GOKTEKIN MUHAMMED EMIN
KOAMA DANY

Rapport : Jeu d'échec

POO2 S4

I)

Le projet :

Ce projet consiste à reproduire le fameux jeu d'échec, constituer de différentes pièces et plusieurs règles spéciales que nous devons prendre en considération comme par exemple : le roque, la prise en passante, la promotion de certaines pièces, et deux règles variantes que nous devons mettre en place en faisant preuve de créativité.

Nos deux variantes sont donc :

-La possibilité d'ajouter des pièces féerique remplaçant d'autre pièces déjà existantes.

-La possibilité pour un joueur de jouer deux fois d'affilée.

II)

Modélisation :

Nous avons mis en place une classe grid dans laquelle un tableau de Piece va être définit, pour ensuite pouvoir récupérer les données et créer un visuel. Nous avons aussi crée un tableau de booléen de la même taille que ce tableau précédent pour pouvoir traiter chaque pièce selon leur position.

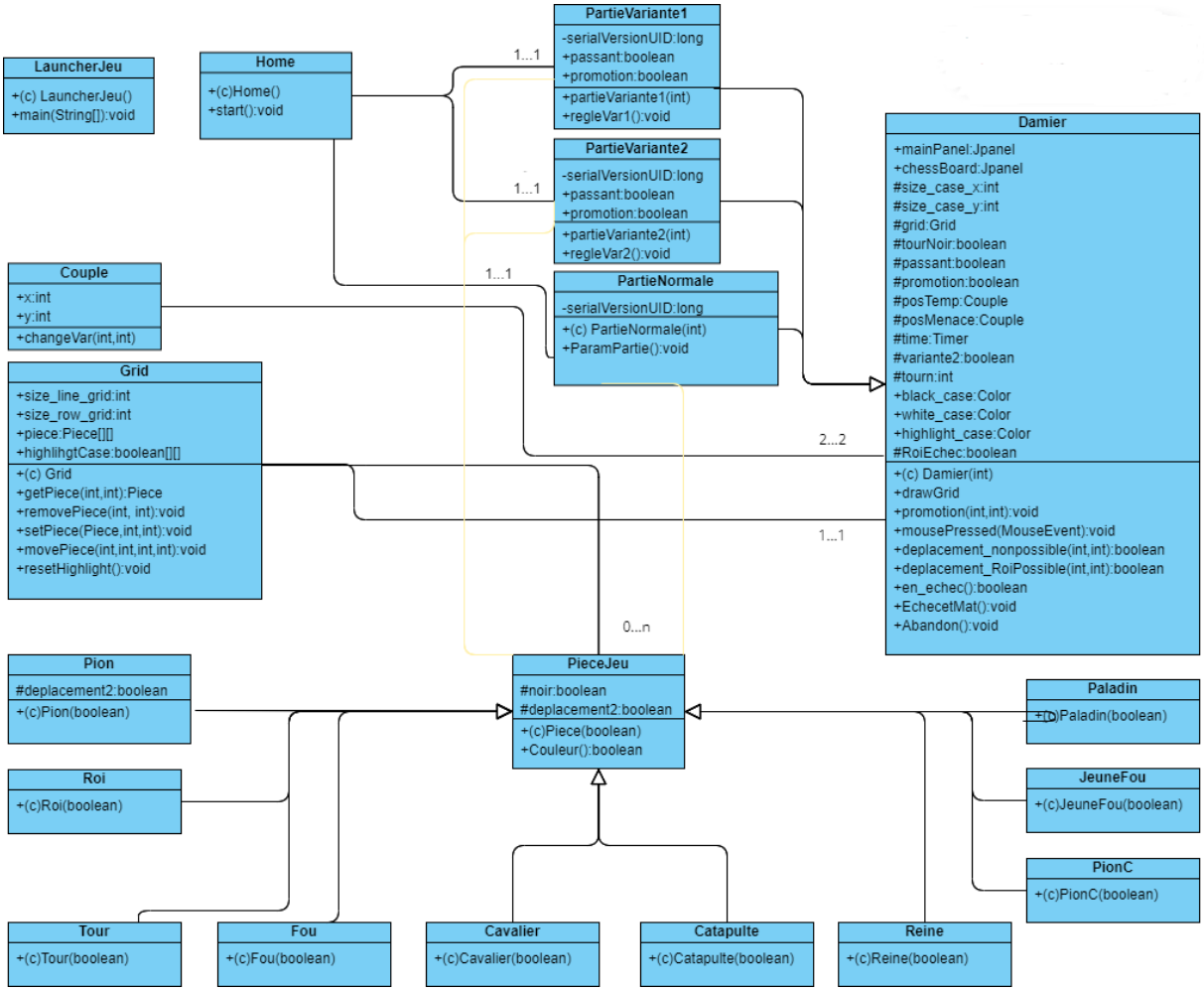
Visualisation :

La partie visuelle du projet a été plutôt structuré dans la classe Damier qui comprend la création du damier en lui-même et dans la classe Home l'interface comprenant le type de partie à jouer et la taille du damier.

Contrôleur :

Nous avons défini la partie Contrôleur principalement dans la classe Damier pour que l'on puisse adapter les mouvements des différentes pièces dans grid par rapport a ce qui est afficher dans Damier. Pour pouvoir faire cela, nous avons créé différente méthode pour le déplacement des pièces et des classes secondaires (par exemple Couple) pour des cas précis.

Voici le diagramme UML de notre projet :



(Liens en jaune : pour ne pas créer de confusion entre différents liens ce sont les mêmes que les noirs)

III)

Explication du code :

Tout d'abord nous avons une classe LauncherJeu contenant une méthode main qui nous permet de lancer le jeu en utilisant la méthode start de la classe Home pour pouvoir lancer une interface permettant de choisir la partie à l'aide de ses sous-classes pour chaque partie en question. Ensuite nous avons une classe PieceJeu, classe parent de plusieurs classes dont Tour, Pion, Cavalier … qui vous l'aurez compris représente les différentes pièces du jeu. Cette classe va nous être très utile car nous l'utiliserons pour mettre en place les pièces dans la classe grid pour les différents types de parties.

La classe grid que nous utiliserons avec les différentes méthodes de getter et setter et même de suppression de pièce mais aussi de mouvement. En effet grid va nous permettre à l'aide de son tableau de booléen de connaitre les possibilités d'actions des différentes pièces.

La classe Damier que l'on peut considérer comme la classe « principale », dans laquelle on crée l'interface de jeu pour y mettre les différentes pièces à l'aide de la classe grid. Pour pouvoir se déplacer avec ces pièces à l'aide des clique émis avec sa souris, que l'on fera dans la méthode mousePressed. Les déplacements seront définis selon la pièce sur laquelle on a cliqué. Les choix possibles de déplacements seront visibles à l'aide d'une différente couleur. Les différentes règles du jeu sont définies dans cette classe. Tout comme les règles de la prise en passante, le roque, la promotion de pièce. Bien sur la mise en échec du roi, l'abandon et effectivement l'échec et mat. Tout cela sera fait dans les différentes méthodes de Damier. (La prise en passante est effectuée dans mousePressed).

Pour la mise en échec nous avons crée une méthode deplacement_nonpossible qui va nous dire selon les déplacements des pièces adverses si le roi est menacé, on utilisera cette méthode dans la méthode en_echec, mousePressed et deplacement_RoiPossible.

Pour l'échec et mat nous utiliserons la méthode deplacement_RoiPossible, pour pouvoir savoir si le roi pourra voir encore des possibilités de déplacements, si non échec et mat.

La méthode abandon étant juste une méthode servant à proposer à un joueur qui joue d'abandonner au bout d'un certain temps.

La méthode promotion faisant appel à une interface, qui permet de changer la nature d'un Pion s'il atteint le bout du damier.

Les différentes pièces féeriques :

1. La catapulte ne peut que se déplacer d'une case vers le haut ou vers le bas. Elle attaque à l'horizontale uniquement et permet d'attaquer des pièces derrière d'autres pièces. (Remplace les pions de chaque cotés).
2. Le jeune fou peut se déplace d'un en diagonale. Par contre, contrairement au fou, il ne peut que manger à l'horizontale et à la verticale. (Remplace les fous).
3. Enfin un pion assez intelligent pour aller en avant et en arrière, et manger en avant et en arrière. (Remplace les pions de chaque cotés +1 case).
4. Le paladin possède les mêmes déplacements que le roi, mais peut aussi manger une pièce deux cases éloignée de lui à l'horizontale et à la verticale. (Remplace les cavaliers).

Répartition du travail :

Nous avons communiqué tout au long à l'aide de discord et avons essayé de partager le travail équitablement entre nous. Puis nous faisions le point ensemble pour travailler de manière rigoureuse et organiser.