graphiquement les formes géométriques car ces fonctionnalités ne sont pas gérées efficacement par la console, et cela surchargerait (visuellement) cette dernière, ainsi que le code source.
La sémantique est suffisante pour reconnaître les mots ; en revanche, nous avons mis de la couleur pour les pions rouges et blancs afin de les différencier et les rendre similaires au MasterMind originel.

Enregistrement et reprise d'un partie :
Ces fonctionnalités ont tout d'abord été intégrées dans le menu et la fonction de jeu. Nous aurions pu les inclure dans notre programme en utilisant des fichiers .txt, comme pour les scores. Toutefois il aurait fallu stocker de nombreuses valeurs correspondant à une même partie et la reprise aurait nécessité un traitement différent selon le type de la variable.