

A31 - Mastermind

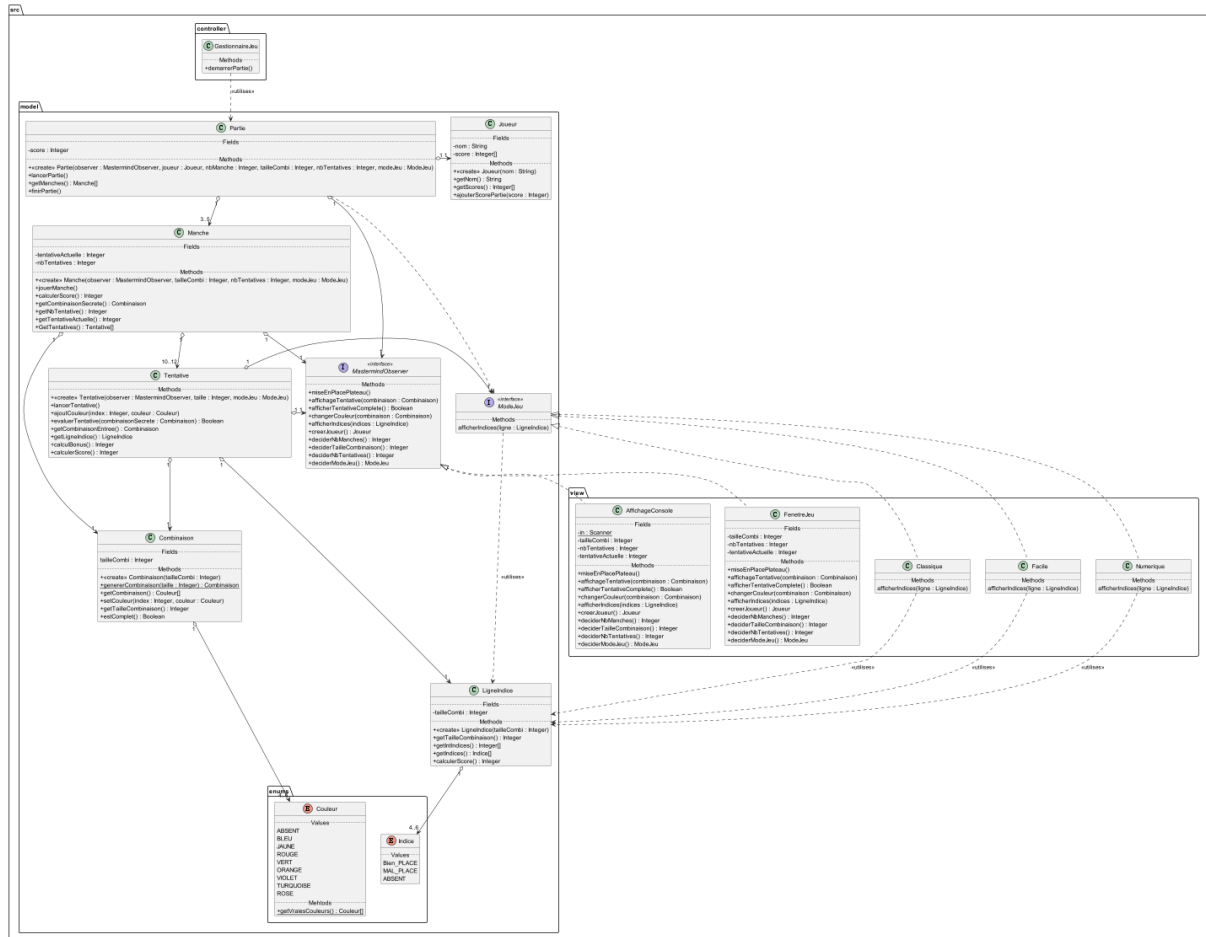
Rapport du 10/12/2023

Sommaire

Sommaire.....	1
DCC produit.....	2
Retranscription du DCC et explication des choix de conception.....	3
Modèle.....	3
Partie :	3
Joueur :	3
Manche :	3
Tentative :	3
Combinaison :	3
Couleur :	3
LigneIndice :	4
Indice :	4
ModeJeu (interface) :	4
MastermindObserver :	4
Vues.....	5
Classique.....	5
Facile.....	5
Numerique.....	5
FenetreJeu.....	5
AffichageConsole.....	5
Contrôleurs.....	6
GestionnaireJeu.....	6

DCC produit

Jusqu'à présent, nous avons consacré toutes les séances d'A32 à la conception du diagramme de classe et au fonctionnement du projet. Voici le diagramme produit :



Nous avons commencé par produire les classes modèles en décomposant le jeu : le jeu est composé d'une partie étant elle-même composée de plusieurs manches qui contiennent un plateau qui est représenté par l'ensemble des tentatives composées d'une combinaison et d'indices.

Après avoir fait les modèles, nous avons créé la classe `GestionnaireJeu` qui sert de contrôleur pour l'ensemble de l'application. A cela, nous avons ajouté les vues, c'est-à-dire une classe observatrice pour l'affichage du jeu sur la console, une autre pour l'affichage avec une fenêtre et trois classes pour les modes de jeu.

Retranscription du DCC et explication des choix de conception

Modèle

Partie :

La classe Partie s'occupe du lancement des manches et de donner le score au joueur. Nous avons décidé que seul la partie aurait accès au joueur puisque les manches et les autres classes en dessous n'ont pas d'intérêt à ce qu'elles le connaissent. Elles sont simplement commandées par la partie et font ce qu'elle veut.

Joueur :

Cette classe représente un joueur avec un nom et un tableau de scores. Il possède le score car le score est une caractéristique intrinsèque au joueur dans le contexte du jeu. Chaque joueur a un score qui reflète sa performance au fil des parties. L'association du score avec la classe est donc naturelle.

Manche :

Elle représente une manche du jeu, avec des tentatives et une combinaison secrète. Nous avons décidé que ce serait à la manche de stocker la combinaison secrète pour faciliter l'évaluation des tentatives du joueur, la gestion de la logique de la manche et la communication avec l'observateur du jeu. Les autres classes n'ont pas à savoir la combinaison secrète et il n'y a que la manche qui la connaît puisqu'il y a une combinaison secrète à chaque manche. Nous avons aussi décidé que ça serait à la manche de retenir le nombre de tentatives car la partie n'en a pas besoin et aussi puisque les tentatives ne veulent pas forcément savoir combien elles sont dans la manche.

Tentative :

La tentative représente une tentative du joueur pour deviner la combinaison secrète. C'est elle qui permet de faire le lien entre la combinaison et la ligne d'indices. Nous avons décidé de mettre l'évaluation de la combinaison dans la tentative puisque l'évaluation a besoin de la combinaison entrée pour la comparer et de la ligne d'indices pour afficher les indices.

Combinaison :

La classe Combinaison encapsule la logique liée à la manipulation d'une liste de couleurs. C'est elle qui a la méthode statique pour générer une combinaison aléatoire car nous avons décidé que ça permet que seule la classe combinaison peut créer et modifier des combinaisons, cela n'aurait pas été logique que la classe manche génère une combinaison secrète elle-même.

Couleur :

L'énumération Couleur représente les couleurs disponibles dans le jeu. Nous avons décidé d'ajouter une valeur "ABSENT" car elle permet d'identifier plus facilement les cases

non initialisées. Notre jeu propose les huit couleurs suivantes: bleu, jaune, rouge, vert, orange, violet, turquoise et rose.

LigneIndice :

La classe LigneIndice encapsule la logique liée à la manipulation des indices. C'est donc elle qui s'occupera de stocker les indices de la tentative et de les changer.

Indice :

L'énumération Indice représente les indices possibles dans le jeu. Tout comme pour la couleur, nous avons décidé d'ajouter une valeur "ABSENT" pour les cases n'ayant aucune valeur. L'énumération reprend bien sûr les deux indicateurs "bien placé" et "mal placé".

ModeJeu (interface) :

Cette interface sert de super-classe pour une stratégie. Elle représente les méthodes que les trois modes de jeu doivent afficher. Seul la partie et la tentative utilisent cette interface parce que la partie doit pouvoir la stocker et la redistribuer ensuite et parce que la tentative en a besoin pour l'affichage des indices

MastermindObserver :

Le MastermindObserver est une interface servant comme son nom l'indique d'interface observatrice. Elle est stockée et appelée dans chaque classe ayant besoin d'afficher des informations à l'utilisateur. Parce qu'elle a été ajoutée à la dernière minute, l'interface et ses classes l'implémentant seront sûrement modifiées.

Vues

Classique

Cette classe est une sous-classe de ModeJeu, son rôle est donc d'afficher des indices. Pour afficher les indices, elle va d'abord afficher les couleurs bien placées puis les mal placées.

Facile

Tout comme la classe précédente, Facile implémente ModeJeu. Sa façon d'afficher montrera les informations sous la même forme mais dans l'ordre de la combinaison.

Numerique

Numerique est la dernière des trois classes implémentant ModeJeu. Cette classe affiche les indices de la même façon que le mode classique sauf qu'à la place d'afficher les pions, il y a le nombre de pions bien ou mal placées.

FenetreJeu

La classe FenetreJeu est une deuxième classe implémentant MasterMindObserver. C'est une classe que l'on a anticipé qui servira à afficher une fenêtre et la partie courante.

AffichageConsole

Cette classe se concentre sur l'affichage du jeu dans la console. Elle permet de jouer directement en entrant les informations dans la console.

Contrôleurs

GestionnaireJeu

La classe joue un rôle essentiel dans la coordination et le démarrage du jeu Mastermind. Elle est responsable de l'initialisation et de la gestion des parties. Ce contrôleur centralise les étapes préliminaires, telles que la création du joueur, la décision du nombre de manches, la détermination de la taille de la combinaison et du nombre de tentatives, ainsi que le choix du mode de jeu en appelant l'observateur qu'il crée.