

TRIFLE – KAMISADO

TEAM CHAMMALOW



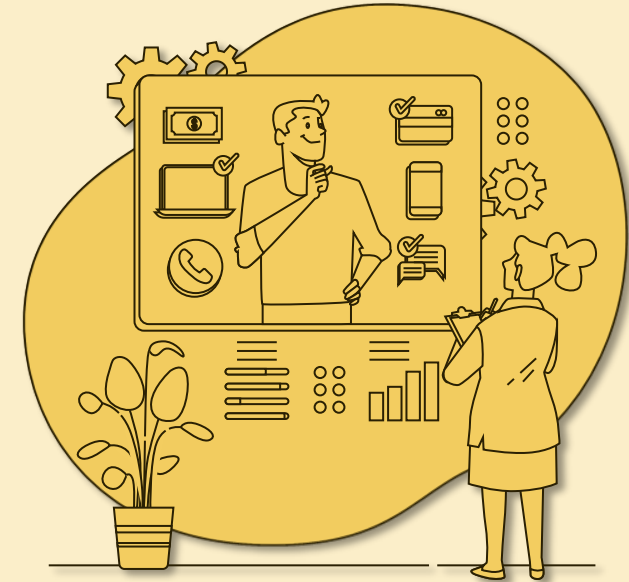
BAPTISTE DULIEUX – CÉDRIC COLIN – HUGUES
ESTRADE – MARVYN LEVIN – TIMOTHÉE MEYER

SOMMAIRE

Introduction

- **L'utilisation de l'IA par notre jeu**
 - Principales techniques
 - Son efficacité
- **L'organisation du travail**
 - Difficultés et impacts sur le projet
 - Bilan de compétence
- **La démonstration**
 - Configuration d'une partie
 - Partie longue avec IA

Conclusion





INTRODUCTION

**MOTIVATIONS – RÈGLES DU KAMISADO
– CE QUI A ÉTÉ FAITS – RESTE À FAIRE
ET CE QUI A ÉTÉ AJOUTÉ**

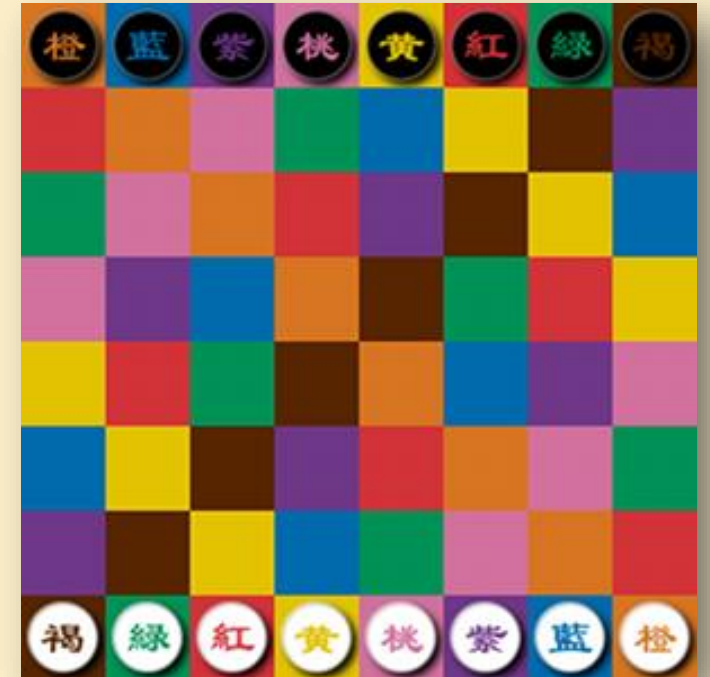
PRÉSENTATION DU KAMISADO

Un peu d'histoire :

- Créer dans les années 1970
- Design par Peter Burley

Kamisado : règles principales

- 2 joueurs
- Objectif : amener une tour sur la rangée des tours adverses
- Déplacements des tours basés sur des règles spécifiques
 - En fonction de la couleur de la case du coup précédent
 - Sumos (dents de dragon)



	Sumo	Double Sumo	Triple Sumo	Quadruple Sumo
				
Movement range	5	3	1	-
Number of towers that can be pushed with a Sumo push	1	2	3	-
Points	1	3	7	15

MOTIVATIONS POUR CHOISIR LE JEU

Kamisado : jeu stratégique complexe

- Choix du jeu Kamisado
 - pour sa complexité stratégique
 - et la richesse de ses règles.
- Environnement ludique
- Et défis techniques.



CE QUI ...

A ÉTÉ FAIT

- Déplacement basique du joueur
- Règles du Oshi et sumos
- Placement des pions de gauche à droite ou de droite à gauche
- Animations des tours
- Configuration d'une partie
- Relancer une partie après avoir terminé
- Jouer dans les 3 modes:
 - Humain vs Humain,
 - Humain vs Bot,
 - Bot vs Bot

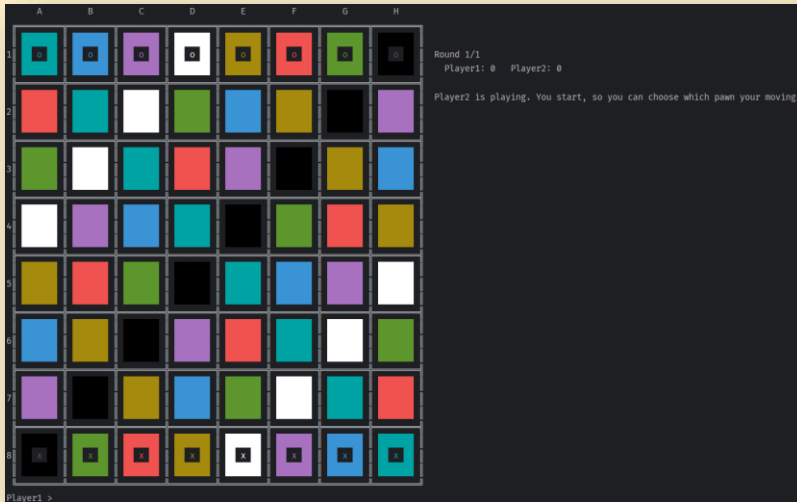
RESTE À FAIRE

- Designer les cases où on peut jouer (70/100%)
- Bot n°2 à réimplémenter (60/100%)
- Bot n°1 revoir le calcul des poids (80/100%)
- Debugging (70/100%)

CE QUI A ÉTÉ RAJOUTÉ

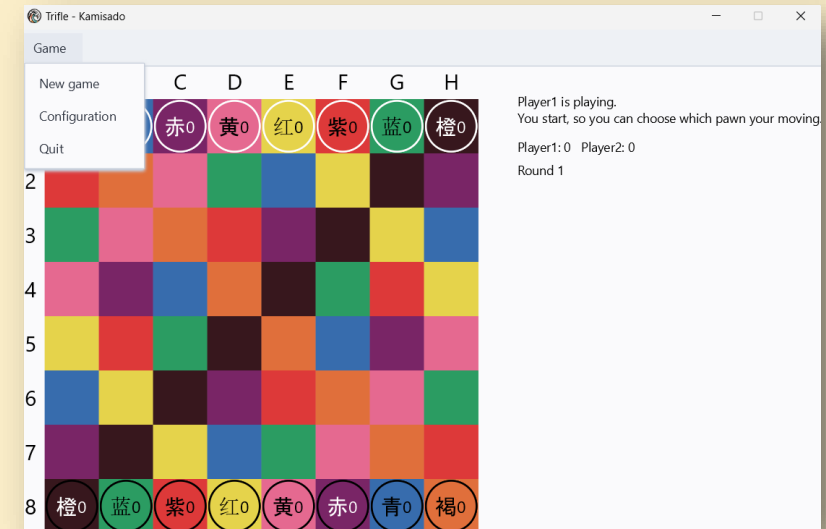
CONSOLE

- Plateau plus lisible avec des cases plus grosses (case 7x3 caractères)
- Tours centrées sur les cases (position 4, 2)
- Historique des tours jouées sur le plateau
- Nombre de parties gagné pour tel joueur



GRAPHIQUE

- Menu de direction (jouer – configurer – quitter)
- Texte pour dire quelle tour on doit jouer
- Nombre de points gagné lors d'une partie
- Musique d'ambiance et sons de déplacements





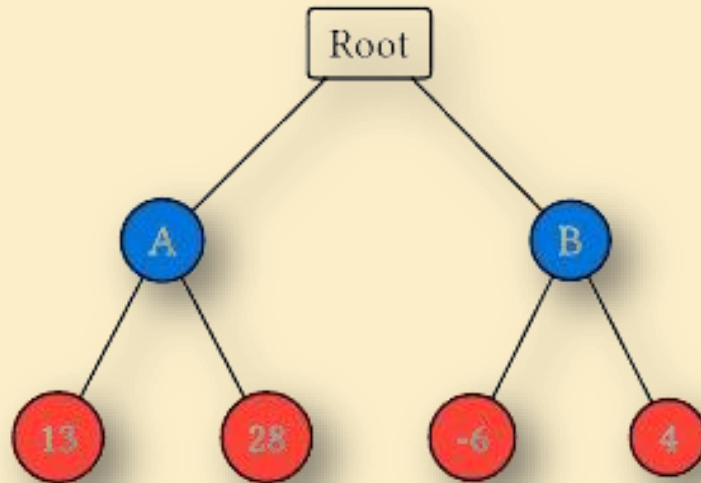
UTILISATION DE L'IA DANS LE JEU

PRINCIPALES TECHNIQUES – LEUR
EFFICACITÉ

PRINCIPALES TECHNIQUES - MÉLIE

Min-Max : algorithme déterministe

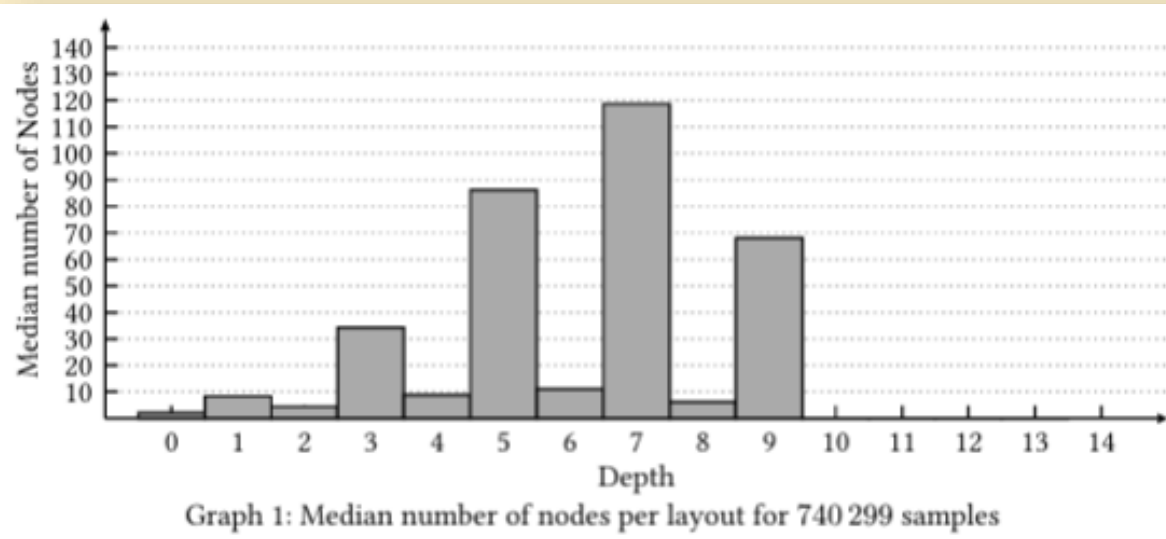
- Repose sur théorie des jeux
- Utilisé par « Deep Blue » (échecs)
- Simple à implémenter



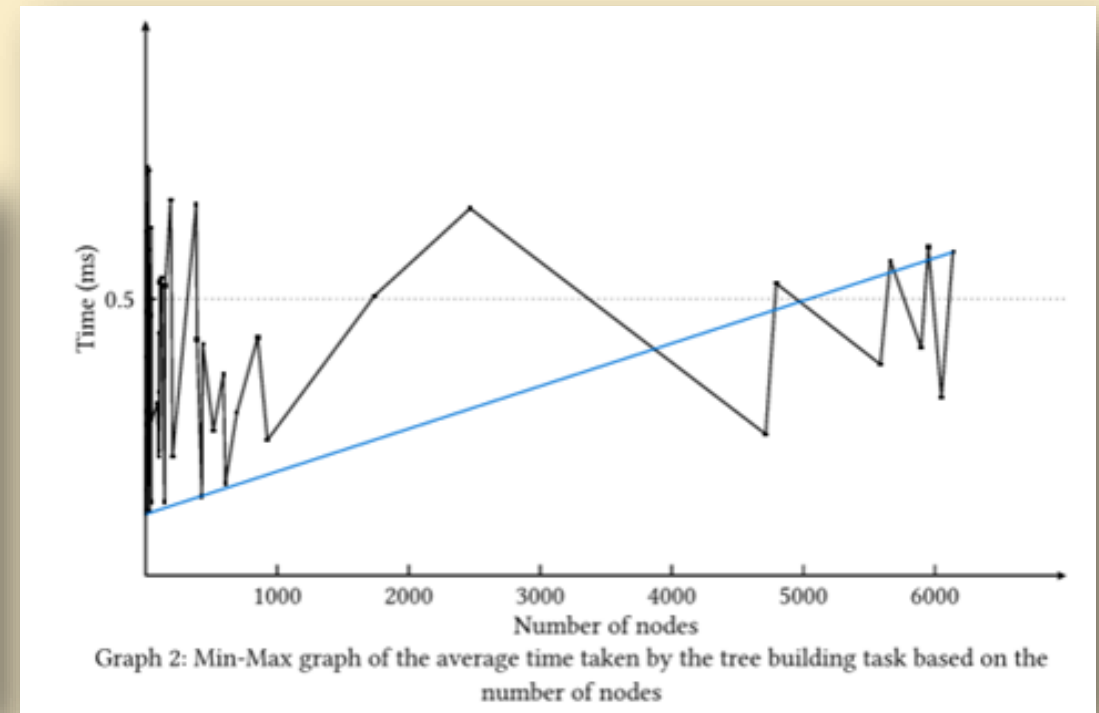
SON EFFICACITÉ - MÉLIE

Min-Max : algorithme déterministe

- Problème utilisation mémoire
- Gagne aucune partie (0,1%)



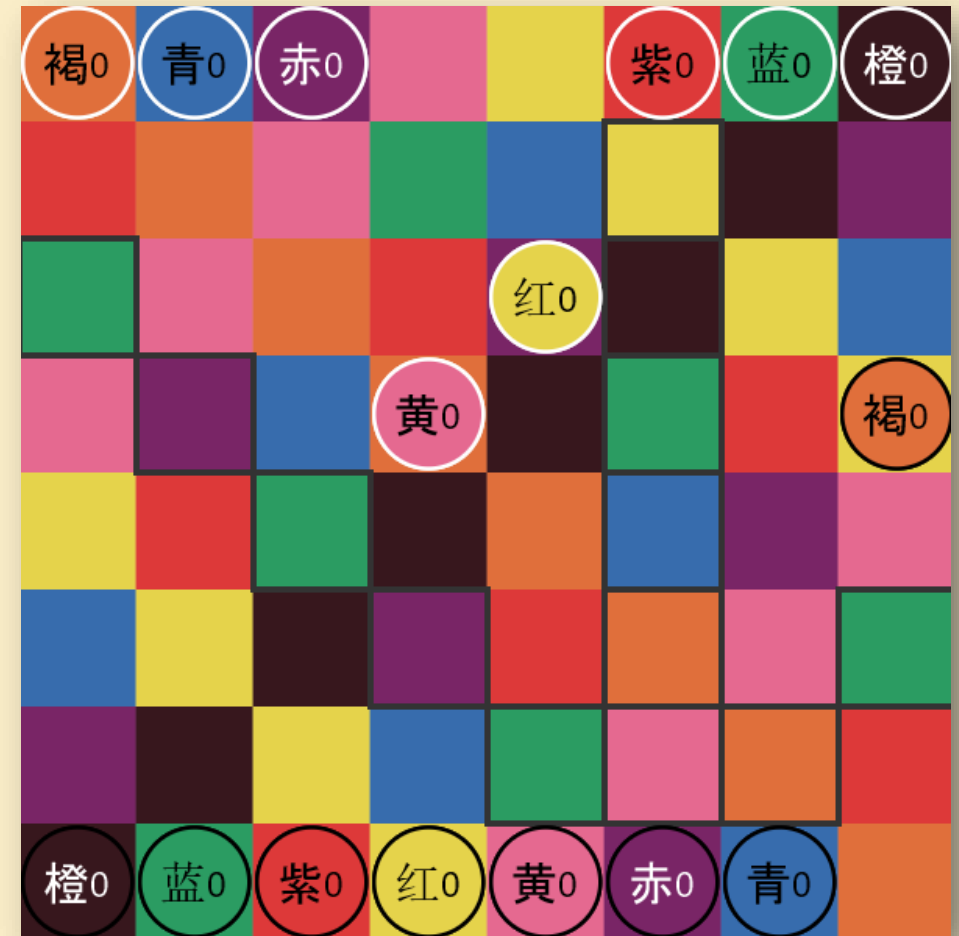
~348 coups par instance



PRINCIPALES TECHNIQUES - EUREDEKU

Bot : algorithme déterministe

- Plusieurs algorithmes testés
- Choix arrêté sur principe suivant :
 1. Chercher un coup gagnant
 2. Chercher à avancer PRESQUE le plus loin possible
 3. Chercher à se rapprocher du centre
 4. Enfin, éviter les coups perdants



SON EFFICACITÉ - EUREDEKU

Bot : algorithme déterministe

- Très efficace pour le nombre d'opérations réalisées
- Fonctionne uniquement si utilisateur ne connaît pas l'algorithme
- Estimation de victoire (face à un joueur innocent)
 - estimé à 40% (selon nos expérimentations)

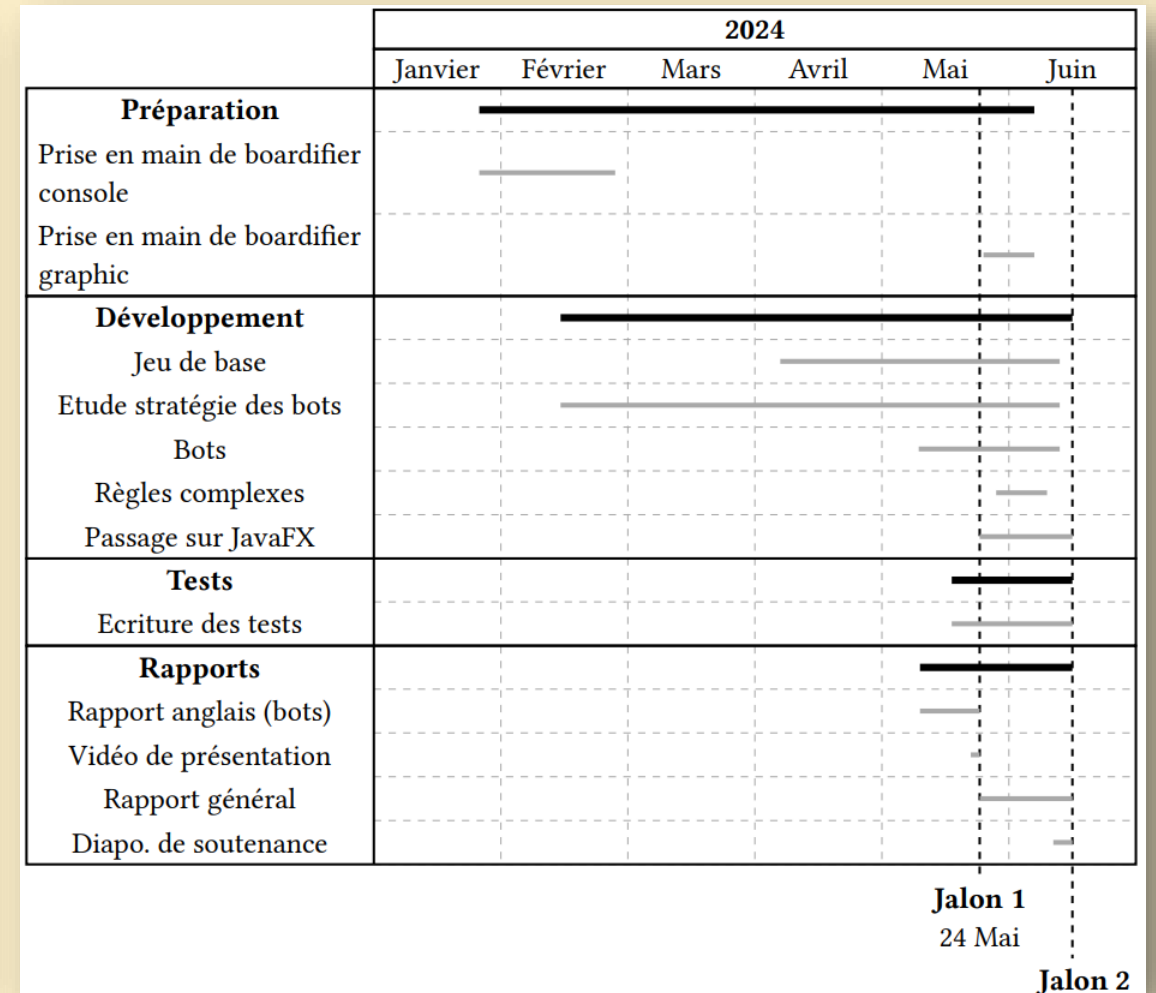
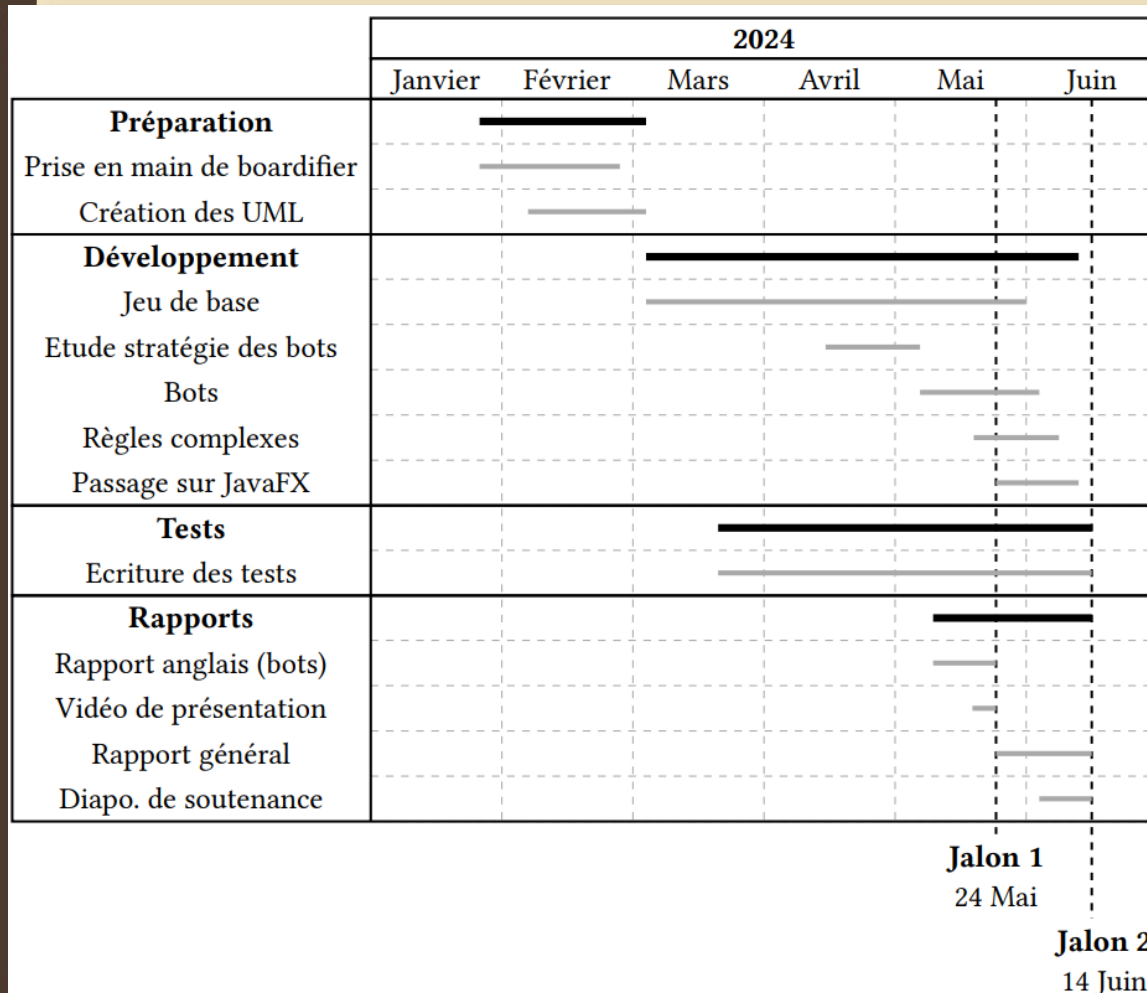




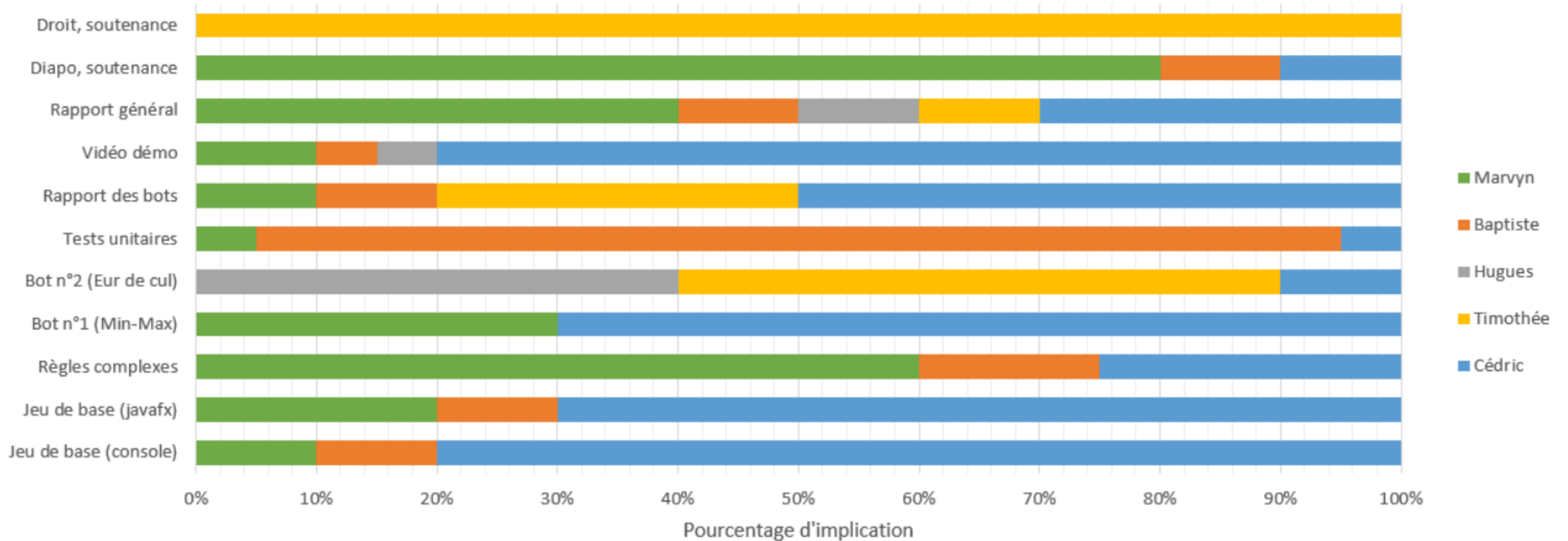
ORGANISATION DU TRAVAIL

DIAGRAMME DE GANTT – IMPACT DE
CES DIFFICULTÉS – BILAN DE
COMPÉTENCES

PLANNING PRÉVISIONNEL - RÉEL



TÂCHES PAR MEMBRE



DIFFICULTÉS

CAUSES

- Mauvaise organisation au début
- Prise en main de Boardifier
- Système de coordonnées
- Mauvaise entente sur l'organisation > réunions

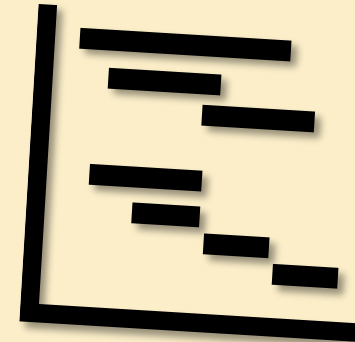
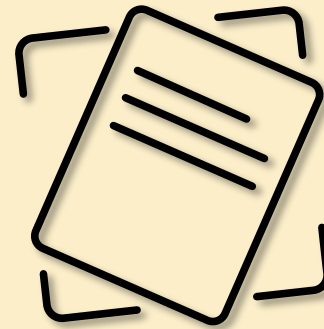
IMPACTS

- Manque de temps (négatif)
- Contrainte dans l'implémentation (négatif)
- Répartition des tâches dans le temps (positif)

BILAN DE COMPÉTENCE

On a développé nos compétences en :

- Java :
 - En **Poo** (Programmation Orientée Objet)
 - En **JavaFx** (Programmation graphique)
- Communication :
 - En rédaction
 - En création de document
- Organisation :
 - A l'aide de réunions
 - A l'aide de diagramme de Gantt
 - Utilisation de Git (commits, branches, merge requests)





DÉMONSTRATION

CE QUI FONCTIONNE – CE QUI NE
FONCTIONNE PAS – CONFIGURATION
& PARTIES



CONCLUSION

**AMÉLIORATIONS POSSIBLES – TEMPS
SUPPLÉMENTAIRE – QUE CHANGER SI
PROJET À REFAIRE**

AMÉLIORATIONS POSSIBLES & TEMPS

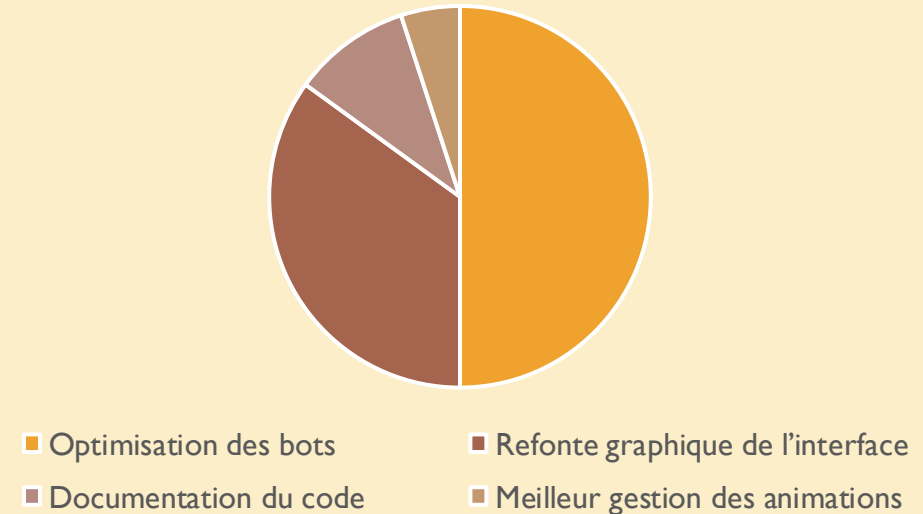
Les choses que l'on pourrait améliorer :

- Optimisation des bots (50%)
- Refonte graphique de l'interface (35%)
- Documentation du code (15%)
- Meilleur gestion des animations (5%)

Le temps nécessaire serait de :

- Environ 1 mois de plus

Temps en fonction des améliorations possibles



QUE CHANGER SI PROJET À REFAIRE

Changement sur quelques points :

- L'organisation :
 - Plus de réunions
 - Plus de communication
- L'ajout de la documentation du Framework
 - Boardifier
- La gestion des poids du Min-Max

