

Rendu 2 : Rapport

L'ajout des vues et du contrôleur représente la majorité du travail effectué entre le premier rendu et le second.

Composants principaux

Ecran de début : donne la possibilité au joueur de saisir ces paramètres, de choisir son mode de jeu, ces informations sont ensuite récoltées par le contrôleur qui à leur tour seront transmis à la classe MastermindGame, cette classe va ensuite instancier les prochaines vues.

Ecran de jeu : La vue principal du jeu, celle-ci affiche les rangés de pions, tous les pions sont toujours affichés à l'écran peu importe la manche et il suffit de cliquer sur un pion pour changer sa couleur. Les indices sont affichés en bas de l'écran. Le contrôleur vérifie la couleur des pions à chaque tentative et transmet ensuite ces informations à la classe de la manche en cours. Les panels sont générés en fonction du mode de jeu.

Ecran de fin : Cette vue affiche seulement le score final du joueur transmis par le contrôleur, s'il a envie de rejouer ou de quitter le jeu. Le bouton rejouer remet le joueur sur l'écran de début et lancera une autre instance de MastermindGame.

Contrôleur : Le MastermindController est responsable du passage d'un round à l'autre et les RoundController qui s'assure du bon déroulement des rounds, les RoundControllers sont stockés dans une liste de round

Design patterns utilisés

Strategy :

-Nous continuons d'utiliser le design pattern strategy pour la génération de nos indices car il nous semble encore être le meilleur choix. La

génération d'indice étant un "mode" qui change le comportement de la manche le fait d'utiliser les stratégies nous semble être la meilleure manière de tacler ce problème

Observer :

-Même chose ici avec le design pattern observer, il nous permet de prévenir des classes sur des changements effectués et avec notre séparation de la manche, du jeu et des rounds avec des classes distinctes, ce design pattern nous reste grandement utile pour faire communiquer ces classes (par exemple pour communiquer la fin d'une manche)

MVC : La structure MVC est entièrement respectée avec les contrôleurs qui interagissent avec la vue, le model qui ne connaît pas la vue et le contrôleur qui vient modifier le model.