

Chu Thomas – S2B1
Guillou Aurélien – S2B1
Giraud Thomas – S2B1
Hernandez Lilian – S2B1
Nguyen Simon – S2C1
Ba Sidi – S2A1

IUT de Belfort
Département Informatique
Groupe 10

Développer un jeu de plateau : Kamisado

RAPPORT FINAL



Sommaire

1 Introduction.....	3
2 Présentation du jeu.....	3
2.1 Les règles du jeu.....	4
2.1.1 Mise en place.....	4
2.1.2 But du jeu.....	4
2.1.3 Déroulement d'une partie.....	4
3 Implémentation et challenges.....	5
3.1 Contraintes.....	5
3.2 Nos choix.....	5
3.3 Challenges.....	8
4 Organisation du travail.....	9
5 Bilan.....	13
5.1 Réalisation technique.....	13
5.2 Connaissances & compétences.....	14
6 Conclusion.....	16
7 Annexes.....	17
7.1 Étude de cas juridique.....	17
7.1.1 Réponse.....	17
7.1.2 Sources.....	18
7.2 Figures.....	18
7.3 Auto-évaluation.....	18
7.3.1 Thomas Giraud.....	18
7.3.2 Lilian Hernandez.....	19
7.3.3 Simon Nguyen.....	19
7.3.4 Sidi Ba.....	19
7.3.5 Aurélien Guillou.....	20
7.3.6 Thomas Chu.....	20

1 Introduction

Dans le cadre d'un projet informatique, il est souvent demandé de travailler en équipe. C'est d'autant plus vrai pour la programmation d'un jeu nécessitant de nombreuses compétences et connaissances. Ce cahier des charges consiste à définir et présenter un projet de développement informatique.

Dans le cadre de notre première année en BUT Informatique, il nous a été demandé de réaliser un jeu de plateau en 8 semaines nous permettant de mettre en pratique nos connaissances et compétences acquises durant l'année.

Les principaux objectifs de ce projet étaient :

- De nous familiariser à travailler en groupe.
- D'appliquer les notions de gestion de projet.
- De respecter les contraintes de programmation imposées.
- De programmer au minimum deux IA sachant jouer au jeu.

Dans le cadre de notre projet, nous avons choisi de programmer un Kamisado, les contraintes imposées pour la programmation étaient de coder en Java, utiliser la bibliothèque graphique Java FX, utiliser le Framework *gamifier*, respecter le paradigme d'architecture MVC (Modèle - Vue - Contrôleur), fournir des batteries de tests assurant la fonctionnalité des méthodes, utiliser le gestionnaire de version GitLab et l'éditeur de texte IntelliJ IDEA.

Durant le projet il nous a été demandé de fournir un rapport technique présentant le fonctionnement de nos IA, un second rapport décrivant l'application mais également une étude droit-économie basée sur la problématique de publier un jeu et enfin le code source final du jeu afin de le présenter à travers une soutenance.

Notre groupe de travail était composé de Thomas CHU, le chef de projet, Aurélien GUILLOU, Lilian HERNANDEZ, Simon NGUYEN, Sidi BA et Thomas GIRAUD. Nous avons choisi de programmer un Kamisado car nous voulions nous challenger pour monter en compétences.

2 Présentation du jeu

Kamisado est un jeu de stratégie à deux joueurs sur un plateau de 8 cases sur 8. Chaque joueur dispose de huit pions noirs ou blancs, portant également la couleur de la case sur laquelle il se trouve au début de la partie. Le but étant de gagner le plus de points. On marque des points en étant le premier à placer un pion de sa couleur sur l'une des cases de la rangée de la base adverse.

2.1 Les règles du jeu

2.1.1 Mise en place

Chaque joueur place ses 8 pions sur les 2 lignes aux extrémités du plateau, une case ne peut comporter qu'un seul pion.

2.1.2 But du jeu

Le jeu se joue en rounds. À chaque tour, l'objectif est d'atteindre une case de départ de l'adversaire avec un de vos pions. Le pion atteignant une case de départ adverse devient « Sumo » et rapporte un point à son joueur. Dans les tours suivants, le "Sumo" peut devenir un "Double" ou un "Triple Sumo" valant 3 et 7 points. Un 'Triple Sumo' qui est à nouveau promu marquera 15 points.

Le jeu est proposé dans les variantes suivantes :

- Le premier joueur qui parvient à amener un de ses pions sur une case de départ adverse remporte la partie.
- Match standard : le premier joueur à 3 points gagne la partie.
- Match long : le premier joueur à 7 points gagne la partie.
- Marathon Match : le premier joueur à 15 points gagne la partie.

2.1.3 Déroulement d'une partie

- Le joueur ayant les pions noirs commence, il peut déplacer n'importe lequel de ses pions
- Ensuite, le joueur ayant les pions blancs doit jouer le pion de la couleur de la case sur laquelle le pion noir a été joué le tour précédent.
- Une pièce peut être déplacée d'autant de cases que le souhaite le joueur en verticale ou diagonale. Cependant, les pions en jeu et les limites du plateau bloquent le déplacement d'un pion.
- On ne peut pas passer par-dessus un autre pion.
- La partie s'arrête lorsqu'un pion a atteint une case de départ adverse.

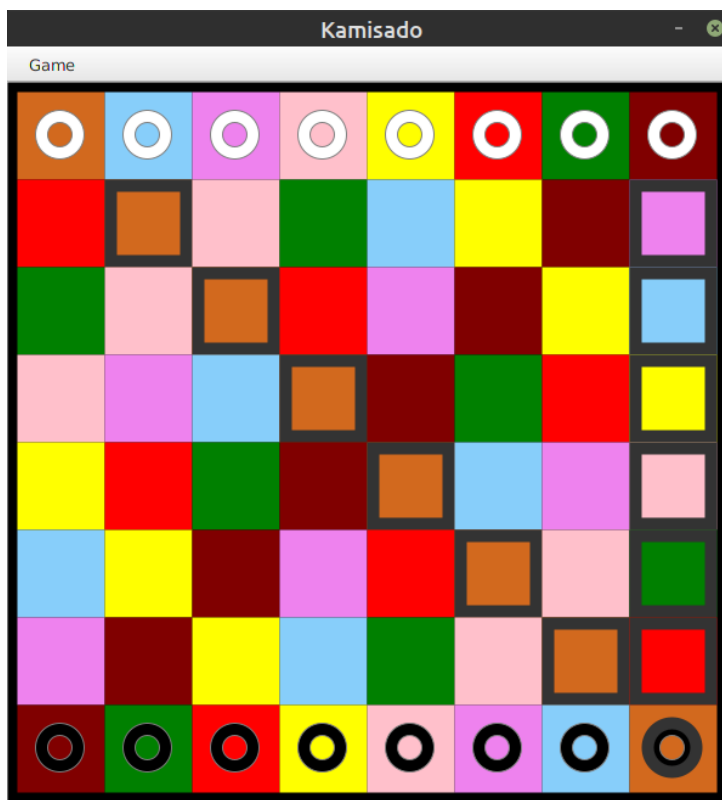
3 Implémentation et challenges

3.1 Contraintes

Pour réaliser notre jeu, nous avons utilisé la structure du paradigme MVC. Dans le cadre du développement d'une application graphique, le principe du paradigme MVC est de séparer les données, leur manipulation et leur visualisation dans des classes différentes. Nous avons utilisé le langage Java pour coder mais également la bibliothèque JavaFx pour créer une interface graphique. De plus, on nous a imposé d'utiliser framework *gamifier* qui concerne également des fonctions et/ou des applications qui n'étaient pas destinées au jeu initialement pour rendre le jeu plus accessible et plus faciles à utiliser. Enfin, nous devons utiliser le gestionnaire de version *gitlab* pour travailler en équipe et gérer le projet.

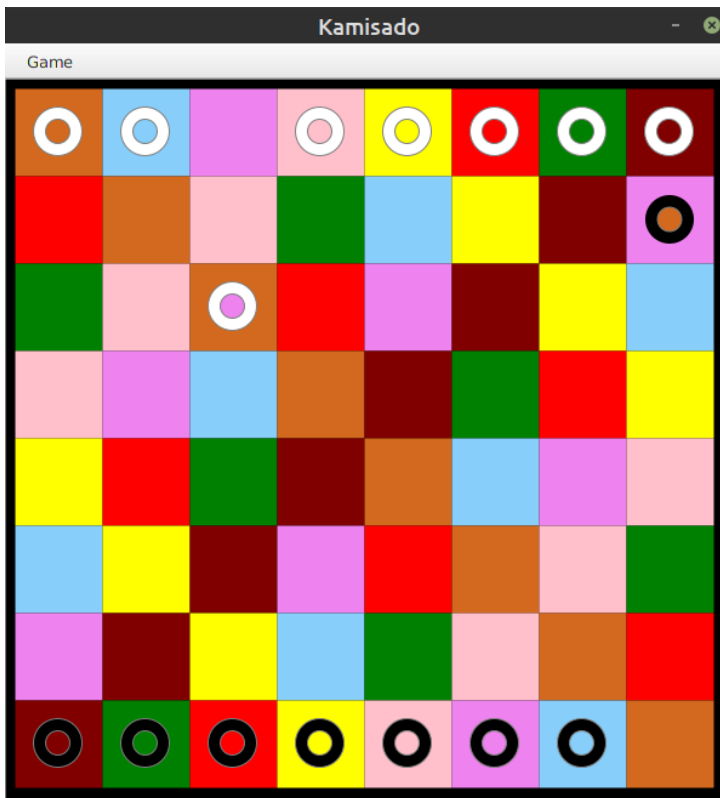
3.2 Nos choix

Le joueur doit pouvoir sélectionner n'importe quel pion au début de la partie pour le déplacement sur une case verticale ou diagonale qui recouvre tout le plateau à l'aide de la liste des cases atteignables nommée *reachable*.



Les cases avec un contour noir sont les cases de reachable que le pion sélectionné peut atteindre pour se déplacer.

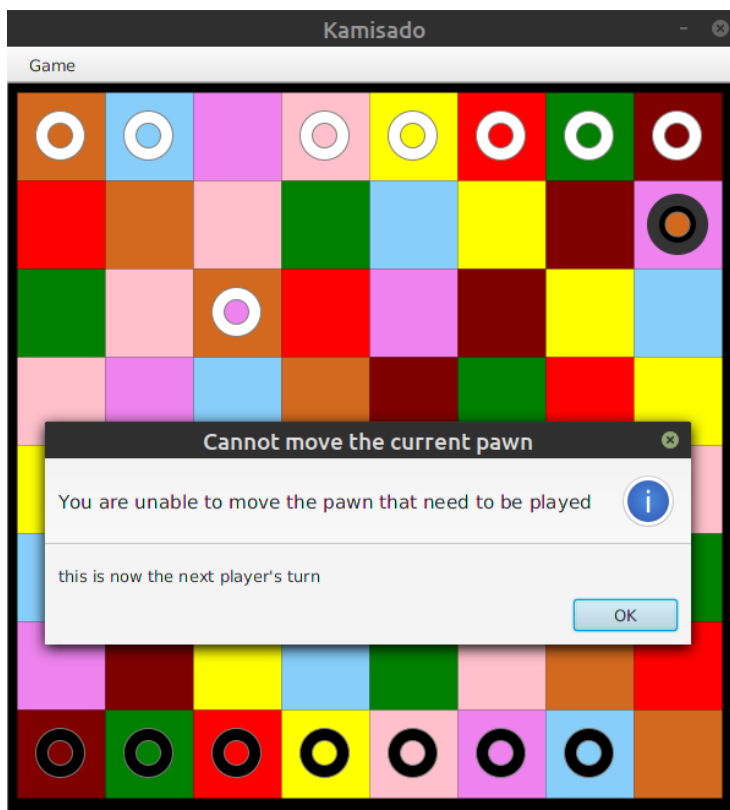
Nous allons par exemple choisir la case rose située en haut à droite à la position (7-1).



Une fois que le joueur déplace le pion, par exemple sur la case rose. L'IA doit obligatoirement jouer à son tour son pion rose.

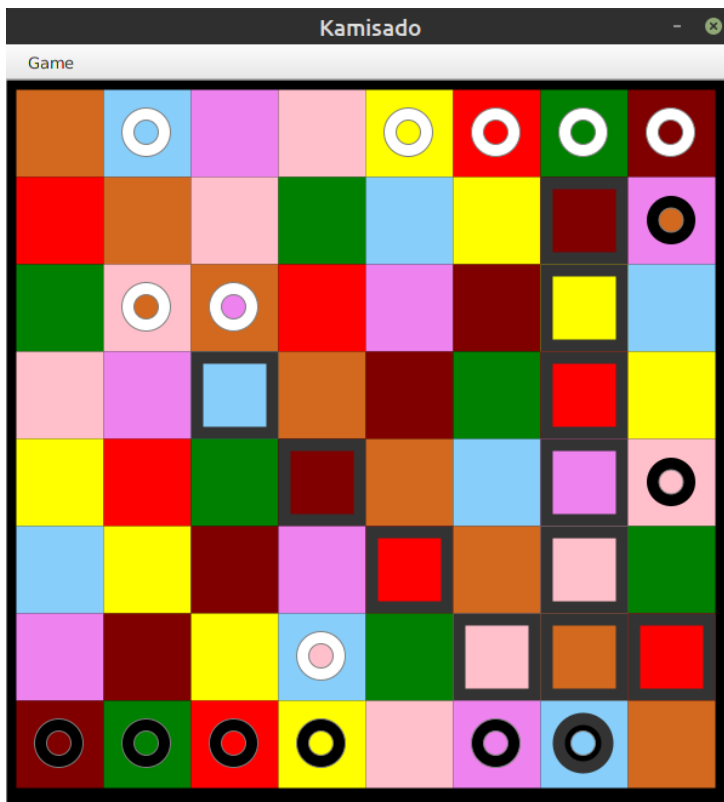
Notre plateau de jeu du Kamisado est basé sur un tableau 2D avec des valeurs que nous avons attribuées pour chaque case.

Nous avons fait une méthode dans le *stageModel* qui permet de sauvegarder la valeur de la dernière case jouée, afin de réutiliser cette valeur lors de la sélection des pions.



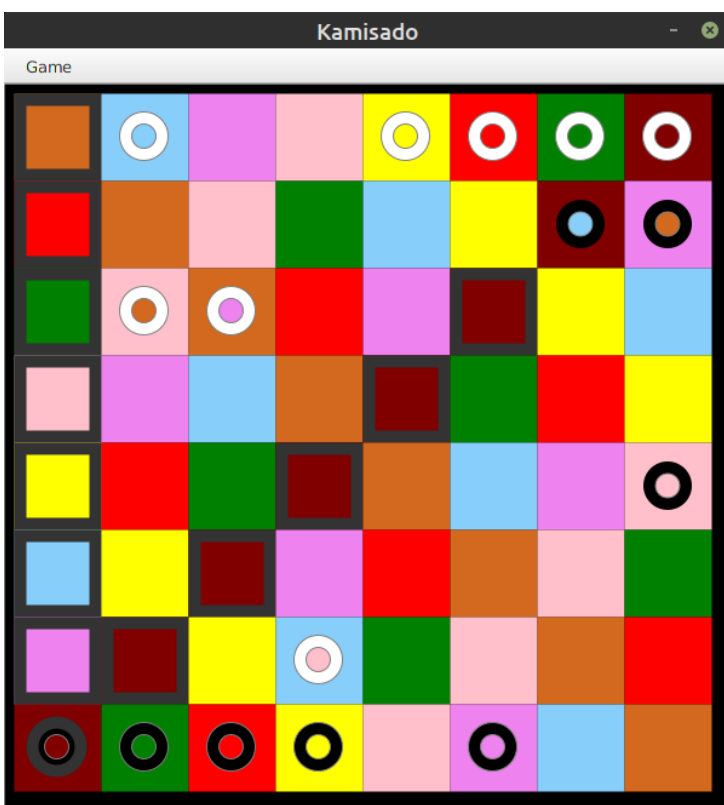
Si la liste *reachable* est vide, le joueur ne peut donc plus bouger.

Ainsi, un message d'alerte apparaît pour prévenir le joueur qu'il ne peut pas bouger son pion et lui fait passer son tour.



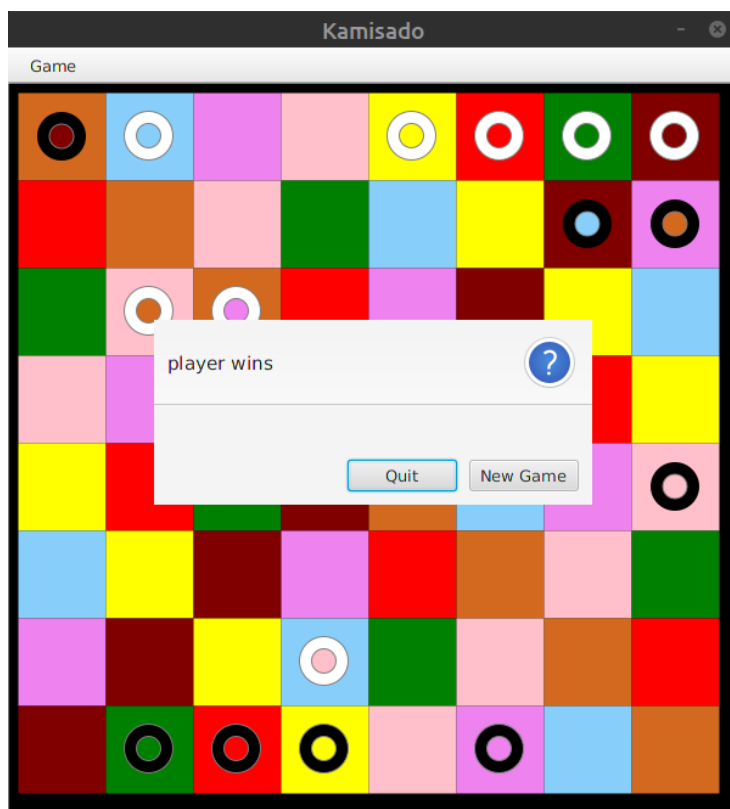
Dans cette situation, c'est au tour du joueur de déplacer son pion bleu sur une des cases du *reachable*.

Pour des raisons de stratégie, nous allons déplacer ce pion sur une case marron pour bloquer le pion marron de l'adversaire, par exemple à la position (6-1).



L'IA ne peut pas jouer son pion marron puisqu'il sa liste *reachable* est vide, son tour est alors passé sans changer la valeur de la dernière case jouée.

C'est donc au joueur au pion marron de jouer. Il peut atteindre le camp de son adversaire à l'aide du *reachable* pour mettre un terme à la partie.



Une fois le pion ayant atteint le camp adverse, la partie s'arrête. Il est possible de quitter la partie ou alors continuer une autre partie avec les sumos.

Les sumos sont une amélioration des pions de base. Le pion qui devient sumo est celui qui a permis de gagner la partie précédente.

3.3 Challenges

Tout d'abord, nous avons dû en apprendre davantage sur le framework gamifier pour coder notre jeu. Le framework gamifier nous a permis de modifier les animations des pions ou encore d'ajouter une image de notre jeu du Kamisado.

Ensuite, la recherche de stratégie gagnante était une partie très délicate. Dans notre groupe, aucun membre ne connaissait le jeu auparavant. Nous avons dû réfléchir et trouver une stratégie en jouant directement sur notre jeu.

Puis, la partie la plus difficile était d'implémenter la stratégie dans notre jeu. Par exemple, nous avons passé énormément de temps sur l'implémentation de l'IA Minimax qui calcule de nombreuses possibilités de coups. Le plus dur lors de l'implémentation de l'IA fut de réussir à adapter l'algorithme au code que nous avons déjà créé. En effet, l'algorithme Minimax étant basé sur une comparaison de valeur associée à des cases, il fallait trouver un moyen d'associer à chaque case du plateau une valeur sans pour autant modifier les différentes représentations du plateau déjà existantes car cela pourrait avoir de graves conséquences. Une autre difficulté apparente lors du codage de l'IA était de savoir comment calculer la valeur à associer à une case, car le calcul d'une valeur heuristique était quelque chose de nouveau pour notre groupe.

Par ailleurs, le développement des sumos était un bonus dans notre jeu du Kamisado. Ainsi, nous avons consacré du temps afin d'ajouter une nouvelle fonctionnalité sur le jeu.

4 Organisation du travail

Vous pourrez trouver ci-dessous, le diagramme de Gantt prévisionnel, le second diagramme pour le déroulement réel, et enfin, six diagrammes circulaires décrivant les pourcentages personnels d'implication pour les grandes parties du projet.

Dans l'ensemble, le projet s'est bien déroulé dans le sens où tous les livrables ont été envoyés à temps, cependant, comme vous le pouvez le remarquer, nous avons dû grandement modifier le planning prévu initialement. Ce projet était plutôt difficile et une première pour tous les membres du groupe, nous avons donc passer une large quantité de temps sur les tutoriels mis à notre disposition par nos enseignants. Le début du développement était plutôt brumeux, nous avons décidé de commencer tôt le développement même si certains n'étaient pas encore accoutumés au framework *gamifier* et au paradigme MVC, en plus d'avoir des difficultés dans la matière, laissant Aurélien Guillou, Thomas Chu et Simon Nguyen sur la plupart du développement. Cette décision a été prise, car nous étions inquiets du temps que nous prendrait l'implémentation du *model*, cette équipe de trois n'étant pas d'excellents développeurs non plus.

L'IA a ensuite été un véritable challenge pour nous et pour les mêmes raisons, la même équipe a été assignée à cette tâche. Nous avons de nombreuses incompréhensions par rapport au framework imposé, nous obligeant à agrandir la taille de la tâche « Decider class ». Lors de cette dernière, nous avons eu de grosse difficultés à tester le comportement des IA, notre solution a été de coder la partie *vue* et la partie *controller* nous permettant ainsi de voir les comportements malgré l'ordre de développement étudié dans le tutoriel de conception d'un jeu de plateau.

Cette décision a ralenti les tests unitaires, ces derniers étant très compliqués, car nos IA étaient complexes et probablement mal optimisées. Nous avons donc poussé la tâche « tests unitaires » jusqu'à la fin du premier jalon, en même temps que l'écriture du rapport des IA.

Ayant réalisé la partie *controller* et *vue* en même temps que le *model*, nous avons pu avancer confortablement pendant le jalon 2 avec l'écriture du rapport final. Tout le monde a pu contribuer à ce livrable, ce dernier ne demandant pas de compétences trop exigeantes.

Globalement, la répartition du travail aurait pu être plus également distribuée, cependant, à cause de la nouveauté, d'un projet très ambitieux en terme de compétences et de gros problèmes de communication pour certains membres, ce résultat n'a pas pu être évité.

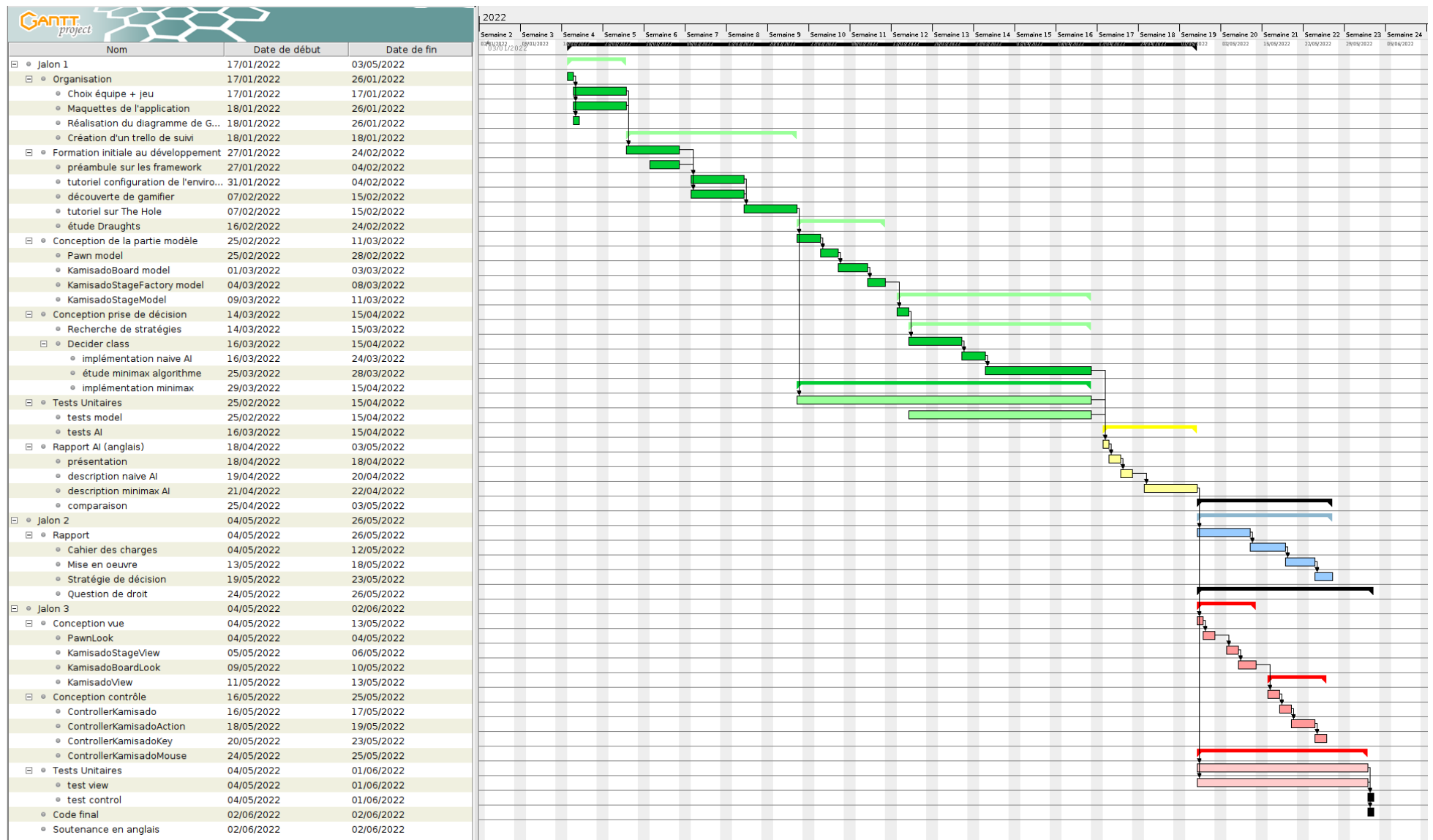


Figure 1: Diagramme de Gantt du déroulement prévu

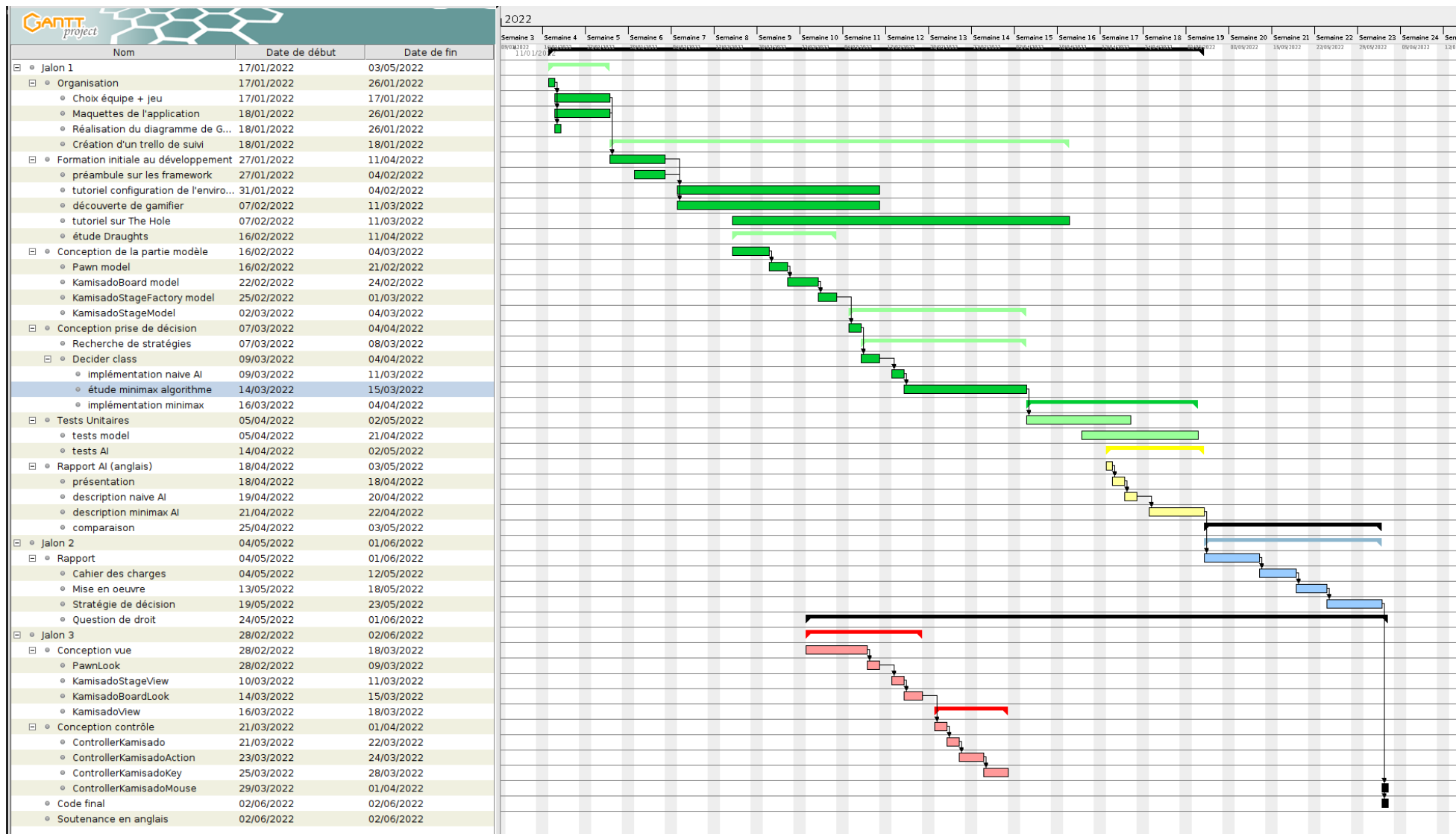
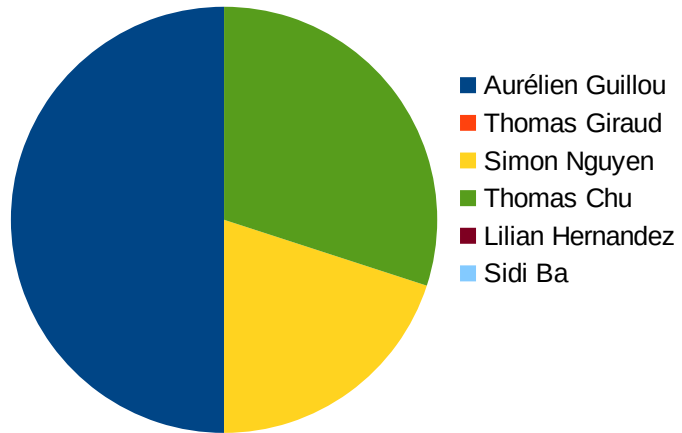
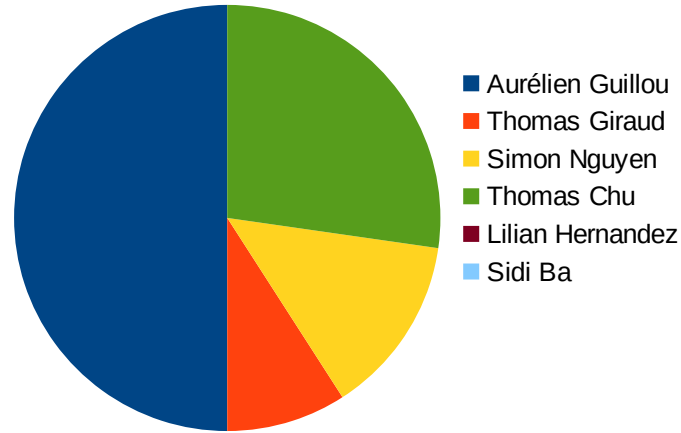


Figure 2: Diagramme de Gantt du déroulement réel

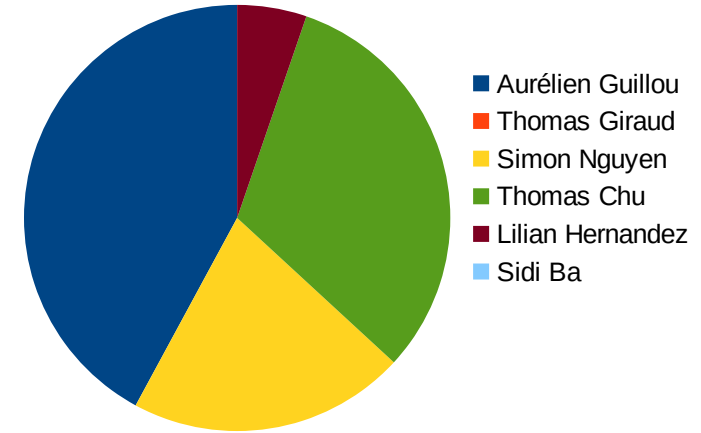
Conception du model (J1)



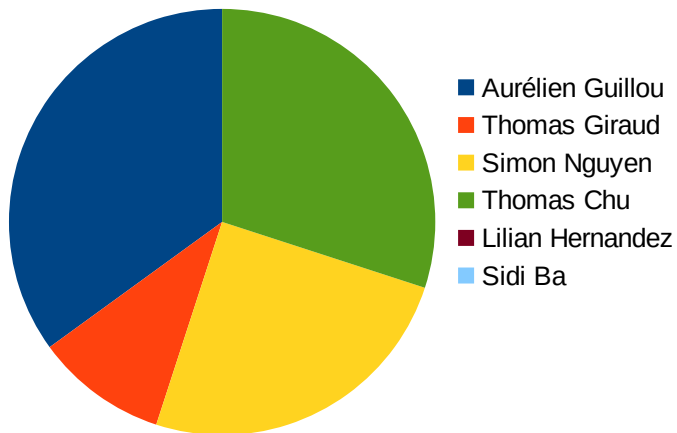
Conception de la vue (J1)



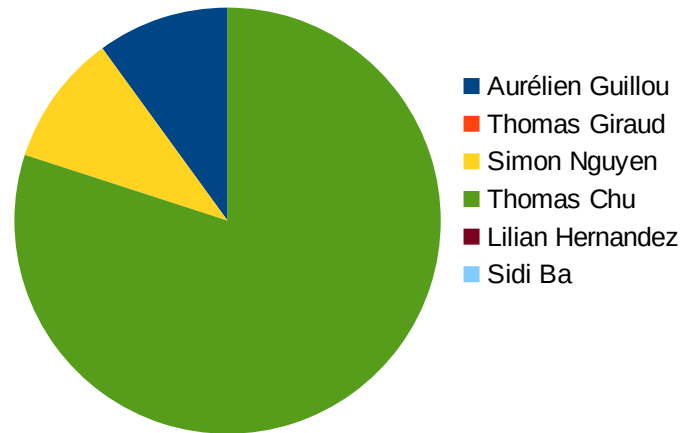
Conception du control (IA) (J1)



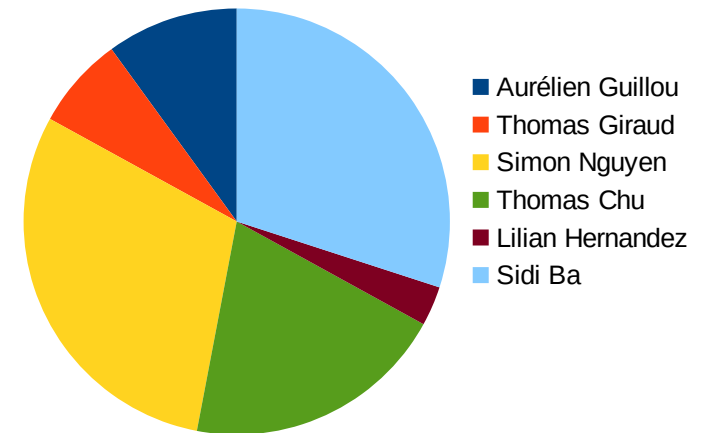
Tests Unitaires (J1)



Rapport IA (J2)



Rapport final (J3)



5 Bilan

5.1 Réalisation technique

Nous avons une interface graphique basique et la plupart des règles du jeu fonctionnelles. C'est-à-dire, les conditions de blocage, l'ordre de sélection des pions ainsi que la condition de victoire... Néanmoins, dans notre jeu du Kamisado, il nous manque la condition « d'impasse »¹ en cas d'égalité. De plus, le joueur doit cliquer sur son pion actif lorsqu'il est bloqué pour faire apparaître le message d'alerte. Enfin, nous ne pouvons pas avoir les doubles ou triples sumos ainsi qu'une interface pour visualiser les points obtenus.

Lors de la réalisation technique de l'IA Minimax, l'une des principales difficultés que nous avons rencontré fut de réussir à appliquer l'algorithme à notre jeu, il fallait comprendre ce qu'était une valeur heuristique et comment la calculer, ce qui n'a pas été simple. L'implémentation de l'algorithme en lui-même n'était pas non plus simple, car même si l'idée n'est pas très complexe, il faut réussir à l'adapter au langage utilisé ainsi qu'au code qui avait déjà été créé.

Ensuite, nous avons également eu un bug sur l'animation des déplacements verticaux. Les pions se téléportaient directement sur la case sélectionnée. Pour y remédier, nous avons passé beaucoup de temps sur la compréhension du framework gamifier pour corriger le problème d'animation. De plus, le gamifier nous a permis d'ajouter une image de notre jeu.

Enfin, les tests unitaires nous ont posé problème. Au moment du premier livrable, il nous manquait quelques méthodes dans le modèle à tester. La réalisation des IA a eu un grand impact sur notre avancé dans notre projet du Kamisado notamment l'IA Minimax qui utilise un algorithme mathématique étant difficile à tester à cause de sa complexité.

¹ En théorie, il peut arriver que les deux joueurs soient complètement bloqués. Dans ce cas, le perdant est le dernier joueur dont le précédent coup a provoqué cette impasse totale

5.2 Connaissances & compétences

Compétence 1 : réalisation un développent application

Dans cette compétence on a développé une application de type jeu de plateau en utilisant le langage Java et la bibliothèque graphique Java Fx en respectant le paradigme d'architecture MVC (Modèle - Vue – Contrôleur). Nous avons fait des tests unitaires qui permettent de vérifier le bon fonctionnement de la partie modèle du paradigme à l'aide des librairies *Mockito* et *JavaFx* . Nous avons implémenter deux IA : une utilisant l'algorithme minimax et une autre dite naïve dont le fonctionnement est précisé dans le rapport les concernant.

Compétence 2 : optimiser des applications informatiques

Cette compétence a été acquise à travers la création des deux IA demandé pour le jeu. Ces dernières nécessitaient d'être optimisées afin de pouvoir respecter le critère imposé qui était que le temps d'exécution, les IA ne devant pas dépasser 1 minute. Nous avons également essayer d'implémenter une IA qui a pour but d'être efficace et qui demande à l'ordinateur un pouvoir de calcul intéressant à comparer.

Compétence 5 : conduire un Projet

Cette compétence nous a permis dans un premier temps, de définir le contexte, les enjeux, les objectifs ainsi que les livrables et les axes de développement. En organisant nos idées, nous avons ainsi pu vérifier la concordance et la faisabilité de notre projet.

Conduire un projet à 6 nous a permis apprendre plus sur la gestion de projet. Nous avons rédiger et suivi un cahier des charges accompagné d'un diagramme de *Gantt* pour partager les tâches. Il nous a permis d'avoir une vue globale sur les tâches à réaliser, les responsabilités et les ressources associées, de l'idée jusqu'à la mise en service en passant par l'analyse des exigences, la conception fonctionnelle, la conception et les tests.

Compétence 6 : travail dans une équipe informatique

Nous avons mis en place une organisation pour ne pas compromettre le travail des membres du groupe, pour cela, nous avons utilisé Trello, un logiciel en ligne gratuit. Tout d'abord, il permet une excellente visualisation des tâches et de leur avancement, notamment grâce à la création de catégories ou la mise en place de dates limites. Par ailleurs, Trello est un outil collaboratif, ce qui signifie que plusieurs personnes peuvent éditer et participer en temps réel au sein du projet. Cela est idéal pour gérer l'avancement de du projet.

Il se présente de la manière suivante :

- Des tableaux
- Des listes
- Des cartes

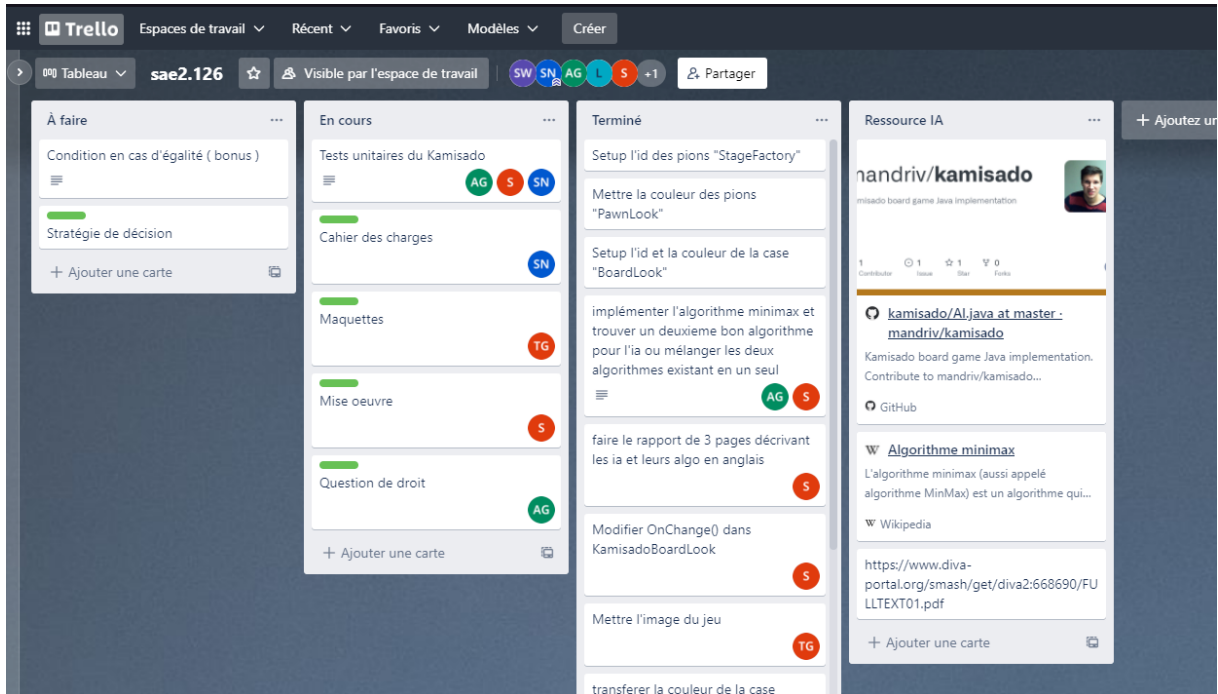


Figure 3: Capture d'écran du Trello post-rapport IA

Nous avons aussi utiliser *git*, un système de gestion de versions qui permet de stocker de manière optimisée et sécurisée des fichiers et toutes leurs modifications dans le temps.

git est décentralisé, et permet à de multiples personnes de travailler ensemble sur le même projet, puis d'intégrer les travaux de chacun.

Gitlab est une plateforme permettant d'héberger et de gérer des projets web de A à Z. Présentée comme la plateforme des développeurs modernes, elle offre la possibilité de gérer ses dépôts *git* et ainsi de mieux appréhender la gestion des versions de nos codes sources.

Tableau de bord interactif du gitlab :

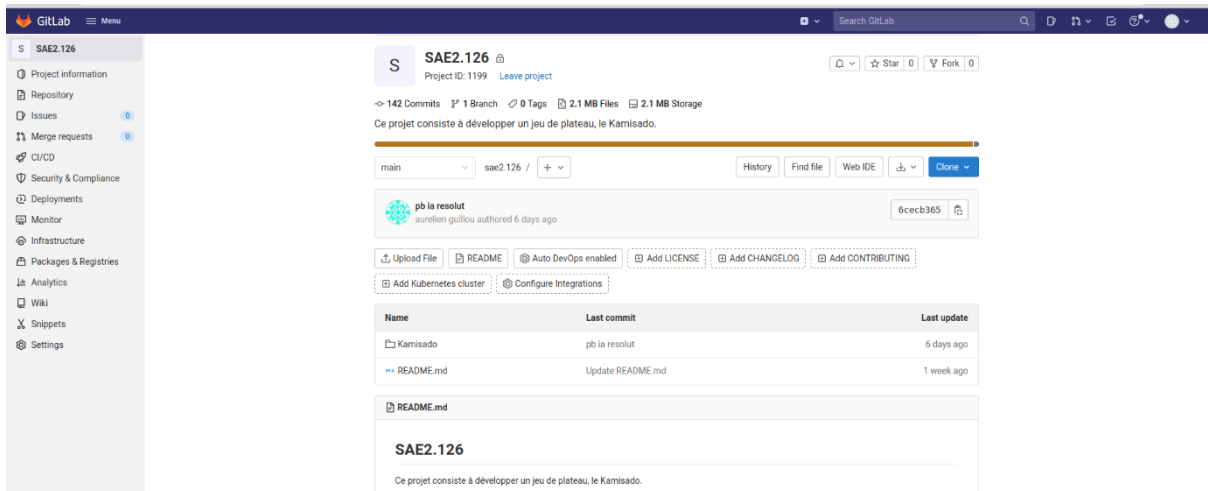


Figure 4: Capture d'écran du gitlab du projet

6 Conclusion

Pour conclure, ce projet nous a permis d'approfondir davantage nos compétences en développement d'application. Nous avons réalisé une application permettant de jouer une partie de Kamisado fonctionnelle à 2 joueurs, ou contre une IA. Par ailleurs, lors de ce projet, nous avons rencontré un nombre non-négligeable de soucis par rapport à la gestion du planning. Un résultat qui s'explique par un projet difficile et une communication faible pour certains membres du projet.

Si le projet devait continuer, nous pourrions corriger les erreurs/bugs déjà présentes énoncées dans le bilan et améliorer le jeu avec une fonctionnalité *drag-and-drop*, un *undo*, une interface plus agréable avec un tableau des scores, des animations, des musiques, mettre le jeu en réseau, etc..

Concernant les remarques sur le projet dans son ensemble, beaucoup d'entre nous sont plutôt frustrés de leur contribution. Le projet était trop ambitieux, l'interface que nous avons obtenu est celle fournie par défaut et notre jeu était pré construit sur la base du *Draught* et les méthodes du *gamifier*. Même aujourd'hui, certaines parties du code sont toujours des zones grises pour nous. De plus, cette grande difficulté a été infranchissable pour certains membres du groupe, rendant l'acquisition des compétences mis en jeu difficile. Sur une note positive, le jeu fonctionne bel et bien, le paradigme MVC est respecté et le cœur du code compris. La difficulté a définitivement permis à certains de s'améliorer en développement d'application et en gestion de projet.

7 Annexes

7.1 Étude de cas juridique

7.1.1 Réponse

Un jeu de plateau sera protégé au titre du droit d'auteur dans le cas où quelque chose d'original sera créé pour le jeu, que cela soit une règle, une forme, un design de plateau, des cartes à jouer, la manière dont est conditionné le jeu, l'univers du jeu voire le nom du jeu qui peut être considéré comme une œuvre de l'esprit et donc être protégé par le droit d'auteur. Cependant, pour être en mesure d'appliquer ce droit, il faudra être capable de prouver que la création est bien originale, notamment via la date de création, en cas de litige tel que le plagiat. Pour ce faire, plusieurs méthodes sont possibles :

- L'utilisation de l'enveloppe Soleau², qui permet de protéger un jeu contre les copies et qui constitue une preuve d'antériorité très fiable devant un tribunal.
- Un dépôt à la SGDL³ qui permet de stocker un manuscrit ayant valeur de preuve d'antériorité avec un volume plus important que l'enveloppe Soleau mais sur une période plus courte.
- L'envoi d'une lettre recommandée à soi-même avec accusé de réception peut également constituer une preuve d'antériorité, mais cela nécessitera de faire attention à ce que l'enveloppe reste bien scellée et ne pourra servir de preuve qu'à travers un constat d'huissier assermenté.
- Présenter le jeu à un huissier assermenté permettra de faire un constat daté qui attestera de la composition du jeu et de ses éléments, permettant de fournir une preuve d'antériorité.
- L'envoi à soi-même est également une possibilité, car la jurisprudence leur a reconnu une certaine légitimité comme élément de preuve d'antériorité.

Il est cependant important de noter que, si le droit d'auteur permet de protéger un jeu, ce n'est pas l'idée elle-même qui est protégée mais son application. En effet, les idées doivent rester dans le domaine public, ce que la cour d'appel de Bruxelles a confirmé en statuant que « le droit d'auteur ne protège pas l'idée en tant que telle, celle-ci appartenant au fond commun de la pensée humaine ».

2 Objet de brevet qui permet de stocker dans une enveloppe sceller 7 à 8 pages format A4 dans une enveloppe et de stocker cette enveloppe pendant 5 ans renouvelables et d'obtenir de manière fiable la date de création d'une œuvre

3 Société des Gens De Lettres qui est une association française de promotion du droit et de défense des intérêts des auteurs

7.1.2 Sources

- « Comment protéger son jeu de société ? – DES DROITS, DES AUTEURS », 2012
- « La protection des jeux de société par la propriété intellectuelle – Le petit juriste », 9 juin 2016
- « Droit d'auteur, propriété intellectuelle et je de société – Ludiconcept »,
- « Droit d'auteur et jeux de société », HEB IESSID

7.2 Figures

Index des figures

Figure 1: Diagramme de Gantt du déroulement prévu.....	10
Figure 2: Diagramme de Gantt du déroulement réel.....	11
Figure 3: Capture d'écran du Trello post-rapport IA.....	15
Figure 4: Capture d'écran du gitlab du projet.....	16

7.3 Auto-évaluation

7.3.1 Thomas Giraud

Durant ce projet j'ai été impliqué dans la création du menu du jeu de l'application (60%) de tests unitaires (10%).

La création d'un Kamisado m'a permis de développer mes connaissances en programmation notamment en Javafx et en Tests Unitaires, l'utilisation d'un framework (Gamifier) et d'un gestionnaire de version (GitLab).

J'ai rencontré beaucoup de difficultés durant la conception du jeu, je pense que notre choix était au delà de mes capacités, l'importante différence de niveau en programmation dans le groupe était un inconvénient car certains membres avaient commencé à programmer alors que j'essayais de me familiariser avec Gamifier, de ce fait, j'ai peu contribué à la programmation du jeu mais j'ai suivi rigoureusement l'avancement du projet.

Pour conclure, je suis satisfait de notre projet car nous avons réussi à programmer un jeu fonctionnel qui répond à toutes les contraintes données. Bien que ma contribution soit faible, j'ai codé quelques méthodes mais je dois m'améliorer pour bien aborder l'année prochaine.

Ce projet m'a permis d'affiner mon choix pour l'année prochaine et m'a fait comprendre que poursuivre dans la programmation n'était pas la bonne solution.

7.3.2 Lilian Hernandez

Durant ce projet, j'ai travaillé sur les règles de déplacement (60 %). Concernant cette contribution, j'ai eu quelques difficultés, car j'ai compris tard l'absence d'animation verticale dans le framework *gamifier*. Je pense avoir reçu de précieuses compétences sur les intelligences artificielles ayant réfléchi au développement de ces dernières.

En conclusion, je pense que le jeu est plutôt fonctionnel au vu de mes compétences.

7.3.3 Simon Nguyen

Durant ce projet, je me suis impliqué dans la correction et dans l'amélioration du code à 30 %, mais également dans la conception de la l'IA dite naïve à 50 %. Lors du projet, je me suis amélioré en java, plus précisément en programmation objet grâce à la découverte du framework Gamifier ou encore sur le paradigme MVC. J'ai su utiliser et comprendre une partie de la bibliothèque JavaFx. De plus, je pense que j'ai contribué à 20 % pour faire les tests de notre jeu du Kamisado. Cependant, je suis frustré de ne pas avoir terminé tous les tests dans le modèle pour le premier livrable. En outre, j'ai rencontré beaucoup de conflits sur *GitLab* qui m'a fait perdre du temps.

C'est la première fois que je suis dans ce groupe d'SAÉ avec des membres qui ne sont pas dans ma classe, c'était même l'une de mes meilleures expériences.

Par ailleurs, je suis fier d'avoir initié le groupe à rejoindre le « Trello » qui nous a permis de mieux nous organiser tout au long de notre projet.

Pendant le déroulement du projet, j'ai trouvé que la conception de l'IA était intéressante. D'ailleurs, je vais peut-être choisir le parcours C « Administration, gestion et exploitation des données » puisqu'il y a des cours sur l'intelligence artificielle.

7.3.4 Sidi Ba

La réalisation du projet a été satisfaisante pour ma part, de même façon au niveau des différents échanges qui ont été nécessaires au niveau du groupe. Ce projet m'a permis d'approfondir mes compétences en langage JAVA+JAVA FX et nous a permis de concrétiser nos compétences, en créant deux IA. Pour ma part, j'ai travaillé plus sur la réalisation du rapport.

Pour conclure sur ce projet, je pense qu'il m'a permis d'acquérir des connaissances intéressantes au niveau du Java et Java fx. D'un point de vue du groupe, je pense que chacun a su s'adapter aux méthodes de travail de chaque membre pour obtenir une meilleure cohésion au niveau de la réalisation du projet. Chacun a effectué le travail demandé. Je pense que ce projet m'a permis un meilleur apprentissage des connaissances de l'Unité d'Enseignement, et m'a permis de m'améliorer en ce qui concerne le travail de groupe et la communication dans un groupe de projet.

Enfin, je pense que le groupe était réparti comme il fallait, tout le monde a réalisé sa partie comme prévu dans les temps donc c'était un bon groupe et comme chaque projet, une bonne expérience.

7.3.5 Aurélien Guillou

Durant ce projet, j'ai été impliqué dans l'ensemble des tâches impliquant du code, que ça soit la conception ou le codage du jeu ou les tests unitaire. Pour ce qui est des autres tâches, j'ai participé au rapport en écrivant différentes parties, mais je n'ai pas participé à la gestion et à la planification du projet. Je pense que dans ce projet, mon implication doit être aux alentours de 50 % du projet. Lors de ce projet, j'ai découvert de nouveaux algorithmes, tels que l'algorithme minimax ou encore l'algorithme de calcul de valeur heuristique dans un jeu de plateau. J'ai également pu améliorer mes compétences de travail avec une équipe lors d'un projet. Les difficultés majeures que j'ai rencontrées viennent notamment de l'implémentation et la création des tests pour l'IA minimax, ainsi que de réussir à travailler correctement dans un groupe. Pour ce qui est du projet en lui-même, je l'ai trouvé très intéressant à réaliser, car il m'a permis de découvrir de nouvelles choses et d'améliorer mes compétences d'une manière que je trouve très sympathique.

7.3.6 Thomas Chu

Durant ce projet, j'ai travaillé sur l'ensemble des livrables et j'étais responsable de la gestion du projet. Je dois dire que la partie du développement était rude, les compétences requises étaient assez hautes, le rythme était dur à prendre, mais j'ai tout de même réussi à participer à 30-40 % de la phase de développement. Le rapport en anglais sur les IA a été écrit par moi-même, avec une contribution des deux membres qui m'ont aidé pour coder la *Decider class* sur l'explication des IA.

La gestion du projet a également été un véritable challenge, diriger une équipe de six personnes est une toute première pour moi, certains ne parlent pas, d'autres codent, mais n'expliquent pas ou mal leur code. L'arborescence du *git* est là, mais cela n'a pas empêché de nombreux conflits à cause de mauvaise communication, surtout au début du projet. Heureusement, la mise en place d'un Trello très tôt dans le projet nous a permis de mieux attribuer les tâches et a permis une bonne cohésion plus tard.

Je considère m'être approché de l'idéal décrit par les principales compétences de développement de la compétence 1 et 2 ainsi que la compétence de gestion de projet.

Pour conclure, ce projet a été éducatif pour moi en tant que chef de projet, au niveau de la gestion du planning et des tâches. Je reste plutôt frustré dans l'ensemble, je n'ai pas réussi à faire participer tout le monde sur tous les livrables et ma progression sur la phase de développement existe, mais n'est pas à la hauteur de mes attentes définies au début du projet. Ceci dit, je ne vois pas comment éviter ces regrets si le projet était à refaire. Pour finir sur une note positive, le jeu est fonctionnel et j'avais une équipe capable dans l'ensemble, ce projet reste une expérience précieuse qui a consolidé ma volonté de prendre le parcours C, le développement étant un de mes traits les plus faibles.