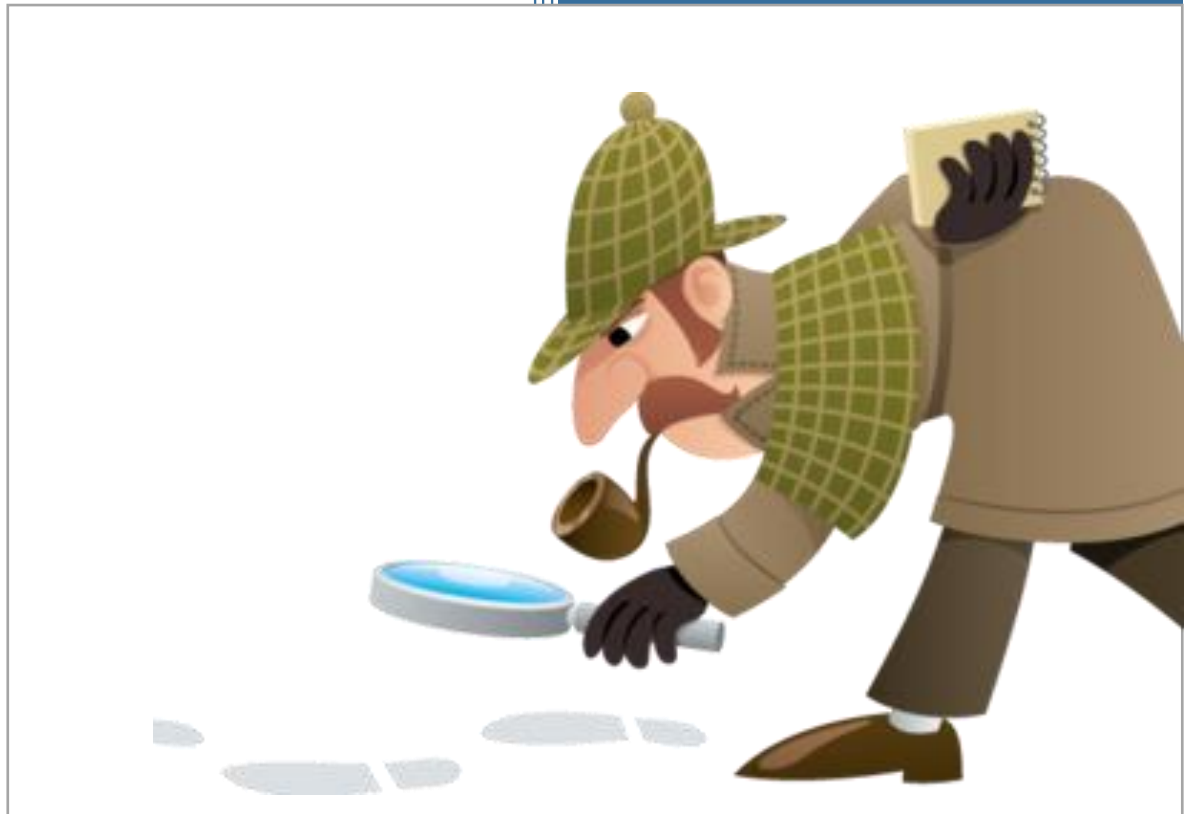




# Rapport de projet



BOMY Clara  
M1 – JAVA  
Novembre 2018

# TABLE DES MATIERES

---

Introduction .....	2
1. Description du projet .....	2
1.1 Brève description.....	2
1.2 Fonctionnalités .....	2
1.3 Déroulement du jeu .....	3
Menu principal .....	3
Commencer une nouvelle partie.....	3
Déroulement d’une enquête.....	4
❖ Les analyses .....	4
❖ Les indices.....	5
❖ Les rencontres avec les suspects.....	5
1.4 Conception architecturale .....	6
1.4.1 Brève description des classes .....	7
❖ Le jeu .....	7
❖ Les personnages .....	7
❖ L’enquete.....	7
❖ La classe conductrice du programme.....	8
1.4.2 Utilisation des interfaces .....	8
1.5 Diagramme de classes .....	8
2. Analyse personnelle .....	8
2.1 Choix faits en termes d’organisation.....	8
2.2 Bilan personnel.....	8
2.2.1 Difficultés rencontrées.....	8
2.2.2 Points de satisfaction .....	9
Conclusion.....	9
Annexe : diagramme de classes.....	10

# INTRODUCTION

---

Mon projet, *JavaSpector*, est un jeu d'enquête en console dans lequel le joueur, l'enquêteur, a pour objectif de trouver le coupable parmi une série de suspects. Mais, cela ne sera pas si simple : en effet, le meurtrier peut compter sur son fidèle partenaire pour tromper le joueur. L'enquêteur pourra interroger les suspects pour obtenir des dépositions qu'il pourra coupler, regrouper et relier aux preuves qu'il trouvera en analysant les quelques éléments de l'enquête : le corps de la victime, l'arme l'ayant tué et la scène de crime.

J'ai décidé de réaliser ce projet car il me plaisait plus que les sujets proposés et m'assurait ainsi l'inspiration et la motivation nécessaire à sa bonne réalisation. De plus, j'ai fait le choix de travailler seule sur ce projet.

## 1. DESCRIPTION DU PROJET

---

### 1.1 BREVE DESCRIPTION

Ce projet est une adaptation du Cluedo classique couplé à un système tiré des jeux de rôle où la réussite d'une action dépend de lancers de dés. Comme dans le jeu Cluedo, nous retrouvons ainsi l'arme du crime, la scène de crime, les différents personnages et les indices que possèdent les autres joueurs, ici remplacés par des suspects.

Une enquête pouvant s'éterniser, le joueur aura la possibilité d'enregistrer sa progression pour y revenir plus tard, ou simplement de s'exercer sur un mode de difficulté moindre sans pour autant perdre la première sauvegarde. Si jamais le joueur se lasse, il pourra sans problème supprimer sa partie ou alors, la transmettre à un autre joueur qui pourra la résoudre à sa place.

Une des spécificités de ce jeu est que sa fin n'implique pas forcément la réussite de l'enquête : en effet, comme dans la vraie vie, des erreurs judiciaires peuvent se produire et rien n'empêchera le joueur d'arrêter un innocent.

### 1.2 FONCTIONNALITES

Sur le jeu en lui-même, le joueur a la possibilité de :

- Créer une nouvelle partie
- Gérer un système de sauvegardes de parties (sauvegarder, charger, supprimer)
- Choisir entre plusieurs niveaux de difficultés
- Vivre une expérience immersive et hors du commun

Au cœur d'une partie,

- Le joueur aura le choix entre :
  - Analyser un élément d'enquête (victime, arme du crime, scène du crime)
  - Interroger, disculper voire arrêter un suspect
  - Gérer un ensemble d'indices résultant des interrogatoires et analyses préalables
- Le meurtrier ainsi que son partenaire de crime auront la possibilité de détourner les soupçons en fournissant de faux alibis et témoignages au joueur

- Les différents suspects seront plus ou moins enclins à donner leurs informations au joueur durant l'enquête, dépendamment de l'attention de l'enquêteur et de l'état d'esprit des suspects

L'enquête se termine lorsque l'enquêteur décide d'arrêter un des suspects. S'il s'agit effectivement du coupable, ce dernier dévoile la raison de son crime, auquel cas, l'enquête est résolue. Dans le cas contraire, le meurtrier court toujours et est libre de réitérer son crime.

### 1.3 DEROULEMENT DU JEU

#### MENU PRINCIPAL

Dans ce jeu, vous incarnez le rôle d'un talentueux enquêteur. Au début du jeu, vous avez le choix entrer commencer une partie ou charger une partie préexistante.

```
(Standardiste) Bienvenue à JavaSpector. Que puis-je pour vous?

(Choix 1) Commencer une nouvelle enquête
(Choix 2) Continuer l'enquête en cours
(Choix 3) Déposer un rapport d'enquête (sauvegarde)
(Choix 4) Reprendre une enquête
(Choix 5) Abandonner une enquête

(Choix 6) Règles du jeu
(Choix 7) Quitter le bureau d'enquête

Votre choix : 5

Choisissez la sauvegarde à supprimer :

(Choix 1) Clara Bomy
(Choix 2) Marine Dael
(Choix 3) Annuler

Votre choix : |
```

#### COMMENCER UNE NOUVELLE PARTIE

Lorsque vous commencez une partie, vous devez tout d'abord saisir vos informations personnelles (nom, prénom et sexe) afin de définir le nom du fichier de sauvegarde et de personnaliser le jeu. Une fois cela fait, vous devez déterminer le niveau de difficulté que vous souhaitez pour votre enquête : selon la difficulté, il y aura cinq, huit ou onze suspects qui seront plus ou moins coopératifs. De plus, vos propres compétences, niveau d'intelligence et niveau de manipulation, seront directement impactées par la difficulté choisie.

```
(Standardiste) Veuillez entrer votre nom :

- Dael

(Standardiste) Et maintenant votre prénom :

- Marine

(Vous) Croyez moi, je suis

(Choix 1) Un talentueux inspecteur !
(Choix 2) Une talentueuse inspectrice !

Votre choix : 2

(Vous) Et je suis tout à fait capable de résoudre des enquêtes

(Choix 1) Basiques.
(Choix 2) Classiques.
(Choix 3) Avancées.

Votre choix : 2
```

Votre supérieur vous fait ensuite passer un test d'aptitude. Vous pourrez ainsi avoir accès à vos niveaux de compétences. Il vous invite ensuite à choisir une affaire : une personne a été tuée et votre mission est de trouver le coupable parmi une liste de suspects.

```
(Standardiste) Mmmh. Votre futur supérieur vous attend à l'intérieur.

-> Vous quittez la réception...

(Lieutenant Chef) Bonjour. J'ai consulté votre dossier, et le moins qu'on puisse dire, c'est que vous êtes une sacrée enquêtrice ! Mais on va quand même vérifier vos capacités.

-> Vous passez un test d'aptitude...

Rapport du test d'aptitude de Marine Dael :

Votre niveau d'intelligence : 40
Votre niveau de manipulation : 40

-> Vous reposez le rapport...

(Votre supérieur) Voici des enquêtes à votre niveau, je vous laisse en choisir une.

-> Vous choisissez une affaire...
```

## DEROULEMENT D'UNE ENQUETE

Vous pouvez dès à présent commencer votre enquête par exemple, en interrogeant des suspects ou en analysant les éléments de l'enquête.

```
(Clara Bomy) Aujourd'hui, je vais...

(Choix 1) Revoir mon test d'aptitude

(Choix 2) Passer en revue les éléments de l'enquête.
(Choix 3) Consulter mes indices.
(Choix 4) Appeler un suspect.

(Choix 5) Retourner au bureau.

Votre choix : |
```

## ❖ LES ANALYSES

Pour vous aider dans votre enquête, vous pourrez analyser l'arme du crime, la scène de crime et le corps de la victime. Cela vous permettra d'obtenir des preuves pouvant s'avérer précieuses pour votre enquête.

```
(Enquêteur) Voyons voir les éléments de l'enquête...

(Choix 1) Autopsier la victime.
(Choix 2) Analyser l'arme du crime.
(Choix 3) Fouiller la scène du crime.

(Choix 4) Retourner à l'enquête.

Votre choix : 3

L'équipe sur le terrain a fouillé la scène de crime. Elle a réuni les indices suivants :

- Preuve découverte : ticket de métro

- Preuve découverte : tâches de sang

- Preuve découverte : empreintes digitales

-> Vous relevez les indices...

|
```

❖ **LES INDICES**

L'ensemble des indices trouvés et des dépositions sont disponibles dans la rubrique « Consulter mes indices ».

```
Indices découverts dans cette enquête :

Indice 1 : Victime - tâches de sang
Indice 2 : Arme du crime - empreintes digitales
Indice 3 : Arme du crime - empreintes digitales
Indice 4 : Scène de crime - mégot de cigarette
Indice 5 : Camille Pasquier - Je suis le criminel que vous recherchez
Indice 6 : Pierre Juzeau - Je me souviens avoir vu Laure Morelle avec une femme de petite taille près du lieu du crime.
Indice 7 : Pierre Juzeau - J'ai travaillé chez moi avec mes enfants. Il y avait aussi Camille Pasquier avec moi.

-> Continuer...
|
```

❖ **LES RENCONTRES AVEC LES SUSPECTS**

Les rencontres avec les suspects sont capitales pour votre enquête car elles peuvent vous permettre de relever des dépositions parfois cruciales pour votre enquête.

Avant de commencer, vous devez choisir, parmi la liste de suspects, celui que vous souhaitez voir. Vous accéderez ensuite à la fiche de présentation du suspect :

<pre>(Enquêteur) Je veux voir  (Choix 1) Marine Juzeau - potentiellement innocent (Choix 2) Laure Juzeau (Choix 3) Luc Dang (Choix 4) Pierre Dael (Choix 5) Alice Dang - potentiellement innocent (Choix 6) Guillaume Morelle (Choix 7) Luc Pasquier (Choix 8) Marine Boore  (Choix 9) Annuler  Votre choix : 6</pre>	<pre>Informations sur le suspect :  Nom, prénom      : Morelle Guillaume Sexe             : homme Age              : 54 ans Etat             : stressé Description physique : petit, gothique  -&gt; Vous reposez le rapport...  (Guillaume Morelle) Bonjour, vous avez demandé à me voir ?  -&gt; Lui présenter une chaise...</pre>
---	--

Vous pouvez ensuite choisir d'interroger le suspect, de l'innocenter ou de l'arrêter.

```
Que voulez vous faire?

(Choix 1) L'interroger.
(Choix 2) L'innocenter.
(Choix 3) L'arrêter.

(Choix 4) Retourner à l'enquête.

Votre choix : |
```

**Interroger un suspect**

L'interrogatoire se déroule en quatre étapes :

- Vous choisissez le genre d'informations (alibi ou témoignage) que vous désirez obtenir.
- Vous lancez vos dés d'intelligence puis de manipulation.
- Le suspect lance ses dés de son côté selon ses propres statistiques.
- Selon le résultat global, vous aurez ou non la chance d'obtenir l'information désirée.

Le résultat d'un lancer se répercute sur les suivants et les jets critiques (résultat inférieur à 15 ou supérieur à 85) ont des conséquences extrêmes : rien n'est joué jusqu'au dernier jet.

Vous voyez les résultats de vos lancers au format :

« *catégorie : valeur\_du\_lancer / palier\_de\_réussite – résultat* »

```
(Enquêteur) Dites moi...

(Choix 1) Que faisiez-vous pendant le crime?
(Choix 2) Avez-vous vu ou entendu quelque chose de suspect?
(Choix 3) Vous pouvez partir.

Votre choix : 2

-> Vous lancez vos dés...

Jet d'intelligence : 92 / 69    FAILURE
Jet de manipulation : 3 / 23    CRITIC_SUCCESS

-> Guillaume Morelle lance ses dés...

(Enquêteur) Le suspect refuse de parler.

-> Vous prenez des notes...
```

## Innocenter un suspect

Lorsque vous pensez qu'un suspect est innocent, vous pouvez le considérer comme tel. Mais, cela ne vous empêchera pas de l'arrêter par la suite.

```
(Enquêteur) Je veux voir

(Choix 1) Pierre Dael - potentiellement innocent
(Choix 2) Luc Pasquier
(Choix 3) Alice Dang - potentiellement innocent
(Choix 4) Luc Dang
(Choix 5) Marine Boore
(Choix 6) Marine Juzeau - potentiellement innocent
(Choix 7) Laure Juzeau
(Choix 8) Guillaume Morelle - potentiellement innocent

(Choix 9) Annuler

Votre choix : 1
```

```
(Pierre Dael) Bonjour, vous avez demandé à me voir ?

-> Lui présenter une chaise...

Que voulez vous faire?

(Choix 1) L'interroger.
(Choix 2) L'arrêter.

(Choix 3) Retourner à l'enquête.

Votre choix : |
```

## Arrêter un suspect

Lorsque vous pensez avoir trouvé le coupable, vous pourrez l'arrêter. Mais, faites attention, ce choix est définitif : une fois qu'un suspect est arrêté, le jeu se terminera et vous ne pourrez pas revenir en arrière. Si vous avez arrêté la mauvaise personne, le jeu est perdu. Dans tous les cas, la sauvegarde est supprimée.

```
Que voulez vous faire?

(Choix 1) L'interroger.
(Choix 2) L'innocenter.
(Choix 3) L'arrêter.

(Choix 4) Retourner à l'enquête.

Votre choix : 3

Vous avez envoyé un innocent en prison.

-> Les crimes ont donc continués...

-> J'ai été renvoyé par mon supérieur pour incompétence...

-> Me revoilà donc sur le marché du travail... Toujours à la recherche d'une enquête à résoudre...

-> Merci d'avoir joué à JavaSpector ! ...
```

Ce jeu comporte trois fins possibles : allez-vous toutes les trouver ?

## 1.4 CONCEPTION ARCHITECTURALE

La conception architecturale comprend toutes les classes et leurs relations les unes avec les autres. Pour rendre les choses plus visibles, chaque classe est présentée séparément avec ses attributs et ses méthodes. Les setters et les getters ne sont pas inclus.

Afin de faciliter la modularité et la visibilité du programme, j'ai créé différents packages regroupant les classes par thème ou par famille. Les regroupements par package ne seront pas représentés dans le diagramme de classes.

### 1.4.1 Brève description des classes

#### ❖ LE JEU

Game
<ul style="list-style-type: none"> <li>#static <code>UI m_console</code></li> <li>#static <code>FileManager m_files</code></li> <li>#static <code>DataManager m_dataSave</code></li> <li>#static <code>Difficulty m_levelChoice</code></li> <li>#static <code>Investigation m_currentInvestigation</code></li> <li>#static <code>boolean m_endedGame</code></li> </ul>
<ul style="list-style-type: none"> <li>+ <code>Game()</code></li> <li>+ static <code>UI getConsole()</code></li> <li>+ static <code>Difficulty getLevelChoice()</code></li> <li>+ static <code>boolean isEndedGame()</code></li> <li>+ static <code>void setEndedGame()</code></li> <li>+ static <code>String[] convertArrayList(ArrayList&lt;String&gt; choices)</code></li> <li>+ <code>void gameMenu()</code></li> <li>+ <code>void newInvestigation()</code></li> </ul>

La classe *Game* permet d’afficher les règles du jeu et de gérer, par fichier, le système de parties : initialiser, charger, sauvegarder, abandonner une partie. Les niveaux de difficultés sont modélisés par l’énumération *Difficulty*.

La classe *UserInterface* est utilisée pour formaliser l’affichage textuel en console et gérer les entrées du joueur.

UserInterface
<ul style="list-style-type: none"> <li>#static <code>int m_nbChoices</code></li> <li>#final <code>int M_CONSOLE_SIZE</code></li> </ul>
<ul style="list-style-type: none"> <li>+ <code>UserInterface()</code></li> <li>+ <code>String concatList(String[] choicesList)</code></li> <li>+ <code>UserInterface clean()</code></li> <li>+ <code>UserInterface display(String text)</code></li> <li>+ <code>UserInterface display(String text, String[] choices)</code></li> <li>+ <code>UserInterface display(String speaker, String text)</code></li> <li>+ <code>UserInterface display(String speaker, String text, String[] choices)</code></li> <li>+ <code>UserInterface execContinue(String text)</code></li> <li>+ <code>int execChoice()</code></li> <li>+ <code>String execInput()</code></li> </ul>

La classe *FileManager* se charge de toutes les opérations liées aux fichiers.

La classe *DataManager* sert à initialiser, charger ou sauvegarder l’ensemble des données du jeu.

#### ❖ LES PERSONNAGES

Les classes *Victim*, *Investigator*, *Murderer*, *CrimePartner* et *Innocent* représentent les différents personnages présents dans le jeu, dont le sexe est défini par l’énumération éponyme. Ainsi, elles sont toutes héritées de la classe abstraite *Character*.

Notons que, hormis la classe *Victim*, toutes ces classes héritent également de la classe abstraite *LiveCharacter* qui contient toutes les méthodes et attributs nécessaires aux lancers de dés - les résultats de ces lancers étant définis par l’énumération *DiceResult*.

Les classes *Murderer*, *CrimePartner* et *Innocent* héritent ensuite de la classe abstraite *Suspect* contenant les méthodes associées à la présentation et à l’interrogatoire des suspects.

#### ❖ L’ENQUETE

La classe *Investigation* est utilisée pour la gestion des menus de l’enquête.

La classe *InvestElement* symbolise les différents éléments d’enquêtes (arme et scène de crime).

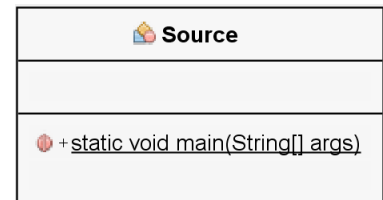
Les indices de l’enquête sont hérités de la classe abstraite *Clue* et peuvent être de deux types, ce sont :

- Soit des preuves trouvées sur les éléments de l’enquête ou sur la victime (classe *Proof*)
- Soit des dépositions fournies par les suspects (classe *Deposition*). Notons que les types de dépositions (*Alibi*, *Heard*, *Seen*, *Role*) sont définis par l’énumération *DepositionType*.



### ❖ LA CLASSE CONDUCTRICE DU PROGRAMME

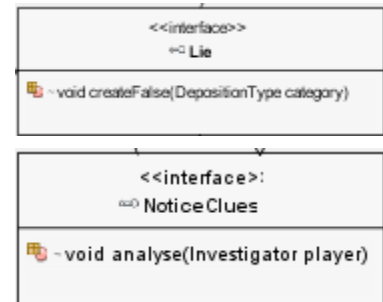
La classe Source est la classe conductrice du programme : elle contient le point d'entrée du programme pour lancer le jeu.



### 1.4.2 Utilisation des interfaces

Pour ce projet, j'ai utilisé deux interfaces :

- L'interface *Lie*, implémentée par les classes *Murderer* et *CrimePartner*, leur permettant de pouvoir fournir de fausses dépositions lors des interrogatoires.
- L'interface *NoticeClue*, implémentée par les classes *Victim* et *InvestElement*, qui permet d'analyser les instances de ces classes pour en extraire les preuves associées.



## 1.5 DIAGRAMME DE CLASSES

J'ai réalisé le diagramme de classes avec le plugging easyUML téléchargeable directement sur NetBeans. Le diagramme de classes est disponible en annexe.

## 2. ANALYSE PERSONNELLE

### 2.1 CHOIX FAITS EN TERMES D'ORGANISATION

Avant de programmer, j'ai commencé à réfléchir à un sujet original, cohérent et dans lequel je pouvais mettre à profit toutes les notions que j'ai pu acquérir lors du cours de Java. Une fois le concept trouvé, je me suis intéressée aux classes et aux liens entre les différentes classes. Suite à cela, j'ai établi le diagramme de classe global du projet dans lequel j'ai y placé les différentes méthodes et attributs des classes. Bien que ce travail de préparation m'ait pris environ deux semaines, j'ai pu avoir une vue d'ensemble sur le projet.

Une fois mon projet validé, j'ai ensuite réfléchi en détail aux fonctionnements de chacune des méthodes afin de pouvoir les programmer rapidement et efficacement. Cela m'a pris environ une semaine, toutefois, il m'a permis d'économiser du temps pour la suite.

Concernant la programmation, j'ai souhaité commencer par coder puis débbugger les fonctions terminales avant de remonter aux fonctions plus générales de façon à gagner du temps.

### 2.2 BILAN PERSONNEL

#### 2.2.1 Difficultés rencontrées

Je ne peux pas nier le fait que travailler seule sur ce projet m'a ajouté une quantité de travail non négligeable. En ayant d'autres projets en parallèle, j'ai eu ainsi des difficultés à trouver du temps à consacrer à ce projet.

Lors des séances de cours consacrées au projet de Java, j'ai également parfois eu l'impression de ne pas être très productive et de perdre du temps sur des choses que je considérais simples – ce qui pouvait être pour moi source de stress et de frustration.

Lorsque j’ai commencé la phase de débogage des fonctions terminales, j’ai parfois connu des baisses de motivation et j’ai souvent cru que je n’arriverais pas à terme ce projet.

Pour la gestion des indices, j’ai dû repenser le système d’accès aux preuves et aux dépositions lorsque j’ai commencé à coder les méthodes plus générales des classes *Investigator* et *Investigation*.

D’un point de vue technique, la lecture en fichier s’est avérée plus compliquée que prévue. En effet, il faut gérer un nombre dynamique de lignes, constitué d’un duo clé : valeur que je stocke dans un `ArrayList`, à la lecture du fichier. Ensuite, la liste est convertie en tableau de `String` et passée à une fonction chargée d’instancier le jeu. Je me suis rendu compte vers la fin qu’il aurait été possible de faire directement un tableau associatif pour récupérer les données de mon tableau, mais, à défaut, j’ai combiné une boucle `while` pour parcourir tout le tableau et un `switch` pour traiter la valeur selon sa clé. Pour un souci de performances, j’ai ensuite regroupé les lignes du fichier selon des catégories (GLOBAL, SUSPECTS, VICTIM, WEAPON, SCENE et INVESTIGATOR) et dédié le premier niveau de `while` – `switch` à la sélection du bon « bloc » d’initialisation. Chaque bloc est alors chargé de remplir les variables nécessaires à l’instanciation des classes concernées puis d’appeler le constructeur dédié au chargement.

### 2.2.2 Points de satisfaction

Pour moi, mon plus grand point de satisfaction a été de voir que, malgré les difficultés rencontrées, j’ai été capable, seule, de mener à terme un projet de développement informatique d’une envergure non négligeable.

J’ai choisi de travailler seule sur ce projet afin de me challenger et de pouvoir être évaluée à ma juste valeur. Pour moi, cela m’a apporté une totale liberté de développement, une connaissance et une maîtrise de l’ensemble du code. De plus, ce projet m’a permis d’approfondir mes connaissances en gestion de projet, d’identifier mes faiblesses en termes d’organisation de projet et de programmation et de consolider mes compétences en Java.

## CONCLUSION

---

D’une façon générale, ce projet présente l’ensemble des notions présentées en cours de JAVA : écriture et lecture en fichier, utilisation des classes, de l’hérité, des interfaces, des exceptions, etc. Ainsi, ce projet m’a apporté une expérience non négligeable dans le domaine du développement.

En guise d’améliorations du projet, il serait intéressant de :

- Permettre au meurtrier de tuer d’autres innocents et de s’enfuir – ce qui implique une gestion du temps
- Modifier les méthodes de la classe *UserInterface* pour avoir un affichage fenêtre
- Pouvoir interroger les suspects dans différents lieux qui auraient une influence sur l’interrogatoire
- Approfondir le système de création d’alibi et de témoignages, en utilisant les noms et caractéristiques des suspects
- Approfondir les indices des éléments de l’enquête pour qu’ils aident davantage à trouver le meurtrier
- Approfondir les interrogatoires. Par exemple, possibilité de brutaliser le suspect, essayer de le mettre en confiance, le menacer, etc. - ce qui ferait évoluer la réputation de l’inspecteur et donc la confiance qu’il inspire aux suspects
- Utiliser un tableau associatif pour la partie instanciation de la lecture en fichier.

## ANNEXE : DIAGRAMME DE CLASSES

