

Rapport Interfaces Graphiques et Design Patterns

Sujet : Jeu de Puzzle à glissières



Sous la responsabilité
De
MATHET Yann

Réalise par
Thierry LOISEAU
Ahmed Oussama Ben Daou
Nathan Lambertz
Elias Kefi



Université de Caen Normandie
UFR des Sciences
L2 Informatique

Table des matières

1	Un peu d'histoire	2
2	Introduction	3
2.1	Qu'est ce qu'un Puzzle à glissières Taquin ?	3
2.2	But de notre jeu	4
3	Objectifs du projet	4
4	Conception	4
4.1	Partie backend :	4
4.1.1	models.piece	5
4.1.2	models.position	5
4.1.3	Mélange du puzzle	6
4.1.4	Téchnique de découpe d'une image	6
4.1.5	Contrôleur de l'application	6
4.2	Partie frontend :	6
5	Architecture du projet	7
6	Diagramme UML	8
7	Rendu	9
7.1	Menu	9
7.2	Puzzle Classique	10
7.3	Puzzle Image	11

1 Un peu d'histoire

Les jeux de puzzle ont une longue histoire, remontant à des siècles, avec des exemples emblématiques tels que le Rubik's Cube et le Tangram, qui ont captivé l'imagination des joueurs du monde entier. Ces jeux ont non seulement divertis, mais aussi stimulé la réflexion, la résolution de problèmes et la créativité.

En créant notre propre jeu de puzzle à glissières, nous avons cherché à offrir une expérience moderne et accessible, tout en restant fidèles aux principes fondamentaux qui font le charme intemporel des jeux de puzzle. Notre objectif était de concevoir un jeu simple à comprendre, mais difficile à maîtriser, offrant ainsi un défi gratifiant pour les joueurs de tous niveaux.

Pour en apprendre davantage sur l'histoire des jeux de puzzle, nous nous sommes appuyés sur des sources variées, telles que des articles de recherche, des livres et des ressources en ligne. Ces références nous ont permis de mieux comprendre le contexte historique et les avantages des jeux de puzzle, et ont guidé notre approche dans le développement de notre propre jeu.

En résumé, notre jeu de Puzzle à glissières s'inscrit dans une longue tradition de jeux de puzzle et cherche à offrir aux joueurs une expérience captivante et enrichissante. En combinant la fascination pour les jeux de puzzle avec les défis de la conception logicielle, nous avons créé un projet qui nous passionne et que nous sommes impatients de partager avec le monde.

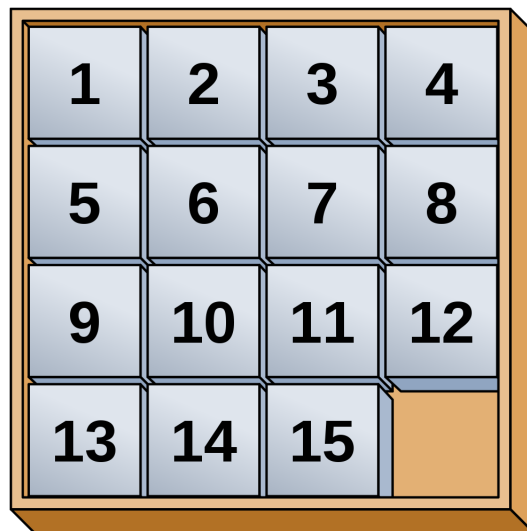
2 Introduction

Dans le cadre de notre projet d’Interfaces Graphiques, nous avons formé un groupe de 4 personnes : **KEFI Elias et LAMBERTZ Nathan** pour le front-end et **Ahmed Oussama BEN DAOU et LOISEAU Thierry** pour le back-end ; Et nous avons découvert le sujet de ce projet qui est : Puzzle à glissières (Taquin).

Dans un premier temps, le sujet nous a paru plutôt simple mais en réalité, il y a plusieurs aspects compliqués. Par exemple, la séparation d’une image en plusieurs morceaux de puzzle, mélange de toutes les pièces du puzzle d’une manière aléatoire, etc ...

2.1 Qu’est ce qu’un Puzzle à glissières Taquin ?

Le Puzzle à glissières, communément appelé Taquin, est un jeu de réflexion à la fois simple et complexe. Ce puzzle a été créé vers 1870 aux États-Unis par Sam Loyd, un compositeur américain de casse-tête numériques et de jeux de logique mathématique. À l’origine, le but de ce jeu était de reconstituer une suite de 15 nombres mélangés sur une grille de 16 morceaux. Voici à quoi pouvait-t-il ressembler : [?]



2.2 But de notre jeu

Dans notre jeu, l'objectif est de reconstituer une image divisée en plusieurs morceaux de taille plus ou moins grande en faisant glisser les pièces. Pour rendre cela possible, une pièce est manquante, ce qui permet aux autres pièces de se déplacer dans l'espace vide. Cette pièce manquante est toujours celle située tout en bas à droite de ce puzzle.

3 Objectifs du projet

Lors de la réalisation de notre projet, notre objectif était de créer une application possédant une interface graphique pour ce puzzle à glissières ; Elle devait être contrôlable à la fois avec la souris et les touches du clavier (W, S, A, D).

Si possible, nous souhaitons mettre en évidence les coups possibles, par exemple en utilisant un contour vert, remplacer les chiffres de la grille par des images, et éventuellement ajouter un compteur de coups.

4 Conception

Nous nous sommes divisés en deux binômes pour réaliser ce projet, un groupe s'occupant de la partie "backend" en créant le puzzle, les mouvements, la MVC etc ...

L'autre groupe s'est occupé de la partie affichage en dessinant les pièces du puzzle et l'intégration des images à la place des cases numérotées.

4.1 Partie backend :

Dans cette section, nous allons détailler la conception de la partie backend de notre jeu de Puzzle à glissières (Taquin). Cette partie est responsable de la logique métier du jeu, y compris la représentation des pièces et des positions sur la grille.

Commençons par parler des modèles qui jouent un rôle essentiel dans la conception de notre jeu de Puzzle à glissières.

4.1.1 models.piece

Classe NumericPieceModel

La classe NumericPieceModel représente une pièce individuelle dans notre jeu de puzzle. Chaque pièce est caractérisée par une valeur numérique, une position actuelle sur la grille, une position finale à atteindre, et peut être soit une pièce normale, soit la case vide (trou) du puzzle.

La classe NumericPieceModel est dotée de méthodes pour obtenir et définir sa valeur, sa position actuelle et sa position finale. Elle fournit également des fonctionnalités pour vérifier si elle représente la case vide du puzzle et si elle est à sa position finale sur la grille.

Classe PieceOfImageModel

La classe PieceOfImageModel représente une pièce d'image individuelle dans notre jeu de puzzle. Chaque pièce est caractérisée par une valeur BufferedImage, une position actuelle sur la grille, une position finale à atteindre, et peut être soit une pièce normale, soit la case vide (trou) du puzzle.

La classe PieceOfImageModel est dotée de méthodes pour obtenir et définir sa valeur, sa position actuelle et sa position finale. Elle fournit également des fonctionnalités pour vérifier si elle représente la case vide du puzzle et si elle est à sa position finale sur la grille.

4.1.2 models.position

Classe Position

La classe Position représente une position sur la grille de jeu. Chaque position est définie par ses coordonnées en termes de lignes et de colonnes.

La classe Position offre des méthodes pour obtenir et définir ses coordonnées horizontales et verticales. Elle est utilisée pour déterminer la position actuelle des pièces sur la grille.

Ces classes sont essentielles pour la logique interne du jeu de puzzle à glissières. Elles sont conçues pour être indépendantes de toute interface utilisateur, ce qui les rend flexibles et réutilisables dans différents contextes d'application.

4.1.3 Mélange du puzzle

Pour avoir un puzzle de taille $n \times n$ bien mélangé, on a besoin de $2^{2 \times n \times n}$ fois de changement aléatoire de pièces.

4.1.4 Technique de découpe d'une image

On a défini par défaut une constante de valeur 4, qui permet de découper n'importe quelle image en un puzzle de dimension de 4×4 et avec cette découpe on obtient la taille d'une pièce, qui permet de dessiner les pièces du puzzle.

4.1.5 Contrôleur de l'application

SwingPuzzleController

La classe SwingPuzzleController est un Contrôleur qui implémente les interfaces suivantes `KeyListener`, `MouseMotionListener`, `MouseListener`; la classe SwingPuzzleController permet de gérer les interactions de l'utilisateur, les touches du clavier, et le mouvement de la souris.

4.2 Partie frontend :

Dans cette deuxième partie, nous allons vous parler de la partie "frontend" de notre projet : Puzzle à glissières (Taquin). Cette partie est chargée de la présentation visuelle du jeu, y compris l'affichage des pièces et des positions sur l'interface utilisateur.

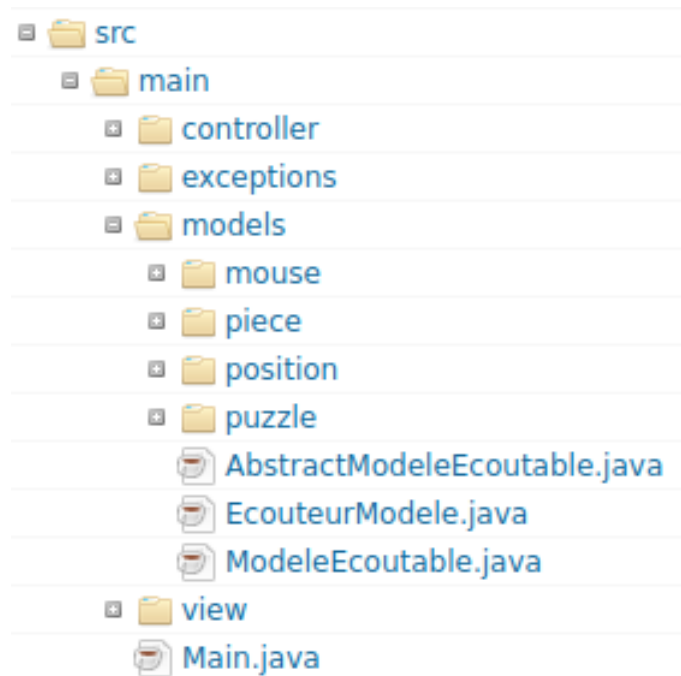
PuzzleView

La class PuzzleView est responsable de la représentation visuelle du puzzle et de la gestion des interactions utilisateur. En implémentant les interfaces (KeyListener,...) elle peut prendre en charge les mouvement souris et touches du clavier. PuzzleView permet également d'intégrer le modèle de notre puzzle (PuzzleModel). Et par conséquent PuzzleView utilise PuzzleModel pour créer un affichage fluide et concret.

MainView

MainView est une sous-classe de JFrame. Elle construit la fenêtre du menu principal, dedans elle crée des boutons. Elle associe des actions aux boutons et définit la logique de construction et d'affichage du puzzle. En s'aidant de PuzzleView, elle gère l'entièreté de la partie graphique.

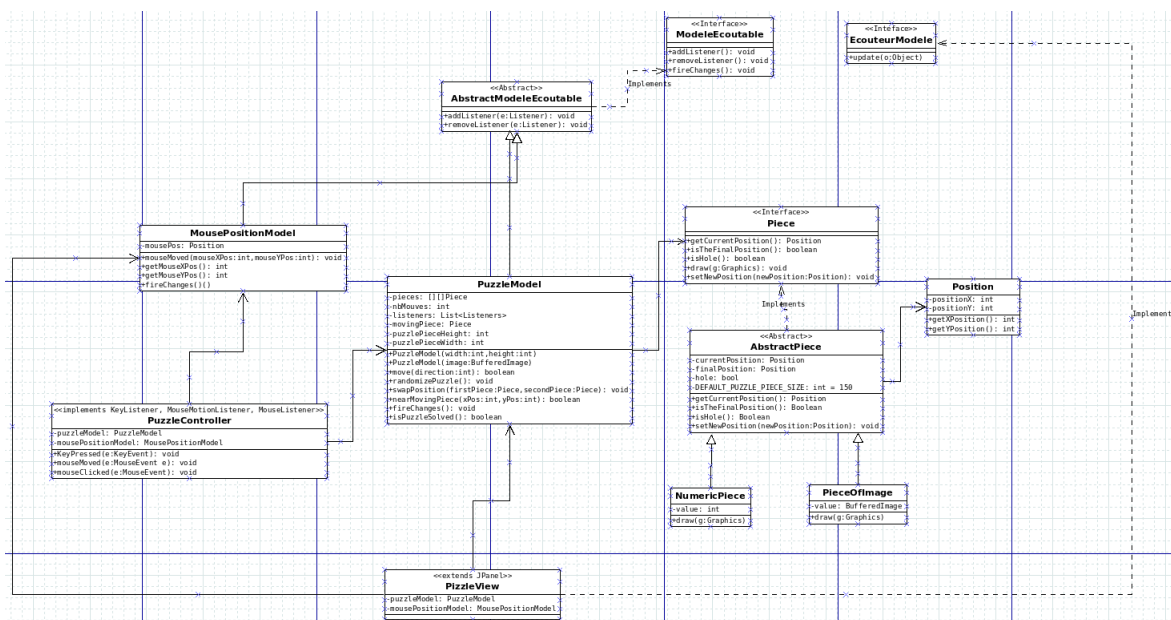
5 Architecture du projet



L'architecture était une phase très importante dans notre projet pour avoir un projet bien structuré. Dans le dossier src on a le code du projet. avec les sous-dossiers ci-dessous :

6 Diagramme UML

Voici notre diagramme UML complet de notre projet :



7 Rendu

7.1 Menu

Ces différents puzzles sont accessible via un menu composé de 3 boutons :

- Classic Puzzle : Lancer le puzzle à cases numérotées.
- Image Puzzle : Lancer la fenêtre pour choisir une image à importé avant de jouer.
- Quit : quitte le jeu.



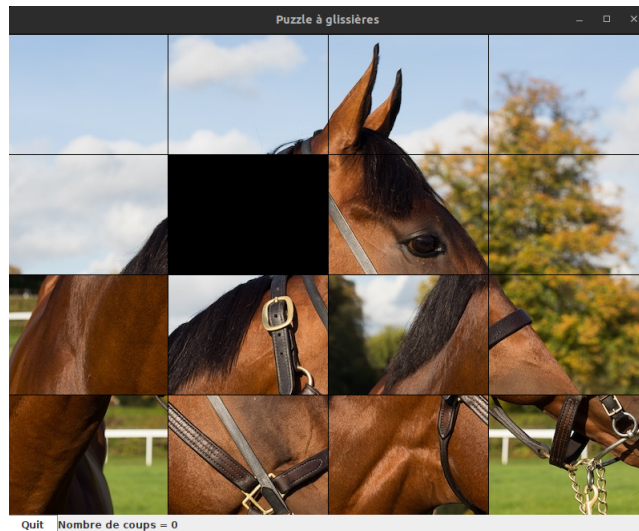
Il y a deux types de puzzle à jouer dans notre application. Un puzzle classique avec des cases numérotées de fonction de la taille du puzzle et un puzzle qui peut se jouer avec une image à importer.

7.2 Puzzle Classique



Avec cette version du puzzle vous pouvez choisir la taille du puzzle au moment de le lancer. En rentrant un entier n , le tableau sera de taille $n \times n$. Les cases qui peuvent être déplacées virent au vert quand la souris passe au-dessus.

7.3 Puzzle Image



La seule différence avec le puzzle classique est que celui-ci n'est pas réglable en taille. Il sera par 4*4 peut-importe l'image importée. Les cases survolées passent aussi au vert, ce qui rend l'image en question cachée par cette couleur mais rend le jeu plus pratique. Les puzzles indiquent aussi en bas de la fenêtre le nombre de coups effectués depuis le début de la partie.

Références

- [1] Creating a GUI With Swing