



Cahier des charges : Hanabi

Université Pierre et Marie Curie
Projet ANDROIDE
2015-2016

Assmann Catalina
Hubert Cédric
Antunes Costa Gonçalves Daniel
Wolfrom Matthieu

1 Présentation du projet

1.1 Contexte

Ce projet se déroule dans le contexte de l'UE Projet de première année de Master ANDROIDE de l'UPMC. Nous sommes un groupe de quatre étudiants qui doivent mener à terme un projet proposé par leurs encadrants. Ce projet repose sur l'utilisation d'une intelligence artificielle qui exploite la logique épistémique afin d'appliquer la meilleure stratégie possible pour le jeu Hanabi.

1.2 Introduction du jeu

Hanabi est un jeu de cartes coopératif. Les cartes représentent des feux d'artifices et les joueurs doivent collaborer afin composer cinq feux d'artifices de couleurs différentes.

Personne ne connaît ses propres cartes, mais voit les cartes des autres. En utilisant des indices et les cartes visibles, les joueurs doivent essayer de poser les feux d'artifices en suites croissantes de cartes de même couleur.

Le score final est la somme des dernières cartes posées de chaque couleur.

Pour réussir ce défi, chaque joueur peut à son tour soit donner un indice à un autre joueur, soit défausser une carte, soit jouer une carte.

Un indice consiste à indiquer quelles cartes sont d'une certaine couleur ou d'une valeur. Un indice négatif peut aussi être donné, indiquant qu'un joueur ne possède aucune carte d'une certaine couleur ou une certaine valeur. A chaque indice, un jeton 'indice' est utilisé et donc retourné. Si le joueur choisit de défausser une carte, il regagne un jeton 'indice'. Par contre, quand un joueur tente sa chance et joue une carte, il est possible que la carte ne complète pas la suite de sa couleur de feu d'artifice, dans ce cas un jeton 'éclair' est retourné. Une fois trois jetons éclairs sont retournés, le jeu est fini et les joueurs ont perdu.

1.3 Objectifs

L'objectif de ce projet est de créer un logiciel qui permet à un joueur humain de jouer à Hanabi avec une ou plusieurs intelligences artificielles (maximum quatre). Nous allons proposer plusieurs modes de jeu :

Facile :

- Avec ou sans jetons
- Avec toutes les cartes visibles

Normale :

- Avec deux à cinq joueurs, toujours au plus un joueur humain. Pour une partie entre 2 ou 3, chaque joueur aura 5 cartes. Pour plus de joueurs,

- chaque joueur aura 4 cartes.
 - Avec tous les IAs partageant la même stratégie ou avec des stratégies différentes.
 - L'utilisateur pourra y jouer via une interface graphique qui lui permettra de voir les cartes de son équipe, mais pas les siennes.
- Il y aura des options pour sauvegarder sa partie et la continuer ultérieurement.

1.4 Description de l'existant

Le logiciel sera réalisé avec le langage Java, et sera déployable sous les principales plateformes le supportant. Nous l'avons choisi car cet un langage orienté objet et donc bien organisé avec des classes bien reparties.

Nous allons donc créer une interface graphique en utilisant Java Swing.

2 Expression des besoins

2.1 Besoins fonctionnels

Le logiciel doit permettre à l'utilisateur de jouer une partie du jeu Hanabi avec d'autres joueurs dirigés par des intelligences artificielles. Pour ceci, nous avons besoin de créer les fonctionnalités suivantes :

L'interface Graphique :

2.1.1 Un menu d'accueil

1. Les règles du jeu

L'utilisateur pourra cliquer sur ce bouton pour qu'on lui affiche les règles du jeu.

2. Continuer une ancienne partie

L'utilisateur pourra cliquer sur ce bouton pour qu'on lui affiche les parties sauvegardées et en choisir une pour la continuer.

3. Commencer une nouvelle partie

En cliquant sur ce bouton, le joueur verra un écran qui lui affichera plusieurs modes de jeu :

- Facile - avec toutes les cartes visibles pour tout joueur
- Facile - sans jetons

— Normal

Dans tous les modes, l'utilisateur va devoir préciser avec combien d'IAs il voudra jouer (maximum 4 IAs). Il devra aussi préciser si les IAs joueront toutes avec la même stratégie ou avec des stratégies différentes.

2.1.2 L'affichage de la partie

1. Visualiser les cartes des autres joueurs

Les cartes des autres joueurs seront visibles à l'utilisateur.

2. Visualiser les jetons indices restants

Les jetons seront affichés sur la table et donc visible à l'utilisateur. Une fois utilisé, le jeton ne sera plus visible.

3. Visualiser les jetons éclairs retournés

Les jetons éclairs seront tous affichés sur la table, soit retournés ou pas.

4. Visualiser les cartes jouées

Les cartes jouées seront visibles sur la table.

5. Visualiser les cartes défaussées

Ce bouton permettra à l'utilisateur de revoir les cartes défaussées pour en déduire des informations sur les siennes. L'interface va juste afficher toutes les couleurs des cartes défaussées et les numéros correspondants.

6. Visualiser les indices donnés pour les cartes de chaque joueur

A chaque indice reçu, l'utilisateur le verra affiché sur la carte concerné. Les indices sont soit une couleur, soit un nombre. L'utilisateur peut aussi voir les indices données aux autres joueurs, pour connaître leur base de connaissance de leur cartes.

7. Jouer une carte

L'utilisateur pourra cliquer sur ce bouton et après choisir la carte qu'il veut utiliser. La carte sera donc affichée et ensuite soit posée sur la table,

soit défaussée si la carte ne convient pas. Si la carte est défaussée, un jeton d'éclair est retourné sur la table.

Une fois que trois jetons d'éclair ont été retournés le jeu est fini et la partie est perdue.

8. Défausser une carte

L'utilisateur pourra cliquer sur ce bouton et après choisir la carte qu'il souhaite défausser. Elle sera donc ajoutée à la pile de cartes défaussées, et un jeton d'indice sera retourné pour pouvoir être réutilisé.

9. Donner un indice

Pour donner un indice, l'utilisateur cliquera sur ce bouton et après sur la main qui l'intéresse. Il choisira ensuite soit une couleur, soit un nombre, et ceci vont être attribués aux cartes correspondant. Il ne pourra pas changer les informations déjà donnés par rapport à une carte. Un jeton d'indice sera donc retourné pour plus être utilisable.

10. Sauvegarder la partie

L'utilisateur pourra donner un nom à sa partie et la sauvegarder pour continuer à un autre moment.

11. Fin du jeu

La fin du jeu s'annoncera via une annonce. L'annonce affichera s'ils ont gagné ou perdu. S'ils ont perdu, la raison sera affiché (3 éclairs). S'ils ont gagné, leur score sera affiché et la réaction de la foule. L'utilisateur pourra ensuite choisir s'il veut faire une nouvelle partie, continuer une autre partie ou quitter le jeu.



FIGURE 1 – Prototype d’interface graphique sur lequel on se basera pour notre implémentation.

2.2 Besoins non fonctionnels

L’intelligence artificielle devra décider de son prochain coup en un temps limité afin que la partie se déroule normalement. Il est donc indispensable que les IAs soient efficaces et optimisées.

Nous avons réfléchi à plusieurs possibilités pour les IAs :

- **Une IA simple** : Cette IA jouera d’une manière basique. Elle va jouer des cartes valides si possible (quand elle connaît la carte, et elle peut être posée), sinon, elle jouera d’une manière plus ou moins aléatoire.
- **Une IA sans risque** : Cette IA jouera des cartes que quand elle les connaît. Dans le cas contraire, elle aura une fonction d’évaluation pour calculer si c’est mieux de défausser une carte ou de donner un indice et lequel.
- **Une IA qui prend des risques** : Cette IA sera moins adverse au risque. En utilisant une heuristique, elle va choisir son coup par rapport à la probabilité de réussite, qui sera forcément moins de 100 pourcent (comme l’IA sans risque).

- **Une IA omnisciente** : Cette IA connaît toutes les cartes (même les siennes), elle se concentrera donc sur les priorités entre les différents coups possibles (jouer une carte qu'on connaît, défausser des cartes pour gagner des jetons 'indice' ou donner des indices aux autres joueurs)

2.3 Critères d'acceptabilité du produit

Notre but est de simuler un jeu entre deux ou plusieurs humains, en utilisant des IAs. Nous avons donc besoin d'avoir des intelligences artificielles efficaces qui ne prennent pas trop de temps pour calculer leur prochain coup. Nos critères d'acceptabilité par rapport aux intelligences artificielles sont donc :

- De permettre à l'utilisateur de jouer une partie de Hanabi sans devoir attendre trop longtemps pour son tour.
- L'intelligence artificielle est considérée efficace si elle atteint systématiquement un score entre 20 et 25 (le score maximum possible) en jouant qu'entre des IAs avec la même stratégie.

Nous voulons aussi que le jeu soit facile et simple à jouer pour le joueur humain, donc une interface claire et compréhensible est indispensable. Il faut que toute personne soit capable de s'en servir. Nous voulons aussi avoir au moins une IA qui joue de façon réaliste plutôt que d'essayer de faire un score maximal.

3 Déroulement du projet

3.1 Planification

- Un cahier des charges devra être rendu environ 3-4 semaines après le début du projet (fin février).
- Un rapport complet du projet ainsi que le code devra être rendu le 9 mai 2016.
- Une soutenance aura lieu fin mai 2016.

3.2 Documentation

Le logiciel devra être accompagné d'un document expliquant les règles du jeu, ainsi qu'un manuel d'utilisation. Il y aura également un rapport sur le déroulement du projet, expliquant en détail les raisonnements utilisés.

3.3 Ressources

Il y a plusieurs équipes qui ont déjà développé Hanabi en Java. Leurs projets sont disponibles sur Github et peuvent nous servir comme exemple. Nous pouvons nous inspirer de leurs interfaces et aussi leurs intelligences artificielles.

3.4 Références

GitHub Hanabi :

<https://github.com/jason17055/play-hanabi>

<https://github.com/kazocsaba/Hanabi>