

## LICENCE 2 INFORMATIQUE

---

# Projet de conception de jeu Taquin

---

*Auteur :*

DIALLO ELHADJ ALSEINY  
DJIGUINE MAMADY  
DIALLO ABDOULAYE DJIBRIL  
DIALLO MAMADOU ALPHA

*Chargé du cours :*

MATHET YANN

*Encadrant TP :*

YOHANN JACQUIER

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Le Jeu Taquin</b>	<b>2</b>
2.1	Généralité . . . . .	2
2.2	Notre Taquin . . . . .	2
<b>3</b>	<b>Organisation du projet</b>	<b>2</b>
3.1	Répartition des tâches . . . . .	2
3.2	Architecture du projet . . . . .	2
<b>4</b>	<b>Éléments Techniques</b>	<b>3</b>
4.1	Taquin : . . . . .	3
4.2	Position : . . . . .	4
4.3	ParseImage . . . . .	4
4.4	TaquinGraphique : . . . . .	4
4.5	TaquinFenGui : . . . . .	4
4.6	ModeleEcoutable : . . . . .	4
4.7	AbstractModeleEcoutable : . . . . .	4
4.8	EcouteurModele : . . . . .	4
<b>5</b>	<b>Expérimentations et Usages</b>	<b>4</b>
5.1	Utilisation . . . . .	4
5.2	Lancement de l'application . . . . .	4
5.3	Fonctionnement en mode console . . . . .	5
5.4	Fonctionnement de l'interface graphique . . . . .	5
5.4.1	Navigation en mode graphique . . . . .	6
5.5	Test de l'application . . . . .	6
<b>6</b>	<b>Conclusion</b>	<b>6</b>

# 1 Introduction

La programmation orientée objet est un paradigme de plus en plus indispensable pour toute personne aspirant à faire du développement. Ainsi afin de bien l'assimiler et se parfaire en appliquant ses principes notamment en java, un projet nous a été donné dans le cadre de l'unité d'enseignement Complément POO.

Le but de ce projet est de réaliser une application de jeu puzzle par groupe de quatre (4) étudiants en appliquant les principes et concepts de la programmation orientée vues durant tout le semestre en respectant le concept du MVC (modèle vue contrôleur). Les grandes lignes de ce projet étant :

- De développer un modèle complètement indépendant de la partie graphique afin que le jeu soit exécutable en mode console ;
- une partie vue-contrôleur venant se greffer sur le modèle.

## 2 Le Jeu Taquin

### 2.1 Généralité

Le jeu du taquin est un puzzle qui a été créé vers 1870, depuis il a attiré l'intérêt de nombreux mathématiciens pour sa valeur en tant qu'un problème combinatoire. Le jeu est composé de  $n \times m - 1$  petits carreaux numérotés à partir de 1 qui glissent dans un cadre du format  $n \times m$  laissant une case vide permettant de modifier la configuration des carreaux. Le jeu consiste à remettre dans l'ordre ces cases à partir d'une configuration initiale quelconque.

Le jeu est souvent connu dans les formats 3x3 ou 4x4, d'où l'appellation anglophone 8-puzzle ou 15-puzzle respectivement.

### 2.2 Notre Taquin

Notre application de jeu taquin est un cas général du taquin jouable en console mais aussi en mode graphique avec des options de paramétrages coté graphique permettant d'insérer une image , la découper ,la mélanger, selon le principe du jeu, ainsi que des options de sauvegarde et de restauration d'un état du jeu .

## 3 Organisation du projet

### 3.1 Répartition des taches

Le projet étant essentiellement composé de deux grandes parties, nous nous sommes dans un premier temps réunis afin de mieux comprendre le principe du jeu et élaborer un plan de travail, ce qui nous a permis de comprendre différentes façons d'implémenter ce jeu avec ou sans image. Mamady djiguiné s'est chargé de développer la partie modèle du jeu et son exécution en console, Abdoulaye Djibril Diallo s'est occupée du traitement sur les images, Mamadou Alpha Diallo quant à lui s'est chargé de la sauvegarde et la restauration d'une partie et Diallo Elhadj Alseiny s'est chargé de la partie interface graphique.

### 3.2 Architecture du projet

Concernant l'architecture de notre projet, nous avons voulu la relier au sujet pour une compréhension facile.Cependant nous avons opté pour une décomposition en trois (3) packages :

- **model** : permet d'implémenter le jeu de façon indépendante ;
- **vue-contrôleur** : permet de gérer tout ce qui est rendu graphique du jeu et évènement ;
- **utils** : contient les éléments supplémentaires indispensables pour une bonne configuration du jeu.

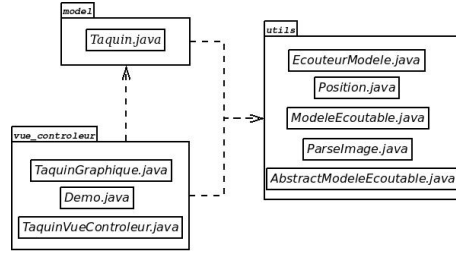


FIGURE 1 – Diagramme de packages

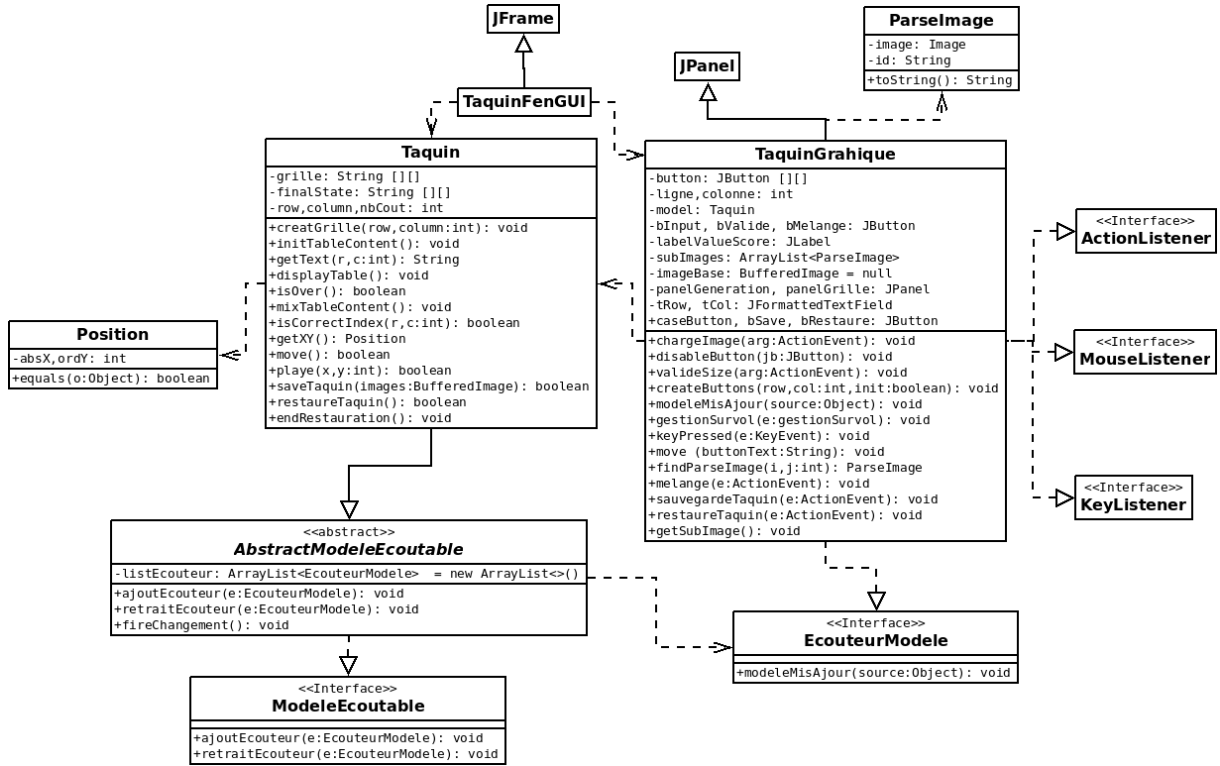


FIGURE 2 – Diagramme de Classes

## 4 Éléments Techniques

### 4.1 Taquin :

Constitue le coeur de notre application .Elle contient toutes les fonctionnalités indispensables pour la conception du jeu dont entre autres les fonctions pertinentes ci-dessous :

- **initTableContent** : Permettant d'initialiser le plateau de jeu ;
- **mixTableData** : Permettant de mélanger le plateau par tirage aléatoire et par permutation de deux valeurs à chaque tour de boucle suivant les dimensions du plateau de jeu ;
- **move** : Gère tout ce qui est déplacement entre le carreau vide qu'on a noté (-) et un voisin direct ;
- **isCorrectIndex** : texte si l'index d'un élément se trouve dans la grille de jeu ;
- **play** : procède à la permutation entre le carreau vide et un voisin direct ;
- **saveTaquin** : permettant de sauvegarder un état du jeu ;
- **restaureTaquin** : permettant de restaurer un état sauvegardé.

## 4.2 Position :

Une classe permettant de représenter un élément dans l'espace utile pour gérer la position du carreau vide.

## 4.3 ParseImage

Est un objet qui regroupe une image et son identifiant afin de faciliter son traitement ;.

## 4.4 TaquinGraphique :

Construit un panel composé des éléments de l'interface graphique du jeu.

## 4.5 TaquinFenGui :

Construit une fenêtre principale à partir d'une instance de TaquinGraphique.

## 4.6 ModeleEcoutable :

Utile pour la définition d'un objet de type (écoutables) via une interface plutôt que par une classe.

## 4.7 AbstractModeleEcoutable :

On utilise cette classe pour ne pas avoir à réécrire la partie événementielle.

## 4.8 EcouteurModele :

est implémenté par TaquinGraphique pour les mises à jour.

# 5 Expérimentations et Usages

## 5.1 Utilisation

Nous avons développé notre application avec du java standard Édition 11 pour des raisons de compatibilité avec les ordinateurs de l'université. Elle peut s'exécuter sur les distributions Linux avec interface graphique telle que : Ubuntu version 20.04.4 LTS, sur mac Os 12.3.1 et windows 10.

## 5.2 Lancement de l'application

Vous devez ouvrir un terminal à l'emplacement du dossier du projet .  
Pour :

— Lancer l'application exécuter : `ant run` ou `java -jar taquin.jar`

Une fois la commande **ant run** ou **java -jar taquin.jar** exécutée, il vous sera demandé de choisir un mode :

- **1** : pour jouer en mode graphique
- **2** : pour jouer en console
- **3** : pour quitter .

```
+----- MENU -----+
+
+ 1 - pour jouer en mode graphique +
+-----+
+ 2 - pour jouer en mode console +
+-----+
+ 3 - pour quitter +
+-----+
Veuillez entrer votre choix(1..3) 2
```

FIGURE 3 – Menu

- Initialiser le projet : `ant init` Cette initialisation créera les dossiers de bases bin, doc et dist
- Compiler le projet : `ant compile`
- Générer le Javadoc : `ant javadoc`
- Générer le fichier jar : `ant packaging`
- Nettoyer le projet : `ant clean`
- Lancer le test : `ant test`

### 5.3 Fonctionnement en mode console

En mode console vous devez entrer les dimensions de la grille de jeu .Quelques contrôles clavier vous servirons afin de pouvoir jouer tels que :

- **z** : Déplace la case vide – vers le haut
- **w** : Déplace la case vide – vers le bas
- **d** : Déplace la case vide – vers la droite
- **q** : Déplace la case vide – vers la gauche

### 5.4 Fonctionnement de l'interface graphique

Au lancement du programme, il se présente avec une grille de 3x3 par défaut.Vous pouvez modifier ce paramétrage. Dans la barre de menu existe deux zones de saisie pour modifier les dimensions de la grille et, afin de valider cette modification clic sur valider. Un label se met à jour à chaque coup joué.

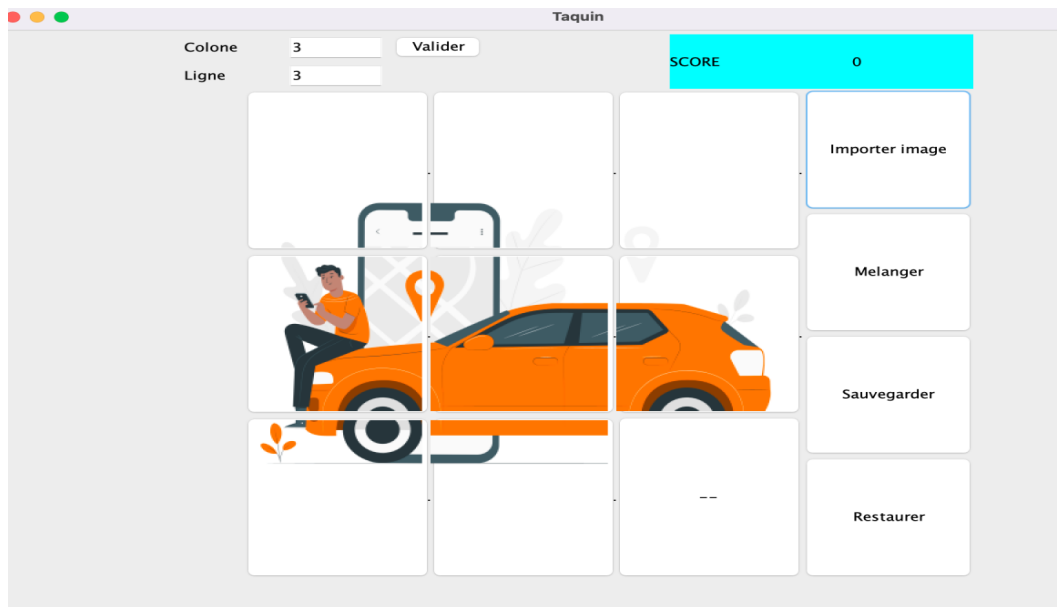


FIGURE 4 – Interface Graphique

### 5.4.1 Navigation en mode graphique

La navigation en mode interface graphique est g  r  e par un clic de la souris sur un carreau le permute avec la case vide si celui-ci est d  pla  able. En plus de la souris vous pouvez naviguez avec les quatre (4) touche de direction du clavier; afin de pouvoir les utiliser vous devez cliquer une fois sur la case vide afin de mettre le focus sur cette derni  re.

## 5.5 Test de l'application

Nous avons r  alis   le test de notre application en utilisant le framework open source Junit-4.12 pour le d  veloppement et l'ex  cution de tests unitaires. Elle teste l'ensemble des m  thodes pertinentes du mod  le.

```
(base) MyMac@MacBook-Pro-de-Djibril tp_completmentpoo % ant test
Buildfile: /Users/MyMac/Documents/Cours/ComplementP00/tp_completmentpoo/build.xml

init:
    [echo] Initialisation du projet Taquin
    [delete] Deleting directory /Users/MyMac/Documents/Cours/ComplementP00/tp_completmentpoo/bin
    [mkdir] Created dir: /Users/MyMac/Documents/Cours/ComplementP00/tp_completmentpoo/bin

compile:
    [javac] Compiling 10 source files to /Users/MyMac/Documents/Cours/ComplementP00/tp_completmentpoo/bin

test:
    [junit] Running test.TestTaquin
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0,017 sec
    [delete] Deleting: /Users/MyMac/Documents/Cours/ComplementP00/tp_completmentpoo/fileTest.txt

BUILD SUCCESSFUL
Total time: 1 second
(base) MyMac@MacBook-Pro-de-Djibril tp_completmentpoo %
```

FIGURE 5 – Capture du resultat du test

## 6 Conclusion

Travailler un sujet aussi passionnant que le taquin nous a appris beaucoup de choses dans ce vaste domaine qui est la conception de jeu allant d'une exp  rience de mod  lisation des diff  rentes   tapes de d  veloppement en passant par des exercices de programmations en Java. Il   tait tr  s agr  able de pouvoir mettre en pratique les enseignements vues durant tout le semestre en programmation orient  e objet. Nous retenons   galement de ce projet, la n  cessit   d'une bonne pr  paration, l'importance de la recherche et la gestion du temps pour la r  alisation d'un programme.

## R  f  rences

- [1] Wikipedia <https://fr.wikipedia.org/wiki/Taquin>
- [2] Java Platform Standard Edition 8 Documentation <https://docs.oracle.com/javase/8/docs/>