



UNIVERSITÉ
CAEN
NORMANDIE

Projet TAQUIN

Complément POO

- ✓ *Mohamed Ba Komara*
- ✓ *Salami Sodiki Olawalé*
- ✓ *Camara Mohamed*
- ✓ *Bamba Alassane*

Table des matières

1	Introduction	3.1-2
2	Objectifs	3.1-3
3	Organisation de projet	3.1-4
3.1	Architecture du projet.....	3.1-4
4	Diagramme de Class.....	3.1-5
5	Explications Techniques	3.1-6
5.1	Mode de fonctionnement des méthodes	5.1-6
5.1.1	melange ()	5.1-6
5.1.2	isFinished ()	5.1-6
5.1.3	decouperImg ().....	5.1-6
6	Conclusion.....	5.1-7
6.1	Améliorations possibles	6.1-7
7	Références	6.1-8

1 Introduction

Dans le cadre de notre deuxième année en Informatique à l'Université de Caen Normandie nous avons eu pour tâche la réalisation d'un projet informatique, intitulé : « **Jeu de Puzzle à glissières (Taquin)** », avec le langage de programmation JAVA, SWING pour la réalisation de L'IHM et le Pattern MVC (Model View Controller) du module Complément POO afin de solidifier notre maîtrise de ce langage. D'après Wikipédia **Taquin** est un jeu solitaire en forme de damier créé vers 1870 aux États-Unis. Sa théorie mathématique a été publiée par l'American Journal of mathématiques pures and applied² en 1879.

Pour mener à bien ce projet nous avons donc été départager par groupe de quatre (4) étudiants.

Dans la partie suivante nous présenterons l'objectif du projet et les différentes étapes importantes à sa réalisation.

2 Objectifs

L'objectif de ce projet en premier lieu, est de réaliser une application de jeu solide, facilement maintenable basé sur une architecture MVC composée de code propre et non de code spaghetti.

Ensuite le programme doit être doté d'une interface graphique, jouable en mode console et mode IHM (Interface Homme Machine).

Pour mener à bien ce projet nous l'avons décomposé de la manière suivante :

- ✓ La mise en place de l'architecture basique de Modèle Vue contrôleur.
- ✓ La mise en place de la partie modèle (Avec le générateur de taquin solvable).
- ✓ La réalisation du compartiment Contrôleur (Avec les types d'écoutes).
- ✓ La mise en place de la vue.

3 Organisation de projet

Le découpage du projet vu dans la partie précédente nous a permis d'être efficaces dans la réalisation de ce projet. Il faut dire qu'au préalable il était prévu que chaque membre s'occupe d'une partie du projet pour une optimisation parfaite, mais par la suite nous avons tous décidés de travailler tous ensemble sur chaque partie du projet pour des raison suivantes :

- ✓ Bien comprendre chaque Notion abordée lors de la réalisation du projet
- ✓ Se confronter à toutes les difficultés rencontrées afin d'améliorer son autonomie.
- ✓ Apprendre d'avantage...

NB : Pour des raisons de confinement nous avons pu travailler de cette manière et très efficacement grâce à l'Application TeamViewer, qui a permis à chaque membre de coder et participer à 100% au projet.

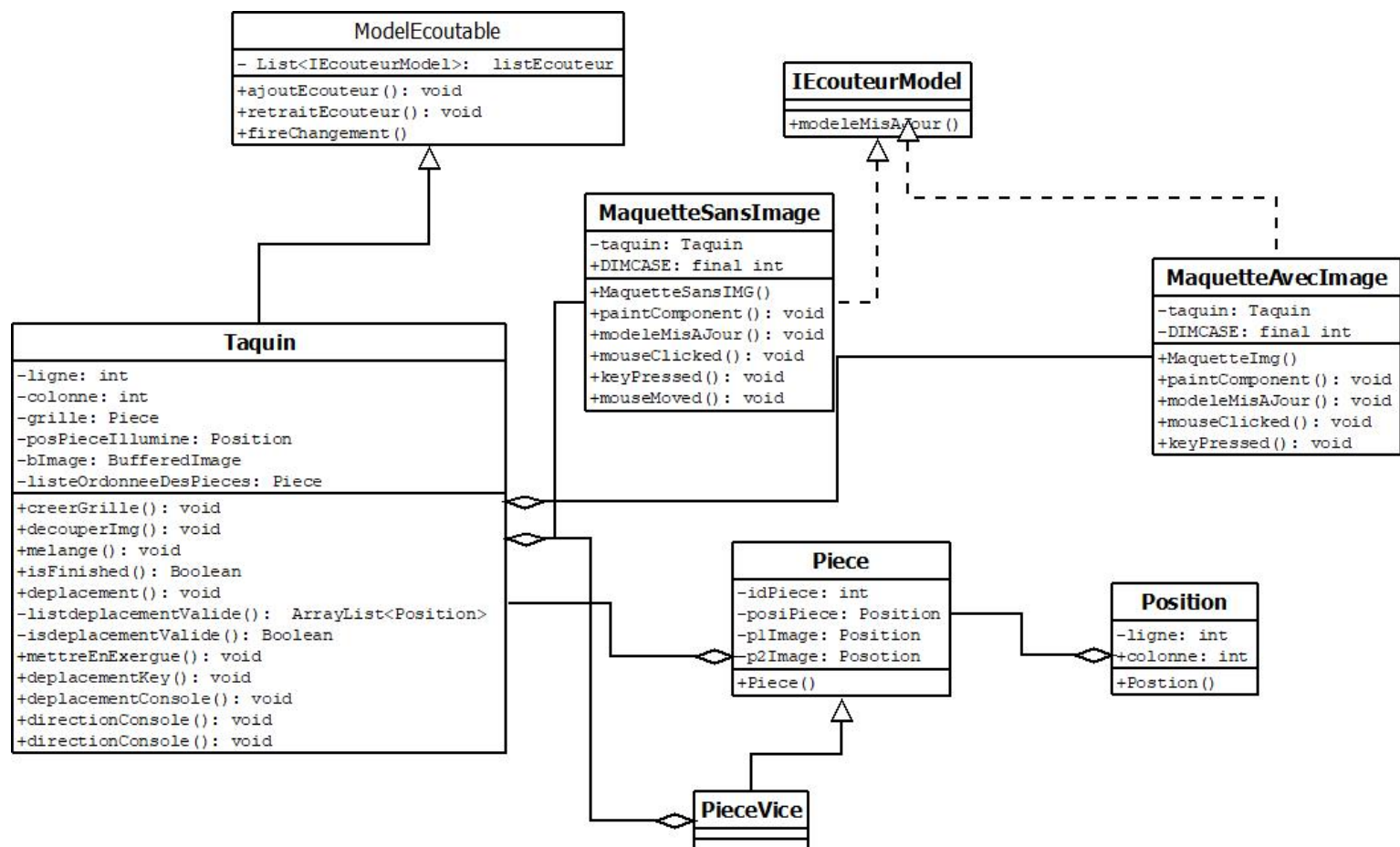
3.1 Architecture du projet

Pour ce qui est du packaging nous avons procédé de la manière suivante :

Vu que le programme doit effectuer trois visualisations différentes c'est-à-dire une partie console, une partie sans image et la dernière avec image. Nous avons donc créé :

- ✓ Un package **Model** : contenant la partie métier de l'application.
- ✓ Un package **Vue** : il contient un Fenêtre héritant de JFrame pour la visualisation du jeu.
- ✓ Un **package Controller** : contient les différents contrôleurs.
- ✓ Un **package Utilitaire**: qui contient en son sein les classes tierces du programme.
- ✓ Un package Main : contient les classes exécutables, une pour l'interface graphique et une autre pour la console

Tous dans le dossier **src**.



4 Diagramme de Class

5 Explications Techniques

Pendant la réalisation de ce projet nous avons effectués plusieurs échanges concernant l'utilisation de la classe **PièceVide**, le mode de déplacement des objets du jeu c'est - à - dire le test du déplacement d'une pièce doit être fait en se focalisant sur la pièce déplaçable ou la pièce Vide etc. Nous avons donc pu grâce à ces différents échanges faire ce que nous pensions être bons au niveau de l'implémentation des différents algorithmes. Ensuite, en ce qui concerne la solvabilité du jeu, il faut dire que nous nous sommes focalisés sur l'explication donnée par le professeur lors de la présentation du projet qui était d'effectuer un mélange à partir d'un jeu déjà résolu.

5.1 Mode de fonctionnement des méthodes

5.1.1 melange ()

Comme son nom l'indique, elle a pour rôle d'effectuer un mélange du jeu. En effet nous partons sur la base de jeux déjà résolus pour effectuer le mélange en déplaçant la pièce vide un nombre de fois aléatoires et en s'assurant que la première pièce soit toujours déplacée. Ainsi peu importe le nombre de mélange effectué nous avons toujours un jeu solvable.

5.1.2 isFinished ()

Elle permet d'identifier la fin de la partie. En effet nous nous sommes basées sur une liste contenant les différentes pièces de façons ordonnées, cela permet donc à la fonction à chaque appel de vérifier si la position des pièces de l'état actuelle du jeu respecte ou non l'ordre de la liste.

5.1.3 decouperImg ()

Elle permet de découper une image en fonction du nombre de ligne et colonne du jeu et attribuer à chaque pièce les positions de la partie d'une image.

6 Conclusion

Durant tout le long de ce projet, il est important de souligner que nous avons été emmené à bien pousser notre réflexion concernant certaines difficultés tant au niveau technique qu'au niveau organisationnel, cela nous a donc permis de bien analyser le projet et mener à bien ce projet. Aussi il faut dire que nous avons appréhendé de nombreux concepts lors de la réalisation de ce projet.

6.1 Améliorations possibles

Vu le temps qui nous avait été imparti pour la réalisation de ce projet nous n'avons malheureusement pas pu ajouter certaines fonctionnalités. Mais il faut dire que de nombreuses améliorations sont possibles pour ce type projet.

Dans un premier temps nous pouvons améliorer le jeu en ajoutant des fonctionnalités telles que le temps, c'est-à-dire mettre en place un chronométrage du temps de résolution du jeu par le joueur. En outre un compteur pour le nombre de coups effectués durant une partie. Enfin l'utilisation d'un algorithme de résolution du jeu **A étoile**.

7 Références

<https://fr.wikipedia.org/wiki/Taquin>

<http://www-lisic.univ-littoral.fr/~fonlupt/Enseignement/2017/aa/taquin.pdf>

<http://villemin.gerard.free.fr/Puzzle/Taquin.htm>

<http://pascal.ortiz.free.fr/contents/tkinter/projets tkinter/taquin/taquin.html>

<https://translate.google.fr/?hl=fr#view=home&op=translate&sl=auto&tl=fr&text=s>

<https://translate.google.fr/?hl=fr#view=home&op=translate&sl=auto&tl=fr&text=s>