

Jeu d'Echec JAVA

Fait avec Véronique Huang

Introduction

Nous avons choisit, avec ma partenaire Véronique Huang, de coder un jeu d'échec pour notre projet de JavaFX. Nous avons donc fait des recherches sur Internet pour débiter notre projet car nous ne savions pas réellement par où commencer, et nous sommes inspirés d'une personne dont je vais citer la source à la fin de ce document. Ce dernier nous a beaucoup aidé et nous a servi de tremplin pour commencer. Il nous a également permis de comprendre de nombreux concepts en JavaFX. Je vais dès lors détailler les fonctions pour lesquelles j'ai le plus contribué, tout en référençant le travail de ma partenaire si besoin est.

Développement

I. La classe Plateau.java

Cette classe est la plus volumineuse en code et contient la plus grosse partie des fonctions que contient notre programme, il s'agit d'un condensé de fonctions faites par moi et Véronique. On peut y trouver les fonctions qui permettent de se déplacer selon les pièces (toutes ces fonctions ont été effectuées par Véronique, les fonctions de mouvements, coupLegaux, et une fonction non aboutie mais qui marche partiellement, celle du roi mis en échec).

Pour ma part, je me suis occupé des fonctions qui permettent de cliquer sur une case et de leur attribuer des actions, comme la fonction jouerCoup qui est appelée dans la fonction casesClic, elle même appelée dans le constructeur du Plateau. Avec ceci nous avons des fonctions comme la fonction « setCurrent » qui permet de changer la couleur d'une case si elle est sélectionnée, et si et seulement si elle n'est pas vide, et également « setPlateau », qui permet de réactualiser le plateau selon la situation. J'ai également fait la fonction coupPossible, qui affiche les coups disponibles pour la pièce sélectionnée. Le jeu d'échec étant un jeu au tour par tour, il est impossible de sélectionner un pion adverse, hormis pour le « manger ».

J'ai réussi à implémenter, non parfaitement, la notion de « Promotion » d'une pièce, en faisant une classe appelée « Promotion », cette classe est utilisée dans la fonction « choisit », qui permet de sélectionner quelle pièce va remplacer notre pion.

II. La classe Promotion.java

La création de cette classe m'a été nécessaire pour implémenter la promotion d'une pièce qui permet, lorsqu'un pion a atteint le bord opposé à sa position, de le promouvoir en dame, fou, cavalier ou tour. Le procédé est exactement le même que pour la classe plateau, sauf que cette fois-ci j'affiche une petite fenêtre contenant 4 cases, avec les pièces qui remplaceront le pion. Lorsqu'un pion est sélectionné, la fenêtre disparaît et le pion se voit remplacé par la pièce choisie, grâce à la méthode

« casesClic » incluse dans la classe Promotion. Je fais appel à cette classe dans la fonction « choisit » de la classe plateau, en créant un nouveau « stage » et une nouvelle « scène », le constructeur de la classe Promotion contient un paramètre booléen, pour connaître la couleur de la pièce qui a atteint le bord.

III. La classe Echec.java

Il s'agit là de la classe principale de notre programme, toute simple, elle contient la méthode « main » et la méthode « start » nécessaires au fonctionnement du programme, j'y définit la taille de la fenêtre (500x500) et l'icône correspondant au programme.

IV. La classe Cases.java

C'est la classe qui définit la couleur des cases, il s'agit d'une classe qui « extends » la classe « buttons », pour que toutes les cases sur le plateau soient en fait des boutons sur lesquels cliquer. Cette classe contient 2 constructeurs, l'un est appelé dans le constructeur du plateau avec comme paramètres 2 entiers, correspondants aux coordonnées de la case, et d'un booléen permettant de déterminer la couleur de la case. Le deuxième constructeur ne contient qu'un entier et sera appelé dans la classe Promotion, pour afficher les 4 boutons contenant les pièces à choisir. Cette classe contient des méthodes, notamment getPiece, pour obtenir la pièce sur la case voulue, getPieceColor également, qui permet de savoir si la pièce est noire ou blanche, la méthode setPiece qui permet de disposer une pièce sur la case, supprimerPiece qui, comme son nom l'indique, permet de supprimer une pièce, puis la méthode estPris, pour savoir si une pièce est occupée ou non. Les autres méthodes sont les setters et getters basiques pour les coordonnées des cases.

V. Utilisation/manipulation d'images

Je me suis également occupé de trouver les images/icônes nécessaires pour la stylisation du plateau et des pièces. J'ai trouvé une petite icône sympathique que j'ai modifiée, qui sert d'icône d'application. Pour les images des pièces, j'ai trouvé une archive contenant plusieurs icônes pour les pièces d'un échiquier. J'ai alors modifié leur taille dans le code car leur taille était beaucoup trop grande par rapport à chaque case. J'ai aussi utilisé ce même procédé pour colorer les cases, je me suis servi d'images de texture de bois pour donner un petit style au plateau, les mêmes images ont été utilisées pour la case de couleur rouge (case sélectionnée), et la case de couleur bleue (coup disponible selon la pièce).

Conclusion

Pour conclure, j'ai apprécié travailler sur ce projet, sincèrement, car il m'a permis de comprendre pas mal de choses en JavaFX, pas nécessairement évidentes. Nous n'avons pas pu aboutir le jeu d'échec avec ma partenaire, concernant l'implémentation de l'échec et mat, mais avons fait le nécessaire pour toute la partie graphique, les pions se déplacent bien, « mangent » leurs opposants, respectent leurs déplacements respectifs, et le joueur bénéficie d'une indication visuelle sur les mouvements disponibles avec la pièce qu'il manipule.

Je pense que je pourrais continuer à travailler sur ce projet en dehors du cadre scolaire, pour voir si je suis capable ou non de le mener à terme.

Références

<https://github.com/stevoisiak/JavaFX-Online-Chess>

http://www.scrapotin.com/boutiq/pop_up.php?im=im/articles/I2989742IT0L.jpg

https://www.adzif.biz/boutique/images_produits/whitewood-z.jpg

<https://marcelk.net/chess/pieces/>

Et d'autres sites/forums pour des problèmes rencontrés dont je ne saurais retrouver la source...