



Projet long - Jeu de société Cluedo
Rapport de l'itération 3 - Groupe IJ01

*Agathe Perrin, Antonin Litschgy, Mickaël Song, Maëlis Marchand,
Thierry Xu, Tom Bonetto, Antoine Dalle-Fratte*

Table des matières

Introduction	3
Travail réalisé au niveau du code	3
Détails de la réalisation	3

Introduction

L'objectif de notre projet est de réaliser un jeu de société type Cluedo qui se joue sur ordinateur. Un utilisateur aura la possibilité de jouer contre des ordinateurs présentant différentes stratégies. Les principales fonctionnalités du jeu original seront retrouvées, avec notamment : le déplacement d'un pion sur le plateau de jeu, la formulation d'hypothèses et d'accusations, l'accès à une grille pour noter ses indices au fil de la partie, etc.

Ce rapport présente ce que nous avons réalisé lors de la troisième et dernière itération du projet.

Travail réalisé au niveau du code

- Formulation d'hypothèses/d'accusations pour le joueur humain
- Remplacement des classes Stratégie par des classes BotNaif, BotExpert, etc.
- Création d'un carnet de jeu pour les bots (classe Carnet)
- Travail sur la classe Jeu
- Stratégie de déplacement pour les bots

Détails de la réalisation

1. Formulation d'hypothèses/d'accusations pour le joueur humain :

Nous avons ajouté à notre interface graphique un bouton intitulé « Formuler une hypothèse ». Lorsque le joueur humain clique dessus, il doit choisir le personnage à soupçonner, puis l'arme et enfin le lieu. Pour sélectionner une carte, il doit simplement cliquer dessus.

Le principe est le même pour le bouton « Formuler une accusation ».

2. Remplacement des classes Stratégie par des classes BotNaif, BotExpert, etc.

Nous avons changé notre conception des stratégies. Au départ, un joueur possédait un attribut « stratégie », mais il a été finalement plus naturel logique de réaliser des classes BotNaif, BotExpert... qui héritent directement de la classe Joueur.

3. Création d'un carnet de jeu pour les bots (classe Carnet)

Nous avons réalisé une classe `Carnet` qui représente la grille du Cluedo utilisable par les bots. Le principe de cette grille est le même que celle du joueur humain (qui le fait manuellement dans l'interface graphique) : quand un bot voit une carte, il coche la case correspondante dans sa grille.

4. Travail sur la classe `Jeu`

Nous avons continué d'implémenter la classe `Jeu`. Nous y avons notamment regroupé les 21 cartes en différents paquets de type `ArrayList<Carte>` : d'abord un paquet qui contient toutes les cartes, puis un paquet pour les cartes `Personnage`, un pour les cartes `Arme` et un pour les cartes `Salle`.

5. Stratégie de déplacement des bots

Les bots possèdent maintenant deux méthodes : une méthode pour trouver la salle la plus proche de leur position actuelle et qui n'est pas la dernière salle qu'ils ont visitée, et une méthode qui les fait se déplacer de leur position actuelle jusqu'à une position de destination passée en paramètre de la méthode, ce déplacement suit le chemin le plus court. Cette méthode prend également en paramètre le score obtenu aux dés, si le score obtenu est suffisant pour que le bot se déplace jusqu'à la case d'arrivée alors la méthode renverra la position de la case d'arrivée, en revanche si le score est insuffisant, elle renverra la case la plus éloignée que le bot est réussi à atteindre.

6. Menu principal & règles du jeu

Nous avons ajouté le menu principal où l'on a mis une option pour mettre ou enlever la musique de fond (il s'agit d'un élément clip). Nous avons également ajouté les règles du jeu qui sont expliquées sur 2 `JFrame` successives.

7. Enchaînement des interfaces graphiques

Nous avons implémenté l'enchaînement des interfaces graphiques, qui est détaillé dans le manuel utilisateur.

8. Plateau de jeu

L'interface graphique du plateau de jeu a été améliorée. Elle comporte maintenant les murs séparant les couloirs des différentes salles. Les zones où il n'y a pas de mur correspondent donc à des portes. Le plateau gère la superposition d'images, notamment d'une case et du pion d'un joueur par exemple. De plus, si l'attribut `surbrillance` de case est vrai, alors l'image de la case sera un peu plus claire. C'est le cas notamment pour les cases accessibles par le joueur après un lancer de dés.