

Presentation du jeu Taquin

Annou Rayane
Mohamed Moudir
Madjid Selmane
Mayas Ould Kaci

April 15, 2024



1 Introduction

Ce rapport présente une analyse détaillée des fonctionnalités des classes qui sont utilisées pour la mise en œuvre d'un jeu de Taquin. Et pour le cote graphique on a utilisées aussi des classes `TaquinController` et `TaquinView`. Le jeu de Taquin est un casse-tête coulissant dans lequel le joueur doit réorganiser des tuiles numérotées dans un cadre en déplaçant une case vide.



Figure 1: Exemple de jeu Taquin

2 Package Modele

2.1 Classe Taquin

La classe **Taquin** représente l'état du jeu de Taquin et contient les fonctionnalités nécessaires pour manipuler et résoudre le puzzle. Voici un aperçu des fonctionnalités principales de la classe **Taquin** :

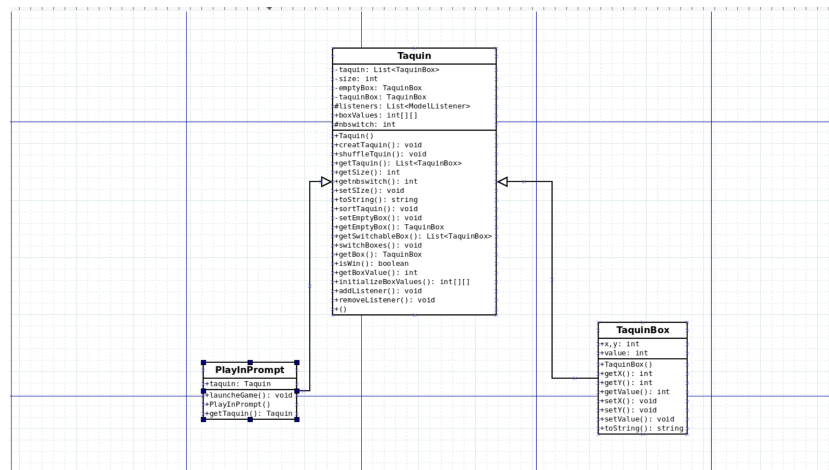


Figure 2: Classe Taquin

2.1.1 Constructeurs

- `public Taquin(int size):` Construit un objet **Taquin** avec une taille spécifiée.
- `public Taquin():` Construit un objet **Taquin** avec une taille par défaut.

2.1.2 Méthodes principales

- `public List<TaquinBox> getTaquin():` Renvoie la liste des boîtes du Taquin.
- `public int getSize():` Renvoie la taille du Taquin.

2.1.3 Méthodes auxiliaires

- `private void creatTaquin():` Initialise le Taquin avec des boîtes numérotées.

- `private void shuffleTquin()`: Mélange aléatoirement les boîtes du Taquin.

2.1.4 Méthodes pour les écouteurs

- `@Override public void addListener(ModelListener e)`: Ajoute un écouteur au Taquin.
- `@Override public void removeListener(ModelListener e)`: Supprime un écouteur du Taquin.

2.2 Classe PlayInPrompt

La classe `PlayInPrompt` est responsable de l'interaction avec l'utilisateur via la console et de la gestion du déroulement du jeu. Voici les principales fonctionnalités de cette classe :

2.2.1 Attributs

- `private Taquin taquin`: Le Taquin associé à la partie en cours.

2.2.2 Constructeur

- `public PlayInPrompt(Taquin taquin)`: Constructeur qui initialise un objet `PlayInPrompt` avec le Taquin spécifié.

2.3 Méthodes

- `public void lanceGame()`: Méthode qui lance le jeu de Taquin et gère les interactions avec l'utilisateur jusqu'à ce que la partie soit gagnée.
- `public Taquin getTaquin()`: Méthode qui renvoie le Taquin associé à la partie.

3 Package vue

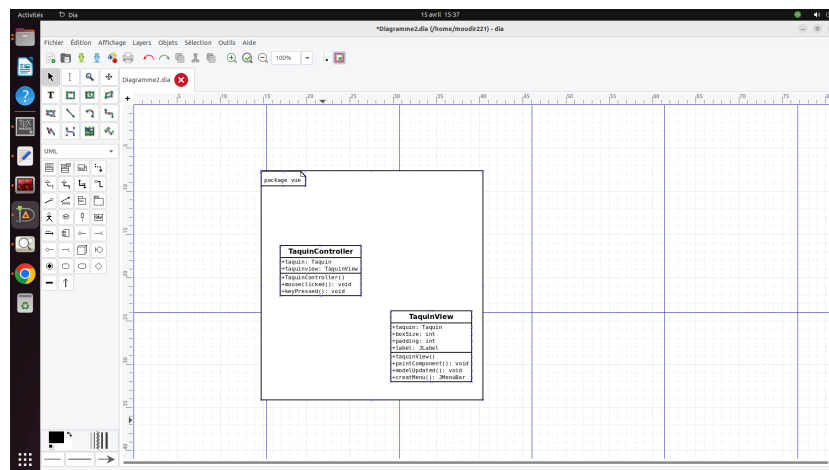


Figure 3: package Vue

3.1 Classe taquinView

3.1.1 Responsabilité

La classe `taquinView` est responsable de l’affichage du jeu de Taquin dans une interface graphique.

3.1.2 Attributs

- `private Taquin taquin`: Cet attribut représente le modèle du jeu de Taquin associé à la vue.
- `protected int boxSize`: La taille d'une case dans l'affichage.
- `protected int padding`: L'espace entre les cases dans l'affichage.
- `private JLabel label`: Un composant JLabel utilisé pour afficher le nombre de coups effectués.

3.1.3 Constructeur

Le constructeur prend en paramètre un objet **Taquin** qui représente le modèle à afficher. Il initialise les attributs de la classe et enregistre la vue comme observateur du modèle.

3.1.4 Méthode `paintComponent(Graphics g)`

Cette méthode est appelée pour dessiner les composants graphiques. Elle est utilisée pour afficher le plateau de jeu avec ses cases et leurs valeurs.

3.1.5 Méthode `modelUpdated(Object source)`

Cette méthode est appelée chaque fois que le modèle est mis à jour. Elle redessine la vue pour refléter les changements et met à jour le texte du label pour afficher le nombre de coups effectués.

3.1.6 Méthode `createMenu(JFrame frame)`

Cette méthode crée une barre de menu permettant à l'utilisateur de sélectionner la taille du Taquin. Elle prend en paramètre le cadre `JFrame` auquel la barre de menu sera ajoutée et renvoie la barre de menu créée.

3.2 Classe `TaquinController`

3.2.1 Responsabilité

La classe `TaquinController` est responsable de la gestion des interactions entre l'utilisateur et le jeu de Taquin.

3.2.2 Attributs

- `private Taquin taquin`: Cet attribut représente le modèle du jeu de Taquin associé au contrôleur.
- `private taquinView taquinView`: Cet attribut représente la vue du jeu de Taquin associée au contrôleur.

3.2.3 Constructeur

Le constructeur prend en paramètre un objet `taquinView` et un objet `Taquin`, puis initialise les attributs de la classe. Il crée également une fenêtre `JFrame` pour afficher la vue et ajoute la vue à cette fenêtre. Enfin, il ajoute le contrôleur en tant qu'écouteur de la souris et du clavier pour la vue.

4 Package observer

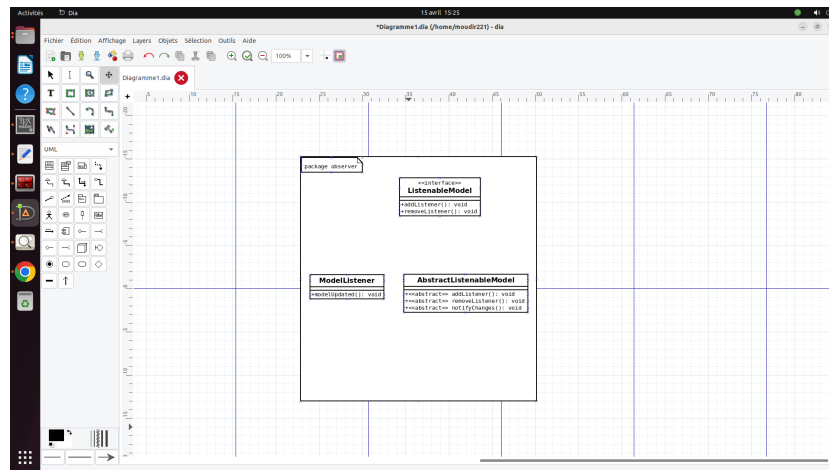


Figure 4: Package observer

4.1 Classe AbstractListenableModel

4.1.1 Responsabilité

La classe abstraite `AbstractListenableModel` fournit une implémentation de base pour un modèle écoutable. Elle permet d'ajouter, de supprimer des écouteurs et de notifier les changements à ces écouteurs.

4.1.2 Méthodes

- `public abstract void addListener(ModelListener listener):` Méthode abstraite permettant d'ajouter un écouteur à ce modèle.
- `public abstract void removeListener(ModelListener listener):` Méthode abstraite permettant de supprimer un écouteur de ce modèle.
- `public abstract void notifyChanges():` Méthode abstraite permettant de notifier tous les écouteurs enregistrés que le modèle a été mis à jour.

4.2 Interface `ListenableModel`

4.2.1 Responsabilité

L'interface `ListenableModel` définit les méthodes pour ajouter et supprimer des écouteurs à un modèle.

4.2.2 Méthodes

- `public void addListener(ModelListener listener)`: Méthode permettant d'ajouter un écouteur à ce modèle.
- `public void removeListener(ModelListener listener)`: Méthode permettant de supprimer un écouteur de ce modèle.

4.3 Interface `ModelListener`

4.3.1 Responsabilité

L'interface `ModelListener` définit une méthode pour notifier les écouteurs lorsqu'un modèle est mis à jour.

4.3.2 Méthodes

- `public void modelUpdated(Object source)`: Méthode appelée pour notifier tous les écouteurs enregistrés que le modèle a été mis à jour. Le paramètre `source` spécifie l'objet qui a déclenché la mise à jour du modèle.

5 Interface graphique

L'interface graphique du jeu de puzzle Taquin est développée en Java, utilisant la bibliothèque Swing pour la conception de l'interface utilisateur. Le code est structuré en trois composants principaux : la vue, le contrôleur et le modèle.

5.1 Vue (`TaquinView`)

La classe `TaquinView` est responsable de l'affichage graphique du jeu. Elle hérite de `JPanel` et implémente l'interface `ModelListener` pour écouter les mises à jour du modèle. Elle utilise les fonctionnalités de Swing pour dessiner les cases du puzzle et afficher le nombre de coups effectués. Voici un aperçu des fonctionnalités principales de cette classe :

- Utilisation de la méthode `paintComponent(Graphics g)` pour dessiner les cases du puzzle en fonction de l'état actuel du modèle.
- Mise à jour de l'affichage à chaque modification du modèle grâce à la méthode `modelUpdated(Object source)`.

5.2 Contrôleur (TaquinController)

Le contrôleur, représenté par la classe `TaquinController`, gère les interactions de l'utilisateur avec l'interface graphique. Il implémente les interfaces `MouseListener` et `KeyListener` pour détecter les clics de souris et les frappes de clavier. Voici les principales fonctionnalités de cette classe :

- Gestion des clics de souris pour permettre au joueur de déplacer les cases du puzzle en cliquant dessus.
- Gestion des frappes de clavier pour permettre au joueur de déplacer les cases en utilisant les touches directionnelles.
- Création d'une barre de menu permettant à l'utilisateur de choisir la taille du puzzle.

5.3 Modèle (Taquin)

Le modèle, représenté par la classe `Taquin`, contient la logique métier du jeu de puzzle Taquin. Il est responsable de la manipulation des cases du puzzle et de la vérification de la condition de victoire. Les principaux aspects de ce modèle incluent :

- Stockage de l'état actuel du puzzle et des opérations permettant de le modifier, telles que le déplacement des cases.
- Vérification de l'état de victoire en comparant l'état actuel avec l'état final du puzzle.

En combinant ces trois composants, l'interface graphique du jeu de puzzle Taquin offre une expérience utilisateur interactive et intuitive.

6 Partie du jeu

Le jeu de puzzle Taquin offre une expérience interactive où les joueurs peuvent choisir la taille du puzzle et tenter de le résoudre en déplaçant les cases pour les remettre dans l'ordre correct. Voici comment se déroule une partie typique du jeu :

6.1 Début de la partie

Au lancement du jeu, une interface graphique s'affiche, permettant au joueur de sélectionner la taille du puzzle. Cette sélection est effectuée via une barre de menu proposant différentes tailles disponibles, telles que 3x3, 4x4, etc.

Une fois la taille choisie, le puzzle est généré avec les cases mélangées aléatoirement, sauf une case vide, créant ainsi un état initial aléatoire.

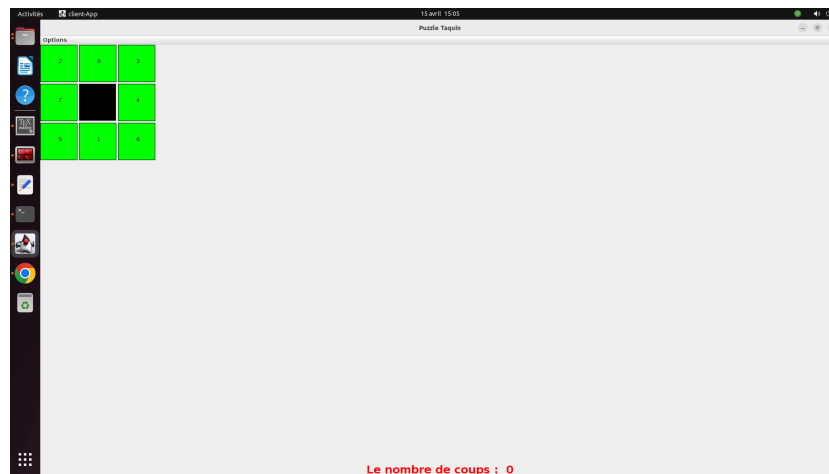


Figure 5: Début de la partie

6.2 Tentative de résolution

Le joueur peut alors commencer à résoudre le puzzle en déplaçant les cases. Deux méthodes sont disponibles pour cela :

- **Clic de souris** : Le joueur peut cliquer sur une case adjacente à la case vide pour la déplacer dans la position vide.
- **Touches du clavier** : Le joueur peut également utiliser les touches directionnelles (z, s, q, d) pour déplacer la case adjacente à la position vide.

Le joueur continue à déplacer les cases jusqu'à ce que le puzzle soit résolu, c'est-à-dire que toutes les cases soient dans l'ordre correct.

6.3 Fin de la partie

Lorsque le puzzle est résolu, un message s'affiche, informant le joueur qu'il a gagné la partie. Ce message peut être une boîte de dialogue.

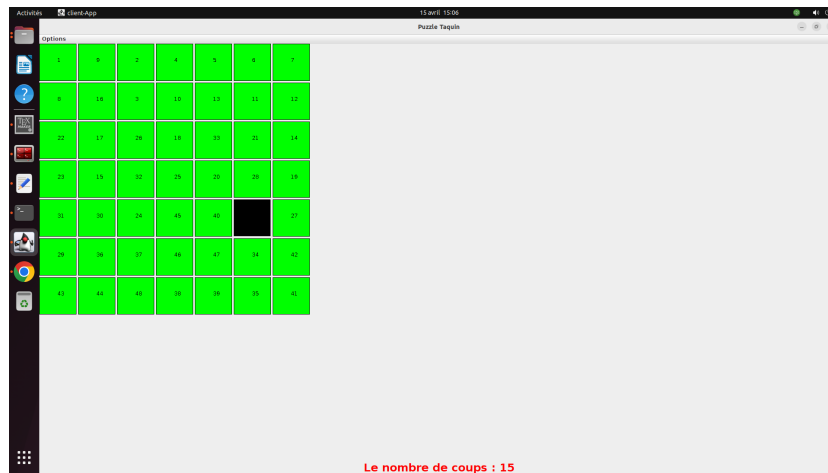


Figure 6: Tentative de résolution

Après avoir affiché le message de victoire, le jeu peut offrir la possibilité de rejouer en générant un nouveau puzzle avec la même taille sélectionnée ou en permettant au joueur de choisir une nouvelle taille pour un défi différent.

En résumé, le jeu de puzzle Taquin offre une expérience divertissante où les joueurs peuvent tester leur résolution de problèmes et leur pensée stratégique pour résoudre des puzzles de différentes tailles.

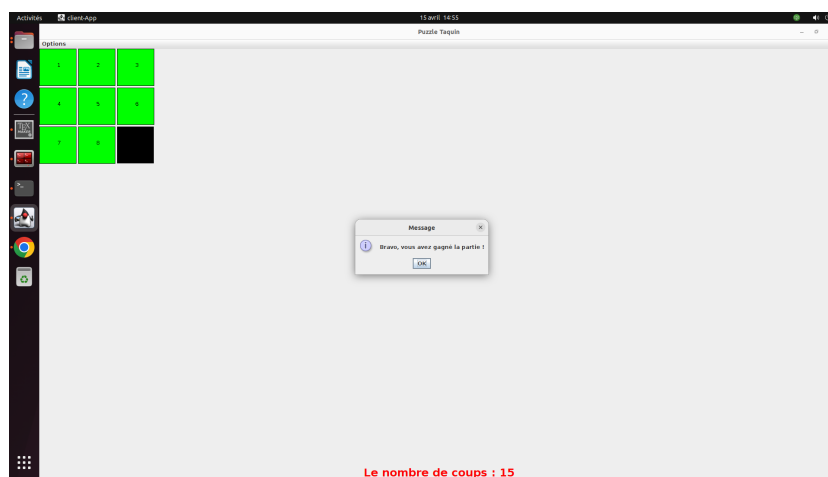


Figure 7: Fin de la partie

7 Conclusion

Ce rapport met en lumière les fonctionnalités essentielles des classes qui sont utilisées pour implémenter un jeu de Taquin. Ces classes fournissent des fonctionnalités pour représenter l'état du jeu, manipuler les boîtes du Taquin, gérer les interactions avec l'utilisateur et vérifier si le puzzle est résolu. En combinant ces fonctionnalités, il est possible de créer une expérience de jeu interactive et stimulante pour les utilisateurs.