

UNIVERSITÉ CAEN NORMANDIE

Rapport Interfaces Graphiques et Design Patterns (L2 informatique)

Soumis par :

Emilien Huron
Florian Pépin
Tom David
Imen Boukraa

14 avril 2024
Campus 2 CAEN
2nd Semestre



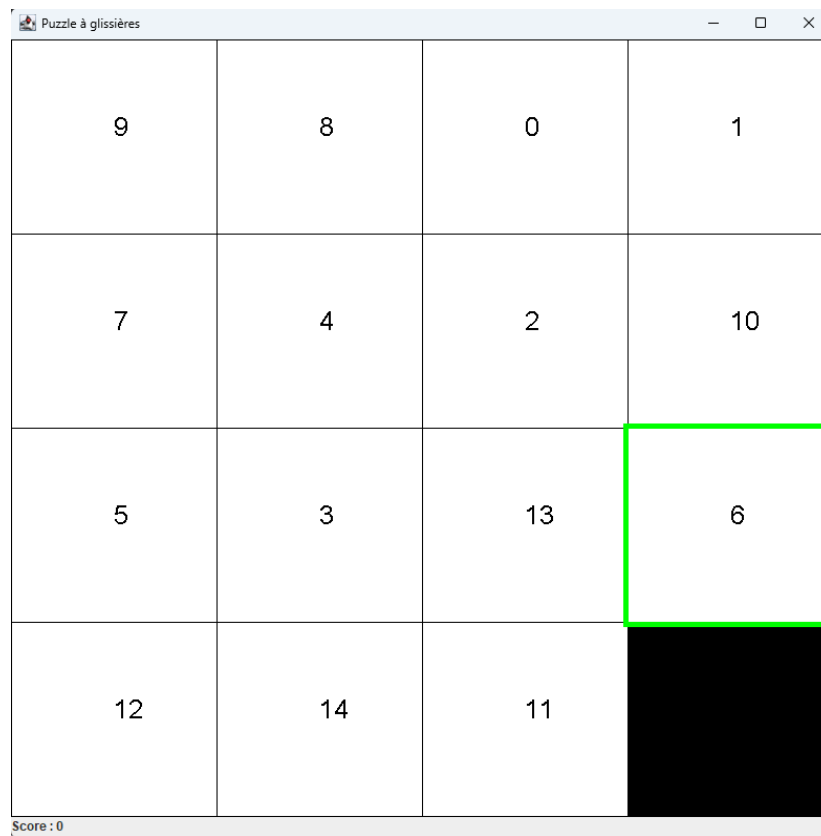
UNIVERSITÉ
CAEN
NORMANDIE

Table des matières

1	Introduction	1
2	Diagramme du projet	2
3	Fonctionnalités implémentées	3
3.1	Classe ImageSplit	3
3.2	Classe GrilleGui	3
3.3	Classe VueScore	4
3.4	Classe VueTerminal	4
3.5	Classe VueGrille	5
3.6	Classe Modele	5
4	Algorithmes	6
5	Répartition des tâches	7
6	Conclusion	7

1 Introduction

Le jeu de Taquin, également connu sous le nom de Puzzle à Glissières, est un jeu de réflexion qui se joue sur une grille variable, souvent arrangée en damier. Chaque case de la grille contient une pièce numérotée ou un fragment d'image, à l'exception d'une case vide qui permet de déplacer les pièces adjacentes. Ce projet vise à développer une application du Taquin offrant à la fois une interface graphique et une interface en ligne de commande. Inspirée du concept classique, notre application permet des déplacements intuitifs des pièces et introduit des fonctionnalités améliorant l'expérience utilisateur.



Taquin

2 Diagramme du projet

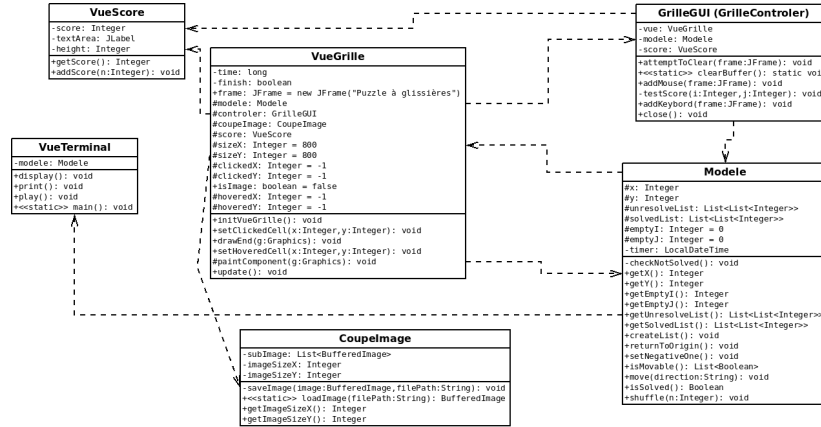


FIGURE 1 – UML - taquin

L'architecture de notre projet de Taquin est fondée sur le modèle de conception Modèle-Vue-Contrôleur (MVC), une approche structurée qui sépare l'application en trois composants principaux : le modèle, la vue, et le contrôleur. Cette séparation favorise la modularité et la flexibilité, permettant à l'équipe de développer, et maintenir chaque partie de manière plus efficace et indépendante.

3 Fonctionnalités implémentées

3.1 Classe ImageSplit

La classe ImageSplit joue un rôle crucial dans la préparation des puzzles d'images. Cette classe est conçue pour découper une image en plusieurs petites images qui s'assemblent pour former la grille du jeu de Taquin. Elle utilise une méthode qui prend une image source et la divise en fonction du nombre de lignes et de colonnes spécifiées, créant ainsi un ensemble de tuiles que le joueur peut ensuite déplacer sur la grille. La classe prend une image, la découpe en $x*y$

3.2 Classe GrilleGui

La classe GrilleGui est le contrôleur principal du jeu, facilitant l'interaction entre l'utilisateur et le modèle du jeu, elle gère les interactions de l'utilisateur avec le jeu. Elle écoute et répond aux clics de souris et aux pressions sur les touches du clavier. Par exemple, un clic de souris sur une tuile déplace cette tuile si un mouvement est possible, et les touches d'irrationnelles permettent de déplacer les tuiles vers le haut, le bas, la gauche ou la droite. Cette classe assure que les actions de l'utilisateur sont transformées en modifications sur la grille de jeu.

3.3 Classe VueScore

La classe VueScore gère l’affichage des scores dans l’interface utilisateur du jeu. Elle utilise un JLabel pour montrer le score actuel, qui est mis à jour à chaque action valide du joueur. Cette mise à jour permet de garder le joueur informé de ses performances en temps réel quand il est sur l’interface graphique.

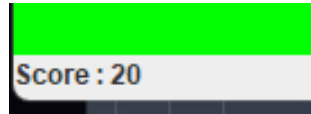


FIGURE 2 – Score

3.4 Classe VueTerminal

La classe VueTerminal fournit une option alternative pour jouer au jeu en utilisant une interface en ligne de commande. Elle montre la grille de jeu directement dans le terminal et permet aux joueurs d’entrer des commandes pour bouger les tuiles. Cette manière de jouer est idéale pour ceux qui préfèrent une approche plus classique ou pour ceux qui n’ont pas accès à une interface graphique.

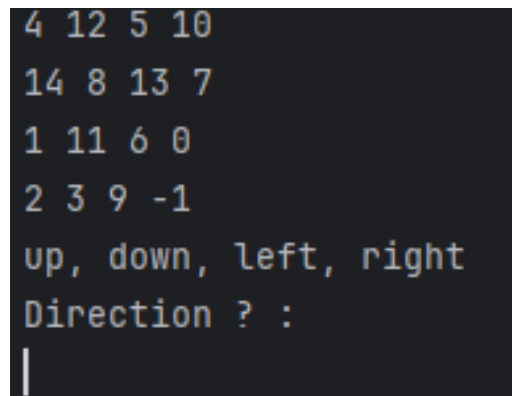


FIGURE 3 – Score

3.5 Classe VueGrille

La classe VueGrille joue un rôle central dans l'interface utilisateur du jeu de Taquin. Elle affiche la grille du jeu, où chaque tuile peut être soit un numéro, soit un fragment d'une image plus grande. Cette classe est également responsable de la gestion des interactions de l'utilisateur avec la grille, mettant à jour l'affichage en conséquence. Elle assure que l'interface utilisateur est intuitive et réactive, offrant une expérience de jeu fluide.

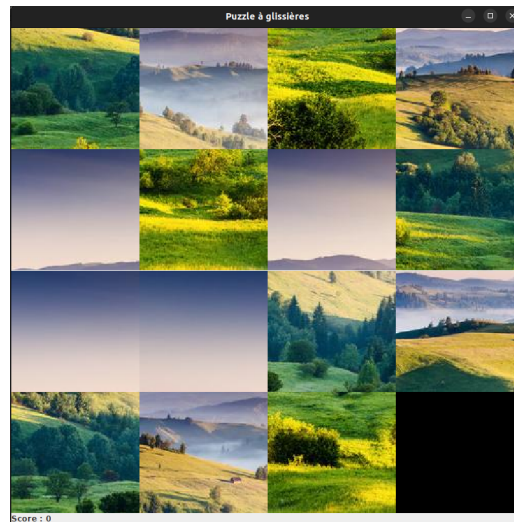


FIGURE 4 – taquin avec image découpé

3.6 Classe Modele

La classe Modele gère tout ce qui concerne les positions des tuiles sur la grille, y compris l'emplacement de la case vide qui permet de bouger les tuiles. Voici comment elle fonctionne :

1. **Suivi des Tuiles :** Elle tient à jour deux listes principales : l'une pour l'arrangement actuel des tuiles (`unresolveList`) et l'autre pour l'arrangement qui représente la solution (`solvedList`). Ces listes aident à contrôler les déplacements des tuiles et à vérifier si le jeu est résolu
2. **Préparation du jeu :** Quand le jeu commence, Modele met en place la grille selon les dimensions choisies (nombre de lignes et de colonnes) et mélange les tuiles pour créer un nouveau puzzle à chaque

partie. Sachant que le mélange n'est pas un simple shuffle, mais bien des déplacement de case, pour éviter de trouver des jeu non solvable.

3. **Déplacement des Tuiles :** Cette classe permet de bouger les tuiles autour de la case vide. Elle vérifie si un mouvement est possible (vers le haut, le bas, la gauche ou la droite) et effectue ce mouvement si c'est le cas.
4. **Vérification de la Victoire** Vérification de la Victoire : La fonction (isSolved) regarde si l'arrangement actuel des tuiles correspond 'a la solution pour déterminer si le joueur a gagné.
5. **Assurance de la Solvabilité :** Modele s'assure que le jeu peut toujours revenir a un état solvable apres avoir melange les tuiles ou apres que les joueurs aient fait des déplacements.

4 Algorithmes

Cette partie du jeu utilise plusieurs algorithmes clés pour fonctionner correctement :

1. **Mélange des tuiles :** Avant chaque partie, les tuiles sont mélangées de manière aléatoire pour préparer le terrain de jeu. Cet algorithme de mélange utilise une méthode assurant que le puzzle généré soit solvable, tout en étant suffisamment complexe pour offrir un défi intéressant. Il évite les configurations initiales qui conduiraient à une solution rapide ou immédiate, préservant ainsi l'intérêt et l'engagement du joueur tout au long de la partie.
2. **Détection de la victoire :** Cet algorithme vérifie constamment si les tuiles sont dans le bon ordre pour terminer le puzzle. A chaque fois que le joueur déplace une tuile, l'algorithme vérifie si le jeu est résolu. Si toutes les tuiles sont correctement placées, le jeu annonce immédiatement la victoire. Cela déclenche aussi des animations ou des messages de félicitations, ce qui rend la victoire encore plus gratifiante pour le joueur.

5 Répartition des tâches

1. **Tom David (85%)** : Participation au PDF, Développement global.
2. **Florian Pépin (85%)** : Participation au PDF, Développement global.
3. **Emilien Huron (65%)** : Participation au PDF, Développement découpe image et survole des cases.
4. **Imen Boukraa (10%)** : Participation au PDF.

6 Conclusion

En guise des conclusions, ce projet de jeu de Taquin illustre efficacement l'application du modèle Modèle-Vue-Contrôleur (MVC) pour créer une expérience utilisateur engageante et fonctionnelle. Les différentes classes développées, notamment *Modele*, *VueGrille*, *VueScore*, et *VueTerminal*, travaillent en harmonie pour offrir à la fois une interface graphique et une interface en ligne de commande. Cela permet de répondre aux préférences de différents utilisateurs.