

# Rapport de Projet : Awélé

Jordan REPOVY et Julien HOFFMANN



## Table des matières

Introduction .....	3
Préparation .....	4
Familiarisation avec l'Awélé .....	4
Début de la programmation .....	4
Passage à l'ajout de données .....	6
Conclusion .....	7

## Introduction

L'Awélé est un jeu traditionnel Africain de la famille Mancala dont le but est de déplacer les graines contenues dans les trous du plateau de jeu de manière à en capturer le plus possible. Ce jeu est pure stratégie et ne laisse aucune place au hasard, l'issue du jeu dépend à 100% des choix faits par le joueur.



Le but de ce projet est de mettre en œuvre des techniques de fouilles de données vu en cours (et autres trouvés sur internet) afin de créer un bot capable de jouer à l'Awélé. Ce bot sera ensuite mis à rude épreuve lors d'un tournoi rassemblant l'ensemble des bots programmés par les autres groupes.

# Préparation

## Familiarisation avec l'Awélé

Les règles de l'Awélé étant simple mais difficile à appréhender en les lisant simplement, nous nous sommes d'abord initié à l'Awélé sur le site <http://s.helan.free.fr/awale/lejeu/jouer/awale.html> afin de se familiariser avec le jeu.

Après quelques parties d'Awélé sur les différents sites nous avons cherché les différentes stratégies du jeu :

<http://www.myriad-online.com/resources/docs/awale/francais/strategy.htm>  
<http://s.helan.free.fr/awale/conseils/>  
<http://abstractstrategygames.blogspot.fr/2010/10/awale.html>  
[http://www.nongnu.org/awale/tutorial\\_fr.html](http://www.nongnu.org/awale/tutorial_fr.html)

Nous y avons appris différentes tactiques tel que le Krou ou la Fourchette et avons réfléchi à une manière de les implémentés dans RStudio.

## Début de la programmation

Nous avons d'abord pris connaissance des trois rapport fournis cependant adlkiller et slaim\_sekri nous ont parus relativement compliqué à mettre en place. En revanche, le bot NaiveBonobo nous a beaucoup plus parler avec son concept plutôt simple mais efficace de modifier les données de manière à combler les faiblesses de Naive Bayes, un algorithme puissant mais incapable de faire des corrélations entre les données.

Ensuite nous avons observé les deux bot fournis (nb1 et nb2), le programme du jeu d'Awélé et les données fournis (awele.data) afin de se familiariser avec l'environnement mis à notre disposition.

Puis nous avons créé différents bots sur la base de nb1 (puisque'il battait nb2) en remplaçant simplement l'algorithme NB par d'autres (KNN, LR, LDA, SVM, CDA, MLP ...). Nous les avons ensuite soumis au test du tournois, duquel MLP et SVMr se sont démarqué.

	CDA	KNN	LDA	LR	MLP	SVM	SVML	SVMR
CDA		0	1	0	0	1	0	1
KNN	1		0	0	1	0	0	1
LDA	0	1		1	1	0	0	0
LR	1	1	0		1	1	0	1
MLP	1	0	0	0		0	0	1
SVM	0	1	1	0	1		1	0.5
SVML	1	1	1	1	1	0		1
SVMR	0	0	1	0	0	0.5	0	
Total	4	4	4	2	5	2.5	1	5.5

Après ce tournoi nous avons commencé une tentative de modification des données fournis. Le principe de cette modification était relativement simple, nous avons implémenté sous forme de condition les différentes stratégies apprises sur internet et avons extrait un sous-set à partir de awele.data avec cette condition.

Ensuite à l'aide d'une fonction créée à cet effet nous avons calculé le pourcentage de défaite de ce sous-set et avons décidé qu'un pourcentage de défaite supérieur à 80% était suffisant pour éliminer cette condition des données originales.

En effet un coup ayant 83% de chance de perdre peut avoir un résultat gagnant dans la matrice de données ne signifie pas qu'il le sera à chaque fois et donc nous préférons le supprimer afin que cet état de jeu ayant 83% de chance de défaite ne soit pas considéré comme bon.

Après de multiples conditions testé nous avons affiné les données et testé Nb utilisant ces données contre les autres bots avec les données originales.

Total	105	42	72	39	51	135	444
% Victoire	23,65	9,46	16,22	8,78	11,49	30,41	100

On peut voir que NB est maintenant le premier du tournoi avec 30.41% de victoire.

## Passage à l'ajout de données

Nous avons vite vu que même si retirer des données fonctionnait, en ajouter fonctionnait encore mieux !

Pour cela nous nous sommes basé sur les observations faites dans le rapport de NaiveBonobo. Nous avons remarqué que ces observations étaient semblables à une partie des conditions que nous avons utilisées pour écrémé les données mais qu'au lieu de les supprimer afin que NB ne les utilise pas comme stratégie gagnant, il valait mieux ajouter des colonnes de données l'informant du plus ou moins bon ou mauvais état du jeu afin de faire le travail de corrélation qu'il ne fait pas.

Nous avons donc tenté d'implémenter ces conditions afin de les ajouter au tableau de données :

- Dans 30 cas sur 36, avoir la première case à 1 et la deuxième à une valeur inférieure à 3 conduit à une défaite
- Dans 18 cas sur 19, avoir la première et la troisième case à 0 conduit à une défaite.
- La différence entre le nombre de graines du joueur et de l'adversaire (semble influencer les chances lorsqu'elle atteint un nombre supérieur à 10 ou inférieur à -10)
- Le nombre de case à zéro d'un joueur semble augmenter grandement les chances de perdre
- Le nombre de graine prises influence les chances de gagner

Pour cela nous avons parcouru le tableau de données et pour chaque ligne nous avons créé une liste contenant les valeurs de la rangée en excluant le résultat puis nous avons calculé les différentes données mentionnées ci-dessus et les avons concaténées à cette liste.

Chaque ligne parcourue augmente la taille de la liste et à la fin de l'itération cette liste est convertie en tableau de résultats contenant toutes les nouvelles données calculées.

Nous avons tenté d'ajouter les conditions que nous avons précédemment utilisées pour écrémé les données cependant celles-ci semblaient induire NB en erreur et nous avons choisi de ne garder que les conditions que nous avons correspondant à celle mentionnées dans le rapport de NaiveBonobo.

Nous avons au cours du développement du bot procédé à de nombreux tournois incluant notre NB avec données écrémé (qui restait relativement fort puisqu'il battait tous les autres algo) et avons réussi au bout du compte à terminer premier !

## Conclusion

Ce projet nous aura beaucoup appris tant sur le développement sur RStudio que sur la fouille de données et avons été capable de mettre en pratique les connaissances vus en cours afin de mieux les appréhender. Nous avons maintenant une meilleure compréhension de ces techniques et sommes satisfait de notre travail !