Rapport de synthèse

BAH Oumoul Kiramy
RICHARD Georges Khaly
HIBA L Moudden
BAAZZI Osama

7 avril 2022

Complement POO
Groupe 1A

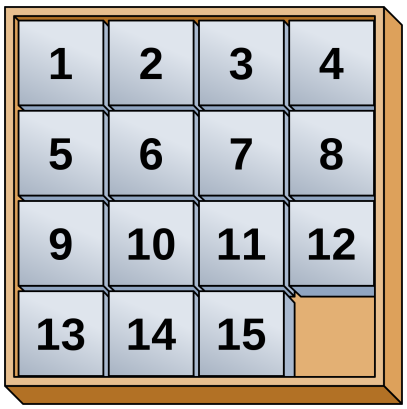

FIGURE 1 – Exemple de taquin résolu

Sommaire

1 Presentation du taquin

Le taquin est un jeu solitaire en forme de damier créé vers 18701 aux États-Unis. Sa théorie mathématique a été publiée par l'American Journal of mathematics pure and applied2 en 1879. En 1891, son invention fut revendiquée par Sam Loyd3, au moment où le jeu connaissait un engouement considérable, tant aux États-Unis qu'en Europe. Il est composé de 15 petits carreaux numérotés de 1 à 15 qui glissent dans un cadre prévu pour 16. Il consiste à remettre dans l'ordre les 15 carreaux à partir d'une configuration initiale quelconque.

Le principe a été étendu à toutes sortes d'autres jeux. La plupart sont à base de blocs rectangulaires plutôt que carrés, mais le but est toujours de disposer les blocs d'une façon déterminée par un nombre minimal de mouvements. Le Rubik's Cube est aujourd'hui considéré comme l'un des « descendants » du taquin. Le but du projet est de réaliser une application de jeu, dotée d'une interface graphique,(mais pouvant être utilisé sans l'interface graphique) qui consiste en un puzzle à glissière comme illustré dans l'exemple figure 1

2 MVC

2.1 Modele

2.1.1 Diagramme de classes

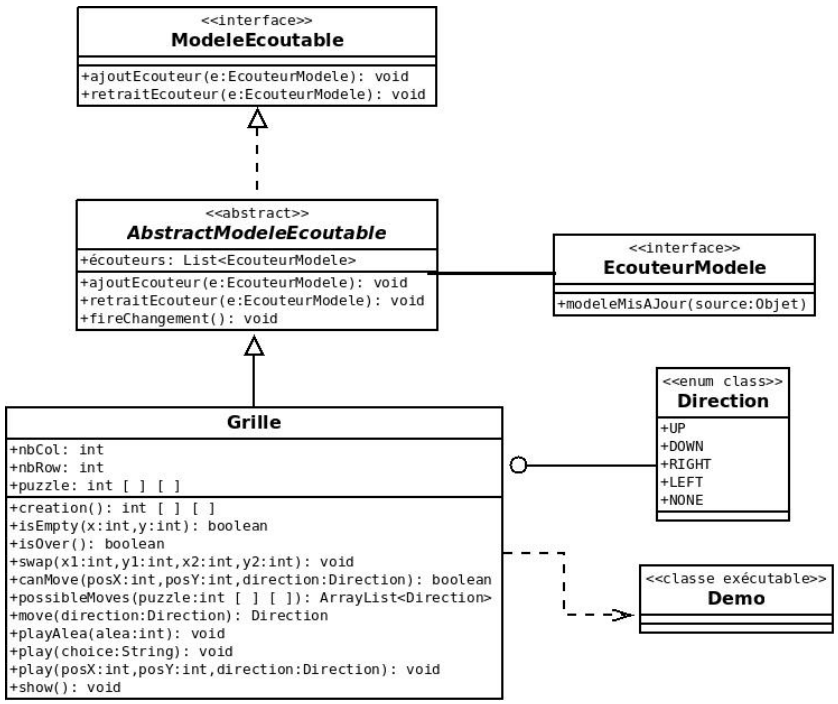

FIGURE 2 – Diagramme de classes du modèle

2.1.2 Fonctionnalité des classes

Dans la partie modele, comme indiqué au diagramme de classe 2 on trouve les deux interfaces ModeleEcoutable et EcouteurModele ainsi que la classe abstraite AbstractModeleEcoutable. Ces derniers permettent de mettre une relation entre la partie Modele et la partie Vue, on peut dire que ça prévient la Vue en cas de changement du Modele comme ça les changements qu'on trouve dans le terminal vont être visualisés.

Ensuite j'ai une classe enum Direction qui donne les directions possibles dans le jeu. Puis la classe Grille qui contient l'ensemble des méthodes pour ouvoir jouer au taquin. La première fonction était pour créer une matrice de nombre rempli de nombres successifs à partir de 1, et le dernier nombre on le remplace par 0 comme ça il représente le trou. Après il y a une fonction qui vérifie si la case est un trou ou pas, une fonction pour déterminer la fin du jeu, et les fonctions des mouvements et des plays. Dans une de ces fonctions, on prévient la vue que le modèle change en utilisant le fireChangement issu de la classe abstraite AbstractModeleEcoutable. Et enfin on trouve la méthode show() qui permet de visualiser les changements de la grille dans le terminal.

Pour jouer au terminal, j'ai ajouter une méthode play qui prend en paramètre une chaîne de caractère ou String, qui joue le mouvement entré en paramètre s'il est possible, et pour mélanger le taquin qui est en ordre au début, j'ai créé une méthode playAlea qui prend en paramètre un entier n, elle permet de faire n mouvements aléatoires dans le taquin comme ça le puzzle va être en désordre et l'utilisateur essayera de le résoudre. C'était ces méthodes que j'ai utilisé dans la Demo pour rendre mon jeu fonctionnel et jouable au sein du terminal.

2.2 Vue et controleur

- La partie vue du jeu de taquin permet d'avoir une version visuelle du jeu, une image découpée en plusieurs morceaux. Cette partie du code est chargée de dessiner toute la grille du jeu à des carrés semblables, de contrôler le jeu avec les touches enfoncées sur le clavier et avec les clics de souris.Elle implement l'interface EcouteurModele (voir fig 2) .Elle est gerée par la class PanelJeu et comporte diverses fonctionnalités d'affichage de la grille.
- La partie controle du jeu est chargé de gerer les differents mouvements des boutons. Les classes Panelscontroles et BoutonGrille contiennent

les fonctionnalités de controle.
Pour deplacer un bouton le controleur est implémenté de tel sorte qu'il suffit de cliquer sur le bouton et de d'utiliser les boutons de directions pour les déplacements .

3 Fonctionnalités de l'interface graphique et mode de fonctionnement

L'interface graphique comporte diverses fonctionnalités. La principale est de donner à l'utilisateur la possibilité de sauvegarder et de charger la dernière partie jouée. Elle s'avere très pratique dans la mesure où l'on est pas obligé de finir la partie lancée. Dans l'interface graphique, le contrôle sera fait de deux façons simultanées :

- un clic sur l'élément que l'on souhaite déplacer (s'il est déplaçable). l'élément selectionné aura une bordure noire.
- Un appui sur l'une des 4 flèches du clavier (par ex. la flèche de gauche signifie que l'on veut déplacer l'élément situé à gauche). L'élément gardra sa position s'il ne peut pas etre deplacé dans la direction choisi.

Le jeu peut egalement etre joué dans un terminale et les deplacement s'effectuent en choisissant juste la direction dans la quelle vous souhaitez bouger la case vide. Le melange aleatoire de la Grille est implémenteé de sorte à faire une permutation entre les cases adjacentes de la case vide et la case vide . Ce faisant , le jeu reste solvable. Une amelioration possible serait de pouvoir couper des images et en faire des puzzle.