



Simon BAY  
Maëlys POIRE

Département INFORMATIQUE  
1<sup>ère</sup> année – groupe B6  
Année universitaire 2017 – 2018

## Rapport final

« Arcanor »

Projet effectué au sein de  
L'IUT de Vannes



21 mai 2018

### Commanditaire

Sébastien Lefèvre  
*IUT de Vannes, 8 rue Montaigne 56000 Vannes*

### Enseignant tuteur

Sébastien Lefèvre

## Table des matières

I. Rappel du contexte .....	3
II. Organisation du projet .....	4
I. Gestion du planning d'actions .....	4
II. Différence entre le planning prévisionnel et réel .....	4
III. Utilité de la planification .....	5
IV. Nos recommandations .....	5
III. Diagrammes de séquences.....	6
I. Lancement d'une partie .....	6
II. Déroulement d'un tour .....	6
III. Sauvegarde d'une partie .....	7
IV. Chargement d'une partie .....	7
IV. Diagrammes de classes .....	8
I. Le motif MVC (Modèle-Vue-Contrôleur) .....	8
II. Notre conception prévu de l'application.....	8
III. Notre conception finale de l'application.....	10
IV. Les tests .....	11
V. Nos choix techniques.....	11
VI. Etat d'avancement du projet.....	11
V. Bilan du projet.....	12
I. Synthèse des difficultés rencontrées et solutions apportées .....	12
II. Bilan personnel du travail réalisé .....	12
1) Maëlys .....	12
2) Simon.....	13

## I. Rappel du contexte

M. Lefèvre aimerait pouvoir rejouer au jeu *Arcanor*.

Il s'agit d'un jeu en bois qui requiert de la mémoire et de la stratégie. *Arcanor* se joue à deux joueurs.

Le but étant d'avoir d'envoyer ses pièces de l'autre côté du damier pour marquer des points. Chaque pièce à des points différents, le but étant d'avoir un total de 12 points pour remporter la partie. Il est possible de capturer des pièces en les emboitant comme des poupées russes.



Le plateau de jeu d'Arcanor

Ce jeu a obtenu l'As d'Or au Festival International des Jeux de Cannes en 1995. La demande de M. Lefèvre est la suivante : avoir la possibilité de jouer à *Arcanor* sur ordinateur, tablette ou smartphone.

Le jeu doit remplir les conditions suivantes :

- Pouvoir être joué en version console et graphique
- Avoir été testé et soit utilisable lors de son rendu
- Offrir la possibilité de jouer à 2 joueurs sur un ordinateur
- Offrir la possibilité de jouer contre l'ordinateur
- Offrir la possibilité d'enregistrer une partie en cours pour pouvoir la reprendre plus tard
- Être rendu accompagné d'un manuel d'utilisateur
- Être présenté lors d'une soutenance afin d'expliquer le fonctionnement de notre logiciel

## II. Organisation du projet

### I. Gestion du planning d'actions

Lors de la première période à quatre, nous nous sommes répartis le travail grâce à l'outil Trello. Cela nous permettait de savoir les tâches restantes et qui s'occupaient des tâches en cours. Ainsi, nous avons pu respecter l'échéancier prévu tel que le rendu du cahier des charges et du cahier d'analyse et de conception. De plus, l'outil git permettait de suivre l'avancée du projet.

Lors de la seconde période à deux, nous avons d'abord commencé par utiliser l'outil Trello. Cependant, nous nous sommes rapidement rendu compte que cet outil n'est pas forcément pertinent lorsqu'il n'y a pas une liste précise de choses à faire.

Ainsi nous avons rapidement décidés de coder tous les deux côte à côte. Nous nous donnions rendez-vous tous les matins. Ainsi, nous nous sommes repartis le travail au fur et à mesure, selon les envies et compétences de chacun. Nous n'avons pas respecté un calendrier précis.

### II. Différence entre le planning prévisionnel et réel

Nous n'avons planifié que très rapidement notre planning de codage. Nous voulions découper notre travail en trois parties.

Au départ nous pensions suivre ce schéma :

- Avant les partiels : codage de tout le modèle et tests du package
- Après les partiels pendant une semaine : codage de tous ce qui permettra de jouer sur la console
- Après les partiels pendant la dernière semaine : gestion de l'interface graphique
- Après les partiels pendant les derniers jours restants : ajout de fonctionnalités (jeu en réseau, intelligence artificielle, mode triche etc).

Finalement nous n'avons pas pu respecter totalement ce planning.

En effet, avant les partiels, nous avons bien codé le modèle qui fonctionnait à la lumière des différents tests. Cela nous avait déjà pris beaucoup de temps de refaire notre diagramme de classe qui s'est révélé obsolète lorsque nous avons commencé à coder

Cependant lorsque nous avons commencé à vouloir jouer en console, beaucoup d'exceptions ont commencé à survenir (par exemple le déplacement d'un pion déjà mangé) et cela nous a pris plus de temps que prévu de régler ces problèmes.

Nous avons donc pris du retard sur le codage de l'interface graphique. En conséquence, nous ne pourrons pas coder des fonctionnalités supplémentaires par manque de temps, et notre interface ne sera pas aussi aboutie que nous le souhaitions.

### III. Utilité de la planification

Notre évaluation de l'utilité de planifier un projet est pour nous de 6/10.

En effet, il est utile de planifier son projet, car cela permet de se donner une idée du temps qu'il faudrait pour réaliser telle ou telle tâche. Un planning permet de savoir en temps réel si nous sommes en avance ou en retard par rapport à ce que nous avons prévu et permet de se fixer des objectifs par jour.

Cependant, les plannings créés sont souvent peu ou pas viables. Avec notre expérience, nous trouvons qu'il est peu intéressant de faire un planning alors que nous n'avions jamais fait de projet de ce type en informatique. Un planning de quelqu'un qui connaît mieux les délais dont ont besoin les développeurs est très utile car facile à suivre.

Cependant, notre planning était beaucoup trop serré par rapport à ce que nous avons dû faire, car nous n'avions pas conscience des différents problèmes que nous devions régler.

En entreprise cependant, nous pensons que la planification est plus importante et plus intéressante notamment pour que le client puisse suivre l'avancée du projet. Cela permet aussi aux personnes qui ne sont pas directement dans le projet de suivre que tout se passe bien.

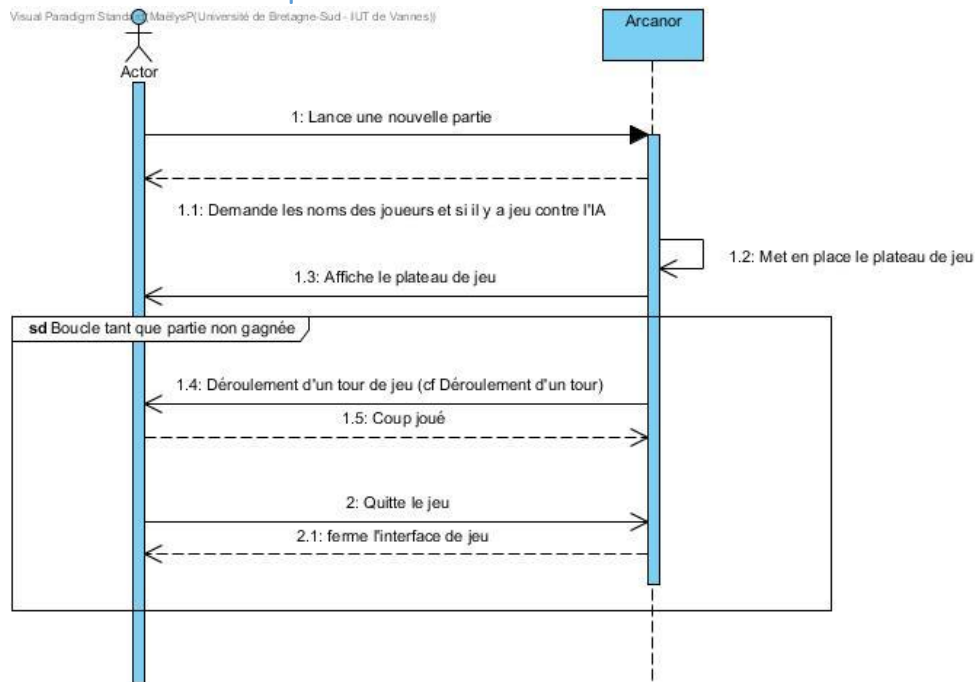
### IV. Nos recommandations

Nos recommandations seraient les suivantes :

- Bien réfléchir à la conception, et la vérifier plusieurs fois en commençant à écrire du code avant le début même du projet. Ce qui semblait marcher sur le papier de notre côté était totalement erroné lorsque nous avons commencé à coder. Cela nous a pris moins d'une semaine de codage pour le voir.
- Travailler avec quelqu'un qu'on peut voir ou contacter très régulièrement, car coder à distance est une difficulté supplémentaire. De plus coder à deux a un côté plus motivant que d'être seul devant son ordinateur.
- S'il s'agit d'un jeu : coder rapidement un moyen de le tester en tant que joueur (pas avec junit) pour voir rapidement toutes les exceptions qu'il est facile d'oublier lorsque l'on fait du test. Il ne faut pas perdre de vue le côté concret du programme.

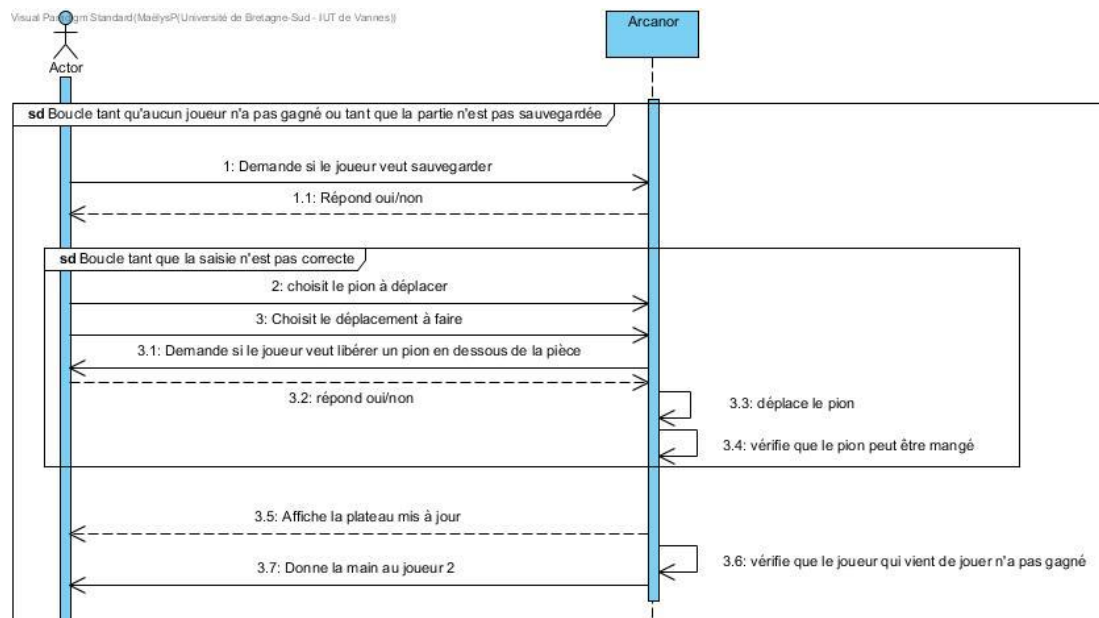
### III. Diagrammes de séquences

#### I. Lancement d'une partie



Il n'y a plus les choix de niveaux de l'intelligence artificielle disponible, puisque nous n'avons pas réussi à la coder.  
Le reste est identique à ce que nous avons prévu lors de la conception.

#### II. Déroulement d'un tour

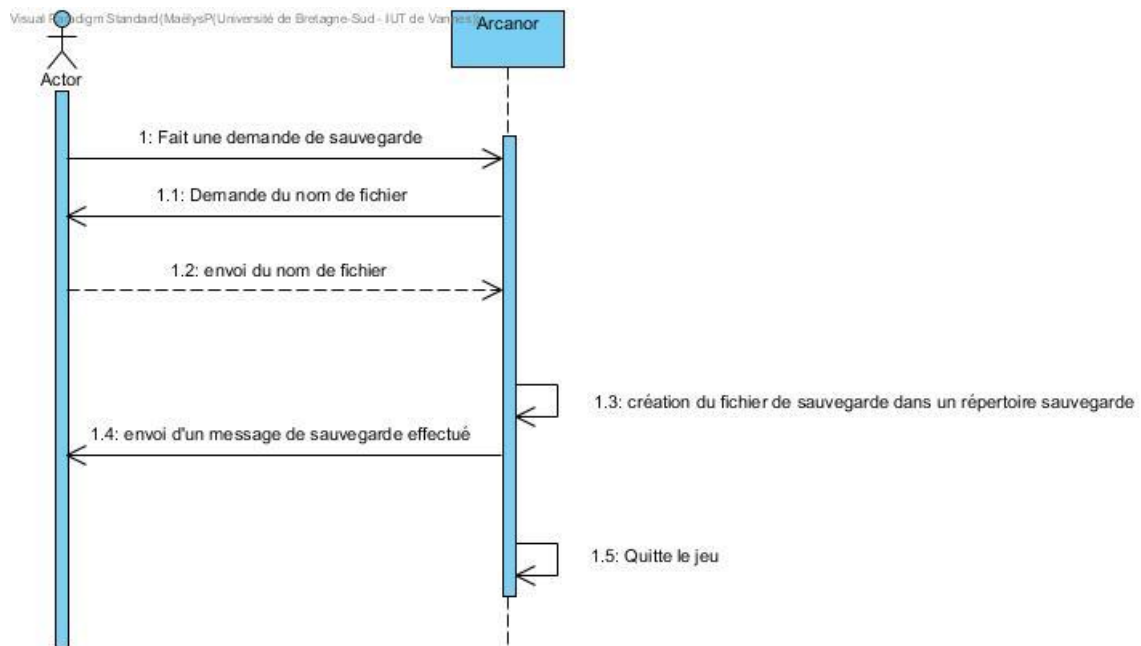


Nous avons retiré le mode facile qui permet de voir quels pions ont des pions en dessous d'eux.

En mode console, le joueur ne peut pas valider sa saisie, cependant cela boucle tant que la saisie n'est pas correcte.

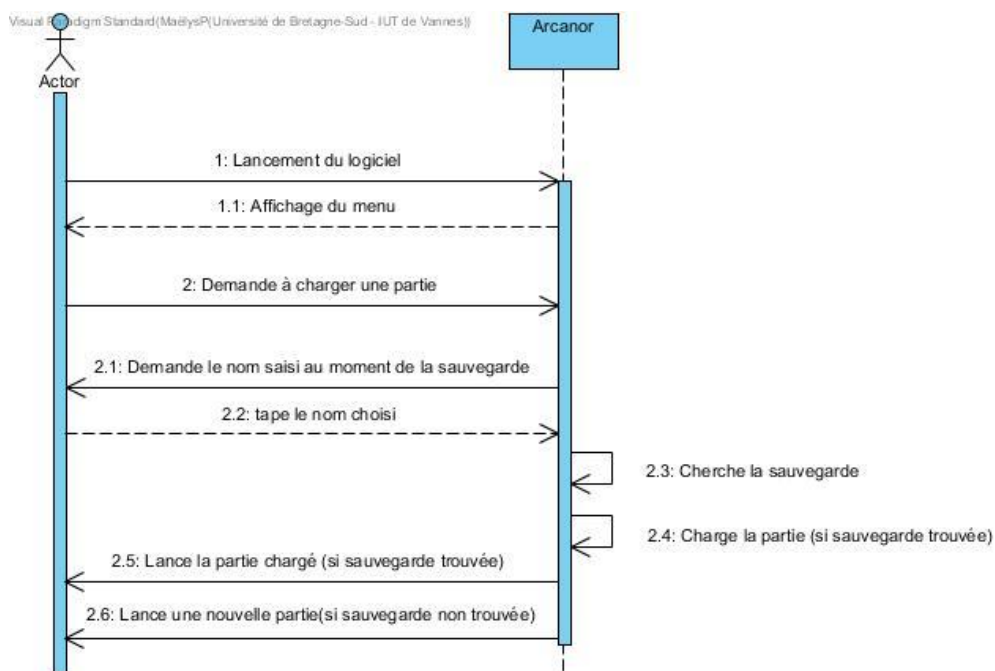
En mode graphique, le joueur doit valider ses choix.

### III. Sauvegarde d'une partie



Nous n'avons pas changé la manière de sauvegarder par rapport à la conception initiale.

### IV. Chargement d'une partie



Le chargement correspond aussi à ce que nous avons pensé lorsque nous avons conçu le logiciel.

## IV. Diagrammes de classes

### I. Le motif MVC (Modèle-Vue-Contrôleur)

L'objectif de ce motif est de séparer l'application en trois parties distinctes :

- Une partie qui permet une gestion de l'information (Modèle)
- Une partie qui permet la visualisation des données (Vue)
- Une partie qui permet d'articuler la gestion de l'information et la visualisation des données (Contrôleur)

Dans une application informatique, ce motif permet de séparer le fonctionnement d'une application en 3 parties distinctes (presque indépendantes). Nous avons suivi ce modèle pour coder notre application

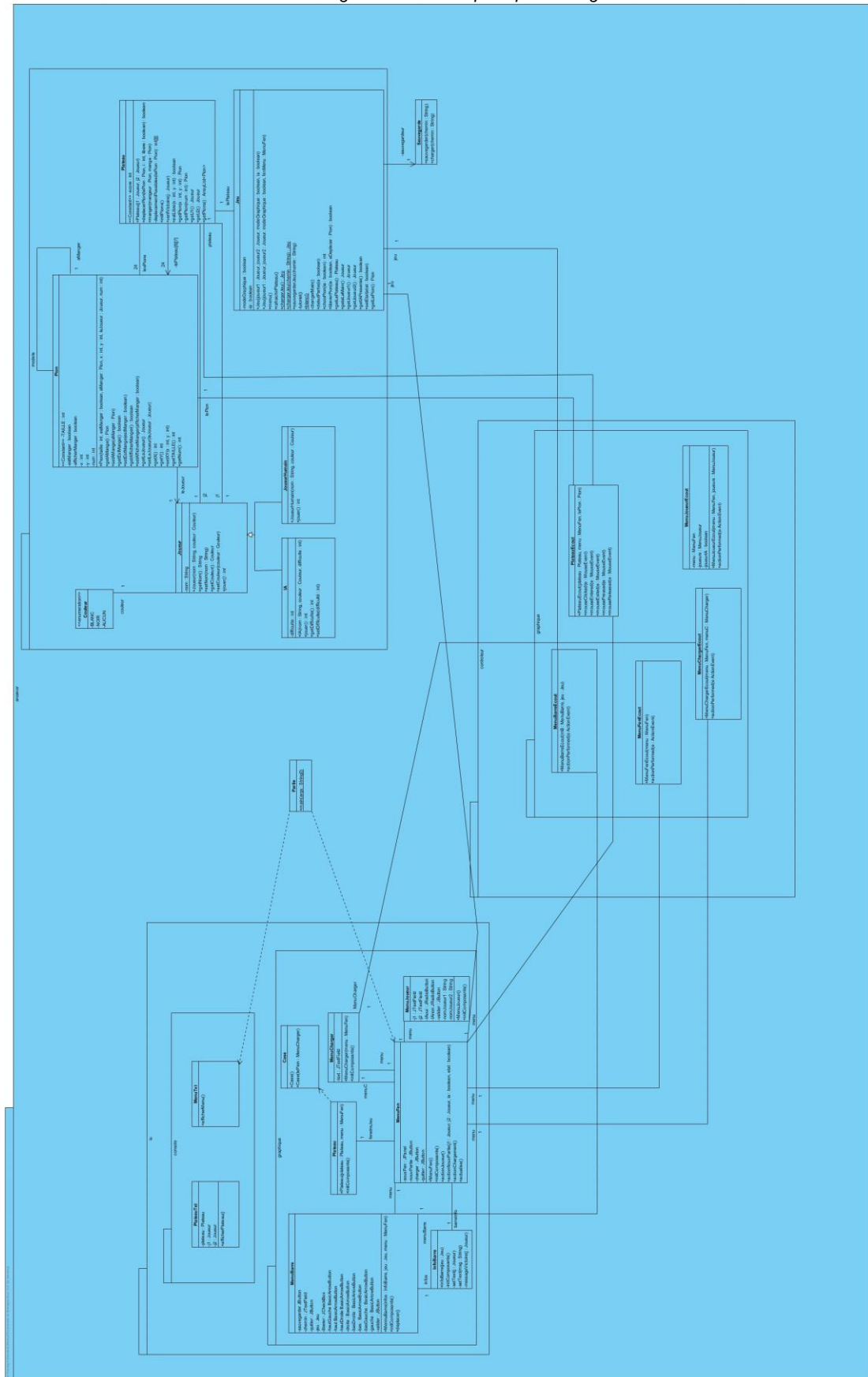
### II. Notre conception prévu de l'application





### III. Notre conception finale de l'application

### Notre diagramme de conception post-codage



## IV. Les tests

[Home](#)

Packages

test

Classes

[TestIA](#)  
[TestJoueurHuman](#)  
[TestPhon](#)  
[TestPlateau](#)  
[TestSauvegarde](#)

Unit Test Results.

Designed for use with [JUnit](#) and [Ant](#).

Summary

Tests	Failures	Errors	Skipped	Success rate	Time
12	0	0	0	100.00%	0.970

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Packages

Name	Tests	Errors	Failures	Skipped	Time(s)	Time Stamp	Host
<a href="#">test</a>	12	0	0	0	0.970	2018-06-26T14:35:16	PC-baysi

Nous avons effectué douze tests sur notre application. Nous avons testé le package modele uniquement car il s'agit du cœur de notre logiciel.

## V. Nos choix techniques

La gestion des écouteurs (scanner) sur la console ne se trouve pas dans le package controleur, car cela compliquait les choses et retirait de la lisibilité. Les scanners sont directement intégrés dans le code du package modele. Ils sont activés lorsque l'application a son booléen mode graphique sur faux.

Les écouteurs de l'interface graphique ne sont plus groupés par interaction (MouseListener, ActionListener etc.) mais par panel. Cela simplifiait la gestion d'évènements et la lisibilité du code.

La partie serveur n'a pas été faite par manque de temps et de connaissances. Les niveaux de difficulté de l'intelligence artificielle ont été supprimés puisque nous n'avons pas été en capacité de les coder.

Nous avons décidé de faire une seule fenêtre dans la partie graphique (package iu) avec des panels qui viennent la compléter. Cela nous a semblé plus propre lors de la conception.

## VI. Etat d'avancement du projet

Notre application est totalement fonctionnelle que ce soit en mode console ou en mode graphique. Cependant aucun ajout optionnel n'a été effectué par manque de temps. Notre logiciel respecte les règles de jeux et permet d'effectuer une partie.

Le cahier des charges de base est respecté.

## V. Bilan du projet

### I. Synthèse des difficultés rencontrées et solutions apportées

Difficulté	Explication	Solution
Déplacement des pions en mode console	Les déplacements choisis ne correspondaient pas à ce qu'on obtenait sur l'affichage console. Le déplacement par coordonnées semblait ne pas convenir à notre modèle.	Nous avons mis plus de temps mais nous avons finalement réussi à ce que cela fonctionne. Nous donnons un schéma des déplacements disponibles et l'utilisateur choisit un chiffre.
Construction d'une intelligence artificielle	Nous souhaitions créer une intelligence artificielle capable de réfléchir au meilleur déplacement possible.	Nous n'avons pas réussi à faire déterminer à l'intelligence artificielle de trouver le meilleur déplacement à faire parmi les déplacements possibles. Nous avons donc abandonné l'idée de faire une vraie intelligence artificielle.
Déplacement des pions en mode graphique	Nous avons rencontrés énormément de difficultés à coder un déplacement des pions en les prenant et en les relâchant sur le plateau à l'endroit souhaité.	<p>Nous avons d'abord passé beaucoup de temps en essayant par tous les moyens. Lorsque cela a commencé à fonctionner, les déplacements ne correspondaient pas du tout à l'endroit où l'on cliquait.</p> <p>Nous avons donc pris le parti d'effectuer les choix de déplacements sur l'interface. L'utilisateur clique sur le pion qu'il souhaite déplacer, puis dans une barre de jeu situé à droite, il effectue tous ses choix de jeux tel que la libération de pions, la sauvegarde etc.</p>

### II. Bilan personnel du travail réalisé

#### 1) Maëlys

Au cours de ce projet, j'ai été amenée à toucher à toutes les facettes de l'application. En effet, j'ai pu coder une partie de l'interface graphique, une partie de l'affichage console et une partie du modèle du logiciel. Cela m'a permis de retravailler beaucoup d'aspects techniques de l'année, comme la gestion des sauvegardes par exemple. Ce projet permet donc de faire un bilan des connaissances acquises.

Il permet aussi de gérer des applications plus grosses que celles faites jusqu'ici. Ce qui nous a forcés à adopter des méthodes de travail plus rigoureuses comme le test de chaque fonctionnalité dès qu'elle a été codée.

Ce travail a été très formateur sur la gestion d'un projet, et sur comment s'organiser en groupe pour répondre aux différents jalons fixés au fur et à mesure.

Enfin, cela a permis de donner un côté concret de nos connaissances et permet de s'approcher d'une situation d'entreprise. Il est intéressant de savoir que nous avons maintenant l'expérience de codage en « situation réelle ».

## 2) Simon

Ce projet m'a permis de m'essayer à différents aspects de la programmation. Grâce à notre bonne répartition des tâches, j'ai pu toucher à tout dans le projet. J'ai ainsi pu mettre en application tout ce que nous avons vu au cours de l'année.

Le projet m'a aussi permis de m'améliorer dans tout ce qui touche aux interfaces homme-machines. De plus, j'ai pu découvrir des technologies tout à fait nouvelles pour moi comme ANT et JUnit.

Enfin, je pense que ce projet aura été très formateur par rapport à ce qu'on nous demandera en entreprise car nous avons beaucoup d'autonomie et n'étions que très peu guidé dès le départ.