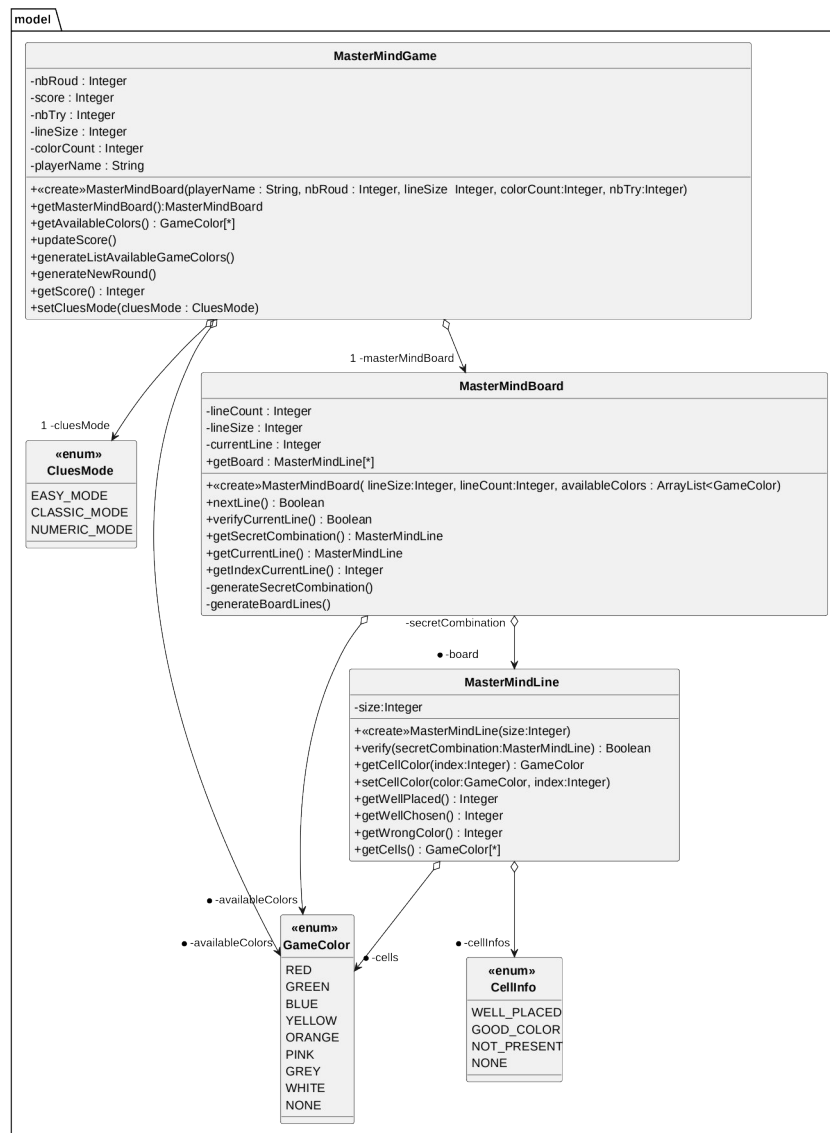


# Rapport du projet A31-MasterMind : Rendu 1

## Introduction :

Pour ce premier jet de notre projet de jeu « MasterMind » en Java nous avons fais de nombreux choix de conceptions que nous allons préciser dans ce rapport qui se concentrera principalement sur le modèle de notre application.

## UML :



### Package model :

Pour notre package model nous avons choisi de séparer notre jeu en 3 classes principales qui selon nous, représentent efficacement ce qu'est un MasterMind.

Nous avons décidé tout d'abord de représenter la partie de MasterMind par une classe MasterMindGame

### Classe MasterMindGame :

Cette classe regroupe tous les attributs et méthodes nécessaires au déroulement d'une partie composé de plusieurs manches, cette classe couvre entre autres tout ce qui est relatif aux informations du joueur, son score, le round sur lequel il est et le mode de jeu choisis.

La classe MasterMindGame gère la partie et passe d'une manche à l'autre.

Les manches sont gérées par la MasterMindBoard qui représente la table de jeu réelle.

### Classe MasterMindBoard :

Cette classe gère la manche courante de MasterMind, elle contient les lignes de la table de jeu ainsi que la combinaison secrète que le joueur doit trouver.

Cette combinaison est vérifiée par la ligne elle même, qui est contenu dans une liste de ligne dans la MasterMindBoard.

### Classe MasterMindLine :

Cette classe est donc une ligne du MasterMind, elle contient une liste de couleurs représentées par une énumération pour plus de clartés. La classe est capable de vérifier sa composition avec la composition secrète de la board dans laquelle elle est, et de générer dans une listes d'indices ,représentés par une énumération aussi, l'état de chacune de ses cellules (bien placé, mal placés, non présente).

### Conclusion :

Cette décomposition du modèle permet de faire se dérouler une partie de MasterMind en ligne de commande de manière fluide et compréhensible d'un point de vue programmation.

Chaque classe à un rôle bien précis et permettra par la suite de mieux intégrer le modèle à l'aide d'un contrôleur et de différents observers à une vue qui pourra l'utiliser de manière simple.