



# Projet BA1 Stratego

## **COURS D'ALGO2 : JAVA.**

Par Thomas Rosi et Charles Raeymaekers

Le rapport reprend, la partition du travail du binôme, l'explication du projet, les points forts ainsi que les points faible du projet (bugs éventuels).

## **Table Des Matières**

### **Introduction**

#### **1. Répartition des tâches**

#### **2. Manuel de l'utilisateur**

2.1 Règles

2.2 Interface

2.3 Options

#### **3. Algorithmique**

3.1 Généralités

3.2 APiece

3.3 Tiles

3.4 Array

3.5 List

#### **4. GUI**

#### **5. Bugs**

5.1 Partie Algorithmique

5.2 Partie Graphique

### **Conclusion**

### **Bibliographie**

# **Introduction**

Dans le cadre du cours « Projet d'informatique » de première année Bachelier en sciences informatiques, nous sommes amenés à réaliser un programme en langage informatique « java ».

Cette année, le projet consiste en l'implémentation d'un jeu de Stratego. Cet exercice nous permet de nous plonger directement dans la conception et l'utilisation d'un code complet en java.

Comme son nom l'indique, le Stratego est un jeu de stratégie.

## **1. Répartition des tâches**

Avec un projet de cette ampleur, nous avons décidé de nous répartir les tâches. Les deux grandes parties de ce projet étant l'algorithmique du jeu, et l'interface graphique, Charles Raeymaekers s'est occupé de l'interface graphique et Thomas Rosi a implémenté le jeu en console.

## **2. Manuel d'utilisateur**

### **2.1 Règles**

Le Stratego est un jeu de stratégie qui se joue à 2 joueurs, en un contre un. Il se compose d'un plateau quadrillé et de 40 pièces par joueur. Ces pièces représentent une armée.

Ces pièces sont les suivantes :

- Un drapeau (rang 0)
- Un maréchal (rang 10)
- Un général (rang 9)
- Deux colonels (rang 8)
- Trois majors (rang 7)
- Quatre capitaines (rang 6)
- Quatre lieutenants (rang 5)
- Quatre sergents (rang 4)
- Cinq démineurs (rang 3)
- Huit éclaireurs (rang 2)
- Un espion (rang 1)
- Six bombes (rang 11)

Le but du jeu est de s'emparer du drapeau adverse tout en protégeant le sien.

Avant de commencer la partie, chaque joueur pose ses quarante pièces, sur le plateau.

Le jeu se déroule en tour par tour. Un tour se déroule de façon très simple : Il suffit de choisir une pièce et de la déplacer d'une case vers le haut, vers le bas, vers la droite ou vers la gauche. Attention, le terrain, sous forme d'une grille de dix cases sur dix, n'est pas uniforme. Il comprend deux obstacles composés de quatre cases, considérés comme étant des lacs sur lesquels les pièces ne peuvent pas se déplacer.

Lors de la partie, les joueurs avancent un par un leurs pions en direction de l'équipe ennemie, sans avoir la connaissance de la répartition adverse. Lors d'un déplacement, une pièce peut avancer d'une case (vers le haut, le bas, la droite, la gauche) sauf l'éclaireur (pièce de rang 2) qui a le droit de se déplacer en ligne ou en colonne, du nombre de cases souhaité. Le drapeau (pièce de rang 0) et les bombes (pièces de rang 11) ne peuvent pas bouger !

Si un joueur se déplace sur une case comprenant un pion adverse, un combat s'engage pour prendre/garder la place. Ce combat est effectué à partir des rangs donnés à chaque pièce (exemple : le rang 10 gagne contre le rang 9, le rang 9 gagne contre le rang 8, ...). Deux exceptions : le démineur (pièce de rang 3) est seul à pouvoir détruire les bombes et si l'espion (pièce de rang 1) attaque le maréchal (pièce de rang 10), c'est ce dernier qui perd le combat. Toutes les pièces peuvent s'emparer du drapeau (pièces de rang 0). Au cours d'un combat, si les deux pièces ont le même rang, elles s'entretuent.

Pour gagner une partie de Stratego, il suffit de s'emparer du drapeau adverse. Pour ce faire, il faut que l'un des joueurs déplace un de ses pions sur la case du drapeau adverse. Néanmoins il se peut qu'un joueur ne sache plus déplacer ses pions, dans ce cas il perd la partie.

## 2.2 Interface

Lors du lancement du jeu, une fenêtre « menu » est à la disposition du joueur. Celle-ci permet de voyager dans le jeu. Elle contient un bouton « Play » qui permet de lancer une partie de Stratego contre l'intelligence artificielle implémentée. Le deuxième bouton lance une seconde fenêtre appelée « Options » (décrite dans la section suivante : Options).

Une fois la partie lancée, l'utilisateur met en place son jeu en cliquant sur les pions disponibles placés dans la colonne de gauche. Une fois les quarante pièces mises en place, un bouton « Play » est disponible sur la droite. Grâce à celui-ci, la partie se lance.

Durant la partie, des fenêtres « fight » s'affichent en indiquant au joueur le pion gagnant et le pion perdant (entouré en rouge). À cet instant un bug se présente (voir partie 5.2 sur les bugs). Garder ouvertes les fenêtres ou les fermer ne change rien au fonctionnement du programme mais nous conseillons de les fermer pour la visibilité du jeu.

En fin de partie, une fenêtre « end » s'affiche avec victoire ou défaite. En fermant cette fenêtre, le programme s'arrête.

## 2.3 Options

Lorsque la fenêtre Options se lance, il est possible de changer la difficulté du jeu. L'intelligence artificielle possède deux niveaux.

Le premier niveau, appelé Random, est une intelligence aléatoire n'ayant aucun but précis. Le but du second, Average, est d'avancer vers l'adversaire, ainsi, l'ia attaque le joueur.

## 3 Algorithmique

### 3.1 Généralités

Dès le projet fixé et après quelques réflexions, nous avons directement divisé le jeu en plusieurs composants. Nous avons créé un tableau représentant le plateau de jeu. Celui-ci, nommé Array, a une taille de 10x10. La variable caractérisant le tableau est une case, définie Tiles et enfin, nous avons créé les pièces à positionner dans les cases, d'où la classe abstraite APiece.

### 3.2 APiece

La classe APiece est une classe abstraite définissant les caractéristiques d'une pièce. Une pièce a 3 caractéristiques : Un nom, un rang et une appartenance. Le nom est défini par une chaîne de caractères, le rang par un entier et un booléen définit l'appartenance.

### 3.3 Tiles

La classe Tiles est une classe définissant les cases retrouvées dans la classe Array. Celle-ci enregistre la pièce qu'elle comprend, s'il y en a une et la position qu'elle occupe via un système de coordonnées.

### 3.4 Array

La classe Array instancie un nouveau tableau de jeu qui gère toutes les méthodes utilisées pour se déplacer et finir la partie. Les méthodes de déplacements sont définies selon une série de boucles if au départ, testant le pion mis dans la Tiles rentrée en paramètre. Ensuite, nous avons implémenté des méthodes qui testent les déplacements possibles, ainsi on peut vérifier si on ne prend pas une Tiles comprenant un pion ne pouvant pas bouger (bombes, drapeau, pièces adverses).

Nous avons essayé de définir dans un deuxième temps ces mêmes méthodes dans la classe APiece, mais cela sans succès. En effet, un bug dans l'instance Array, ne permet pas de récupérer le tableau dans la classe des pions.

### 3.5 List

La classe List est une classe utilisée pour placer les pièces en début de partie. L'intelligence artificielle connaît plusieurs listes d'entiers représentant le rang des pièces et en choisit une au hasard.

## 4 GUI

Le lancement du jeu se fait par le lancement de la classe StratMain qui crée une StratFenetre. StratFenetre est la classe de base de la gui du jeu. Elle contient la majeure partie du code de la gui que nous avons implémenté.

Dans la StratFenetre, nous utilisons un système de changement de panels pour faciliter la fluidité du jeu. Elle contient 3 panels principaux pouvant être intervertis via des boutons:

- un panel représentant le menu principal.
- un panel de placement des pions.
- un panel de la partie en cours.

Pour le menu principal, nous utilisons une image importée d'internet (voir bibliographie ci-dessous) affichée grâce à un PicturePanel.

Pour le placement des pions, nous avons créé une grille grâce à un BoardPanel et une colonne de petits Gamepanels représentant la colonne de pions sur le côté gauche du plateau. Le joueur sélectionne les pions, un à un, pour les placer sur le plateau via un mouseListener. Les images sont également tirées d'internet.

Pour la partie en cours, nous utilisons un panel similaire à celui du placement mais sur lequel est rajouté le tableau affiché en console via un affichage de pions. Le panel de partie en cours est la représentation des actions effectuées par le code en console implémenté.

Nous utilisons un mouseListener pour les déplacements des pions, c'est-à-dire que dans un premier temps nous sélectionnons le pion et ensuite son déplacement. Si un pion adverse se trouve à l'endroit du déplacement du pion, une fenêtre « fight » s'affiche indiquant au joueur si son pion a gagné, perdu ou si les pions se sont entretués (grâce à 3 fenêtres différentes).

En fin de partie, si le drapeau du joueur est pris ou si le joueur s'empare du drapeau adverse, une fenêtre « end » s'affiche pour indiquer la victoire ou la défaite.

Toutes les fenêtres possèdent une taille fixe.

## 5 Bugs

### 5.1 Partie algorithmique

Dans l'intelligence artificielle, l'ordinateur ira toujours le plus loin possible en fonction de la pièce choisie. C'est-à-dire que si la pièce est un éclaireur, il ira le plus loin qu'il le peut mais il avancera que d'une seule case avec tous les autres pions (sauf drapeau et bombes).

La méthode load() permettant de charger le tableau enregistré lors d'une partie précédente ne fonctionne pas.

### 5.2 Partie graphique

Dans la partie graphique plusieurs bugs sont présents.

La fenêtre d'option prend du temps à se lancer quand on clique sur le bouton option.

Lors du placement des pions, on ne sait pas placer un pion sur un autre déjà placé sans récupérer le pion écrasé et le placement des pions s'effectue un à un.

Quand on change de Panel (pour revenir en arrière), le Panel n'est pas relancé, il se compose toujours des éléments contenus avant le changement.

Lorsqu'on clique sur un pion pour effectuer un déplacement, on est obligé de le déplacer.

Lors du choix d'un pion pour un déplacement, si on clique sur une case qui ne contient pas de pions, cela nous retourne une erreur en console qui n'influence cependant pas le bon fonctionnement du programme.

Lors des combats entre les pièces, on ne distingue pas clairement l'appartenance des fenêtres « fight ». L'affichage d'une fenêtre ne stoppe pas le jeu, l'ia joue son tour même si une fenêtre est affichée. Si l'ia prend un des pions du joueur, après la prise d'un de ses pions, la première fenêtre affichée sera celle du dernier mouvement effectué (celle de l'ia) et la seconde sera celle du joueur.

## Conclusion

Lors de ce projet, nous avons pu mettre en pratique ce que nous avons appris durant l'année aux cours de programmation I et II. Nous avons malheureusement eu beaucoup de difficultés pour mettre en place l'interface graphique et relier l'algorithmique à celui-ci. Il est vrai que nous avons travaillé sur nos ordinateurs personnels en java 1.8 (unique version disponible en téléchargement via Oracle) alors que les ordinateurs de la faculté sont en version 1.7.

## **Bibliographie**

Image de fond du menu principal et de la page d'introduction du rapport :

[https://lh5.ggpht.com/X3Y7glAM169C5ugoSe0rM3rez7lwy1FO2\\_Re5nQNPL-bUbQaeOsOLA9GeF7npG\\_PjtM=h900](https://lh5.ggpht.com/X3Y7glAM169C5ugoSe0rM3rez7lwy1FO2_Re5nQNPL-bUbQaeOsOLA9GeF7npG_PjtM=h900)

Image de pions :

[http://www.ludigaume.be/Jeux/S/Stratego/stratego\\_Pieces.jpg](http://www.ludigaume.be/Jeux/S/Stratego/stratego_Pieces.jpg)

Mode d'emploi du jeu original Stratego 1999 Jumbo International, Amsterdam.