

A31 - Mastermind

Rapport final

Choix de conception	2
Observateurs	2
Modes de jeux	2
Première solution :	2
Seconde solution :	2
Boutons de changement de couleur de la combinaison	3
Diagramme de classe du projet	3

Choix de conception

Observateurs

Nous avons décidé de mettre 3 observateurs qui regardent chacun des endroits spécifiques :

- ObservateurManche, qui observe une Manche
- ObservateurTentative, qui observe une Tentative
- ObservateurPartie, qui observe la Partie

Cela permet par exemple à ce que la manche n'ait pas accès à des paramètres de l'observateur de la partie ou des tentatives. Nous avons décidé que ces trois observateurs soient réunis en une seule interface, ObservateurUI, qui servira à unifier le type d'affichage tout le long du jeu (pour éviter qu'il n'y ait un affichage par fenêtre pour la partie et un autre par console pour la manche par exemple).

Modes de jeux

Pour les stratégies d'affichage d'indices, nous avons décidé de les enlever, car le problème est que le patron de conception demande à ce qu'il y ait trois classes par type d'affichage, ce qui finirait par donner un très grand nombre de micros-classes, ce qui aurait pollué le projet. Nous avons réfléchi à deux solutions :

Première solution :

La première solution conserve tous les patrons de conception concernés (observateur et stratégie) : les trois classes Facile, Classique et Numérique deviendraient des classes abstraites et auraient des sous-classes pour chaque type d'affichage.

Nous aurions par exemple les classes ClassiqueFenetre, ClassiqueConsole, FacileFenetre, FacileConsole, etc.

Les deux problèmes de cette solution sont qu'elle ne nous oblige pas à créer une méthode pour chaque type d'affichage et chaque mode de jeu, et aussi que cela entraînerait la création de nombreuses classes pour n'abriter qu'une seule méthode.

Seconde solution :

La seconde solution abandonne l'idée du patron de conception de stratégie et implémente les trois méthodes abstraites afficherIndicesFacile, Classique et Numérique dans ObservateurTentative pour obliger la présence d'une façon d'affichage. Cela impliquerait la présence d'un switch case sur le mode de jeu pour déterminer la méthode à utiliser.

Le seul problème est qu'on aurait un switch case à utiliser pour déterminer le type d'affichage.

On a donc décidé de choisir la Solution 2 car la Solution 1 n'était pas adaptée et qu'il n'est pas favorable d'avoir des erreurs vis-à-vis des classes manquantes.

Boutons de changement de couleur de la combinaison

Pour le changement de couleur, nous voulions mettre en place la possibilité de sélectionner la couleur et l'index dans différents ordres : l'index puis après la couleur, et inversement. Mais, par manque de temps, nous n'avons que réussi uniquement à implémenter le choix d'une couleur puis après l'index. Cependant, nous savons comment faire : il suffit, dans le choix des couleurs, de vérifier si un index a été sélectionné et, si oui, de changer la couleur. Mais cela va de pair avec le fait que nous voulions aussi enlever les boutons de sélection de l'index et les remplacer par la sélection de la case elle-même.

Diagramme de classe du projet

Le diagramme de classe du projet est disponible dans le fichier uml/DCC.puml à la racine du projet.