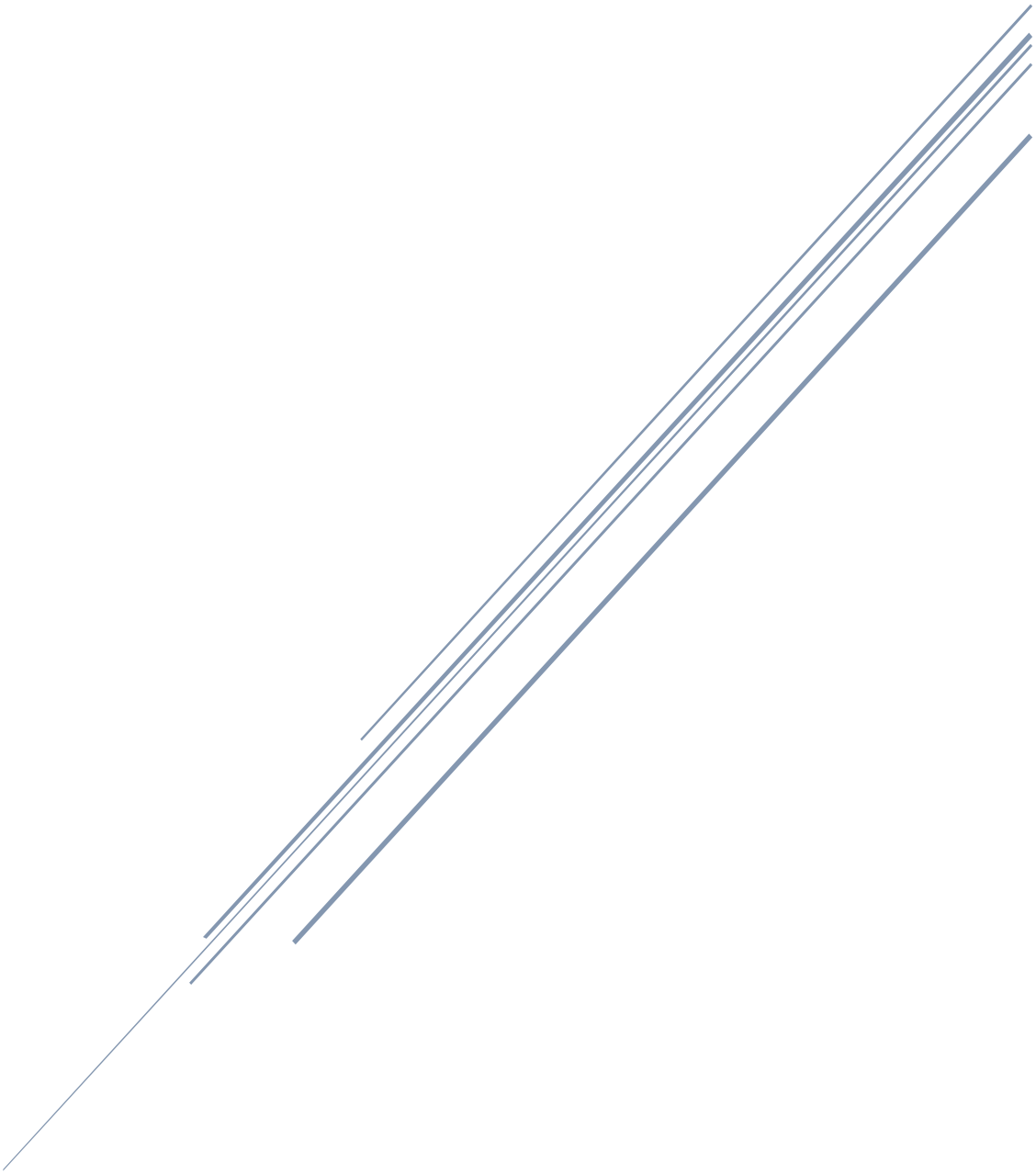
TAQUIN

Projet – Ingénierie Logicielle



Benjamin Barbier – Baptiste Mounier
L3 MIASHS SC

Table des matières

Recette informatique

Objectifs atteints

- Jeu en solo dans la console avec une taille fixée de 4 par 4 blocs, un seul motif numérique allant de 1 à 15 avec un seul bloc sans numéro et déplaçable (z, s, q, d)
- Jeu en solo avec interface graphique au clavier
    - Au clavier (z, s, q, d) : un seul bloc se déplace
- Possibilité de demander de l'aide à une I.A. pour le prochain coup
- Possibilité de laisser l'I.A. jouer toute seule jusqu'au bout
    - Véritablement fonctionnel que sur la version console, pour le mode graphique il n'y a pas d'affichage à chaque coups mais la recherche a lieu
- Possibilité de recharger la partie qui était en cours lors du précédent lancement du jeu (grâce à un stockage dans un fichier)
    - Non implémenté dans le mode graphique

Description de l'application

Installation

Vous pouvez trouver l'exécutable .jar dans le dossier dist, sinon en lançant la méthode main de la classe application.Main vous lancerez le Taquin en console ou en graphique si vous ajoutez un argument quelconque.

Fonctionnalité

Les tests et le fichier Ant seront détaillés un peu plus tard.

Vous pourrez également voir le nombre de coups effectués, le nom du joueur et le score dans la console et le nombre de coups dans l'interface graphique (tous les coups de l'ia comptent).

Le score commence à 50 (nombre de coups de mélange) et diminue de 1 par coup, l'utilisation de l'ia réduit également le score de 1 point supplémentaire par coup (donc deux par coup de l'ia au total).

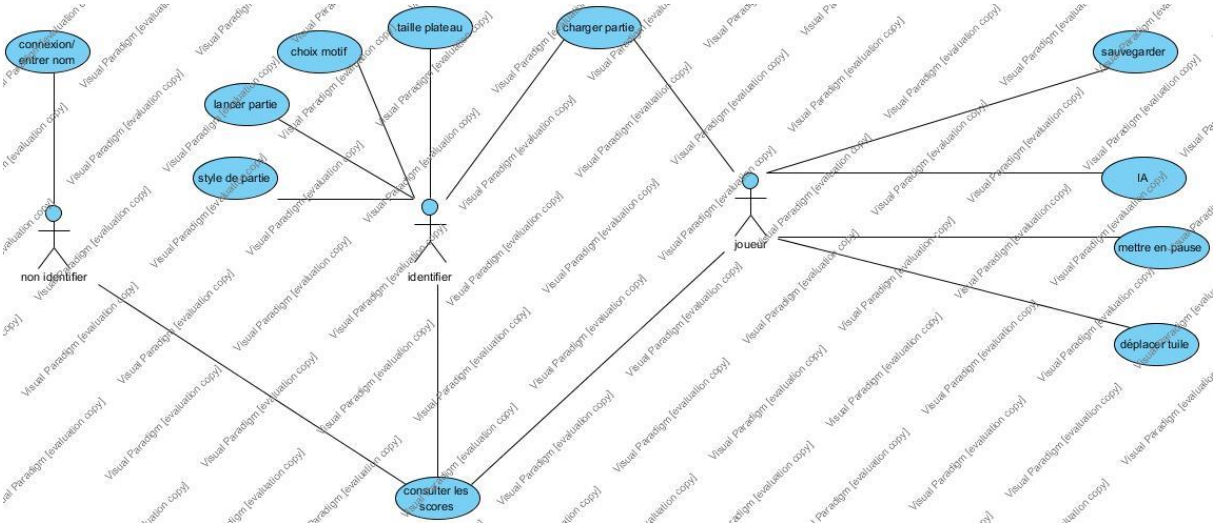
Ensuite les commandes sont simples :

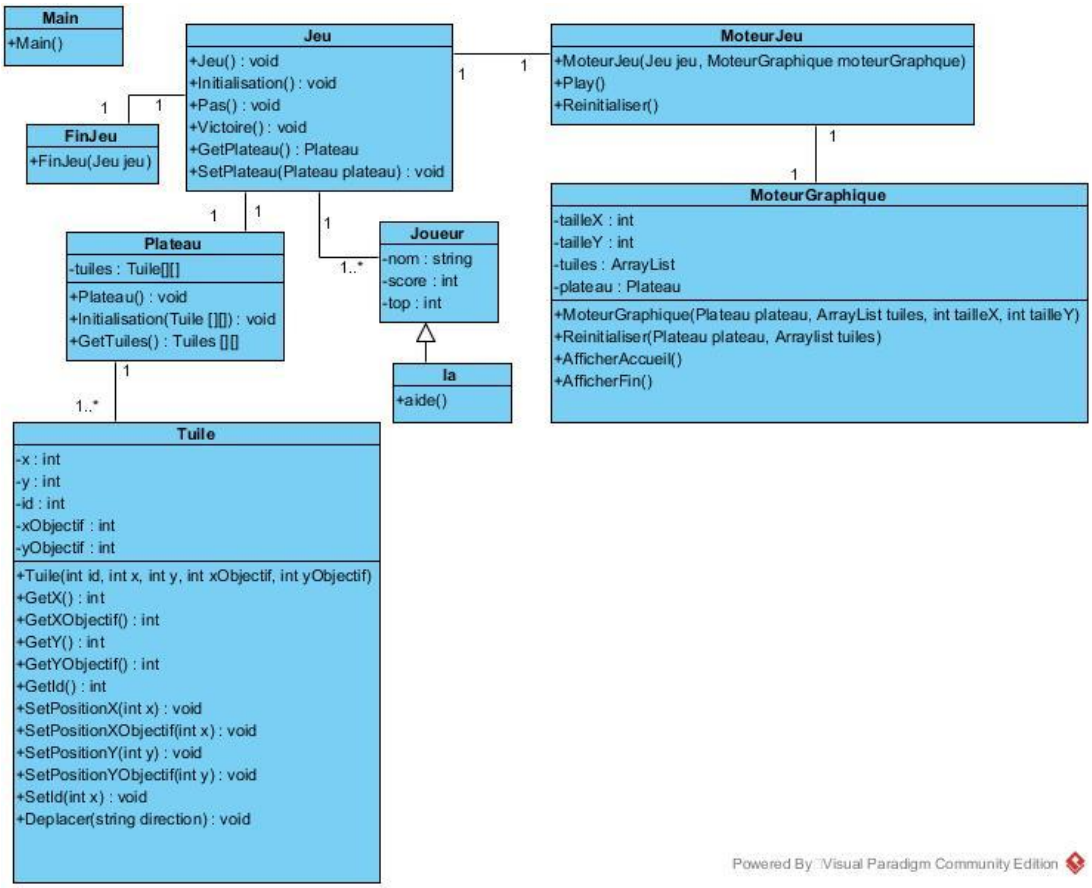
- Console
    - z : pour pousser une tuile vers le haut
    - s : pour pousser une tuile vers le bas
    - q : pour pousser une tuile vers la gauche
    - d : pour pousser une tuile vers la droite
    - ai 0 : pour une aide complète de l'ia
    - ai x : pour une aide de x coups de l'ia
    - save test : pour sauvegarder la partie dans test.taquin situé dans le dossier save
    - load test : pour charger la partie sauvegarder dans test.taquin
- Graphique
    - z : pour pousser une tuile vers le haut
    - s : pour pousser une tuile vers le bas
    - q : pour pousser une tuile vers la gauche
    - d : pour pousser une tuile vers la droite
    - Full Help (bouton) : pour une aide complète de l'ia

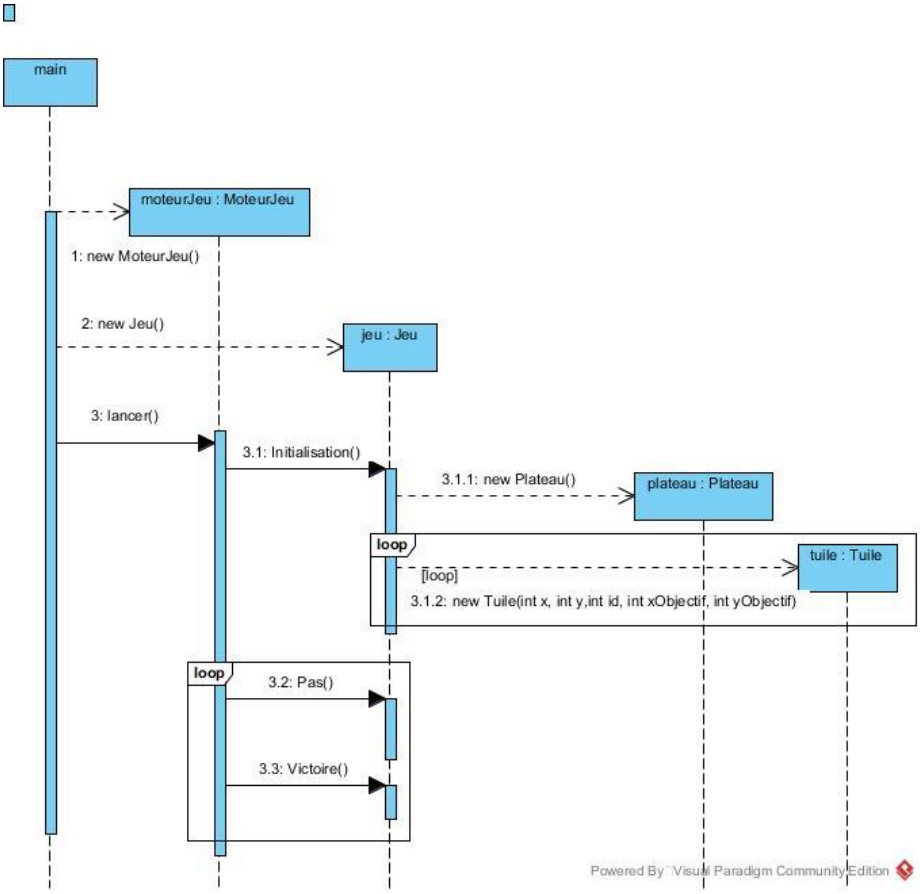
- L'ia résout le Taquin mais ne dispose pas d'affichage intermédiaire et donc le jeu commence une nouvelle partie

- Help (bouton) : pour une aide d'un coup de l'ia
- New Game : pour débuter une nouvelle partie

Conception







I.A.

L'intelligence artificielle est basé sur un algorithme de recherche A* dû au grand nombre de nouvel état possible à chaque itération. Quel que soit le choix fait (aide d'un coup, plusieurs coups ou totale) elle calcule la résolution complète et retourne les actions à effectuer pour le choix fait. De ce fait elle a le désavantage de pouvoir prendre un peu de temps si la malchance est là (le mélange des tuiles sont mélangées à l'aide de 50 déplacement aléatoire depuis un Taquin résolu avant le début de la partie). Mais d'un autre côté, elle apporte un coup sûr même pour une demande d'aide pour seulement 3 coups. L'algorithme utilise deux heuristiques : la somme des distances de Manhattan entre l'état actuel et l'état de résolution et le nombre de tuiles mal placées.

Ant

Nous avons également fait un fichier Ant.xml permettant de nettoyer le dossier bin, compiler le projet (implémentation pour gérer les tests JUnit non réalisée), créer le fichier .far dans le dossier dist et générer la javadoc dans le dossier doc/javadoc.

Tests

Nous avons mis en place quelques tests unitaires, situés dans src/test, testant les points suivants :

- Déplacements
    - Pousser une tuile vers le haut
    - Pousser une tuile vers le bas
    - Pousser une tuile vers la gauche
    - Pousser une tuile vers la droite
- I.A.
    - Distance de Manhattan
    - Nombre de tuiles mal placées
    - Résolution complète
- Victoire
    - Levée d'une exception de victoire

Repository

GitHub : https://github.com/BaptisteMounier/Taquin-BarbierMounier

Trello : https://trello.com/b/TbWChNx6/projet-taquin

Documentation

Toute la Javadoc se trouve dans le dossier doc/javadoc du projet.