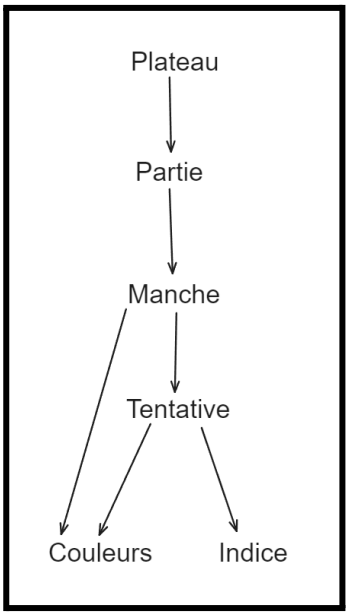
Projet Mastermind A31 : Choix de conception

Eliott Schott - Sylvain Schaefer



Les classes (Modèle) :

- Plateau : La classe Plateau a été créée pour être transmise au contrôleur, qui va créer l'objet **Partie** via la classe **Plateau**, que le contrôleur pourra manipuler. La classe Plateau est un peu le "parent" de notre jeu. Elle permet aussi de stocker le nom du joueur, pour pouvoir l'afficher, ainsi que son meilleur score, par exemple à la fin de la partie ou quand on souhaite en recommencer une. La fonction de notification permet à la **ViewGame** de connaître la bonne taille de combinaison afin que l'affichage soit correct.

- Partie : La Partie est générée lorsque le contrôleur appelle la méthode du Plateau. Et elle permet au contrôleur de créer les **Manche** avec les bons paramètres, via sa méthode "*createManche()*"**.** Elle est "parent" de **Manche** mais "enfant" de **Plateau** car une **Partie** peut créer plusieurs **Manche.**

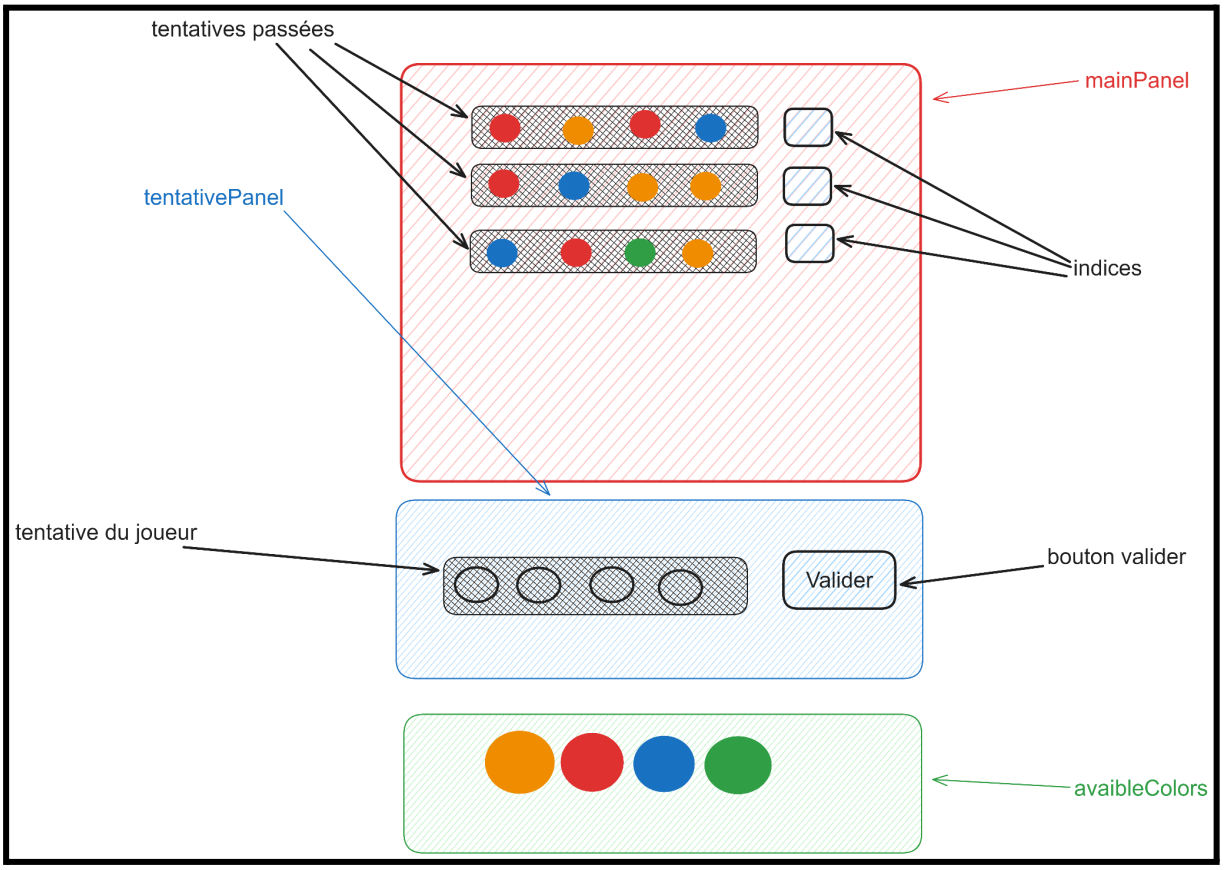
- Manche : La classe **Manche** connait les informations nécessaires à la création des différentes **Tentative.** La **Manche** possède des méthodes afin de créer la combinaison secrète et les **Tentative** afin que lors de la vérification, elle puisse comparer les deux combinaisons. La **Manche** est "parent" de **Tentative** et "enfant" de **Partie**. A chaque fois que le joueur valide sa combinaison, la **Manche** s'occupe de vérifier cette dernière, en actualisant la variable booléenne qui dit si la **Manche** est terminée. Les fonctions de notification des observers permettent d'actualiser l'affichage des indices après la vérification, et permettent au joueur de placer des pions pour son essai.

- Tentative : Nous avons décidé de créer cette classe car elle permet de stocker pour une **Tentative** donnée un tableau de ***Couleurs*** qui représente la combinaison entrée par le joueur, ainsi que le tableau d'***Indice*** qui sera rempli en cohérence lors de la vérification. Donc **Tentative** est "enfant" de **Manche** et utilise les enums ***Indice*** et ***Couleurs***

- (Enum) Indice : Enumération d'***Indice*** avec laquelle on peut construire les tableaux stockés dans la **Tentative**

- (Enum) Couleurs : Énumération de couleurs avec laquelle on peut construire les tableaux stockés dans la **Tentative** et dans la **Manche**.

- (Interface) mastermindObserver :
    - "init" permet de fixer la taille des panels. Nous avons utilisé le patron Observer car il permet d'actualiser la vue tout en restant dans un modèle de conception MVC. L'interface fait le lien entre les classes **Plateau**, **Partie** et **Manche** et la classe **ViewGame**.



Les classes (Contrôleur) :

- GameController : Le GameController permet de créer la partie, en appelant la méthode du Plateau. La fonction GameStart permet de créer les manches via la partie et les tentatives via les manches. Lors de l'appel à valider la tentative dans la vue, le GameController appelle les fonctions de la Manche.

Les classes (Vues) :

- ViewStart :
    - permet de donner les paramètres tels que le nombre de tentatives, la taille des combinaisons, le nombre de pions disponibles, le nombre de manche et le nom du joueur
- ViewGame (implémente mastermindObserver) :
    - Variable globale entier “indiceType” (0, 1 et 2) pour indices (facile, classique et numérique) donc la méthode “*updateIndice()*” va afficher l’indice selon la variable gobale “indiceType”.
    - Nous avons fait le choix de pouvoir changer le mode d’affichage des indices directement dans ViewGame, soit en plein partie et non dans ViewStart
- ViewEnd : Nous n’avons pas encore décidé du rôle de ViewEnd. Elle permettra probablement de relancer une partie via un appel du contrôleur, ou simplement d’afficher les statistiques du **Plateau**.