

2018

Léa Chemoul – Antoine Ganne

Polytech Lyon – 3A Informatique

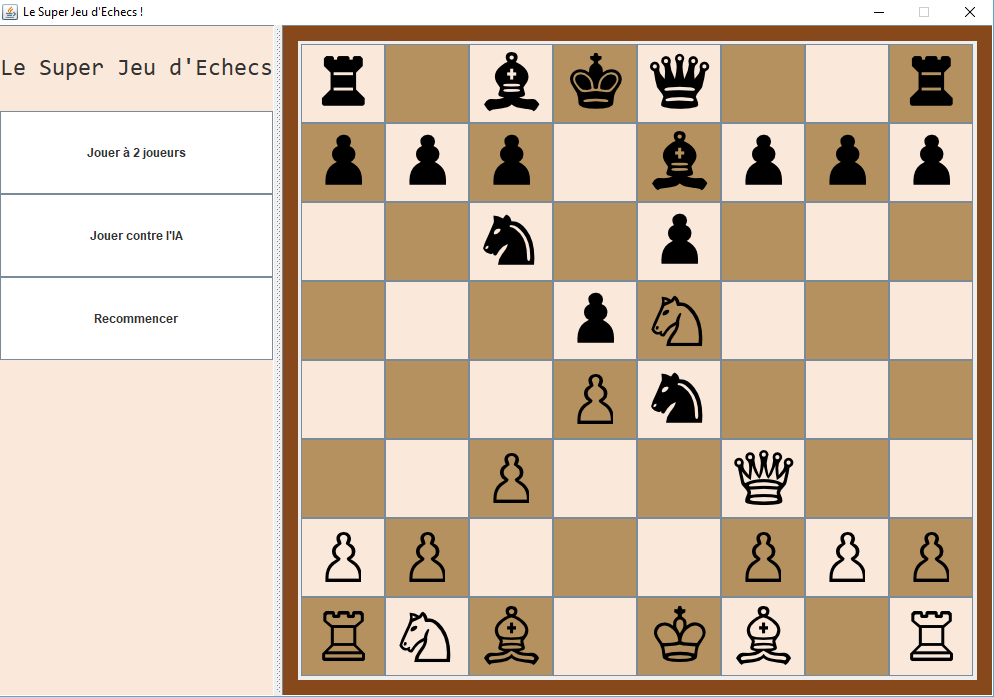
04/02/2018

RAPPORT APO – CHESS GAME



RAPPORT DE PROJET

### Contexte

Le but de ce projet est de réaliser un jeu d’échec en java. Nous avons dans un premier temps développé le jeu en mode console, puis nous l’avons reporté en graphique en utilisant la bibliothèque java swing. La majeure partie des fonctions ont été codées pour pouvoir être soumises à un jeu console ou à un jeu graphique.

Modes de jeu :

* 2 joueurs peuvent jouer au tour par tour sur le même ordinateur.
* 1 joueur peut jouer seul contre une Intelligence artificielle basique.

Vous pouvez vous référer au README.md se situant à la racine du projet pour trouver plus d’informations.

### Fonctionnalités développées

**Fonctionnalités liées aux règles du jeu :**

* La mise en échec de son propre Roi est impossible
* Grand et petit Roque
* Prise en passant
* Promotion du pion lorsqu’il arrive à la dernière ligne
* Déplacements des pièces selon les règles de déplacements qui leurs sont propres

**Fonctionnalités liées à la fin d’une partie :**

* Si l’un des rois est en échec et mat
* Si l’un des rois est en Pat
* Si 50 tours sans prises sont joués, la partie est finie en match nul.
* Si l’un des rois est mangé

**Fonctionnalités de l’IA :**

* Soustrait son Roi s’il est en échec
* Choisie une pièce aléatoirement parmi ses pièces et la déplace aléatoirement parmi les déplacements possibles. Sauf si le déplacement met son Roi en échec.
* Réalise la promotion de son pion
* Réalise une prise en passant des pièces de l’adversaire avec une probabilité de ½ (lorsque l’IA choisie de ne pas prendre en passant cela n’est pas affiché sur la console).

### Déroulement d’une partie

* Initialisation de l’échiquier par un fichier de configuration (se place directement à la racine du projet).
* Choix de jouer contre l’IA ou un autre joueur (pour ce dernier choix : saisi du nom des joueurs)
* En réponse aux cliques sur une case tu plateau, la classe caseListener est appelée. C’est dans cette classe que réside une bonne partie des traitements.

On y teste si :

* Le pion appartient bien au joueur. Si c’est le cas on affiche ne vert les possibilités de déplacement.
* Si une pièce a déjà été sélectionnée (principe de « pièce touchée pièce jouée » sauf si la pièce en question met obligatoirement le Roi en échec) alors on peut cliquer sur une case qui appartiens à ses déplacements possibles (sauf si cela met en échec notre Roi)

### Choix de conception

* Héritage des pièces et polymorphisme. La classe Pièce est une classe abstraite dont héritent les classes Pion, Cavalier, Tour, Fou, Roi, Dame.
* Les pièces sont considérées comme « mangées » si elles n’apparaissent plus sur le plateau. En effet, lorsqu’un déplacement permet au joueur de manger une pièce de l’adversaire, il prend tout simplement sa place et la pièce est détruite. Cela permet d’économiser les calculs.
* Le jeu a été conçu de sorte qu’il n’est pas nécessaire de quitter l’application pour tester les différentes configurations ; Il suffit de cliquer sur recommencer.

### Diagramme des classes

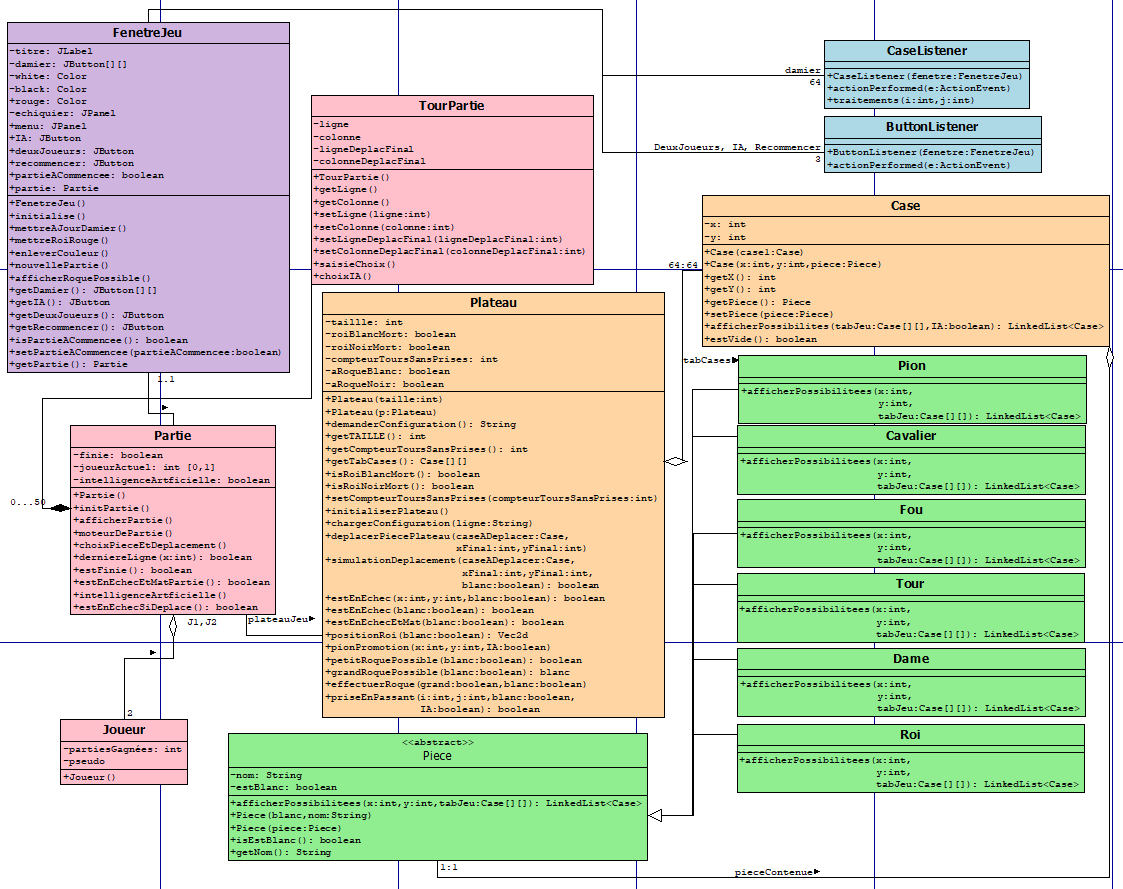
**Légende** :

Rose : classes relatives à une partie

Orange : classes relatives au damier

Vert : classes relatives aux pièces, à savoir Pièces et les classes qui en hérite.

Violet : classe principale contenant l’entrée du programme et contenant les objets java swing.

Bleu : classes de gestion des évènements de clique à la sourie. Elles héritent d’Action Listenner.

### Retour sur le cahier des charges

Cahier des charges :

|  |  |  |
| --- | --- | --- |
| Fonctionnalité | Oui/Non | Choix de conception |
| Visualisation du jeu en console | Oui | En graphique : Java Swing |
| Déplacement des pièces | Oui | Polymorphisme : fonction déplacementsPossibles() renvoie les déplacements possibles pour une pièce donnée. |
| Chargement d’une conﬁguration de jeu | Oui | A travers un JFileChooser initialisé à la racine du projet |
| Garde-fous des déplacements | Oui | Mise en échec de son propre Roi impossible.  Déplacements selon les règles standards. |
| Intelligence artiﬁcielle basique. | Oui | Soustrait son Roi lorsqu’il est en echec. |

### Fonctionnalités futures à développer

* Une intelligence artificielle un peu plus développée : capable d’empêcher ses pièces d’êtres mangées.
* Réutiliser le fait d’avoir saisi le nom des joueurs pour afficher des messages personnalisés.
* Exporter en .txt la configuration d’une partie encours pour pouvoir la récupérer.

### RETOUR SUR EXperience

Ce projet nous aura permis d’apprendre à développer une application graphique java grâce à java swing. Nous avons pu appliquer certains concepts du cours comme les classes abstraites, l’héritage, le polymorphisme, …

