

## Langage Orienté Objet

### Projet

L'objectif du projet est de concevoir une application Java permettant de jouer au jeu de plateau **Shazamm !** de Philippe des Pallières et François Bruel<sup>1</sup>. Ce jeu est également jouable en ligne depuis le site [www.boiteajeux.net](http://www.boiteajeux.net). Si vous avez un doute sur le fonctionnement du jeu, n'hésitez pas à aller l'essayer sur ce site !

#### 1. Présentation du jeu

(Source : <http://www.trictrac.net/jeu-de-societe/shazamm-/regle>)

##### Synthèse :

Nombre de joueurs : 2 joueurs (sorcier rouge ou sorcier vert)

Nombre de cartes sortilèges : 2 paquets de 14 cartes (28 cartes en tout)

Nombre de points de mana par joueur : 50

*Allez-vous céder le passage au sorcier d'en face ?! Plutôt griller !*

#### Un pont trop loin...

Le plateau de jeu représente un pont que vont vouloir emprunter deux sorciers au même moment ! Le pont est composé de 19 **plaques**. En début de partie, chaque joueur est affecté aléatoirement à l'une des deux couleurs (sorcier rouge ou vert). Le but du jeu est de repousser le joueur adverse dans la lave située à l'extrémité du pont. Une **partie** se déroule en plusieurs **manches**. Chaque manche est composée de plusieurs **tours**.



Figure 1 - Le sorcier rouge a gagné la partie en poussant le sorcier vert dans les flammes.

<sup>1</sup> <http://www.trictrac.net/jeu-de-societe/shazamm-/infos>

## Principe général :

En début de partie, un **mur de feu** est placé au centre du pont. Les sorciers sont, quant à eux, placés à 3 cases du mur de feu (le sorcier rouge du côté gauche et le sorcier vert du côté droit).

A chaque tour, les joueurs définissent en secret et simultanément une stratégie d'attaque (à partir des points de Mana et des cartes sortilèges dont ils disposent). Lorsque les deux joueurs ont choisi leur attaque, le système détermine le vainqueur éventuel du tour et déplace le mur de feu en conséquence. S'il n'y a pas de vainqueur pendant un tour (égalité), le mur de feu ne bouge pas.

Une manche se termine lorsque le mur de feu atteint la case où se trouve l'un des deux sorciers. A la fin de la manche, les joueurs sont replacés à trois cases du mur de feu par rapport à sa position actuelle (le perdant recule et le gagnant avance). Une partie du pont s'effondre : on retire 1 plaque à chaque extrémité du pont, que l'on remplace par de la lave. Ainsi, lors de la deuxième manche, le pont ne comporte plus que 17 plaques ; puis 15 à la troisième manche, etc.

La partie se termine lorsque l'un des joueurs, à force d'être repoussé par le mur de feu, tombe dans la lave à l'une des extrémités du pont.

## Utilisation du Mana :

Chaque joueur possède des **points de Mana**. Cette réserve de Mana est utilisée pour définir la puissance de chaque attaque. Concrètement, à chaque manche, les joueurs vont avoir un capital de 50 points de Mana. À chaque tour, les joueurs vont indiquer, simultanément et en secret, la quantité de Mana qu'ils veulent dépenser afin de pousser le "mur de feu" vers l'adversaire. Un joueur doit miser au minimum 1 point de mana, le maximum étant la quantité lui restant.

Si la réserve de Mana d'un joueur tombe à zéro, il perd la manche et le mur de feu atteint automatiquement sa position actuelle.

## Utilisation des cartes sortilèges :

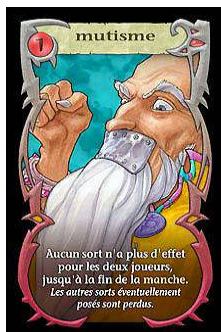
Chaque joueur dispose d'un paquet de **14 cartes sortilèges** numérotées dans lequel il pourra piocher (il y a donc 2 paquets distincts et 28 cartes sortilèges au total). A chaque début de manche, le paquet est re-mélangé.

En début de partie, chaque joueur pioche 5 cartes sortilèges dans son paquet. Le duel peut commencer !

Durant chaque tour, les joueurs peuvent accompagner leur mise (nombre de points de Mana utilisés) d'une ou plusieurs cartes "sortilège". Une fois que les deux joueurs sont prêts (i.e. ils ont défini la mise en nombre de points de Mana et ont sélectionné les cartes sortilèges à jouer), ils révèlent leur mise et leurs cartes jouées ce tour ci, le tout en même temps. On applique les effets des cartes "sortilège". Si plusieurs cartes sont jouées, les effets sont résolus en suivant l'ordre de priorité indiqué sur chacune d'elle. Le sorcier dont l'attaque est la plus forte pousse le mur d'une case vers son adversaire. Certaines cartes "sortilège" modifient le résultat de l'attaque...

A la fin d'une manche (sous réserve que le perdant ne se retrouve pas dans la lave et perde ainsi la partie), chaque joueur pioche 3 cartes "sortilège". Une nouvelle manche débute. La réserve de Mana de chaque joueur est remise à 50 points.

Les 14 cartes sortilèges sont identiques d'un joueur à l'autre, à savoir :



### 1 - Mutisme

Aucun sort n'a plus d'effet pour les deux joueurs, jusqu'à la fin de la manche. Les autres sorts éventuellement posés sont perdus.



### 2 - Clone

Je pose devant moi **une** des cartes jouées par l'adversaire au tour **précédent**. Cette carte est appliquée à ce tour, comme si je l'avais jouée normalement.



### 3 - Larcin

**Tous** les sorts joués à ce tour sont sous mon contrôle. Chacun est, à mon choix, appliqué comme si je l'avais joué, ou annulé et défaussé.



### 4 – Fin de manche

La manche est finie ! Les sorciers se replacent à 3 pas du mur de feu (dans sa position actuelle), et on commence une nouvelle manche.



## 5 – Milieu

Je replace immédiatement le mur de feu à égale distance des deux sorciers. Le tour continue ensuite normalement.



## 6 – Recyclage

Je peux rectifier ma mise, en ajoutant ou retranchant jusqu'à 5 points de mana.



## 7 – Boost attaque

La puissance de mon attaque est augmentée de 7.



## 8 – Double dose

La puissance de mon attaque est multipliée par deux.



## 9 – Qui perd gagne

Le mur de feu avance en sens inverse : vers celui qui a gagné ce tour. N'a pas d'effet si le mur de feu ne devait pas bouger.



## 10 – Brasier

Le mur de feu se déplace de deux cases au lieu d'une. Seulement s'il devait se déplacer, bien sûr.



## 11 – Résistance

Si le mur de feu devait avancer vers moi, il ne bouge pas.



## 12 – Harpagon

Si je perds ce tour (i.e. si le mur de feu avance effectivement vers moi), ma mise n'est pas retranchée de ma réserve de Mana.





### 13 – Boost réserve

Ma réserve de Mana s'augmente de 13 points. Après que j'ai payé ce que je dois.



### 14 – Aspiration

Ma réserve de Mana s'augmente du montant de la mise de l'adversaire.

## Comment je gagne ?

Si, en fin de manche, un sorcier se retrouve sur une dalle "lave", il tombe dans le fleuve de lave. Le sorcier restant sur le pont, et son joueur donc, est déclaré "Grand Vainqueur" !

## Historique de la partie

Un tableau présent dans l'interface graphique permet de visualiser l'historique de la partie (ex. : cartes sortilèges et quantité de Mana utilisées à chaque tour, vainqueur de chaque manche, date et heure de chaque évènement, etc.).

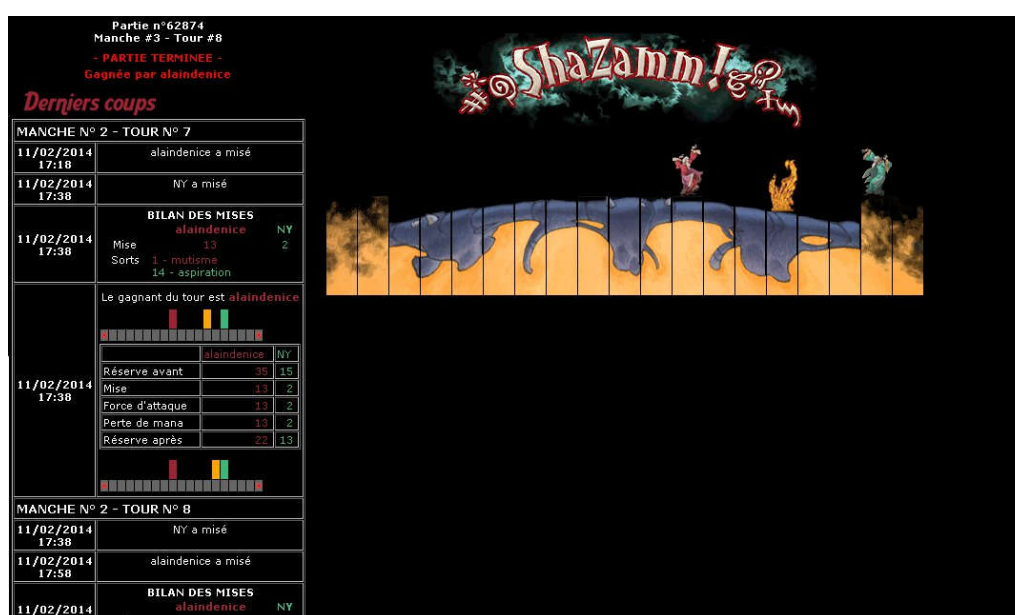


Figure 2 - Capture d'écran avec l'historique de la partie.

## 2. Consignes pour le projet

Ce projet est à réaliser **par groupe de 3 étudiants**. Le travail est à rendre pour le **mardi 9 mai à 23h55 sur Arche**. Il s'agit d'un projet assez long, n'attendez pas la dernière minute pour vous y mettre. Vous devrez rendre le **code source commenté** (le dossier du projet NetBeans compressé), ainsi qu'un **rapport** justifiant vos choix d'implémentation. Sous réserve de modification d'emploi du temps, les **soutenances** auront lieu le jeudi 11 mai après-midi (15 minutes de présentation incluant une démonstration + 10 minutes de questions). Un rendu intermédiaire est attendu pour **l'achievement 1a le vendredi 07 avril**.

Une attention particulière sera portée à la **qualité du code** (choix des structures de données adaptées, réutilisation des classes vues en cours, portée des variables appropriée private/public, structure des classes, utilisation de l'héritage et des interfaces en Java, etc.). En outre, **la note sera également déterminée en fonction du niveau d'achievement que vous atteindrez**. Lorsque vous terminez un achievement, faites une sauvegarde de votre projet pour éviter de faire planter tout votre programme en tentant de réaliser l'achievement suivant.

**Les achievements peuvent être réalisés dans l'ordre de votre choix. Les achievement 1a et 1b** correspondent à un minimum strict à atteindre respectivement pour le cours de Programmation Avancée (Java) et le cours d'Intelligence Artificielle d'Armelle Brun.

- **Achievement 1a (dépôt intermédiaire attendu sur Arche le vendredi 7 avril) :** mettre en place la partie modèle dans votre programme Java, i.e. implémenter toute la mécanique du jeu sans l'interface graphique. Dans cette version du jeu, les informations sont affichées dans la console sous forme de texte et où les joueurs saisissent leur mise et choisissent leurs cartes au clavier. Dans cette version, on considérera qu'il n'y a que des joueurs humains.

La saisie et l'affiche dans la console sont facultatifs si vous arrivez à réaliser l'achievement 2 avec interface graphique. Vous pouvez néanmoins prévoir les deux alternatives dans votre programme en laissant la possibilité de lancer le programme avec ou sans interface graphique (ce qui vous octroiera un bonus).

- **Achievement 1b (pour ceux qui suivent le cours d'Armelle Brun) :** la version 1a + la possibilité qu'il y ait un joueur ordinateur. Ce dernier sera doté d'une intelligence artificielle et doit essayer de gagner à tout prix. Vous expliquerez dans votre rapport et lors de la soutenance la stratégie mise en œuvre par votre programme pour gagner (parcours de graphe et gain attendu dans chaque état, algorithme min-max, élagage alpha-beta, ou en fonction de la mise du joueur humain au tour précédent, ou système expert à base de règles prédéfinies, etc.) et la représentation des données utilisée (graphe, ou tableau de probabilités, etc.).
- **Achievement 2 :** la version 1a (et éventuellement la version 1b pour ceux qui suivent le cours d'Armelle Brun) + une (belle) interface graphique dans une seule et même fenêtre. Vous pouvez vous inspirer de l'interface que vous trouverez en ligne sur [boiteajeux.net](http://boiteajeux.net) et essayez de la reproduire en Java, ou bien définir votre propre interface graphique du moment que tous les éléments nécessaires au jeu sont représentés. Dans cette version, les interactions ne se font plus via la console, mais via l'interface graphique. Vous devriez normalement tous être capables d'atteindre ce niveau, au vu des TD réalisés. Dans cette version, on suppose que les deux joueurs jouent sur le même ordinateur et regarde l'écran à tour de rôle (Ex. : le joueur rouge mise puis valide son choix en secret, puis il tourne la

tête pendant que le joueur vert mise à son tour, etc.). Il faudra trouver une mécanique de jeu pour que le joueur adverse ne puisse pas voir votre main.

- **Achievement 3 :** la version 2 + une interface graphique multi-fenêtres. Chaque joueur dispose d'une fenêtre. De cette manière, même si les deux joueurs jouent sur le même ordinateur, on peut faire en sorte que les fenêtres s'affichent sur des écrans différents (bureau étendu). Ainsi, chaque joueur ne verra que sa main (cartes sortilèges et mises secrètes) et l'historique du jeu.
- **Achievement 4 :** la version 3 + une durée limitée pour jouer (la durée de chaque tour est paramétrable en début de partie). A la fin du délai, si le joueur n'a pas fait de choix, le système choisi par défaut d'appliquer la mise minimale et de ne pas utiliser de carte sortilège.
- **Achievement 5 :** la version 4 + l'utilisation d'une base de données MySQL pour gérer l'historique des joueurs (nom, prénom, avatar, nombre de parties gagnées, etc.). Il faudra donc créer et installer la base et les tables appropriées, faire communiquer votre programme avec la base en JDBC (cf. aide fournie sur Arche). A défaut d'une base MySQL, vous pouvez également utiliser un fichier XML pour stocker/récupérer les informations si vous préférez.

Pour réussir ce projet, il faut se poser les bonnes questions : Comment représenter intelligemment les données dans votre programme ? Quelle structure adopter (les variables, les classes nécessaires, héritage, etc.) ? Votre rapport devra faire état de vos réflexions et de vos choix sur ce projet.

Dans le rapport, il doit y avoir le descriptif des fichiers et des classes (éventuellement accompagné d'un diagramme de classes), la présentation et le fonctionnement de votre application (notamment comment l'installer et le lancer), la présentation de votre interface graphique (avec des captures d'écran) et des explications quant à la stratégie adoptée par votre programme d'intelligence artificielle (joueur ordinateur) pour gagner.

Un forum d'échange, divers documents utiles au projet et les images du jeu sont disponibles sur Arche dans la section projet.

#### Sites utiles :

- <https://docs.oracle.com/javase/7/docs/api/>
- <http://docs.oracle.com/javase/tutorial/jdbc/basics/gettingstarted.html>
- <http://boiteajeux.net/>