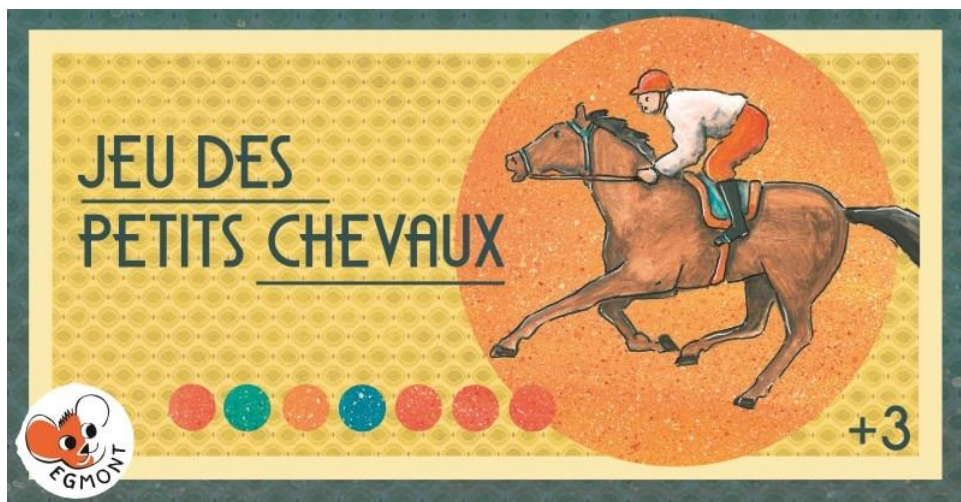


GAVOILLE Quentin

CARON Maxime

Projet POA

Le jeu des petits chevaux



IER SCIENCES &
J TECHNIQUES

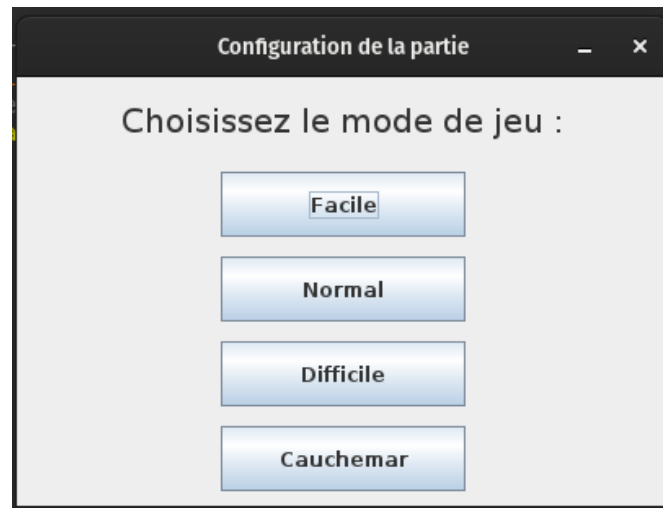


UNIVERSITÉ DE
FRANCHE-COMTÉ

Année 2021

Comment le jeu se présente-t-il :

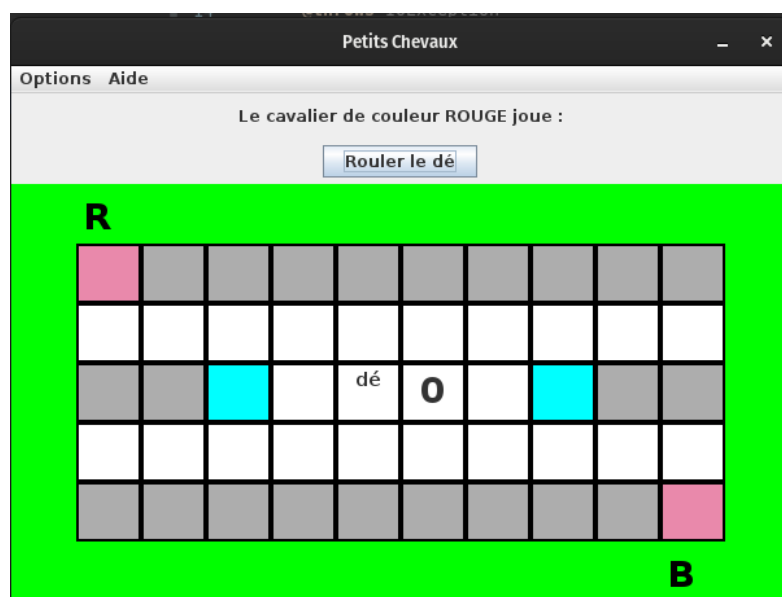
Quand vous lancez le jeu, la première chose que vous verrez et une fenêtre vous demandant de choisir la difficulté avec laquelle vous souhaitez jouer.



Ici on voit qu'il y a 4 niveaux de difficulté qui sont :

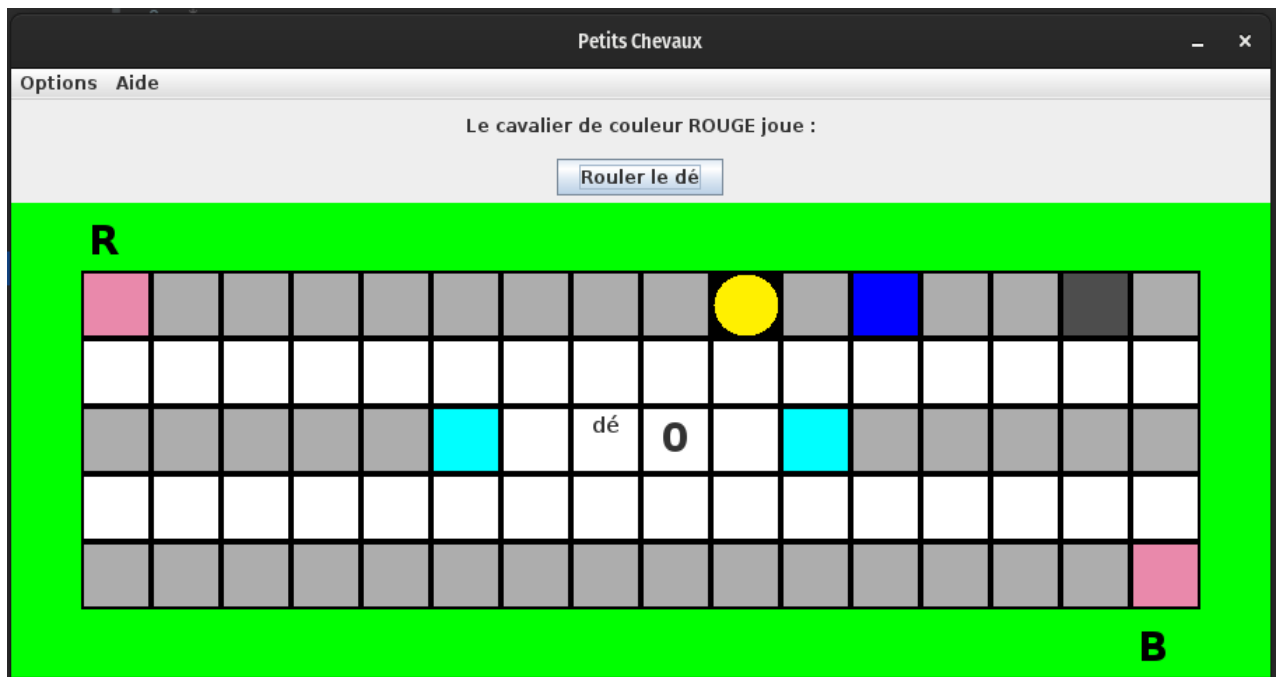
-Facile : dans ce mode les ligne font 10 cellules (jouables) et il n'y a aucuns pièges.

Ainsi si vous avez choisi le mode facile vous verrez ce type de plateau :



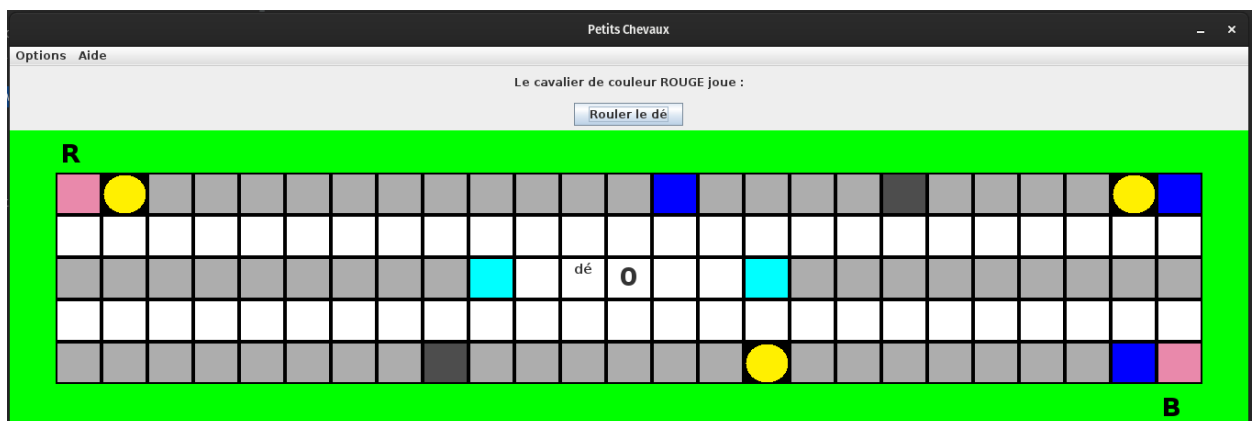
-**Normal** : dans ce mode les ligne font 16 cellules (jouables) et il y a 3 pièges placés aléatoirement sur le plateau.

Ainsi si vous avez choisi le mode normal vous verrez ce type de plateau :



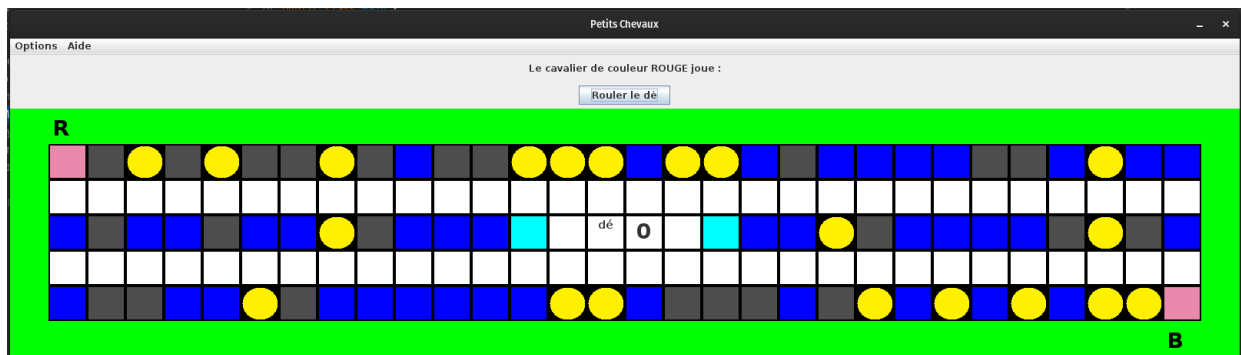
-**Difficile** : dans ce mode les ligne font 25 cellules (jouables) et il y a 10 pièges placés aléatoirement sur le plateau.

Ainsi si vous avez choisi le mode difficile vous verrez ce type de plateau :



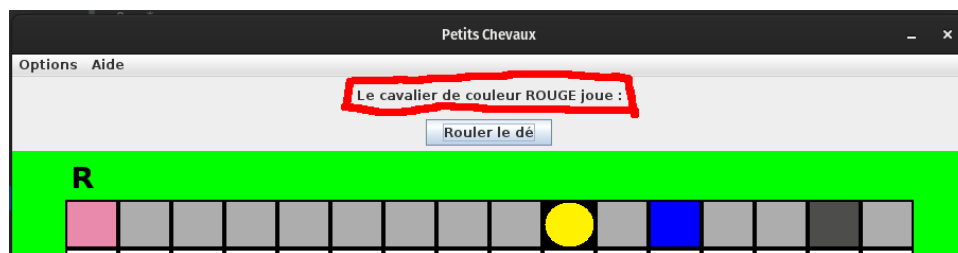
-Cauchemar : dans ce mode les ligne font 30 cellules (jouables) et il toute les cases neutres sont des pièges.

Ainsi si vous avez choisi le mode cauchemar vous verrez ce type de plateau :



Pour chaque partie vous incarnerez le cavalier rouge identifié par la lettre 'R', votre adversaire le cavalier bleu lui est incarné par le joueur bleu symbolisé par la lettre 'B'.

À chaque tour le jeu indique quel joueur doit jouer dans le message tout en haut, ainsi que la valeur du dé au milieu du plateau.

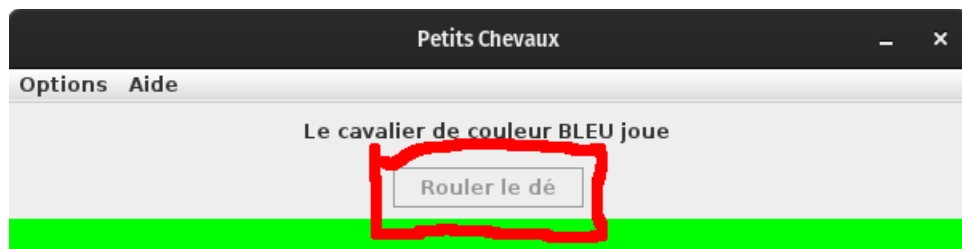


Pour lancer le dé le joueur doit cliquer sur le bouton "rouler le dé", attention le bouton se bloque lorsque que c'est au tour de l'adversaire.

Peut lancer le dé :



Ne peut pas lancer le dé :



Si le joueur souhaite quitter ou recommencer la partie, il peut cliquer sur le menu déroulant "Options" en haut à gauche pour accéder à ces fonctionnalités :

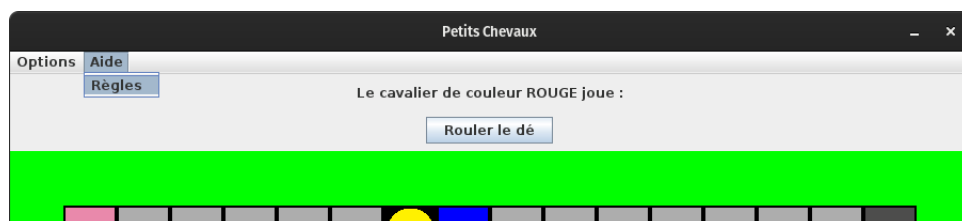
Quitter :



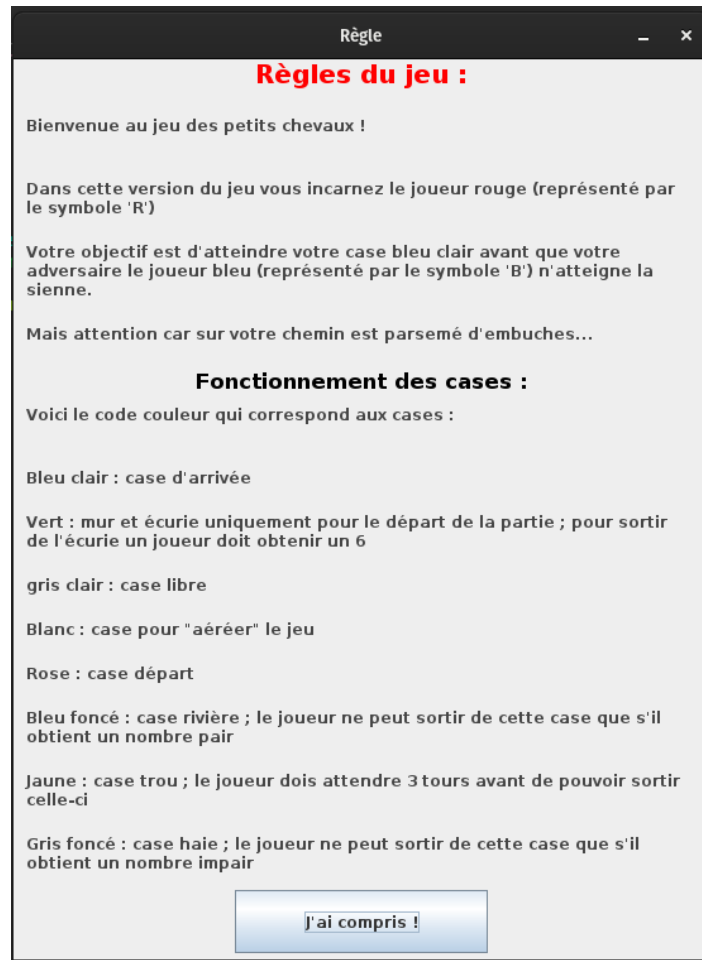
Recommencer :



Vous pouvez aussi accéder aux règles du jeu en cliquant sur "Aide", "Règles" :



Fenêtre d'affichage des règles :



Fonctionnement global :

La fenêtre de jeu : elle est constituée de trois parties qui sont :

- "GTopBoard" le haut de la fenêtre avec le bouton pour rouler le dé ainsi que l'indication du joueur à qui c'est le tour.

- "GGameBoard" le plateau de jeux avec les joueurs ainsi que la valeur du dé.

- "GBotBoard" le bas de la fenêtre avec les légendes des cases où sont les joueurs.

Ces parties de la fenêtre sont assemblé à l'aide de la classe "Controller" en la fenêtre. L'actualisation de ces parties est faite indépendamment et à chaque tour dans la classe "Party" avec la méthode "PlayGraphic".

La partie haute de la fenêtre : cette partie est faite simplement d'un bouton "Rouler le dé" qui se désactive lorsque c'est au tour de l'adversaire, ainsi que d'un "JLabel" qui en se basant sur le numéro du tour indique quel joueur doit jouer.

Le plateau de jeu : le plateau est basé sur le modèle console (comme le prévoit la programmation MVC), il recrée la grille du plateau avec "GridLayout" et en se basant sur la liste de "Cell" du modèle console attribut à chaque cellule de la grille l'équivalent graphique des cellules (les GCell comme GraphicCell) du mode console.

Chaque cellule graphique possède un "JLabel" qui permettra d'actualiser la position des joueurs sans avoir à redessiner tout le plateau de jeu.

L'actualisation du dé se fera de la même façon.

La partie basse de la fenêtre : cette partie est constituée de deux "JLabel" qui en se basant sur les informations de la classe "Board" affiche les significations des cases sur lesquels les joueurs se trouvent.

La classe "Party" : cette classe lance le jeu, si lors de son initialisation dans la classe "Main" on met en paramètre un chiffre entre 1 et 4 inclus la classe "Party" lancera le jeu en mode console avec le niveau indiqué en paramètre (1 étant le niveau facile et 4 le niveau cauchemar). Si le chiffre donné en paramètre est supérieur à 4 alors le programme lancera le jeu en mode graphique et l'utilisateur saisira le niveau de jeu qu'il souhaite.

Chaque partie finis générera un historique d'elle même dans le fichier "Backup.txt".

.

Nos choix et pourquoi :

Points de vue utilisateur :

La difficulté prédéfinis : nous avons décidés de faire une fenêtre avec des cases de difficulté prédéfinis pour rendre le jeu plus "User proof", cela nous permet aussi d'éviter un maximum d'erreur auxquelles nous n'aurions pas pensé et de devoir perdre beaucoup de temps à tester toutes les différentes configurations possibles, en revanche avec une fenêtre de saisie le programme gère parfaitement tout type de paramètres entré du moment qu'ils sont "justes".

Le jeu n'a pas beaucoup d'options : ne pas mettre beaucoup d'options et aller à l'essentiel est aussi un choix, d'abord parce que plus le programme est simple plus il est stable, mais aussi parce que chaque option prend beaucoup de temps à intégrer proprement au code.

Point de vue développement :

La variable public static "sizeCell" dans la classe "GGameBoard" : mettre cette variable en "public static" permet à toutes les classes de type "GCell" d'avoir la taille par défaut d'une cellule et ainsi de pouvoir homogénéiser la taille des cellules de de pouvoir centrer les joueurs dans les cases où ils passeront.

Le "BackupParty.txt" : au départ nous voulions utiliser le principe de cette option pour pouvoir sauvegarder chaque tour indépendamment et les "réinjecter" dans le jeu lorsque le joueur voudrait revenir aux tours précédents, mais finalement par manque de temps nous avons privilégié des options simples. Néanmoins nous avons gardé cette sauvegarde car elle s'est avérée très pratique pour le débogage du programme.

La classe "Button" : cette classe est la même pour tous les boutons excepté le bouton "rouler le dé", elle est simple est utilisable partout car elle retourne simplement si on a cliqué sur le bouton ou non, cela permet de n'avoir qu'une classe pour plusieurs boutons.

Les commentaires en anglais : par principe nous faisons tous nos commentaires en anglais, nous nous excusons d'avances pour ceux qui ne seront pas écrits dans un anglais correct.

Ce que nous n'avons pas fait et pourquoi :

Le système de sauvegarde et de retour au tours précédents : comme expliqué précédemment le système de sauvegarde n'a pas été fait pour des raisons de manque de temps.

Le système de placement des pièges aléatoire sur les cases du milieu : le placement des pièges aléatoirement au milieu n'a pas été fait car la version de base était trop instable.

Ce qui beug ou qui marche moyennement :

Petit beug à l'actualisation : lors de l'actualisation du jeu à chaque tour si le cavalier rouge est dans l'écurie on remarque un petit beug d'affichage, ce beug n'influe néanmoins pas le bon fonctionnement du programme.

L'aléatoire sur la disposition des pièges : l'aléatoire est fait de sorte que chaque fois que le programme place un piège le suivant est placé après sur le plateau cela évite les pièges au même endroit et d'avoir à retriier la liste des positions mais si un piège est placé sir la dernière case valide du plateau les suivants prendront automatiquement tous la même position. Cela implique que parfois le programme ne met pas exactement le nombre de pièges désiré sur le plateau.

Conclusion :

Le programme est parfaitement fonctionnel malgré les quelques défauts mineurs cités précédemment, vous trouverez joint à celui-ci une javadoc complète.

Pour vous faire gagner du temps :

Avoir deux joueurs sur une même case lors du cas où un joueur en mage un autre alors que celui-ci est déjà sur sa case départ fonctionne :

Exemple du cas où deux joueurs sont sur la même case :

