

Rapport Général SAE2.126 | Trifle - Kamisado

S2 B1 Groupe 2



Cédric Colin
Marvyn Levin
Baptiste Dulieux
Timothée Meyer
Hugues Estrade

18/06/2024

Sommaire

1 Introduction	4
2 Présentation du jeu	5
2.1 Déplacements	5
2.2 Sumos	6
2.3 Fin d'une manche/partie	7
3 Implémentations et challenges	8
3.1 Contraintes	8
3.1.1 Paradigme MVC	8
3.1.2 Fonctionnement global du jeu	8
3.2 Choix	9
3.3 Challenges	10
4 Organisation du travail	11
4.1 Planning prévisionnel	11
4.2 Planning réel	13
5 Bilan	15
5.1 Réalisation technique	15
5.1.1 Fonctionnalités présentes	16
5.2 Connaissances et compétences	16
6 Conclusion	18
7 Annexes	20
7.1 Étude de cas juridique	20
7.1.1 Propriété Intellectuelle	20
7.1.2 Licences et Contrats	20
7.1.3 Régulation des Contenus et Protection des Consommateurs	21
7.1.4 Responsabilité Juridique et Protection des Données Personnelles	21
7.2 Auto-évaluation	22
7.2.1 Hugues	23
7.2.2 Baptiste	26
7.2.3 Timothée	28
7.2.4 Marvyn	31
7.2.5 Cédric	34
7.3 Figures	37

1 Introduction

Depuis toujours, les jeux ont fasciné l'humanité, offrant divertissement, réflexion et découverte du monde. Des jeux de société classiques comme Monopoly ou Puissance 4 à l'explosion des jeux vidéo en 2D et 3D, ces divertissements ne cessent d'évoluer et de captiver.

Pour les étudiants en informatique, concevoir un jeu est un défi passionnant et formateur. Cela permet de mettre en pratique leurs connaissances et d'acquérir de nouvelles compétences, tant techniques que transversales.

C'est dans ce contexte que notre équipe a choisi de développer le jeu de société **Kamisado**. Créé en **1970**, ce jeu de stratégie abstrait pour deux joueurs se distingue par sa simplicité de règles et sa profondeur tactique. Son originalité et son accessibilité en font un choix idéal pour notre initiation à la conception de jeux vidéo.

Kamisado (Fig. 1) nous a séduit par son univers de possibilités caché derrière une apparence simple. Maîtriser le jeu demande réflexion et anticipation, un challenge qui nous a motivés à développer des algorithmes d'intelligence artificielle performants. L'aspect visuel épuré du jeu nous a également laissé une grande liberté créative pour l'interface graphique.

Ce rapport présente les différentes étapes de notre projet, depuis l'analyse du jeu original jusqu'à la réalisation de notre propre version numérique. Il s'articule autour de plusieurs chapitres qui décrivent en détail notre démarche.



Fig. 1. – Jeu du Kamisado

2 Présentation du jeu

Le Kamisado est un jeu de plateau très simple qui se présente de cette manière:

Dans la Fig. 2, nous avons un plateau coloré de dimensions 8x8 cases. Chaque pion est une **tour** et est associé à un des deux joueurs, elle est identifiée par sa couleur - noir ou blanc.

Le joueur qui démarre pourra alors déplacer une tour parmi les 8 présentes.

Vous pourrez remarquer que le plateau est construit selon une symétrie centrale - le point situé au centre. Le paterne permet alors d'établir diverses stratégies intéressantes, sur lesquels nous reviendront juste après.



Fig. 2. – Situation de départ

2.1 Déplacements

La règle primordiale est la suivante:

La tour que chaque joueur doit jouer doit être de la couleur de la case sur laquelle l'adversaire a joué. Cette tour pourra alors se déplacer en diagonale et verticalement, mais toujours en direction de son adversaire. Il est impossible de passer par-dessus une autre tour, que ce soit la sienne ou celle de son adversaire.

Pour gagner, il est nécessaire de placer sa tour dans la rangée adverse.

Voyons un exemple:

Dans la Fig. 3, le joueur 1 a commencé et a déplacé sa tour jaune sur une case violette.

Le joueur 2 doit alors déplacer sa tour violette, nous pouvons alors remarquer que la tour violette peut se déplacer sur toute la rangée verticale et en diagonale.

Si une tour **ne peut pas être déplacée**, on dit alors qu'elle est bloquée. Il sera donc considéré que la tour a avancé de 0 cases.

Par exemple, dans la Fig. 18, le joueur 1 a joué la tour **rose en rouge**, et le pion **rouge** adverse est allé en **rose**. Le joueur 2 doit alors **rejouer** son pion rouge, il pourra gagner.



Fig. 3. – Un exemple de plateau en jeu

2.2 Sumos

La règle des dents de dragon est la suivante:

Le joueur qui remporte une manche place un marqueur Dent de dragon sur la tour avec laquelle il a atteint la rangée adverse. Cette tour est désormais un Sumo. Au cours des manches suivantes, un Sumo ne peut plus se déplacer d'autant de cases qu'il veut, mais peut en revanche effectuer un oshi.

Un oshi consiste à repousser une tour par un Sumo. La tour qui est repoussée par un Sumo doit se trouver sur la case directement devant lui, pas en diagonale. La case derrière la tour repoussée doit être libre avant le oshi et la tour repoussée doit appartenir à l'adversaire.

Voyons un exemple:

Dans la Fig. 4 le joueur 1 a commencé et a déplacé sa tour orange sur une case rouge, devant la tour rouge du joueur 2.

Le joueur 2 doit alors déplacer sa tour rouge, nous pouvons alors remarquer que la tour rouge peut se déplacer sur toute la diagonale gauche et peut aussi exécuter un oshi car il possède une dent de dragon (Sumo).

Une tour peut avoir plusieurs dent de dragons et ainsi devenir un Sumo, un Double-Sumo, un Triple-Sumo ou même un Quadruple-Sumo.



Fig. 4. – Coup précédant le oshi

Dans la Fig. 5, le joueur 2 exécute un **Oshi**. Il déplace alors la tour orange de son adversaire sur une case verte. Cela laisse donc le joueur 2 à rejouer en fonction de la couleur de la case sur laquelle la tour rouge de l'adversaire a fini sa route.

Le joueur 2 peut donc se **déplacer à nouveau** avec sa tour de couleur verte sur toute la longueur horizontale ou en diagonale droite.

Un Sumo simple peut pousser une seule tour ennemie, un Double-Sumo 2 tours, un Triple-Sumo 3 tours. Un quadruple-Sumo, lui, ne peut plus bouger du tout. Les Sumos remportent en cas de victoire respectivement 1, 3, 7, 15 points à leur propriétaire. Voici un exemple de coup de Triple-Sumo (Fig. 19 - Fig. 20).



Fig. 5. – Exécution du oshi par un Sumo

2.3 Fin d'une manche/partie

Une manche se termine de deux manières:

- Les deux adversaires sont bloqués et ne peuvent plus jouer. Le joueur qui a provoqué le blocage est alors perdant (Fig. 6).
- Un des deux joueurs atteint le camp adverse, il gagne alors cette manche (Fig. 7).



Fig. 6. – Partie sans impasse



Fig. 7. – Joueur atteint le camp de l'autre joueur

Dans le cas d'une partie **rapide** (1 manche), le joueur qui gagne la manche gagne aussi la partie (**1** point). Au contraire, dans une partie **standard**, le gagnant est le premier joueur qui atteint **3** points. Dans une partie **longue** il faut **7** points à une personne pour gagner et une partie **marathon** se termine lorsque le gagnant cumule **15** points au total.

3 Implémentations et challenges

3.1 Contraintes

3.1.1 Paradigme MVC

La contrainte principale de ce jeu a été de l'implémenter de façon à respecter le paradigme MVC.

Afin d'illustrer ce paradigme, nous avons choisi schéma:

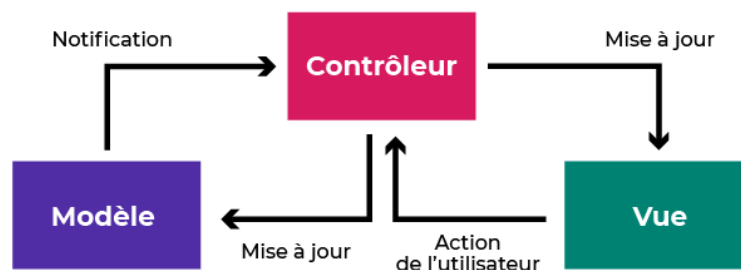


Fig. 8. – Schéma du paradigme MVC

Le modèle MVC consiste à diviser le code en 3 parties: Le contrôleur, le modèle et la vue.

La vue est responsable de l'affichage, c'est elle qui va afficher les textes, le plateau, les pions et en général toute l'interface, tout en récupérant les actions de l'utilisateur (clics, entrée clavier...). Ce sont les yeux du programme.

En parallèle, le modèle peut être vu comme la mémoire, il va stocker les informations utiles au jeu, tel que les joueurs, pions, le plateau et tous ces éléments requis pour faire fonctionner le jeu.

Enfin, le contrôleur est le cerveau du jeu. Il va appliquer la logique du jeu, en lisant les infos du modèle et de la vue, il va alors pouvoir modifier les infos du modèle et mettre à jour la vue.

3.1.2 Fonctionnement global du jeu

Maintenant que nous avons vu comment le paradigme MVC fonctionne, nous pouvons nous intéresser au jeu en lui-même.

La vue du jeu est séparée en différentes parties:

- Le **RootPane**: C'est l'interface affichée au démarrage, utilisée en guise de configuration;
- La **View**: Plus globale, elle va gérer la barre de navigation que vous retrouvez en haut du jeu afin de lancer une partie, configurer une partie ou quitter le jeu;
- Le **stage View**: Plus complexe, cette partie est destinée à la vue **en jeu**, il va gérer l'affichage du plateau de jeu, des pions et des textes affichés lors d'une partie.

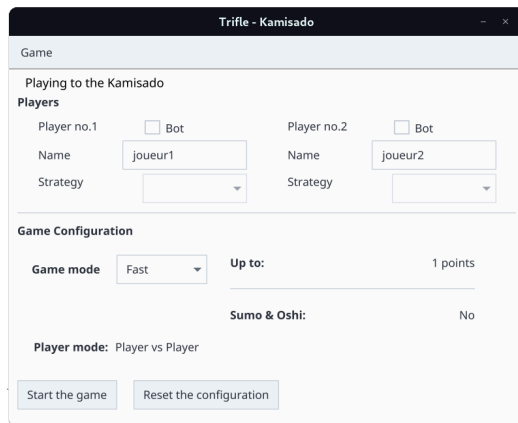


Fig. 9. – Capture d'écran de la vue au démarrage du jeu

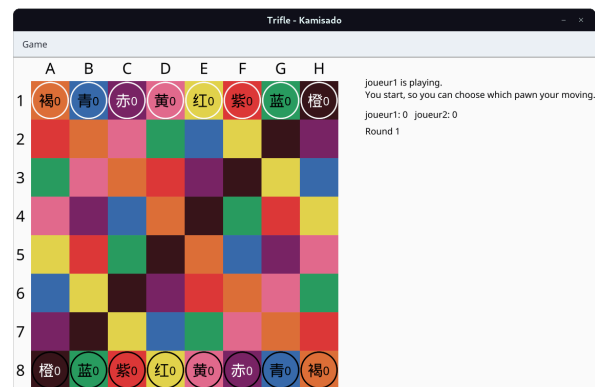


Fig. 10. – Capture d'écran de la vue en jeu

En parallèle, le modèle va enregistrer les pions, les derniers coups de chaque joueur et toutes les informations utiles au contrôleur.

Finalement, le contrôleur est séparé en plusieurs parties:

- **GameController** : Ce contrôleur va avoir pour mission de gérer les fins de chaque coup et détecter les victoires. Il s'occupe également des situations de victoire.
- **Contrôleur des actions** : Ce contrôleur permet de gérer les actions liées à la barre de navigation.
- **Contrôleur des clic de la souris** : Ce contrôleur est de loin le plus important. Il va permettre de déplacer les pions. Il va également être celui qui va déterminer les coups possibles, et si le joueur est bloqué.

La contrainte principale a été de gérer les mouvements avec la souris et le framework **Boardifier** fourni.

3.2 Choix

Pour transposer le jeu et ses règles dans notre application, nous avons pris plusieurs décisions clés :

- **Interface utilisateur** : Nous avons choisi une interface simple et intuitive. Au démarrage, une interface de configuration permet aux utilisateurs de choisir les paramètres de la partie. Une fois la partie lancée, l'interface principale affiche le plateau de jeu, les pions et les options de navigation.
- **Interactions utilisateur** : L'utilisateur interagit principalement avec la souris. Les clics de la souris sont capturés pour déplacer les pions sur le plateau. Des notifications sur le plateau informent l'utilisateur des actions possibles, et des informations à droite du plateau l'informent à propos des événements du jeu.
- **Gestion des règles** : Les règles du jeu sont implémentées dans les différents contrôleurs. Ils vérifient les coups possibles, détectent les situations de victoire et appliquent les règles complexes.

3.3 Challenges

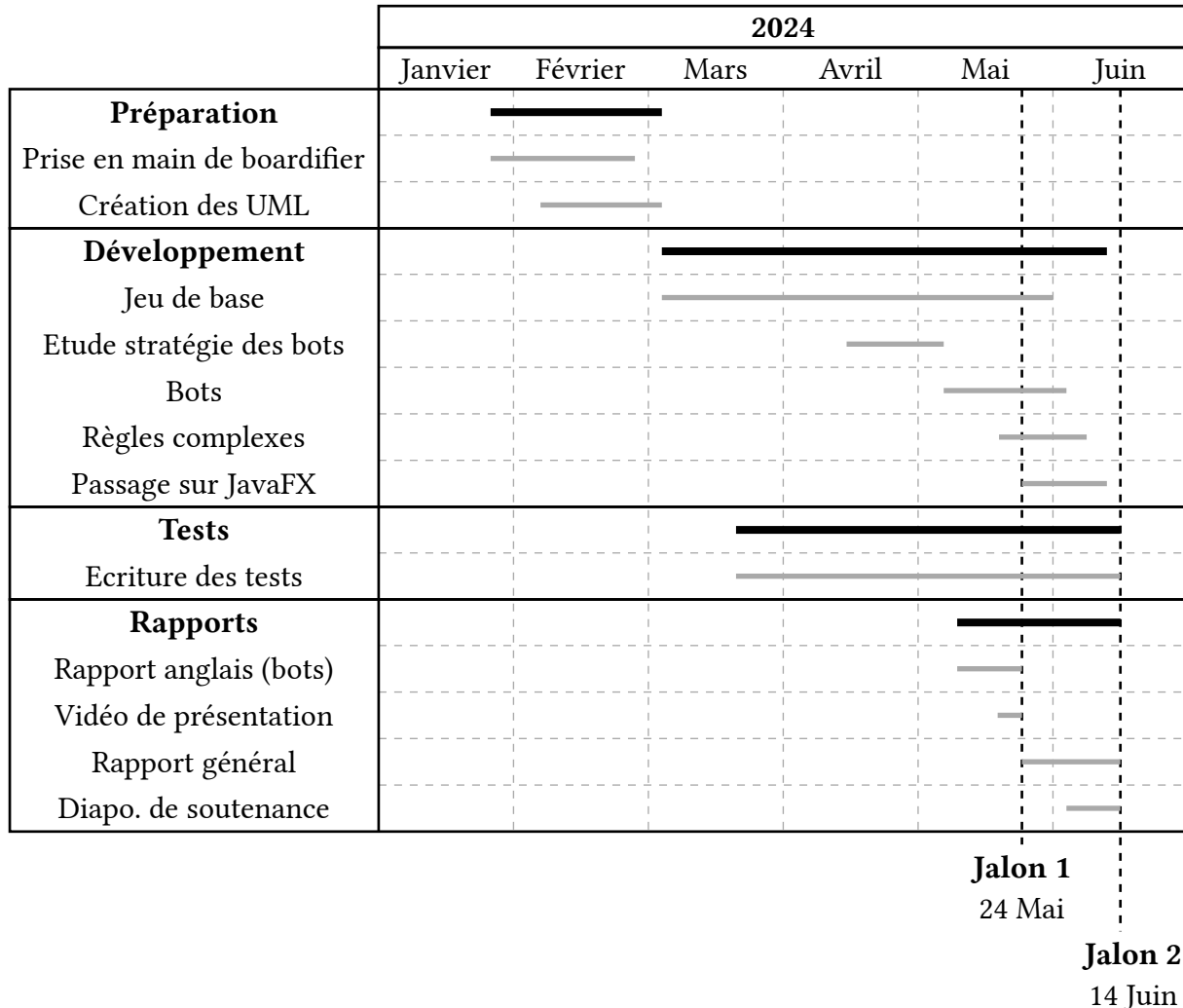
Voici les challenges que nous avons du relever pour transposer le jeu en une application. Cela implique le fait de souligner les difficultés informatiques et les obstacles rencontrés:

- **Gestion des mouvements de la souris** : La précision et la réactivité des mouvements de la souris sont cruciales pour une bonne expérience utilisateur. Il est difficile de capturer et d'interpréter correctement les clics de la souris, surtout lorsque des règles complexes doivent être appliquées en temps réel.
- **Complexité des règles du jeu** : Les règles du jeu peuvent devenir très complexes, notamment pour détecter les situations de blocage ou de victoire. Implémenter ces règles de manière fiable et efficace nécessite une compréhension approfondie du jeu et une programmation minutieuse.
- **Utilisation d'un framework inconnu** : L'utilisation du framework **Boardifier**, que nous ne connaissions pas avant ce projet, a été un réel défi. Il a fallu du temps pour comprendre comment utiliser ce framework efficacement et intégrer ses fonctionnalités dans notre application.
- **Respect du paradigme MVC** : Assurer une séparation claire entre le modèle, la vue et le contrôleur tout en garantissant une interaction fluide et efficace entre ces composants a été un défi constant (*Chapitre 3.1*).

Ces challenges soulignent l'importance d'une bonne planification, d'une compréhension claire des exigences du projet et d'une collaboration efficace au sein de l'équipe.

4 Organisation du travail

4.1 Planning prévisionnel



Dans ce planning, nous nous sommes concertés dès la prise de connaissance de la SAE, conscients de la complexité et du temps requis pour le développement du jeu, afin de rendre un travail de qualité.

Dans cet objectif, nous avons alloué un mois à la prise en main du framework Boardifier (notamment à travers le tutoriel fourni) et à la définition des UML, afin de déterminer nos besoins et options. Pour ces deux étapes, l'implication de tous les membres de l'équipe était requise.

Une fois ces étapes réalisées, le développement du jeu de base avec les règles basiques débutera début mars et se poursuivra jusqu'à fin mai. L'étude des stratégies possibles pour les deux bots commencera en parallèle mi-avril et sera suivie du début du développement des bots pendant le mois de mai. Les règles complexes et le passage à JavaFX et Boardifier-graphic interviendront peu après le premier jalon et dureront jusqu'au 15 juin environ.

En parallèle, des tests seront réalisés tout au long de la phase de développement.

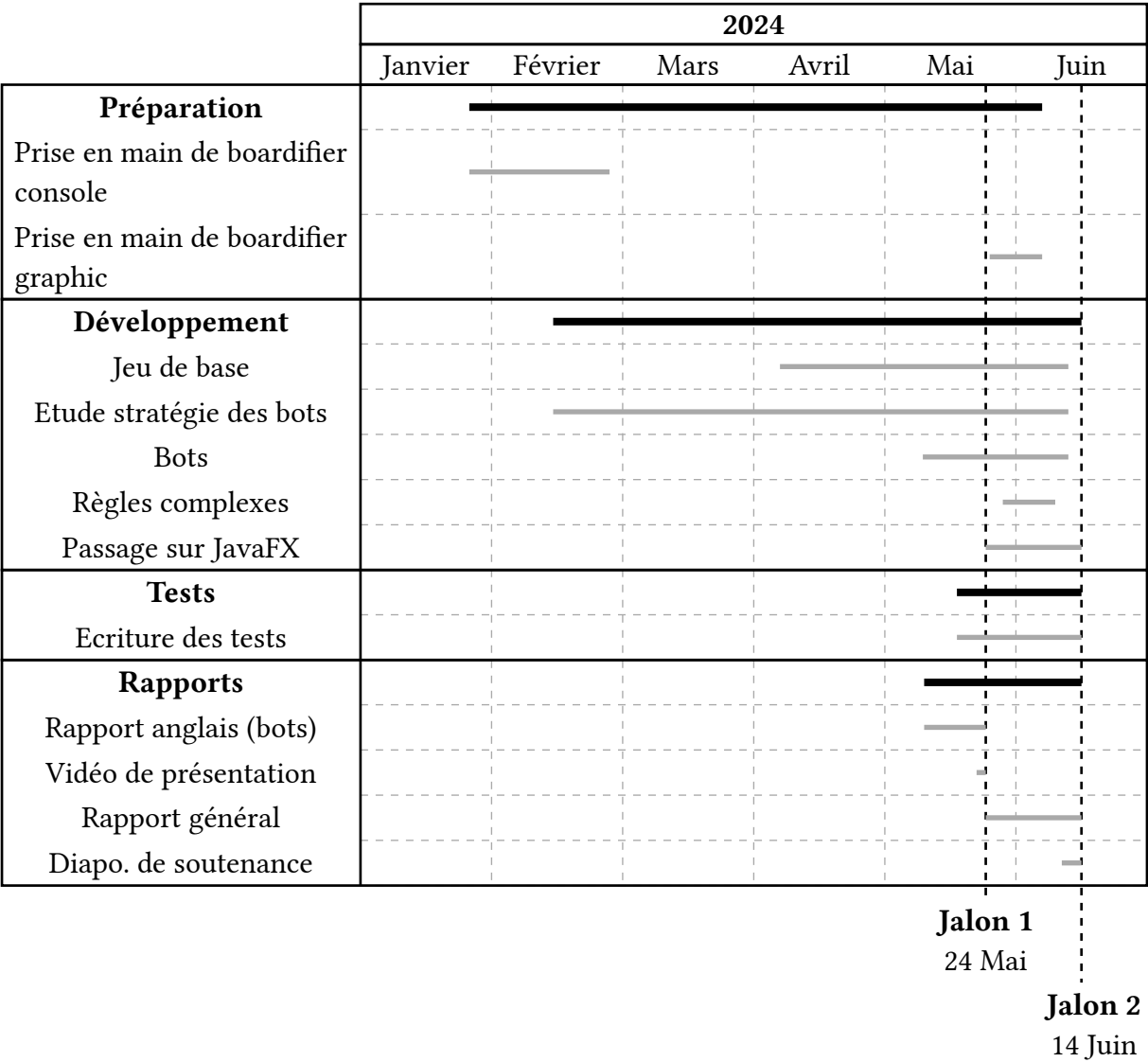
Les rapports commenceront mi-mai avec le rapport anglais sur les bots et la vidéo, remis avant le premier jalon, suivis du rapport général et de la préparation de la soutenance entre le jalon 1 et 2.

L'objectif est de s'y prendre suffisamment à l'avance dans le développement du jeu afin de pouvoir pallier à tout bug, contre-temps ou problème technique.

Pour la répartition des tâches, nous nous sommes répartis de cette façon:

- **Cédric:** Bot n°1, rapports et logique basique du jeu
- **Marvyn:** Bot n°1, rapports, logique des règles de base
- **Baptiste:** Tests unitaires et règles complexes
- **Timothée:** Bot n°2, droit et diapo de soutenance
- **Hugues:** Bot n°2 et vidéo

4.2 Planning réel



En comparaison au planning prévisionnel (*Chapitre 4.1*), nous pouvons remarquer de nombreuses différences.

Sur le plan du temps de travail, le développement a commencé très tard et de manière irrégulière. L'origine de cet écart provient d'un manque de travail durant les vacances, où certains n'étaient pas disponibles et par conséquent le planning prenait du retard chaque jour. L'étude de la stratégie des bots a commencé beaucoup plus tôt afin de comprendre le jeu et ses mécanismes complexes. En revanche, la prise en main de Boardifier-graphic a été reportée à la fin du projet, ce qui a perturbé le développement des fonctionnalités graphiques et entraîné des retards supplémentaires.

Le développement du jeu de base s'est prolongé bien au-delà de la période prévue initialement, ce qui a eu un impact direct sur les phases suivantes, notamment le développement des bots et des règles complexes. De plus, la rédaction des tests unitaires a été retardée, nécessitant une concentration accrue sur cette tâche dans les dernières semaines avant la date limite.

Les rapports ont également été affectés par ce retard général. Alors que le rapport anglais sur les bots et la vidéo de présentation ont été remis à temps pour le premier jalon, le rapport général et la préparation de la soutenance ont dû être réalisés en parallèle avec la finalisation du développement, augmentant ainsi la charge de travail et le stress pour l'équipe.

L'organisation réelle du projet a montré une nécessité d'amélioration dans la gestion du temps et la coordination entre les membres de l'équipe. Il aurait été bénéfique de maintenir un suivi régulier et de prévoir des points d'étape pour ajuster le planning et s'assurer du respect des délais. Malgré ces défis, l'équipe a su s'adapter et finaliser les tâches essentielles pour respecter les échéances imposées.

Pour la finalisation des tâches, nous nous sommes répartis de cette façon:

- **Cédric:** Bot n°1, rapports et logique basique du jeu, vidéo de démonstration
- **Marvyn:** Bot n°1, rapports, logique des règles de base
- **Baptiste:** Tests unitaires et règles complexes
- **Timothée:** Bot n°2, droit et diapo de soutenance
- **Hugues:** Bot n°2, stratégies des bots

5 Bilan

5.1 Réalisation technique

1. Niveau fonctionnel :

- Le jeu est jouable en mode humain-humain, humain-bot, et bot-bot sur les deux versions (console et graphique).
- Certains coups ont un bug dans la détection mais cela est impossible à corriger ou bien encore à reproduire. Ce bug est inexpliqué et arrive dans de rares cas.
- Les règles de base du jeu sont entièrement implémentées dans la version graphique, incluant:
 - La possibilité de jouer un pion en fonction de la dernière case de couleur de l'ennemi.
 - Les règles des sumos.
 - La sélection de l'ordre des pions par les joueurs.
 - Les règles de blocages partiels et complets.
 - Les conditions de victoire.
- Dans la version console, seules les règles de blocage et les règles de base sont implémentées.

2. Bots :

- Un seul bot est fonctionnel sur les deux versions : Mellie, basé sur un algorithme Min-Max et des poids. Actuellement, Mellie a un taux de victoire de 20% en raison d'un mauvais calcul des poids (sans les sumos).
- Le second bot, EureDeku, n'a pas pu être développé à temps. Cependant, il est théoriquement complet sur papier avec un taux de victoire estimé à 40% contre un humain en jeu de société (sans les sumos).

3. Configuration de la partie :

- La configuration de la partie est réalisée sur les deux versions.
 - La version console utilise une interface utilisateur en mode texte (TUI).
 - La version graphique utilise des champs de texte, menus de sélection et des cases à cocher avec JavaFX.

4. Déplacements :

- Les déplacements se font grâce aux coordonnées dans la version console et grâce à la souris dans la version graphique.

5. Autres fonctionnalités :

- Des animations sont implémentées dans la version graphique.
- Un thème personnalisé pour JavaFX a été utilisé afin de rendre l'interface plus conviviale.
- Des informations contextuelles sont fournies aux joueurs pendant une partie (sur les deux versions).
- Des tests unitaires (JUnit5 - Mockito) ont été réalisés pour les deux versions.

5.1.1 Fonctionnalités présentes

Nous avons rencontré plusieurs difficultés critiques au cours du projet, ayant des impacts variés sur l'avancement:

1. Gestion des mouvements de la souris et du framework Boardifier :

- La précision et la réactivité des mouvements de la souris étaient cruciales pour une bonne expérience utilisateur.
- Le framework Boardifier était inconnu de l'équipe, nécessitant un temps d'apprentissage important. L'analyse parallèle du jeu « The Hole » et de « Trifle » (Kamisado) a permis de comprendre les méthodes de classe nécessaires, mais cela a causé des retards importants.
- Point positif: meilleure compréhension du framework.
- Point négatif: retards dans l'implémentation, impactant notamment le développement du second bot.

2. Coordonnées du plateau 2D :

- Nous avons rencontré des problèmes avec les coordonnées du plateau 2D du jeu, car nous avons commencé le développement avant une mise à jour importante de Boardifier.
- Point négatif: mise en difficulté et retards, notamment dans l'implémentation des bots.

3. Gestion de deux projets en simultané :

- La gestion parallèle de deux projets (Projet de Virtualité Machine en SAE2.345 et Trifle-Kamisado en SAE2.126) a été complexe.
- Point négatif: manque de travail sur SAE2.126 durant SAE2.345 et périodes de partiels, causant des retards.

5.2 Connaissances et compétences

Lors de ce projet Trifle, nous avons développé de nombreuses connaissances et compétences:

1. Informatique :

- Utilisation intensive de la programmation orientée objet (POO) en Java.
- Application du paradigme MVC dans un projet complexe.
- Gestion et organisation de nombreux fichiers et répertoires dans un même projet.
- Développement de bots basés sur des algorithmes comme Min-Max.
- Réalisation de tests unitaires avec JUnit5 et Mockito.
- Utilisation du framework JavaFX pour l'interface graphique.

2. Gestion d'équipe et communication :

- Mise en place de réunions régulières, en présentiel et à distance via un serveur Discord dédié.
- Distribution claire des tâches et des responsabilités au sein de l'équipe.
- Amélioration de la communication et de la collaboration, même si cela a été difficile au début.

3. Rédaction et gestion de projet :

- Rédaction de rapports scientifiques et complets en anglais, incluant des comparaisons d'algorithmes.
- Gestion de projets à long terme avec des échéances claires et des livrables définis.

- Préparation de la soutenance de fin d'année, développant ainsi nos compétences en présentation orale.

Ce projet nous a permis de grandir en tant que développeurs et en tant qu'équipe, renforçant notre capacité à travailler ensemble pour surmonter les défis techniques et organisationnels.

Une version personnelle de cette partie est mise en annexe (*Chapitre 7.2*)

6 Conclusion

Le projet de développement du jeu Kamisado a représenté un réel défi, marqué par des découvertes enrichissantes et des défis techniques à surmonter. À travers une analyse approfondie des règles du jeu, une conception rigoureuse basée sur le paradigme MVC, et une mise en œuvre soignée des fonctionnalités graphiques et des algorithmes de jeu, notre équipe a réussi à créer une application fonctionnelle et immersive.

Nous avons exploré les différentes stratégies du Kamisado et adapté ses règles complexes dans une interface utilisateur intuitive, facilitant ainsi une expérience de jeu agréable pour les utilisateurs. La gestion des Sumos, les déplacements diagonaux et verticaux, ainsi que la résolution de blocages ont été fidèlement retranscrits dans notre application, garantissant une jouabilité conforme aux attentes des joueurs.

Cependant, le projet n'a pas été sans défis. La gestion des mouvements de la souris, la complexité des règles du jeu et l'adaptation au framework Boardifier ont constitué des obstacles significatifs, nécessitant une collaboration étroite et une résolution créative des problèmes rencontrés.

Malgré les écarts par rapport au planning initial, notre équipe a su s'adapter et relever les défis, aboutissant à une application robuste capable de supporter différents modes de jeu et offrant une expérience utilisateur cohérente et fluide.

Finalement, le développement du jeu Kamisado a été une bonne expérience, combinant compétences techniques, créativité et travail d'équipe pour aboutir à un produit final qui reflète notre engagement envers l'excellence dans le domaine du développement de jeux.



Fig. 11. – Boîte du jeu Kamisado

Table des illustrations

Fig. 1: Jeu du Kamisado	4
Fig. 2: Situation de départ	5
Fig. 3: Un exemple de plateau en jeu	5
Fig. 4: Coup précédant le oshi	6
Fig. 5: Exécution du oshi par un Sumo	6
Fig. 6: Partie sans impasse	7
Fig. 7: Joueur atteint le camp de l'autre joueur	7
Fig. 8: Schéma du paradigme MVC	8
Fig. 9: Capture d'écran de la vue au démarrage du jeu	9
Fig. 10: Capture d'écran de la vue en jeu	9
Fig. 11: Boite du jeu Kamisado	18
Fig. 12: Etude juridique	20
Fig. 13: Participation de Hugues sur la SAE2.126	23
Fig. 14: Participation de Baptiste sur la SAE2.126	26
Fig. 15: Participation de Timothée sur la SAE2.126	28
Fig. 16: Participation de Marvyn sur la SAE2.126	31
Fig. 17: Participation de Cédric sur la SAE2.126	34
Fig. 18: Une situation où la tour rose du joueur 1 est bloquée.	37
Fig. 19: Une situation de oshi par un Triple-Sumo	37
Fig. 20: Une situation de oshi par un Triple-Sumo	37

7 Annexes

7.1 Étude de cas juridique

Les jeux de société et les jeux vidéo, bien que tous deux destinés à divertir, présentent des enjeux juridiques distincts. Ce rapport examine ces différences sous les angles de la propriété intellectuelle, des licences et contrats, de la régulation des contenus et de la protection des consommateurs, ainsi que de la responsabilité juridique et de la protection des données personnelles.



Fig. 12. – Etude juridique

7.1.1 Propriété Intellectuelle

La propriété intellectuelle constitue un enjeu majeur pour les deux types de jeux, mais les aspects protégés et les conséquences des infractions varient.

Droits d’auteur : Les jeux de société voient leurs règles, illustrations et scénarios protégés par les droits d’auteur, avec des sanctions civiles et pénales en cas d’infraction. Pour les jeux vidéo, les droits d’auteur couvrent les scripts, graphismes, musiques et logiciels, impliquant des sanctions similaires.

Marques : Les noms et logos des jeux de société peuvent être enregistrés comme marques, avec des peines civiles et pénales en cas de violation. Dans le domaine des jeux vidéo, la protection des marques est cruciale pour se différencier sur un marché numérique saturé, les peines étant similaires.

Brevets et Dessins/Modèles : Les mécanismes uniques et designs esthétiques des jeux de société peuvent être protégés, avec des sanctions civiles et des amendes en cas d’infraction. Pour les jeux vidéo, les technologies spécifiques et les designs peuvent être brevetés ou protégés par des dessins/modèles, entraînant des peines analogues.

7.1.2 Licences et Contrats

Les accords de licence et les contrats de développement sont essentiels dans les deux secteurs, bien que leur complexité diffère.

Accords de licence : Dans les jeux de société, ils sont utilisés pour les adaptations en jeux vidéo ou pour l’utilisation de propriétés intellectuelles, avec des sanctions civiles en cas de

rupture. Les jeux vidéo, quant à eux, nécessitent des licences cruciales pour les collaborations trans-média, avec des peines similaires en cas de non-respect.

Contrats de développement : Moins complexes pour les jeux de société, ces contrats définissent les droits et rémunérations, avec des sanctions civiles en cas de non-respect. Les jeux vidéo impliquent des contrats très complexes avec de multiples parties, les sanctions étant identiques.

7.1.3 Régulation des Contenus et Protection des Consommateurs

La régulation des contenus et la protection des consommateurs diffèrent également entre les deux types de jeux.

Classification des jeux : La classification des jeux de société est moins formalisée mais soumise à des normes de sécurité strictes, les infractions entraînant des sanctions administratives et des amendes. Pour les jeux vidéo, les systèmes PEGI et ESRB influencent la vente et la distribution, avec des sanctions administratives et des amendes en cas d'infraction.

Protection des consommateurs : Les jeux de société doivent respecter des normes de sécurité strictes pour les pièces de jeu, avec des amendes et des rappels de produits en cas d'infraction. Les jeux vidéo, en revanche, doivent garantir les droits sur les achats numériques et protéger contre les bugs, les non-respects entraînant des sanctions civiles.

7.1.4 Responsabilité Juridique et Protection des Données Personnelles

Les responsabilités juridiques et la protection des données personnelles varient considérablement entre les jeux de société et les jeux vidéo.

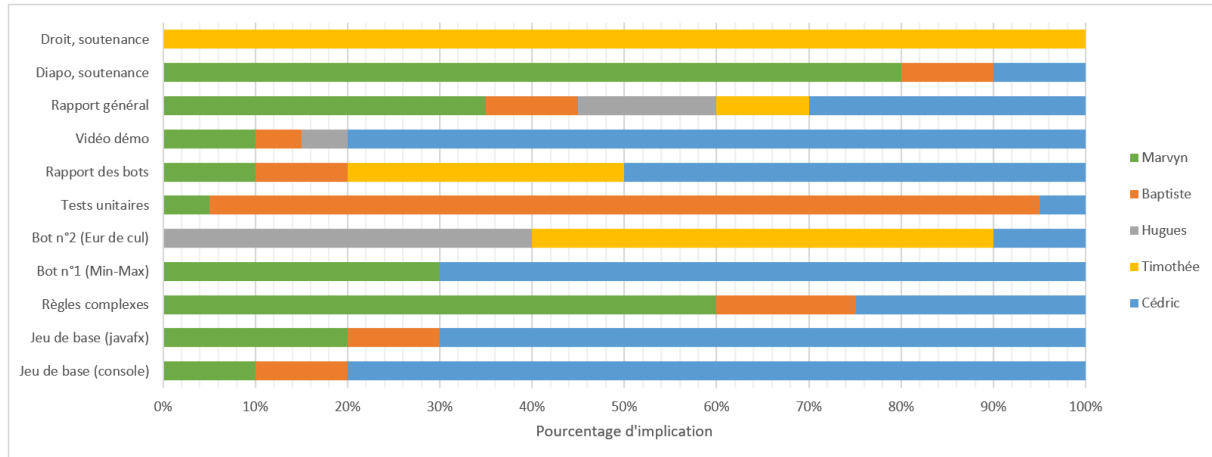
Responsabilité des fabricants et développeurs : Les jeux de société impliquent une responsabilité pour les défauts physiques, avec des sanctions civiles en cas d'infraction. Les jeux vidéo sont responsables des bugs et dysfonctionnements, les peines étant similaires.

Protection des données personnelles : Les jeux de société sont moins concernés par ces questions, tandis que les jeux vidéo doivent se conformer à des réglementations strictes (RGPD, CCPA), avec des amendes lourdes et des sanctions administratives en cas d'infraction.

Les jeux de société et les jeux vidéo présentent des défis juridiques distincts, principalement dus à la nature de leur médium respectif. Les jeux vidéo, en particulier, nécessitent une attention accrue en matière de protection des données personnelles et de régulation des contenus. L'évolution des technologies et des législations continuera d'influencer ces domaines, posant de nouveaux défis pour les créateurs, les consommateurs et les entreprises.

7.2 Auto-évaluation

Voici un résumé global de la participation de chaque membre du groupe dans les différentes tâches principales du projet:



Vous pourrez trouver une version personnelle de ce graphique pour chaque étudiant dans leur auto-évaluation (Hugues: *Fig. 13* - Baptiste: *Fig. 14* - Timothée: *Fig. 15* - Marvyn: *Fig. 16* - Cédric: *Fig. 17*).

7.2.1 Hugues

7.2.1.1 Bilan

7.2.1.2 Mon implication

Durant cette SAE, j'ai travaillé sur les tâches suivantes :

- Le Bot n°2 et la définition de sa stratégie de jeu
- La vidéo de présentation du projet
- Le rapport général (présent document)

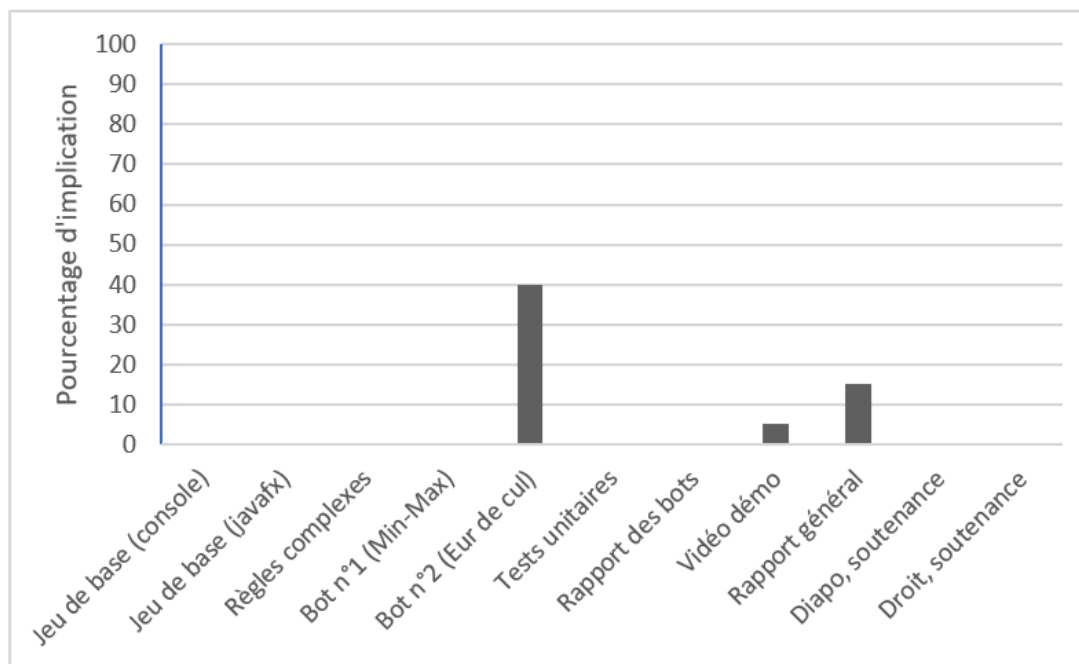


Fig. 13. – Participation de Hugues sur la SAE2.126

Bot n°2 et stratégie de jeu (40%)

Pour le bot n°2, j'ai été en charge de définir sa stratégie de jeu. J'ai procédé à l'élaboration de l'algorithme final de deux manières :

- en testant différents algorithmes et en choisissant les stratégies les plus gagnantes
- en partant d'un algorithme « naturel » mais complexe, puis en le simplifiant au maximum pour éviter les calculs superflus à la machine

J'ai également aidé Timothée à déboguer le code du bot, que nous avons eu beaucoup de mal à implémenter (sans succès).

Vidéo de présentation (5%)

Pour ce qui est de la vidéo de présentation, j'ai aidé à l'organisation et l'orchestration. J'ai aidé à écrire le texte et à préparer le déroulement de la vidéo. Ensuite je me suis occupé d'une partie du montage de la vidéo.

Rapport général (15%)

Pour le rapport général, en plus d'écrire mon auto-évaluation, je me suis occupé d'une petite partie de la rédaction et j'ai également procédé à une relecture générale du document afin de corriger les potentielles fautes, reformuler certains passages, et harmoniser la mise en page.

7.2.1.2.1 Connaissances et compétences acquises

Durant cette SAE, j'ai considérablement enrichi mes connaissances et compétences dans plusieurs domaines clés.

Tout d'abord, j'ai fait des progrès significatifs en Java et JavaFX. En travaillant sur le Bot n°2, j'ai approfondi ma compréhension de la programmation orientée objet, des structures de données et des algorithmes en Java. La nécessité de simplifier et d'optimiser les algorithmes pour le bot m'a permis de mieux maîtriser les concepts de complexité algorithmique et d'optimisation des performances. De plus, l'implémentation de l'interface graphique avec JavaFX m'a donné une première expérience pratique de ce framework, améliorant ainsi mes compétences en développement d'applications GUI en Java.

Ensuite, bien que je n'aie pas participé activement au projet Boardifier, j'ai observé l'évolution du projet et cela m'a permis d'acquérir une meilleure connaissance de cet outil. J'ai pu voir comment Boardifier peut être utilisé pour structurer et organiser des projets complexes, et j'ai appris à apprécier son potentiel pour la gestion de projets informatiques.

En termes de compétences interpersonnelles, cette SAE a été particulièrement enrichissante. La communication avec les autres membres du groupe n'a pas toujours été facile, et nous avons rencontré plusieurs moments de désaccord. Cependant, ces défis m'ont appris l'importance de la communication claire et de l'écoute active. J'ai également compris l'importance de rester patient et diplomate, surtout dans des situations de conflit. Ces expériences m'ont aidé à développer mes compétences en résolution de conflits et en collaboration au sein d'une équipe.

Enfin, cette SAE m'a offert une excellente opportunité de développer mes compétences en gestion de projet. J'ai appris à utiliser des outils comme le diagramme de Gantt pour planifier et suivre l'avancement du projet. J'ai également acquis de l'expérience dans l'organisation et la tenue de réunions efficaces, ainsi que dans la planification et l'exécution des tâches. Cette expérience m'a permis de comprendre l'importance de la rigueur et de l'organisation pour mener à bien un projet, et m'a aidé à développer une approche plus structurée et méthodique du travail en groupe.

7.2.1.2.2 Difficultés majeures rencontrées

Durant cette SAE, plusieurs difficultés majeures ont entravé le bon déroulement de notre projet, affectant tant notre organisation que notre dynamique de groupe.

L'un des principaux problèmes a été l'organisation au sein du groupe. Nous avons commencé à travailler de manière active beaucoup trop tard dans le processus. Une grande partie de notre temps a été consacrée à la planification et à l'apprentissage des nouveaux outils mis à notre disposition, ce qui a considérablement retardé le début effectif du travail sur le projet. Cette procrastination initiale a créé une pression accrue à mesure que l'échéance approchait, nous obligeant à travailler dans l'urgence pour rattraper le temps perdu.

En outre, l'entente au sein du groupe a été compliquée pendant une bonne partie de la SAE. Les divergences quant aux objectifs du projet ont provoqué quelques désaccords. Certains membres de l'équipe souhaitaient se lancer dans des travaux plus complexes que nécessaire, avec une volonté de dépasser largement les attentes du projet. Cette ambition, bien qu'admirable, n'était pas partagée par tout le monde et a parfois entraîné des discussions sur la direc-

tion à prendre. Ces discussions, bien qu'animées, étaient toujours respectueuses et constructives, mais elles ont tout de même ralenti notre rythme de travail.

Ces divergences ont mis en évidence le besoin d'une meilleure coordination et d'un alignement clair des objectifs dès le début du projet. Bien que ces défis aient complexifié la répartition des tâches et la prise de décisions, ils m'ont également offert une précieuse expérience en gestion de conflits et en collaboration au sein d'une équipe. J'ai appris l'importance de la communication et de la flexibilité pour maintenir un environnement de travail harmonieux et productif.

En conclusion, ces difficultés ont mis en lumière l'importance cruciale d'une bonne organisation dès le début du projet et d'une communication claire et respectueuse au sein de l'équipe. Elles m'ont également appris à mieux gérer les désaccords et à rester concentré sur les objectifs malgré les défis.

7.2.1.2.3 Conclusion

Au final, malgré les difficultés rencontrées en cours de route, notre groupe a su se réunifier en fin de projet et aboutir à un résultat satisfaisant. Les efforts de chacun ont permis de surmonter les obstacles organisationnels et relationnels, et nous avons réussi à présenter un projet cohérent et bien structuré.

J'aurais souhaité participer plus activement au développement, en particulier pour le bot, et je regrette de ne pas avoir été plus efficace dans mon travail pour que celui-ci fonctionne correctement. Néanmoins, cette expérience m'a permis de comprendre l'importance de l'organisation et de la collaboration au sein d'une équipe, ainsi que la nécessité de rester flexible et ouvert aux compromis.

Dans l'ensemble, je suis satisfait du projet. Il n'a pas toujours été facile, mais il a été une expérience extrêmement enrichissante. J'ai acquis de nouvelles compétences techniques et humaines, et j'ai appris à mieux gérer les défis d'un projet en groupe. Cette SAE m'a permis de grandir tant sur le plan personnel que professionnel, et je suis reconnaissant pour les leçons tirées de cette aventure.

7.2.2 Baptiste

7.2.2.1 Bilan

7.2.2.2 Mon implication

Durant cette SAE j'ai travaillé sur plusieurs aspects:

- Principalement la réalisation des tests unitaires
- La réalisation de quelques règles élémentaires du jeu
- En partie la réalisation de rapports (pour les bots et le rapport général)
- La réalisation de la vidéo de démonstration

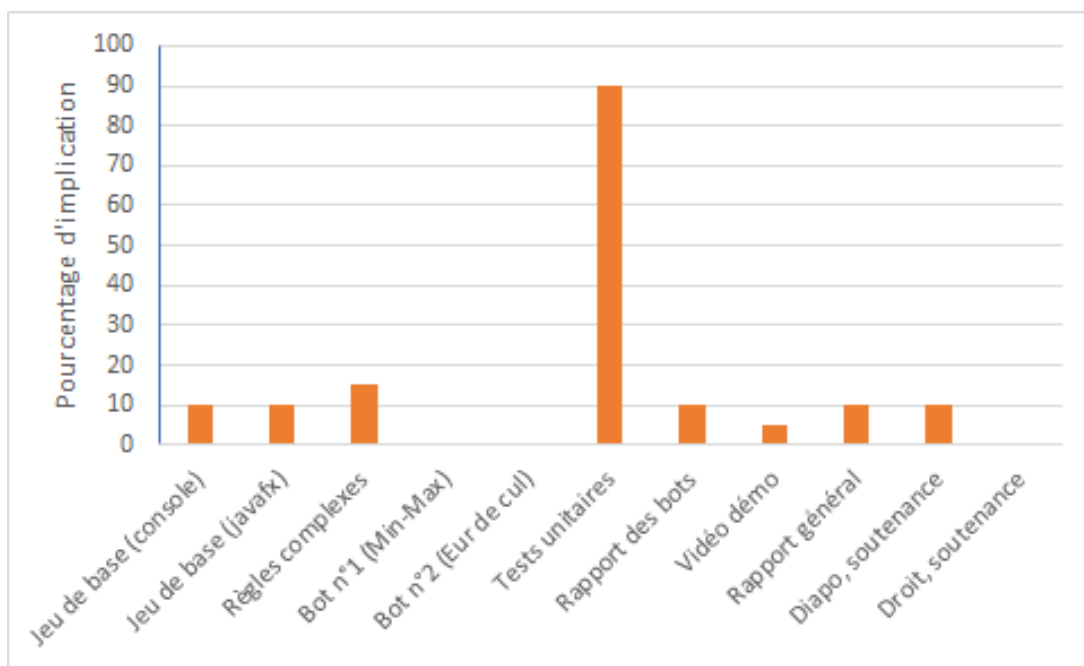


Fig. 14. – Participation de Baptiste sur la SAE2.126

7.2.2.3 Différentes parties réalisées

Tests unitaires (90%):

J'ai réalisé en grande majorité les tests unitaires nécessaire au projet : notamment les tests des Controllers et du Model.

Jeu de base (console et javaFx) (10%)

J'ai réalisé quelques règles de base du jeu et apporté mon aide pour la détection et la correction de bugs de certains éléments du code de base.

Règles complexes (15%)

J'ai réalisé avec mes camarades Marvyn et Cédric des règles plus complexes du jeu : à savoir la règle du Oshi et du Sumo.

Rapports et diaporama (30%)

J'ai aidé l'ensemble du groupe sur les travaux de rédaction et réalisation de rapport et de diaporama.

7.2.2.3.1 Connaissances et compétences acquises

Ce projet m'a permis d'accroître mes connaissances en java, notamment de la librairie JavaFx, et d'utiliser le paradigme MVC lors de la réalisation d'application graphique.

Ce projet m'a également permis d'en apprendre plus sur la réalisation d'intelligence artificielle, notamment le fonctionnement d'un algorithme Min-Max, ou simplement la réalisation d'une stratégie pour un jeu de plateau.

Le travail d'équipe et ma relation avec mes camarades a également été impactée par ce projet. Le fait d'avoir aidé mes camarades à la plupart des tâches m'a permis d'en apprendre plus sur leurs manières de réfléchir et cela à même pu renforcer nos liens.

7.2.2.3.2 Difficultés majeures rencontrées

J'ai rencontré des difficultés sur la réalisation des tests unitaires, notamment lors des tests sur les Controllers car je n'arrivais pas à utiliser Mockito (une librairie permettant de faire des tests) sur la View de l'application.

7.2.2.3.3 Conclusion

En conclusion, durant cette SAE, je me suis impliqué dans plusieurs tâches pour aider à la réalisation de ce projet, principalement dans la réalisation de tests unitaires, mais également la réalisation de règles de bases et complexes, la réalisation de rapports et de diaporama. Ce projet m'a également permis de progresser dans le développement d'application graphique en java avec l'utilisation de JavaFx et la réalisation de tests unitaire. Malgré les problèmes rencontrés, ce projet m'a également permis de renforcer mes compétences en travail d'équipe.

7.2.3 Timothée

7.2.3.1 Bilan

7.2.3.1.1 Mon implication

Durant cette SAE j'ai travaillé sur différentes thématiques :

- La réalisation d'une IA au niveau conceptuel et l'implémentation technique dans boardifier
- La question de droit qui portait sur le point de vue juridique des jeux de sociétés et des jeux vidéos
- Une partie de la réalisation du diaporama et du contenu de la soutenance

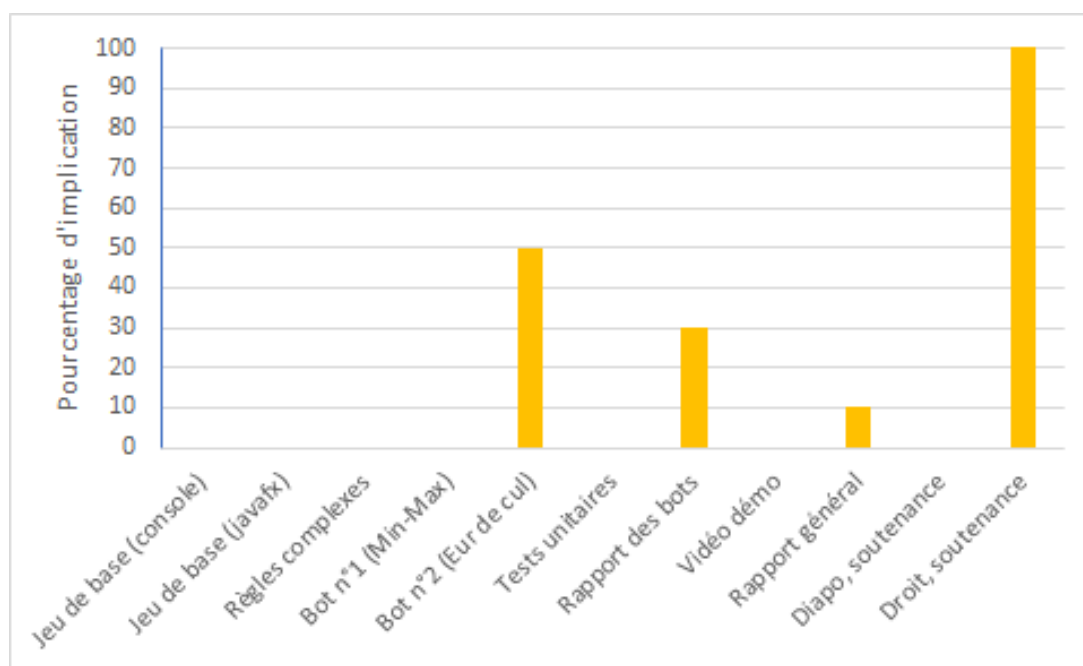


Fig. 15. – Participation de Timothée sur la SAE2.126

7.2.3.1.2 Implication dans ces parties

Droit (100%) :

J'ai réalisé seul la question de droit : « Jeu de société, jeux vidéo : point de vue juridique ». Cette partie comprenait une présentation orale sur cette question ainsi qu'une présentation dans ce rapport. J'y ai principalement parlé de la propriété intellectuelle, des licences et contrats, de la régulation des contenus et de la protection des consommateurs, ainsi que de la responsabilité juridique des données personnelles.

BOT n°2 (60%) :

Avec l'aide de Hugues nous avons testé différentes possibilités d'algorithme de bot afin d'obtenir une IA de niveau débutant/intermédiaire. De plus, j'ai réalisé l'implémentation de l'IA dans le programme en utilisant les outils adaptés, fournis par boardifier.

La stratégie du bot nous est venue en testant diverses méthodes sur un plateau physique. Le but était de trouver un nombre de règles restreintes, permettant à l'IA d'obtenir un bot aux actions cohérentes posant des difficultés aux joueurs débutants et intermédiaires.

Diaporama et soutenance (30%) :

J'ai travaillé avec l'ensemble du groupe sur la réalisation du diaporama de soutenance et le contenu de cette soutenance. En particulier sur la structure générale de la soutenance puis dans un deuxième temps sur les parties concernant mon travail et les sujets que j'ai traités.

7.2.3.1.3 Connaissances et compétences acquises

La question de droit m'a permis de comprendre en surface les différentes notions qui doivent être prise en compte par une entreprise lors de la réalisation d'un jeu vidéo ou d'un jeu de société. Ces deux types de jeu sont, d'un point de vue juridique, plutôt similaire, les différences principales portent surtout sur le type de support utilisé, à savoir les supports numériques (console, application) et les supports physiques (plateau, jeu de cartes, éléments personnalisés).

Le bot n°2 nous a permis d'étudier la création de stratégies dans un jeu mais surtout la transposition de nos idées dans une application et l'implémentation de l'algorithme dans du code impératif. Bien que le bot ne fonctionne pas à cause d'un problème de coordonnées, l'algorithme semble avoir bien été transcrit.

De manière plus générale, le travail d'équipe et les relations avec mes camarades ont été renforcées. Le fait de donner des feedbacks nous a permis de nous améliorer dans nos projets.

De plus, le fait de travailler dans un environnement de développement plus complet en Java, m'a permis de m'améliorer dans ce langage et dans l'étude plus complète d'un projet de grande envergure.

7.2.3.1.4 Difficultés majeures rencontrées

Sur la question de droit, il y avait beaucoup de ressources en ligne, même si les articles sur les jeux vidéos était plus faciles à trouver que ceux sur les jeux de société.

La difficulté majeur que j'ai rencontré se situait au niveau du second bot. Le programme Java possédait 4 couches : le langage Java, le framework JavaFX, le framework Bordifier, et enfin la surcouche Trifle (notre couche personnalisée). Le travail entre ces différentes couches a été compliqué. Il a créé un problème dans le système de coordonnées.

Le système de coordonnées nous a posé diverses difficultés. Dans certaines classes, le premier pion était situé en (0, 0), d'autres fois en (1, 1). De plus, dans certains cas, les axes X et Y étaient échangés. Tous cela nous a empêché de finir le bot n°2 à temps. Avec plus de délai, il aurait été possible d'investiguer les changements de coordonnées pour rendre le bot fonctionnel. A noter que le bot ne crée pas d'erreur lorsqu'il joue, mais joue à la place dans des positions illégales.

7.2.3.1.5 Conclusion

En somme, durant cette SAE, j'ai été impliqué dans diverses thématiques, notamment la conceptualisation et l'implémentation technique d'une IA dans Boardifier, l'exploration juridique des jeux de société et vidéos, ainsi que la préparation du diaporama et de la soutenance. J'ai pris en charge seul la question juridique, abordant des sujets comme la propriété intellectuelle, les licences, la régulation des contenus, et la protection des consommateurs. Pour le développement de l'IA, en collaboration avec Hugues, nous avons testé différents algorithmes pour créer un bot de niveau débutant/intermédiaire, en transposant nos idées dans une appli-

cation concrète malgré un problème de coordonnées non résolu. J'ai également contribué à l'élaboration du diaporama et à la structuration de la soutenance. Ce projet m'a permis d'acquérir des connaissances sur les aspects juridiques des jeux, de renforcer mes compétences en Java, et de consolider le travail d'équipe. Cependant, j'ai rencontré des difficultés majeures, notamment avec le système de coordonnées du bot, compliquées par les multiples couches de programmation impliquées.

7.2.4 Marvyn

7.2.4.1 Bilan

7.2.4.1.1 Mon implication

Au cours de cette SAE, j'ai été impliqué dans les tâches suivantes :

- La réalisation d'une IA au niveau conceptuel et l'implémentation des poids du Min-Max.
- Les règles techniques des Sumos, des déplacements et des victoires.
- La gestion des caractéristiques des pions (classe Pawn).
- La rédaction des rapports techniques (bots et général).
- La réalisation du diaporama et du contenu de la soutenance de clôture de projet.

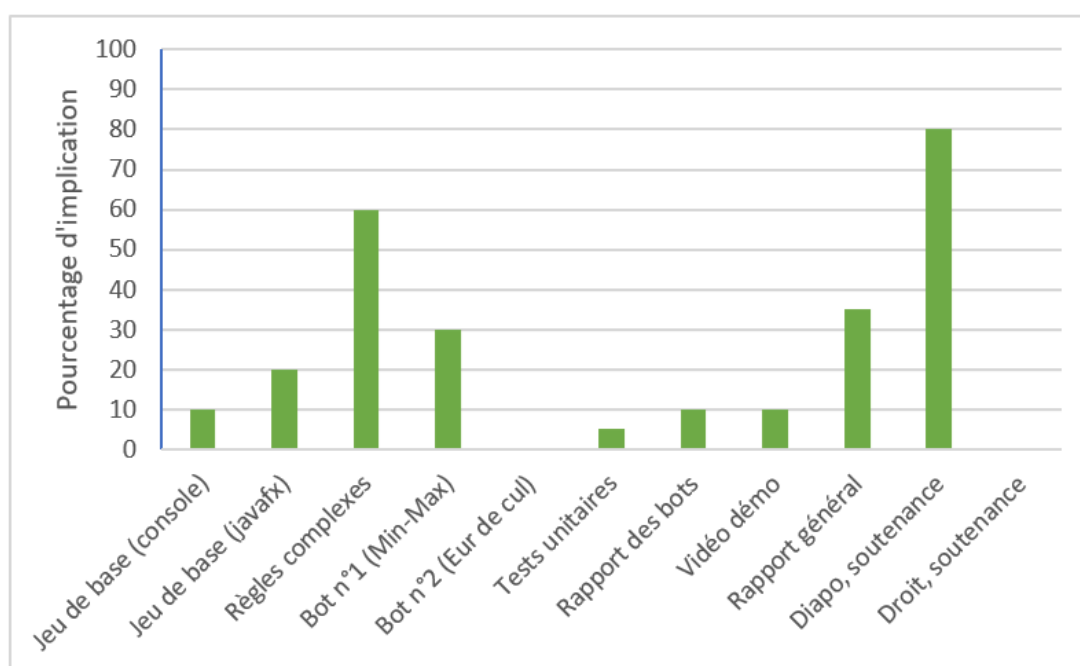


Fig. 16. – Participation de Marvyn sur la SAE2.126

Règles techniques des Sumos et des déplacements (60%) :

J'ai collaboré avec Cédric et Baptiste pour définir les règles de déplacement des pions, les conditions de victoire et les interactions des Sumos dans le jeu. J'ai travaillé principalement sur le développement des règles complexes, leur implémentation dans le TrifleBoard, TrifleStageModel et GameMouseController.

Gestion des pions (70%) :

J'ai pris en charge la gestion de la classe Pawn, définissant toutes les caractéristiques des pions, incluant leurs mouvements, interactions et états spécifiques.

Développement de l'IA - BOT n°1 (30%) :

Avec l'aide de Cédric, nous avons exploré différentes stratégies d'algorithmes pour concevoir une IA de niveau expert. J'ai implémenté les poids du Min-Max dans le programme en utilisant les outils fournis par Boardifier. La stratégie du bot a été élaborée en se basant sur des recherches en ligne sur la création d'IA. Bien que le bot n'ait pas un taux de victoire optimal

en raison des poids pas totalement bien calculés, cette expérience a été enrichissante pour comprendre la création de stratégies dans les jeux.

Rédaction des rapports techniques (45%) :

J'ai rédigé une partie du rapport sur les bots en anglais, en me concentrant sur l'algorithme Min-Max, ainsi qu'une grande partie du rapport général sur la SAE globale.

Réalisation du diaporama et de la soutenance (80%):

J'ai travaillé avec l'ensemble du groupe sur la création du diaporama de soutenance et le contenu de cette présentation. Cela comprenait la structuration des informations, la préparation des slides et la coordination pour la présentation finale.

7.2.4.1.2 Connaissances et compétences acquises

Au cours de ce projet, j'ai acquis diverses connaissances et compétences. D'un point de vue informatique, j'ai approfondi mes compétences en programmation, en particulier avec les algorithmes d'IA comme le Min-Max. J'ai appris à gérer des règles de jeu complexes et à les implémenter de manière efficace dans un code impératif. L'utilisation d'outils de développement collaboratif comme Git m'a permis de gérer et suivre les modifications du code de manière organisée.

Sur le plan de l'organisation du travail, j'ai appris à structurer mon travail de manière plus efficace et à coordonner les tâches avec mes coéquipiers. L'utilisation d'outils de gestion de projet comme Gantt a été cruciale pour suivre l'avancement des tâches et assurer une bonne coordination au sein de l'équipe.

En termes de compétences relationnelles, j'ai amélioré mes compétences en communication et en travail d'équipe. La coordination avec mes coéquipiers pour les règles et les stratégies du bot m'a permis de développer des techniques de collaboration efficaces. J'ai également appris à donner et recevoir des feedbacks constructifs, ce qui a contribué à améliorer la qualité du travail collectif.

Ce projet m'a permis de développer mes compétences en programmation Java et en conception de jeux vidéo. J'ai appris l'importance d'une communication efficace et d'une gestion de projet rigoureuse. De plus, ce projet a renforcé ma capacité à travailler en équipe et à répartir les tâches de manière efficiente. Enfin, j'ai découvert l'importance d'une documentation claire et concise, essentielle pour assurer une bonne compréhension et un suivi efficace des travaux réalisés.

7.2.4.1.3 Difficultés majeures rencontrées

Le projet a présenté plusieurs défis majeurs. Sur le plan informatique, l'implémentation des poids du Min-Max s'est révélée complexe, et malgré nos efforts, le bot n'a pas atteint un taux de victoire satisfaisant. Les tests et le débogage ont pris plus de temps que prévu en raison de la complexité des algorithmes. De plus, le framework Boardifier a posé des problèmes de coordination, notamment avec l'utilisation du format de coordonnées (ligne, colonne), ce qui a entraîné des bugs difficiles à résoudre.

En ce qui concerne l'organisation du travail, la gestion du temps et la répartition des tâches ont parfois été difficiles, surtout lors de la phase finale de rédaction des rapports et de la pré-

paration de la soutenance. Il a fallu ajuster constamment les priorités pour s'assurer que tous les aspects du projet étaient couverts.

Sur le plan relationnel, la coordination avec le groupe a parfois été un défi, surtout lors de la prise de décisions techniques cruciales. Des divergences d'opinion ont nécessité des discussions approfondies pour arriver à un consensus, ce qui a parfois ralenti le processus de prise de décision et la progression du projet.

7.2.4.1.4 Conclusion

Ce projet (SAE2.126) a constitué une expérience extrêmement enrichissante, dépassant mes attentes en termes d'apprentissage et de développement de compétences. La complexité des tâches m'a permis de pousser mes connaissances en informatique plus loin que prévu, notamment dans le domaine de l'IA et de la gestion des règles de jeu.

Sur le plan des critiques positives, le projet était pertinent et aligné avec mes attentes personnelles, me permettant de développer des compétences techniques précieuses en Java (Poo - JavaFx). J'ai particulièrement apprécié l'aspect pratique du projet, qui m'a offert l'opportunité de mettre en application des concepts théoriques de manière concrète. De plus, la collaboration avec mes coéquipiers a été enrichissante et m'a permis de développer des compétences en matière de gestion de projet, renforçant ma capacité à travailler efficacement en équipe.

Cependant, le projet n'a pas été sans défis. Parfois, il s'est avéré trop ambitieux par rapport au temps imparti, ce qui a engendré du stress et des difficultés organisationnelles. La répartition des tâches aurait pu être plus équilibrée pour éviter une surcharge de travail sur certains membres du groupe. De plus, il aurait été bénéfique d'avoir des réunions de suivi plus fréquentes avec tous les membres du groupe pour s'assurer que nous étions sur la bonne voie et pour recevoir des conseils lorsque nous rencontrions des obstacles techniques ou organisationnels.

Pour améliorer les futurs projets, je suggère de prévoir plus de temps pour la phase de test et d'optimisation, en particulier pour les composants complexes comme les algorithmes d'IA. Il serait également utile d'inclure des réunions de commencement en début de projet pour mieux définir les rôles et les responsabilités de chaque membre du groupe (Ce qu'on a vu lors de la SAE2.345). Enfin, mettre en place des revues de code régulières pourrait améliorer la qualité du code et faciliter la détection des erreurs potentielles, tout en encourageant une culture de l'amélioration continue.

En conclusion, ce projet a été une expérience formatrice qui m'a permis de développer mes compétences et de repousser mes limites. Malgré les défis rencontrés, j'ai acquis une compréhension approfondie des techniques de programmation, de la gestion de projet et du travail en équipe. Je suis déterminé à continuer d'apprendre et à me perfectionner dans ce domaine, et je suis convaincu que les compétences et les leçons tirées de ce projet me seront précieuses pour mes futurs projets académiques et professionnels.

7.2.5 Cédric

7.2.5.1 Bilan

7.2.5.1.1 Mon implication

Au cours de cette SAE, j'ai été notamment impliqué dans:

- La conception du jeu de base en mode console et en mode graphique
- La rédaction des rapports, de la soutenance et l'enregistrement + montage de la vidéo
- La réalisation d'une IA au niveau conceptuel et l'implémentation de la logique derrière le Min-Max
- J'ai participé au développement des règles complexes et de la logique derrière les déplacements et conditions de fin de partie
- Interface sous JavaFX, notamment la configuration d'une partie et les boîtes d'alerte

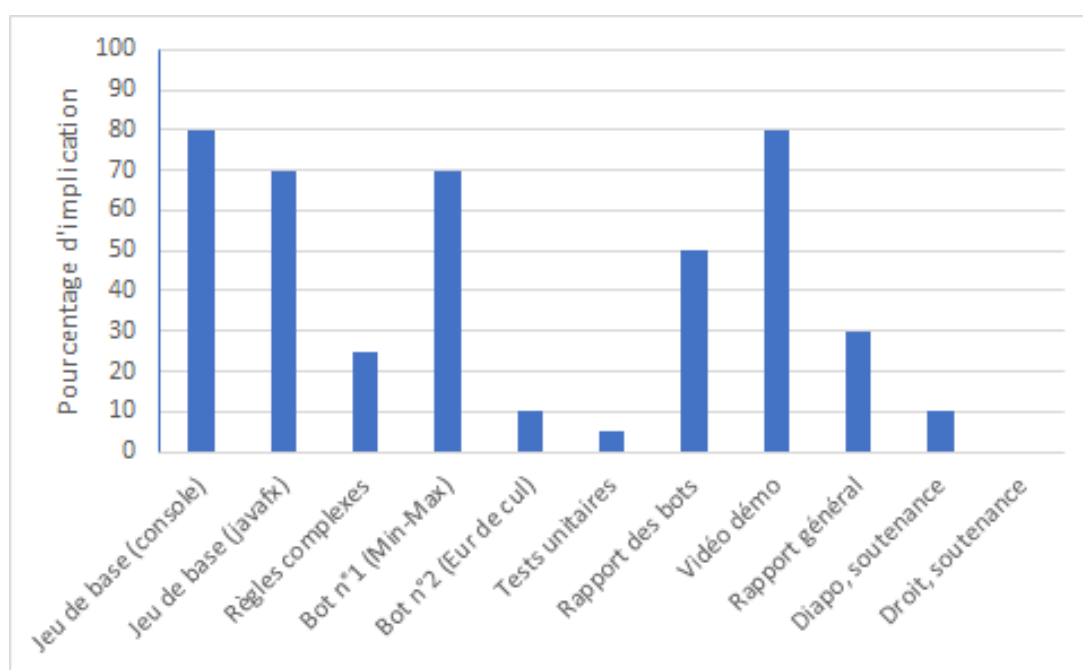


Fig. 17. – Participation de Cédric sur la SAE2.126

Conception du jeu de base console (80%)

Avec l'aide de Marvyn, nous avons réalisé à deux la logique et l'interface du jeu en mode console, incluant le système de configuration, le jeu, les déplacements et l'affichage.

Rédaction des rapports, soutenance et vidéo (170%)

Avec Marvyn, nous avons réalisé les rapports en utilisant le site *Typst* afin de pouvoir réaliser des rapports techniques de meilleur qualité. Nous avons passé beaucoup de temps à tester les bots afin de déterminer leurs capacités et limitations, tout en analysant leurs performances par le biais d'un « benchmark » comptant plus de 300k parties en *Ordinateur vs Ordinateur*.

Nous avons également réalisé la vidéo à quatre, où Marvyn, Baptiste et Hugues m'ont aidé à lister les choses à présenter dans la vidéo pendant l'enregistrement et lors du montage.

Réalisation de l'ordinateur n°1 (Mellie = Min-Max) (70%)

Dès Février, nous avons étudié différentes solutions afin de concevoir un algorithme capable de jouer au jeu. Avec Marvyn nous avons étudié tout d'abord le fonctionnement du Min-Max puis nous nous sommes penchés sur les *Méthodes Heuristiques* ([wikipedia.org](https://fr.wikipedia.org/wiki/M%C3%A9thodes_Heuristiques)), notamment à travers la lecture de diverses thèses et articles scientifiques, accompagné de nos propres expérimentations, dont la création d'un réseau neuronal. *(je reviendrais sur le réseau neuronal dans les difficultés rencontrées).*

Une fois la théorie finie, nous avons entamé le développement, principalement par moi, où Marvyn est ensuite intervenu pour les calculs heuristiques. La combinaison de nos compétences nous a alors permis de réaliser un travail remarquable en peu de temps et dont nous sommes satisfaits.

Je suis également intervenu sur le second bot de Timothée, Hugues et Baptiste afin de leur permettre d'utiliser diverses méthodes et algorithmes utilisés par le Min-Max pour leur bot.

Développement des règles complexes (25%)

Je suis intervenu aux cotés de Marvyn et Baptiste afin de régler divers problèmes dans les règles complexes, notamment au niveau du Oshi et de la détection des blocages ainsi que des bugs liés aux contrôleurs.

Interface sous JavaFX (70%)

Avec Marvyn, j'ai réalisé diverses maquettes liées à l'interface de configuration. Nous avons également ré-implémenté la totalité du jeu sous le mode graphique.

7.2.5.1.2 Connaissances et compétences acquises

Au cours de ce projet, j'ai pu appliquer diverses connaissances et compétences techniques.

Tout d'abord, d'un point de vue informatique, j'ai pu appliquer le modèle MVC à un jeu, là où j'ai plutôt l'habitude de concevoir des jeux sous le paradigme fonctionnel et *ECS (Entity Component System)*. J'ai également pu découvrir l'algorithme du Min-Max et un ensemble de connaissances et d'applications autour.

Lors de mes recherches sur les différentes façon de concevoir un bot, j'ai pu expérimenter avec les réseaux neuronaux, sur lesquels je n'avait que très peu de connaissances auparavant. J'ai pu réaliser divers réseaux, dont un qui permet de savoir si une couleur tend vers le noir ou blanc, ou encore un réseau de ma création contre lequel j'ai joué au morpion, et qui m'a bien battu je dois l'admettre. Ces expérimentations, accompagnées de leur lot de thèses et articles scientifiques, ont été très enrichissantes pour moi et m'ont permis de mieux percevoir et comprendre tout ces systèmes « intelligents ». Si je n'ai pas pu appliquer ces découvertes à ce projet, je compte bien y dédier du temps personnel afin de l'intégrer à mes projets personnels et aux prochaines SAE au cours de cette formation.

Finalement, d'un point de vue organisationnel et relationnel, j'ai pu véritablement comprendre l'importance d'avoir une équipe, où chaque membre a des facilités dans certains domaines et va pouvoir y appliquer ses connaissances dans certaines tâches précises du projet.

7.2.5.1.3 Difficultés majeures rencontrées

Dans les connaissances et compétences acquises (*Chapitre 7.2.5.1.2*), j'ai mentionné que je n'ai pas pu appliquer mes nouvelles connaissances sur les réseaux neuronaux. Nous avons tenté de l'implémenter dans le Min-Max, cependant j'ai rencontré des problèmes sur l'entraînement, notamment car je manquait de données utilisables. J'ai également rencontré un très gros problème sur la structure du réseau. En effet, envoyer les coordonnées des 16 pions à chaque calcul a été très problématique, et le réseau me renvoyait toujours le même nombre. Avec plus de temps, j'aurais pu résoudre ce problème, cependant le manque de temps accompagné d'une série de bugs gênants nous a contraint à abandonner cette stratégie et se contenter d'un Min-Max avec un algorithme *déterministe*.

Une autre difficulté a été le framework **boardifier**, où nous avons eu de très gros problèmes sur les coordonnées. En effet, le framework utilise le format (ligne, colonne) cependant cela se traduit par (y, x), nous avons eu énormément de problèmes à cause de ce détail, où il nous a fallu inverser les coordonnées à tout va, générant son lot de bugs très gênants.

Pour finir, sur le plan organisationnel, nous avons eu de très grandes difficultés sur l'organisation du travail. En effet, certains membres n'ont pas souhaité y consacrer au moins une heure pendant les vacances d'avril, provoquant des retards importants sur le planning. Ayant déjà conduit de nombreux projets personnels avec d'autres personnes, j'avais alerté à répétition sur le manque de temps chronique auquel nous allions faire face. Après un bon mois à me répéter, j'ai décidé de prendre les devants et commencer, ce qui nous a permis d'avoir une avance très confortable à la reprise des cours, nous permettant alors de consacrer plus de temps aux ordinateurs, rapports et à la correction de bugs.

Cette problématique a fini par être réglée en interne pour le mode graphique, où nous nous sommes réellement coordonnés et avons pu être plus productifs.

7.2.5.2 Conclusion

Au cours de cette SAE, j'ai été impliqué dans de nombreux aspects du projet, notamment la conception du jeu, la rédaction des rapports, la réalisation de l'IA et le développement de l'interface. J'ai pu mettre à profit mes compétences en programmation et en conception de jeux, et j'ai également acquis de nouvelles connaissances en intelligence artificielle et en développement d'interfaces graphiques.

Cette SAE m'a permis d'apprendre beaucoup de choses, tant sur le plan technique que personnel. J'ai pu approfondir mes connaissances en programmation et en conception de jeux, et j'ai également découvert de nouveaux concepts tels que l'algorithme du Min-Max et les réseaux neuronaux. J'ai également appris l'importance du travail en équipe et de la communication.

Je suis globalement satisfait de mon expérience sur cette SAE. J'ai beaucoup appris et j'ai pu réaliser un projet dont je suis fier.

J'aimerais continuer à apprendre et à me perfectionner dans le domaine du développement de jeux vidéo. Je suis également intéressé par l'exploration de l'intelligence artificielle et des réseaux neuronaux.

7.3 Figures



Fig. 18. – Une situation où la tour rose du joueur 1 est bloquée.



Fig. 19. – Une situation de oshi par un Triple-Sumo



Fig. 20. – Une situation de oshi par un Triple-Sumo