



**Projet long - Jeu de société Cluedo**  
**Rapport de l'itération 1 - Groupe IJ01**

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

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>Travaux réalisés en méthodes agiles</b>	<b>3</b>
Elevator pitch	3
Charte produit	3
Backlog produit	4
<b>Travail réalisé au niveau du code</b>	<b>6</b>
<b>Détails de la réalisation</b>	<b>6</b>
<b>Diagramme de classes</b>	<b>8</b>

## 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 première itération du projet.

## Travaux réalisés en méthodes agiles

Lors de cette première itération, nous avons d'abord réalisé plusieurs exercices en méthodes agiles. Cela nous a permis de mieux définir notre sujet et de nous rendre compte du travail à faire. Nous avons d'abord réalisé un **“elevator pitch”**, puis une **charte produit** et enfin un **backlog produit**. Ce dernier nous a ensuite permis de nous répartir le travail plus simplement.

### 1. Elevator pitch

Voici notre elevator pitch, réalisé lors de la première séance de TD de méthodes agiles :

**“Pour** les personnes à partir de 8 ans **qui veulent** se divertir. **Notre produit est** un programme de jeu d'enquête **qui** permet de jouer à un jeu de type cluedo. **À la différence des** autres jeux de type cluedo, celui-ci **permet de** jouer à plusieurs ou seul avec des ordinateurs proposant différentes difficultés/stratégies.”

### 2. Charte produit

En méthodes agiles, nous avons également réalisé une charte produit de notre projet. Cela nous a permis de mieux définir la cible de notre futur produit, ainsi que les performances que nous souhaitons atteindre, mais aussi les difficultés que nous pouvons rencontrer.

Vision produit	Critères de succès (métriques)																				
Pour les personnes à partir de 8 ans <b>qui veulent</b> se divertir. <b>Notre produit est</b> un programme de jeu d'enquête <b>qui</b> permet de jouer à un jeu de type Cluedo. <b>À la différence</b> des autres jeux de type Cluedo, celui-ci <b>permet de</b> jouer à plusieurs ou seul avec des ordinateurs proposant différentes difficultés/stratégies.	<ul style="list-style-type: none"><li>• Jeu fonctionnel</li><li>• Utilisateurs satisfaits</li></ul>																				
Clients / Utilisateurs	Risques																				
<ul style="list-style-type: none"><li>• Cible marketing : personnes de plus de 8 ans</li><li>• Catégorie professionnelle : toutes</li></ul>	<ul style="list-style-type: none"><li>• Produit non terminé ou non fonctionnel (bogué)</li></ul>																				
Valeur métier	Performances																				
<ol style="list-style-type: none"><li>1. Choisir la stratégie de l'ordinateur</li><li>2. Jouer seul</li><li>3. Avoir un historique de ses parties</li><li>4. Thème du plateau de jeu en DLC</li></ol>	<ul style="list-style-type: none"><li>• Répondre en un temps raisonnable</li></ul>																				
Jalons : dates ou fonctionnalités	<< Tradeoff matrix>>																				
<ul style="list-style-type: none"><li>• 9/04 : rendu de l'itération 1</li><li>• 23/04 : rendu de l'itération 2</li><li>• 21/05 : rendu de l'itération 3</li><li>• 31/05 : oral de projet</li></ul>	<table><tr><th></th><th>Fixé</th><th>Ferme</th><th>Flexible</th></tr><tr><td>Périmètre</td><td></td><td>X</td><td></td></tr><tr><td>Délai</td><td>X</td><td></td><td></td></tr><tr><td>Coût</td><td></td><td></td><td>X</td></tr><tr><td>Qualité</td><td></td><td>X</td><td></td></tr></table>		Fixé	Ferme	Flexible	Périmètre		X		Délai	X			Coût			X	Qualité		X	
	Fixé	Ferme	Flexible																		
Périmètre		X																			
Délai	X																				
Coût			X																		
Qualité		X																			

### 3. Backlog produit

Enfin, lors de la seconde séance de méthodes agiles, nous avons réalisé notre backlog produit. Ce travail a été essentiel pour nous aider à définir les différents *items* à réaliser lors de notre projet long, en faisant le point sur toutes les fonctionnalités.

Lancer la partie - 88									
Choisir le nombre d'adversaires - 7		Choisir son personnage - 3	Mettre en place la partie - 10						
Afficher les stratégies des bots - 3	Choisir la stratégie du bot - 4		Tirer aléatoirement les 3 cartes de l'enquête - 4	Distribuer les cartes aux joueurs - 4	Installer les pions - 2	Se déplacer - 26			Formuler une
Difficultés allant de 1 à 10						Lancer les dés et afficher le résultat - 5	Choisir un déplacement possible avec son score aux dés - 7	Déplacer son pion - 8	Prendre un raccourci - 6
						Enoncer une hypothèse - 9			

## Jouer au cluedo - 110

### Jouer la partie - 68

hypothèse - 17

Formuler une accusation - 10

Ouvrir le carnet - 4

Mettre la partie en pause - 11

Chacun leur tour les joueurs  
montrent une carte de leur jeu  
si la carte est dans les  
hypothèses - 8

Si l'accusation est vraie :  
le joueur gagne et la  
partie est finie - 3

Si l'accusation est fausse :  
le joueur est éliminé et la  
partie continue - 7

Ajouter des  
notes - 2

Effacer des  
notes - 2

Retour au menu - 4

Changer les options - 7

Changer la  
langue - 4

Couper le son / la  
musique - 3

Jouer en  
français - 2

Jouer en  
anglais -  
2

**Lire les règles - 3**

**Options du jeu - 18**

Quitter le jeu - 1

Afficher les règles  
- 2

Quitter l'affichage  
des règles - 1

Revenir sur le  
menu principal - 4

Changer la  
langue - 4

Couper le son  
/ la musique -  
3

Changer le thème -  
7

Jouer en  
français -  
2

Jouer en  
anglais -  
2

## Travail réalisé au niveau du code

Après avoir terminé notre backlog produit, nous avons pu répartir le code à réaliser au sein du groupe. Nous avons identifié les principaux objectifs à coder :

- Mise en place de la partie : initialisation, distribution des cartes aux joueurs, position des pions sur le plateau ;
- Affichage du menu de démarrage ;
- Interface graphique d'une partie ;
- Plateau de jeu : affichage des cases et de salles, déplacement des pions sur le plateau ;
- Grille de jeu : l'utilisateur doit avoir à sa disposition une grille pour cocher les différents indices qu'il possède au fil de la partie ;
- Arbitrage d'une partie : lors d'une partie, il est nécessaire de gérer le passage d'un joueur à l'autre ;
- Stratégies de jeu : l'utilisateur humain aura la possibilité d'affronter des bots adoptant des stratégies plus ou moins élaborées.

Nous avons donc réparti ces différents objectifs au sein de l'équipe.

## Détails de la réalisation

### 1. Mise en place de la partie

Plusieurs classes ont été créées.

- Une classe carte : chaque carte possède un nom (cf classe NomCarte) et un type (cf classe TypeCarte).
- Une classe joueur : chaque joueur possède un nom, une stratégie et une main de plusieurs cartes (qui va dépendre du nombre de joueurs).
- Une classe MiseEnPlace : pour l'instant elle contient notamment la méthode servant à tirer une carte de chaque type en début de partie: `tirageCrime()`. Il reste à coder les méthodes concernant la mise de côté de ces trois cartes, la disposition des pions sur la plateau, et la distribution des cartes aux joueurs.

### 2. Stratégies de jeu

Afin de réaliser les stratégies de jeu, une classe hypothèse a été créée. Une hypothèse est composée d'une carte arme, d'une carte salle et d'une carte personnage. Une méthode `afficher()` permet d'afficher sur le terminal l'hypothèse du joueur.

Ensuite, une interface stratégie a été mise en place avec comme méthodes `formulerHypothese`, `formulerAccusation` et `montrerCarte` (permet au joueur de montrer, suite à l'hypothèse d'un autre joueur, une des cartes proposées s'il le possède).

Ces méthodes seront implantées de façon différente selon la stratégie : facile, expert et humain.

### 3. Grille de jeu

Afin de réaliser la grille, nous avons d'abord créé un par un, la grille des lieux, la grille des armes, la grille des personnages et la grille des joueurs.

Les 3 premières grilles ont été réalisées de la même manière, nous avons d'abord créé une méthode ligne(string) qui crée une ligne avec un nom et 7 JCheckBox.

Puis chaque grille a été réalisée par un GridLayout(n,1) (n = 6 pour les personnages et les armes et n = 10 pour le lieu) où l'on place n lignes avec les noms correspondant.

La grille Joueur a été réalisée de la même manière que la méthode ligne, sauf qu'à la place des JCheckBox, des JTextField ont été mis.

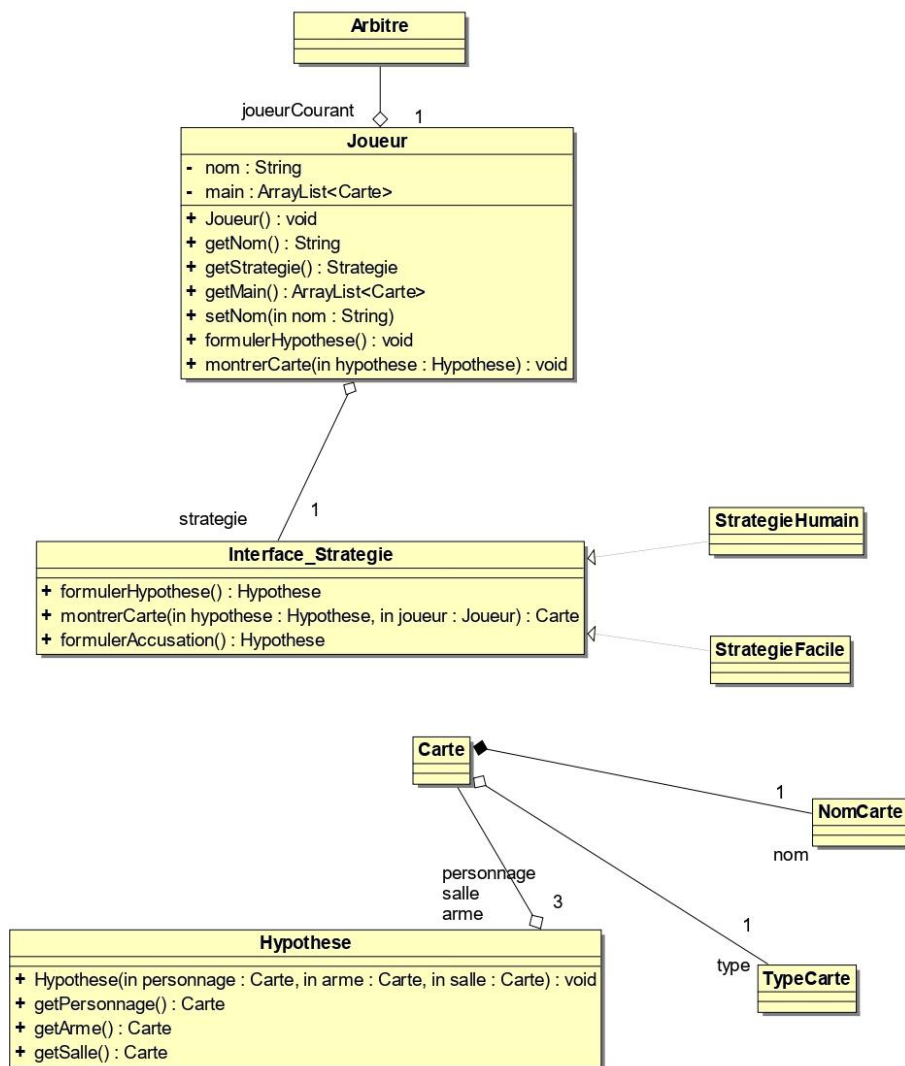
### 4. Plateau de jeu

Le plateau de jeu a été créé avec les différentes cases qui composent : les salles, les portes, les murs et enfin les cases du couloirs. Une classe position permet aux différents objets du plateau d'avoir des coordonnées. Le plateau a été implémenté avec une matrice de Case, les salles et les différentes cases qui les composent ont été renseignées dans une HashMap, liste avec clé (Type énumération Salles) et données (ArrayList de Case).

### 5. Interface graphique

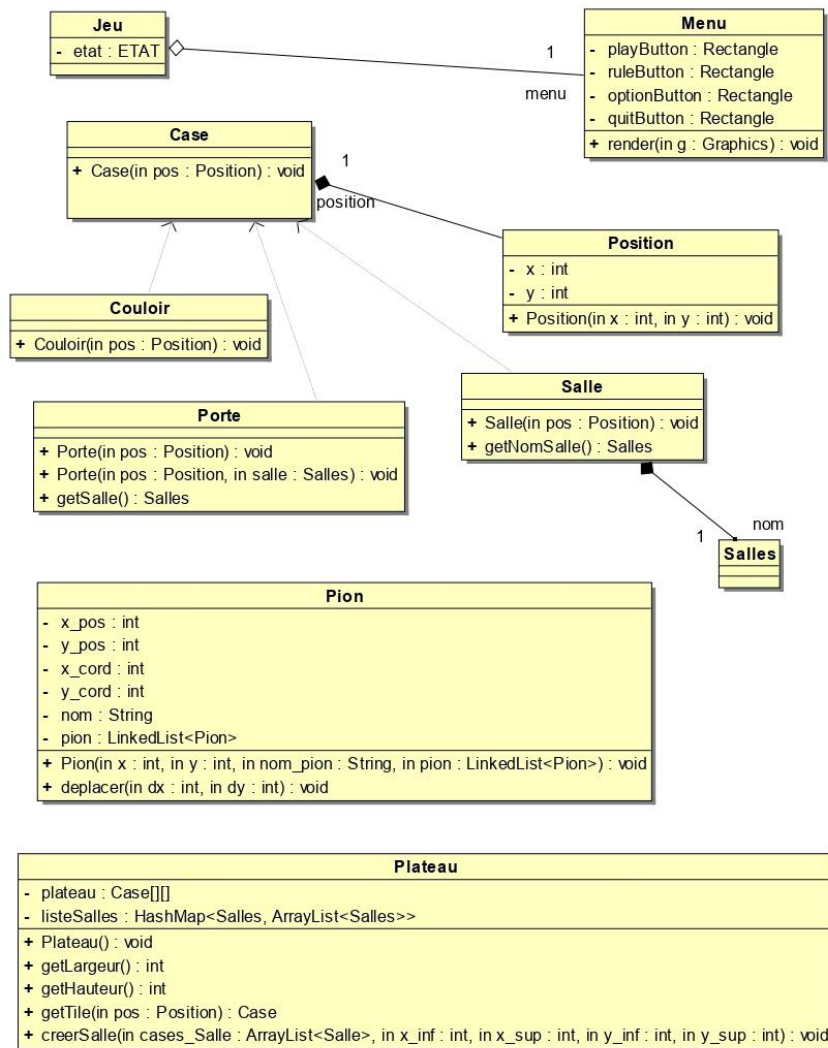
Du point de vue interface utilisateur, pour l'instant c'est une esquisse du menu du jeu qui a été créée, un menu sur le côté droit de la fenêtre pour accéder à la grille, les cartes et la console, un menu à gauche pour les accusations et les hypothèses et au centre l'emplacement pour le plateau de jeu.

## Diagramme de classes



Cette première partie du diagramme de classes présente les classes en lien avec les cartes du jeu.





Cette deuxième partie du diagramme de classes présente les classes en lien avec le plateau de jeu.

Les classes GrilleCluedo, GUI et MiseEnPlace ne sont pas représentées sur le diagramme par manque de place et car la classe MiseEnPlace allie les deux aspects du jeu (cartes et plateau).