

HLIN405 - Projets de Programmation de L2

Blob Wars

Allouch Yanis Roux Jérémie Villaroya Kévin

2018 - 2019

Université de Montpellier - Faculté des Sciences



1 Introduction

- Règles du jeu
- Mécaniques de jeu

2 Démonstration

3 Développement

- Structure globale
- Fichiers de configuration
- Musique et bruitages

4 Intelligence artificielle

- Fonction d'évaluation
- Algorithme MinMax
- Élagage Alpha/Bêta de l'algorithme MinMax

5 Conclusion

- Compétences mises en œuvre et logiciels utilisés
- Diagramme de Gantt

6 Ouverture

- Comparaison de notre code avec l'exemple
- Implémentations possibles

Introduction : Règles du jeu

Blob Wars, un jeu de plateau de type Othello

Condition de victoire : Avoir le plus de pions à la fin de la partie

Mécanique principale : Se déplacer ou se cloner sur une case adjacente à un pion ennemi pour le convertir

Fin de la partie : Plateau rempli ou $n - 1$ joueurs ne peuvent plus bouger (soit n le nombre de joueurs au total)

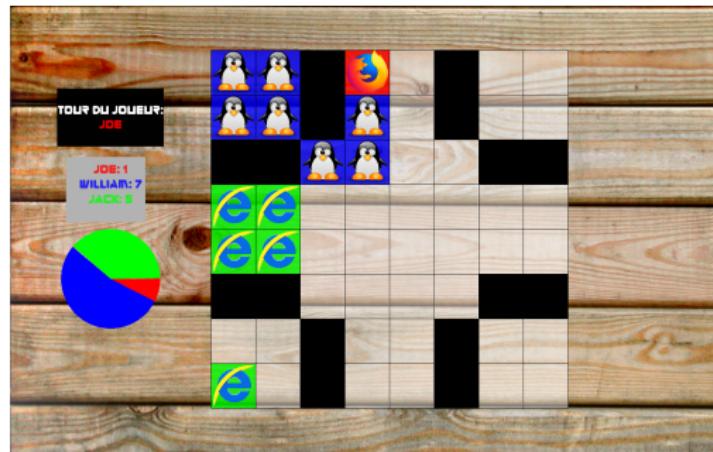


Figure 1 – Capture d'écran d'une partie avec notre application Blob Wars

Introduction : Mécaniques de jeu

Analysons les 4 types de coups possibles du pingouin situé au centre de l'écran.

Mouvements possibles

- 1 : Simple déplacement
- 2 : Déplacement et attaque
- 3 : Simple clonage
- 4 : Clonage et attaque

Zones

Grise claire : Déplacement
Grise foncée : Clonage

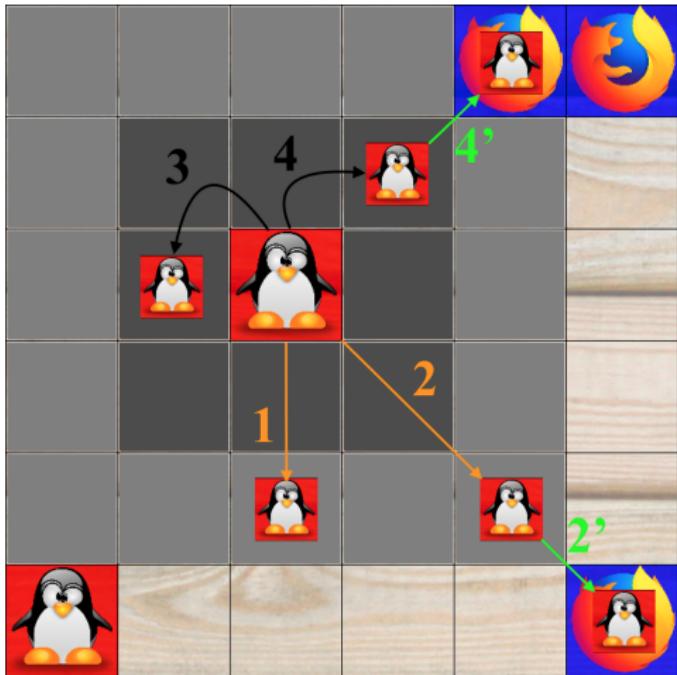


Figure 2 – 4 types de coups possibles

Démonstration

Voici une démonstration de l'application Blob Wars (développée en Java) que nous avons conçue.



Figure 3 – Logo de notre application Blob Wars

Dans notre version du Blob Wars, les "blobs" sont des logos de moteurs de recherche, de systèmes d'exploitations ou de marques.

Développement : Structure globale

Structure du programme

La fenêtre de jeu (interface graphique) dialogue avec le moteur du jeu et le jeu en lui-même.

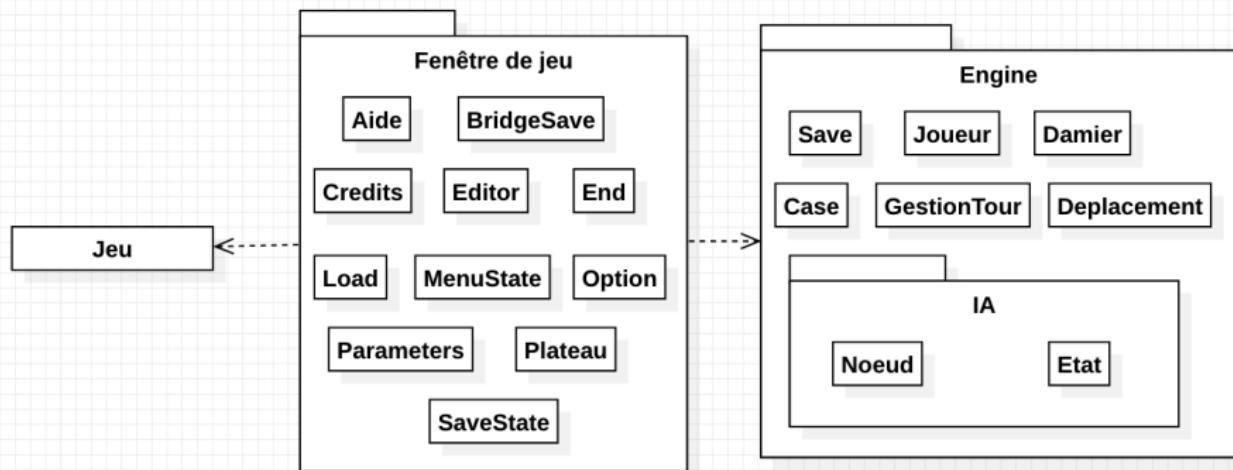


Figure 4 – Diagramme de classes simplifié global

Développement : Fichiers de configuration

L'utilisateur peut générer ou charger des fichiers de configuration afin de sauvegarder ou charger des parties.

```

1 2019/04/02 00:23:58 ) Date et heure
2
3 4) Nombre de joueurs
4
5 1 tres_facile Joe red firefox 2
6 2 joueur William blue ie 5
7 3 moyen Jack green windows 6
8 4 joueur Averell yellow chrome 4
9 2) Index du joueur Propriétés joueurs :
10 qui doit jouer
11 - index
12 - niveau IA
13 ou nom joueur
14 - équipe
15 - avatar
16 - index avatar
17 Plateau :
18 - numéro = index joueur
19 - 0 = case vide
20 - X = case inaccessible
  
```

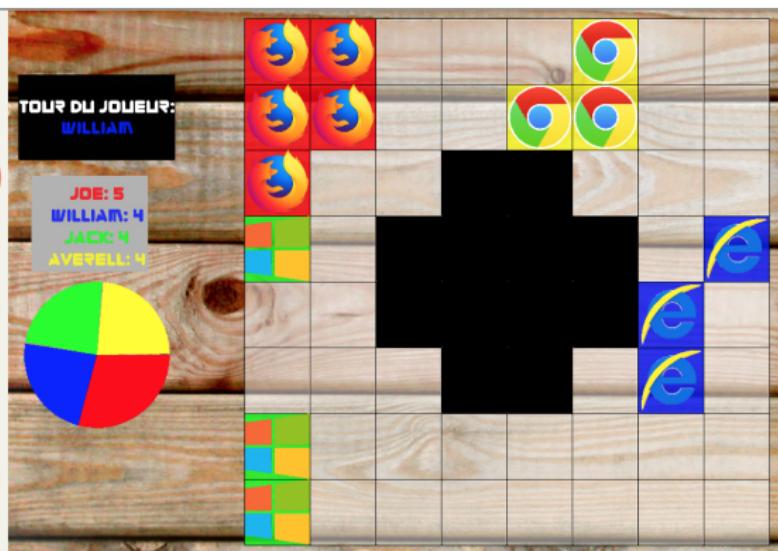


Figure 5 – Exemple d'un fichier de configuration et de la partie correspondante

Développement : Musique et bruitages

Garage Band

Il s'agit d'un logiciel de MAO (Musique Assistée par Ordinateur) avec lequel nous avons composé la musique et fabriqué les bruitages du jeu.



Figure 6 – Aperçu dans Garage Band de la bande originale du jeu Blob Wars

Intelligence artificielle : Fonction d'évaluation

Eval()

La fonction attribue une valeur entière à un état de jeu pour un joueur donné.

Gains et pertes calculés

- $+1$ = pour chaque "blob" créé
- -1 = pour un de ses "blob" détruit
- $+100$ = lorsqu'on peut atteindre la victoire
- -100 = lorsque le joueur en question est éliminé

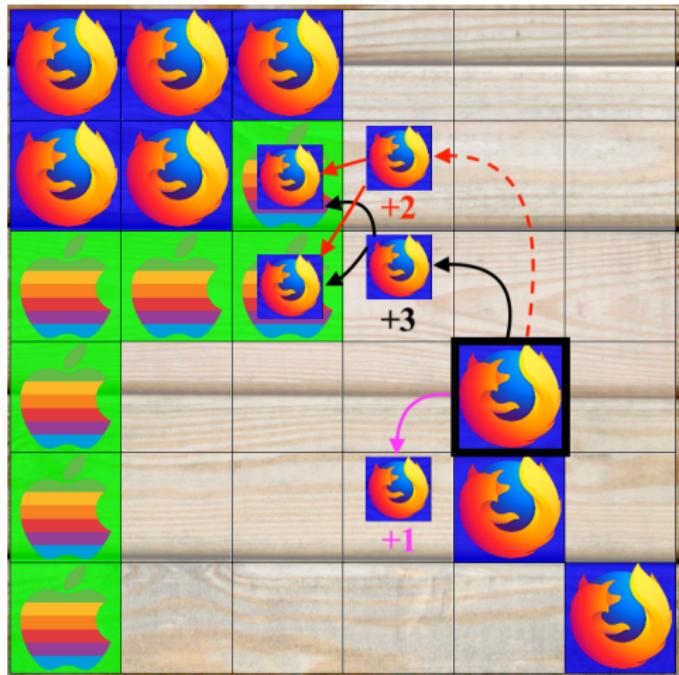


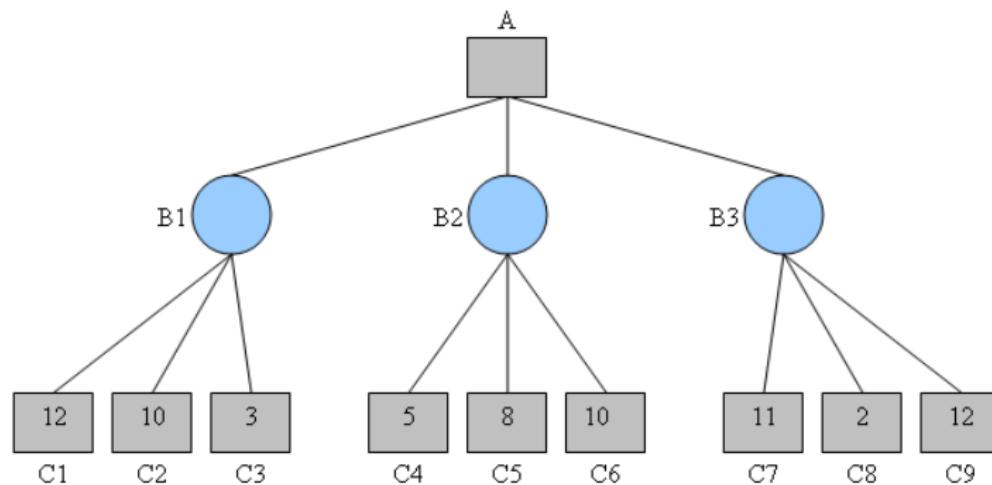
Figure 7 – 3 coups possibles et valeurs associées

Intelligence artificielle : Algorithme MinMax

MinMax

Cet algorithme renvoie le meilleur coup que le joueur peut effectuer (en créant l'arbre des possibilités par récursion et le remontant en appliquant le max ou min si c'est le tour d'un ennemi ou d'un allié).

Figure 8 – Exemple d'un arbre étiqueté avec les valeurs d'un MinMax (1/3) (Wikipédia)

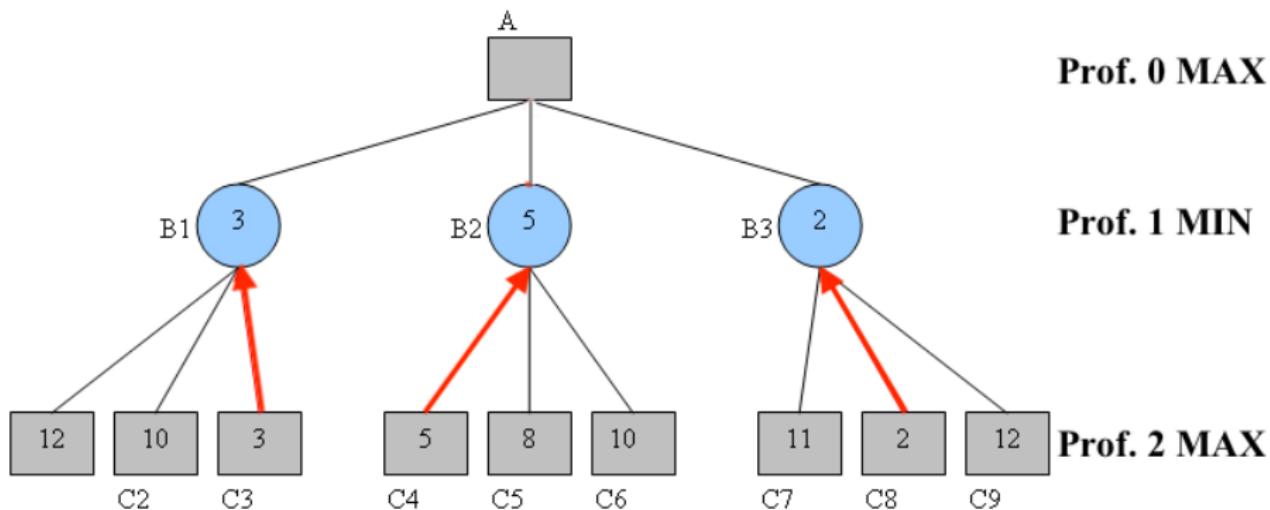


Intelligence artificielle : Algorithme MinMax

Valeurs des nœuds B

Les nœuds B reçoivent chacun la valeur minimum stockée dans leurs fils C.

Figure 9 – Exemple d'un arbre étiqueté avec les valeurs d'un MinMax (2/3) (Wikipédia)

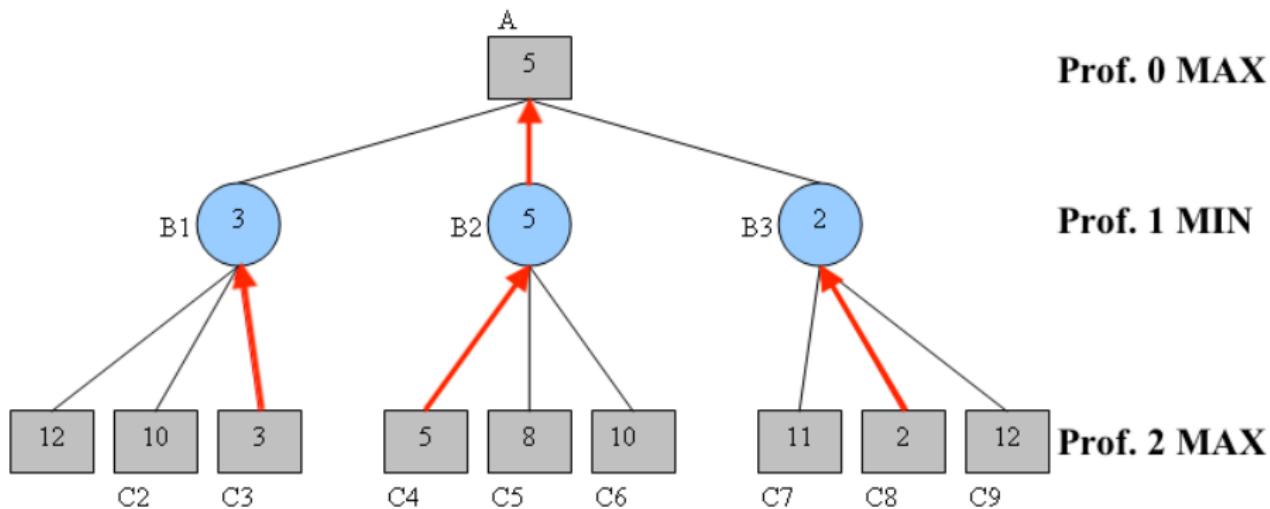


Intelligence artificielle : Algorithme MinMax

Valeur du nœud A

Pour déterminer la valeur du nœud A, on choisit la valeur maximum de l'ensemble des nœuds B.

Figure 10 – Exemple d'un arbre étiqueté avec les valeurs d'un MinMax (3/3) (Wikipédia)



Intelligence artificielle : Algorithme MinMax

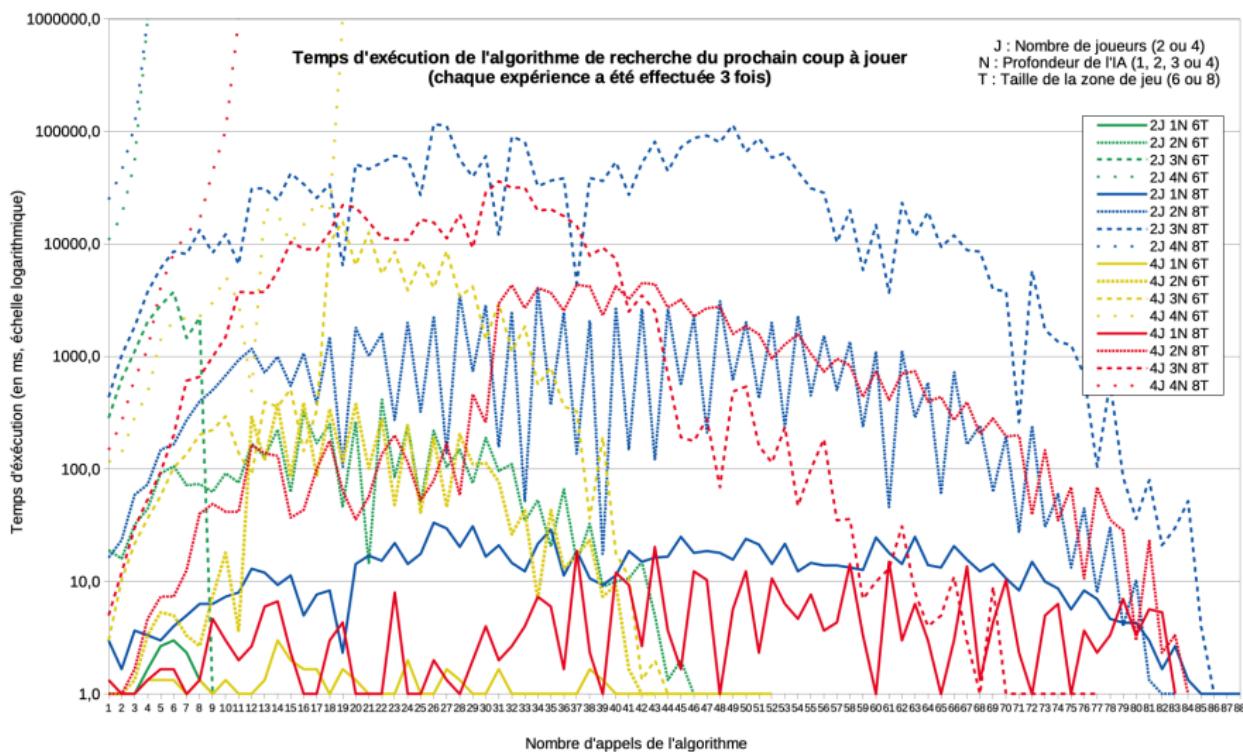


Figure 11 – Calculs de temps d'exécution de l'algorithme MinMax

Intelligence artificielle : Élagage Alpha/Bêta de l'algorithme MinMax

Élagage Alpha/Bêta

L'objectif de cette optimisation est de couper (ou élaguer) les branches de notre arbre pour avoir moins de contenu à explorer. Qui dit moins de contenu à explorer dit moins de calculs à effectuer. Cette opération est communément appelée un élagage.

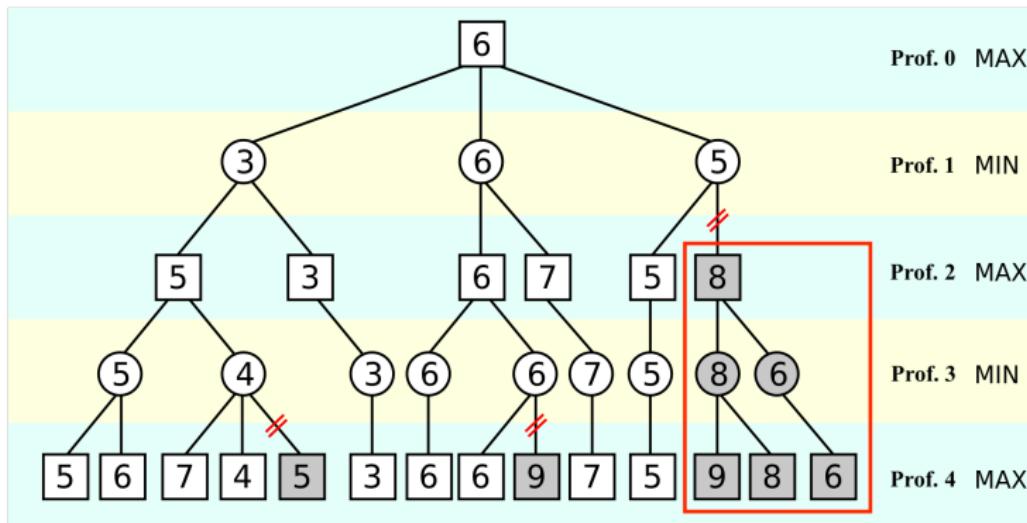


Figure 12 – Arbre étiqueté avec les valeurs d'un MinMax avec élagage Alpha/Bêta (Wikipédia)

Intelligence artificielle : Élagage Alpha/Bêta de l'algorithme MinMax

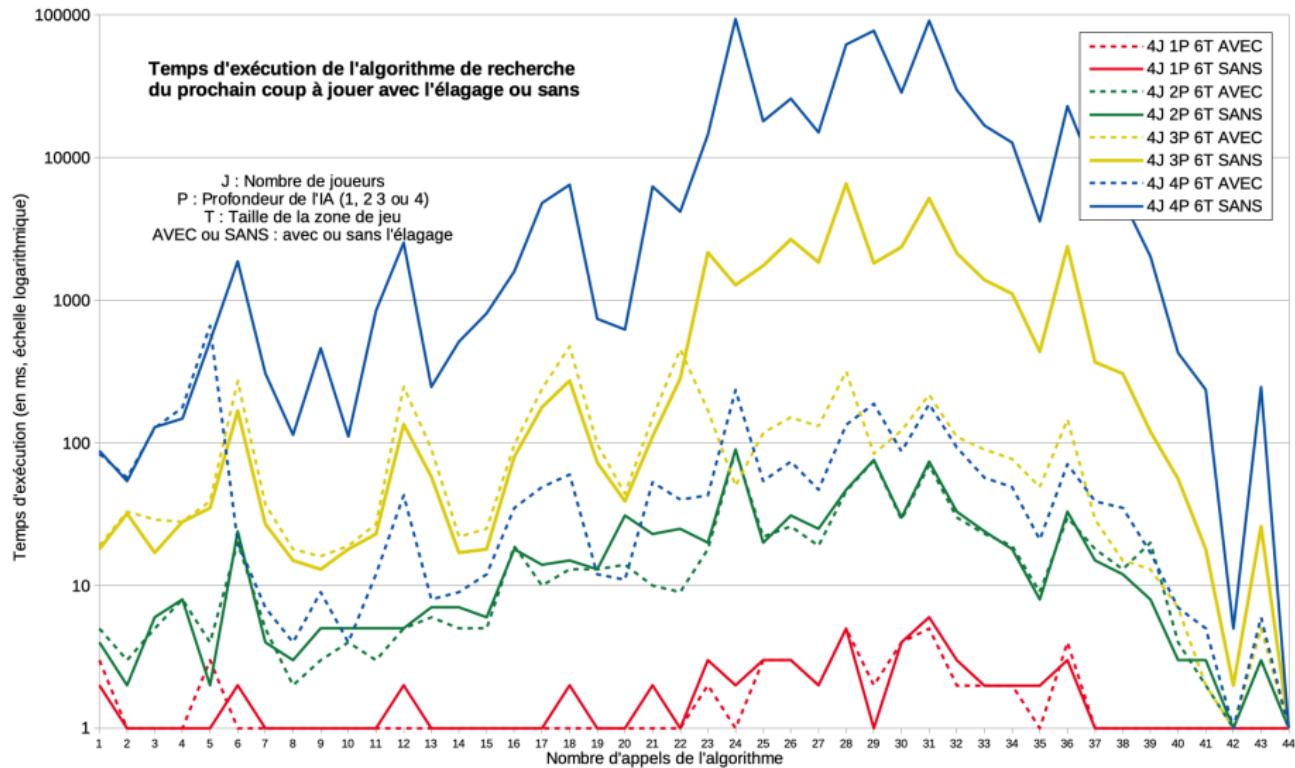


Figure 13 – Calculs de temps d'exécution de l'algorithme MinMax avec ou sans élagage

Conclusion : Compétences mises en œuvre et logiciels utilisés



GitLab



GarageBand



- Savoir s'organiser et se répartir le travail en groupe
- Utiliser des outils de travail adaptés (Librairie Slick2D, Gantt, Git, L^AT_EX, Eclipse, ...)
- Mettre en pratique les connaissances des modules de L1 et L2
- Favoriser l'open-source et le gratuit
- Apprendre des autres et de leurs expériences

Conclusion : Diagramme de Gantt

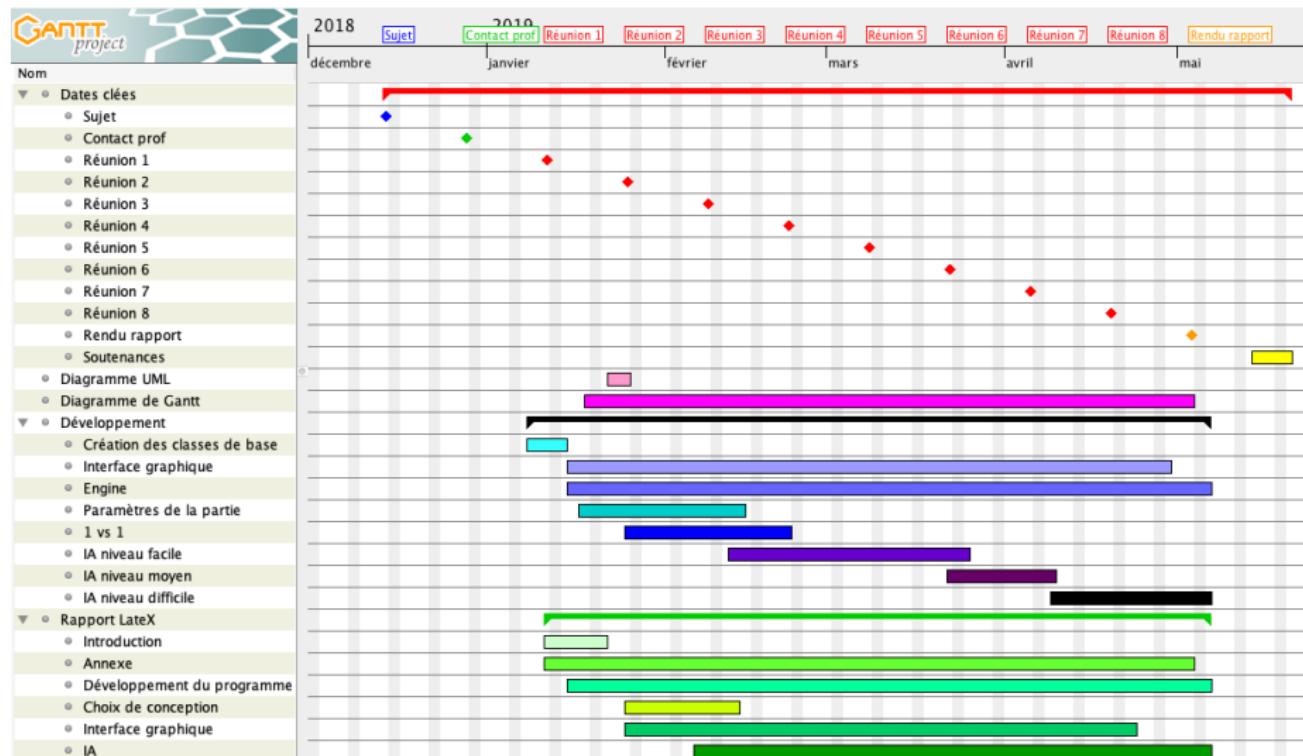


Figure 14 – Diagramme de Gantt de Blob Wars

Ouverture : Comparaison de notre code avec l'exemple

Récupération du code du Blob Wars en ligne

À l'aide du logiciel *Sothink SWF Decompiler*, nous avons récupéré l'algorithme du jeu donné dans le sujet (Blob Wars en ligne). Il s'agit d'un enchaînement de boucles conditionnelles (*If/Else*).



Figure 15 – Capture d'écran du Blob Wars en ligne donné en exemple

Ouverture : Implémentations possibles

- Développement pour le web et/ou téléphone
- Pouvoir customiser l'apparence du jeu
- Ajouter des animations pour les pions
- Pouvoir sauvegarder pendant le calcul des noeuds
- Un bot qui pourrait jouer avec notre IA et contre l'IA du Blob Wars donné en exemple
- Ajouter d'autres fonctions d'évaluations

Blob Wars

Allouch Yanis Roux Jérémie Villaroya Kévin

2018 - 2019

Université de Montpellier - Faculté des Sciences

