



UNIVERSITÉ
DE MONTPELLIER



Rapport de projet

Profilage par essais et erreurs au poker

Réalisé par
Manuel Chataigner, Morgane Vidal
et Benjamin Commandré

Encadré par
Pierre Pompidor et Guillaume Tisserant

Pour l'obtention du master informatique
Année universitaire 2014-2015

Remerciements

Nous tenons tout d'abord à remercier Jimmy Lopez et Sébastien Beugnon pour nous avoir conseillés tout au long de notre projet.

Nous remercions aussi nos tuteurs Guillaume Tisserant et Pierre Pompidor pour nous avoir accompagnés tout au long de la réalisation du projet, en particulier pour les choix des méthodes de profilage.

Table des matières

1	Introduction	9
1.1	Généralités	9
1.2	Sujet	9
1.3	Rapide description des règles du Texas Hold'Em Poker	10
1.3.1	Mise en place du jeu	10
1.3.2	Préflop	10
1.3.3	Flop	10
1.3.4	Turn	10
1.3.5	River	11
1.3.6	Les combinaisons	11
1.4	Cahier des charges	14
1.4.1	Problématique	14
1.4.2	Fonctionnalités obligatoires	14
1.4.3	Fonctionnalités optionnelles	15
1.4.4	Spécifications techniques	15
1.4.5	Organisation prévisionnelle	15
2	Organisation du projet	17
2.1	Organisation du travail	17
2.2	Choix des outils de travail	17
2.2.1	Gestionnaire de versions	17
2.2.2	Création de l'interface graphique	17
2.2.3	Rédaction et mise en page du rapport	17
2.2.4	Normes de programmation	17
2.2.5	Langage	18
3	Analyse du projet	19
3.1	Système de jeu	19
3.2	Profilage	19
3.2.1	Choix d'enregistrement des données	19
3.3	Formule de calcul du taux de rationalité d'un joueur	19
3.3.1	Formule de calcul du taux d'agressivité d'un joueur	20
3.3.2	Formule de calcul du taux de bluff d'un joueur	21
3.3.3	Formule de calcul du taux de passivité d'un joueur	21

3.4	Profilage statique	21
3.5	Réutilisation des résultats du profilage	21
3.6	Profilage dynamique	21
3.7	Réutilisation des résultats du profilage	21
4	Mise en place des méthodes de profilage	22
4.1	Calibrage des intelligences artificielles	22
4.2	Profilage statique	23
4.2.1	Scénarios de tests	23
4.2.2	Établissement des profils attendus	23
4.2.3	Réutilisation des résultats du profilage	23
4.3	Profilage dynamique	23
4.3.1	Scénarios de tests	23
4.3.2	Établissement des profils attendus	23
4.3.3	Réutilisation des résultats du profilage	23
5	Analyse des résultats obtenus	24
5.1	Profilage statique	24
5.1.1	Scénarios de tests	24
5.1.2	Analyse des gains de parties	24
5.2	Profilage dynamique	24
5.2.1	Analyse des gains de parties	24
6	Perspectives et conclusion	25
6.1	Perspectives d'amélioration du profilage	25
6.2	Profilage obtenu	25
A	Test annexe	27

Liste des figures

1.1	Quinte Flush Royale	11
1.2	Quinte Flush	11
1.3	Carré	12
1.4	Full	12
1.5	Couleur	12
1.6	Suite	13
1.7	Brelan	13
1.8	Double Paire	13
1.9	Paire	14
1.10	Carte Haute	14

Glossaire

Les termes définis dans ce glossaire sont identifiables dans le corps du texte au moyen d'un astérisque ().*

Bluff : Le bluff est une technique de jeu qui consiste à jouer comme si l'on possédait un jeu différent de celui détenu en réalité.

Rationalité : La rationalité appliquée au poker stipule que toutes les actions d'un joueur ont une logique.

Agressité : Un joueur agressif va jouer de façon à prendre l'initiative et continuer à miser dans le but d'intimider son adversaire et ainsi prendre l'avantage.

Profilage statique : Méthode consistant à profiler un joueur lors de chaque partie. Le profil établi n'est pas modifié à chaque fois que le joueur profilé effectue une action.

Profilage dynamique : Méthode consistant à profiler un joueur tout au long d'une partie, en mettant à jour le profil établi en fonction des actions effectuées.

Cave : La cave correspond à la somme possédée par chaque joueur au début de la partie.

Pré-flop : Instant du jeu où le joueur possède deux cartes en main avant que les cartes communes n'aient été révélées.

Flop : Correspond aux trois premières cartes communes posées sur la table.

Checker : Correspond au moment où un joueur reste dans le jeu mais ne place pas d'enchères. Un joueur ne peut checker que si aucun joueur n'a misé avant lui.

Dealer : Joueur se trouvant sur le siège d'où les cartes vont être distribuées.

- Blinde :** Terme correspondant aux mises obligatoires pour les deux joueurs situés à gauche du dealer.
- Bouton :** Le bouton représente le dealer qui distribue les cartes.
- Relancer :** Une relance correspond au moment où un joueur va miser plus que ce que ses adversaires viennent de miser.
- Coup :** Correspond à la distribution courante d'un jeu.
- Mise :** Montant placé sur la table par un joueur à un instant donné.
- Se coucher :** Correspond au moment où le joueur abandonne le coup. Ses mises sont alors perdues.
- Turn :** Quatrième carte commune.
- River :** Cinquième carte commune.
- Parler :** Effectuer une action.

Chapitre 1

Introduction

1.1 Généralités

Alors qu'actuellement, de plus en plus de méthodes de profilage sont établies dans le monde des jeux afin de trouver le profil d'un joueur adverse, le poker est un jeu qui pose problème dans ce domaine. En effet, ce jeu est basé sur la chance et sur le bluff. De ce fait, un joueur ne peut jamais être sûr à 100% de pouvoir gagner et donc, son adversaire ne peut pas deviner le jeu qu'il a de manière certaine. C'est pourquoi, il est dur de profiler un joueur de manière certaine. Il y a donc des recherches dans le développement de méthodes permettant de profiler un joueur de manière efficace.

1.2 Sujet

Le but de notre projet est de mettre en place une façon de profiler de façon efficace un joueur. Nous devons tout d'abord établir une méthode de profilage statique. C'est à dire, une technique pour profiler un joueur en fonction des actions qu'il a effectuées pendant toute une partie, de manière globale. Après avoir implémenté cette première méthode de profilage, nous devons mettre en place une intelligence artificielle pouvant jouer en fonction des résultats obtenus. Ainsi, nous pourrions voir si notre intelligence artificielle est capable de gagner plus de parties une fois le profil du joueur adverse établi.

S'il nous reste du temps, nous devons mettre en place une méthode de profilage dynamique, avec par conséquent, un profil qui s'établit tout au long de la partie et qui est modifié à chaque nouvelle action du joueur adverse. Nous devons alors, de même que précédemment, mettre en place une intelligence artificielle capable de jouer en exploitant les données de profilage. Ainsi, nous pourrions observer la différence du nombre de parties gagnées en fonction des deux types de profilage, à savoir, statique ou dynamique, mais aussi en fonction des profils établis. En effet, un joueur considéré comme agressif et rationnel est-il plus facile à profiler et à battre qu'un joueur considéré comme non agressif

et irrationnel ?

1.3 Rapide description des règles du Texas Hold’Em Poker

1.3.1 Mise en place du jeu

Le Texas Hold’Em Poker est un jeu de cartes dans lequel le but d’un joueur est de gagner le plus d’argent.

Afin de commencer une partie, on désigne tout d’abord le bouton* qui sert à désigner le donneur ou dealer*. Le joueur choisi sera donneur pour une main, puis cela sera au tour du joueur à sa gauche une fois cette main terminée et ainsi de suite.

Une fois le donneur sélectionné, il faut poser les blindes*. Le joueur directement à gauche du dealer* pose la petite blinde* alors que celui situé à sa gauche pose la grosse blinde* qui correspond souvent à exactement le double de la petite.

Le dealer* doit alors distribuer les cartes aux joueurs, à savoir, deux cartes par joueur.

1.3.2 Préflop

Lorsque les cartes ont été distribuées, l’étape préflop* commence. Le tour de mises préflop* démarre donc par le joueur à la gauche de celui ayant mis la grosse blinde*. Celui-ci peut se coucher*, suivre* la grosse blinde* afin de pouvoir rentrer dans le coup ou encore relancer* d’un montant au moins égal à deux fois la grosse blinde*. Une fois le tour de mises préflop* terminé, on passe à l’étape suivante : le flop.

1.3.3 Flop

Lors du flop*, le dealer* pose trois cartes faces apparentes au milieu de la table. Le premier tour de mises post-flop commence alors. Les règles sont les mêmes que lors du préflop* mis à part le fait que le premier joueur à parler* peut soit checker* soit miser*.

Une fois le tour de mises fini, on passe à l’étape du turn*.

1.3.4 Turn

Lors du turn*, le dealer* pose une carte découverte, à la suite des cartes du flop*.

Le troisième tour de mises commence alors. Celui-ci est identique au flop*.

Une fois celui-ci fini, on passe à la dernière étape, à savoir la river*.

1.3.5 River

La dernière carte est alors posée par le dealer* au milieu, à la suite des autres cartes découvertes pendant les autres étapes.

De même que pour l'étape précédente, un tour de mises est effectué, avec les mêmes règles.

Une fois ce tour de mises terminé, les joueurs restants abattent leurs cartes et la meilleure main l'emporte.

1.3.6 Les combinaisons

Lorsque l'on compare les mains au poker, la main gagnante est celle ayant la combinaison la plus forte. Au poker, il y a 9 combinaisons, de la quinte flush royale, correspondant à la combinaison la plus forte à la carte haute, correspondant à la combinaison la plus faible. Ainsi, une quinte flush royale gagne face à une quinte flush qui elle-même gagne contre un carré et caetera jusqu'à la combinaison la plus faible.

La combinaison la plus forte, la quinte flush royale est une suite d'une seule couleur, allant de l'as au dix. Par exemple, la main suivante correspond à une quinte flush royale.



Figure 1.1: Quinte Flush Royale

Après la combinaison la plus forte, il y a la quinte flush, correspondant, comme on peut le voir dans la figure suivante, à une suite de cartes de la même couleur.



Figure 1.2: Quinte Flush

Par la suite, un joueur peut avoir un carré, à savoir, quatre cartes identiques. Si jamais deux joueurs ont un même carré, par exemple si le carré se trouve sur la table, le joueur gagnant sera celui ayant pour cinquième carte la carte la plus haute.



Figure 1.3: Carré

Après le carré, vient le full qui correspond à trois cartes identiques et deux autres cartes identiques, comme on peut le voir dans la figure 1.4 suivante.



Figure 1.4: Full

La combinaison correspondant à une couleur, ou flush, vient ensuite. Un joueur ayant une couleur doit donc avoir une main composée de cinq cartes de la même couleur.



Figure 1.5: Couleur

Il y a ensuite une simple suite, donc une main composée de cartes qui se suivent, comme on peut le voir dans l'exemple suivant.



Figure 1.6: Suite

Lorsqu'un joueur a un brelan, on compte uniquement ses trois cartes. Si jamais il y a égalité entre deux joueurs, on va regarder la carte haute suivante. Si celle-ci a de nouveau le même poids pour les deux joueurs, on regarde la deuxième carte la plus haute. La figure suivante correspond à un exemple de main ayant pour combinaison un brelan.



Figure 1.7: Brelan

De même que précédemment, lorsqu'un joueur a une double paire, on compare les paires puis la carte haute.



Figure 1.8: Double Paire

De même que pour la combinaison précédente, lorsqu'un joueur a une paire, si jamais il y a égalité entre deux joueurs pour la paire, on va comparer les cartes suivantes, en comparant chaque carte suivante.



Figure 1.9: Paire

Enfin, lorsqu'un joueur n'a pas d'autre combinaison, on va regarder sa carte la plus haute. De ce fait, si deux joueurs ont la même combinaison, une carte haute, et qu'ils ont la même carte haute, on va comparer la carte haute suivante et de même jusqu'à ce qu'on ait deux cartes différentes ou alors que l'on ait comparé les cinq cartes de la main.



Figure 1.10: Carte Haute

1.4 Cahier des charges

1.4.1 Problématique

Durant les 10 semaines de développement du projet, nous devrons donc implémenter de façon intelligente une simple application permettant de jouer au poker puis, une intelligence artificielle capable de profiler différents types de joueur, tout d'abord de façon dynamique et éventuellement de façon statique.

1.4.2 Fonctionnalités obligatoires

Le projet qui nous a été attribué a pour but de développer une intelligence artificielle de poker permettant d'établir les différents profils des joueurs.

L'intelligence artificielle devra donc profiler ses adversaires en se basant sur différents critères comme la rationalité ou l'agressivité. Elle catégorisera alors les différents types de joueurs et leur attribuera des comportements.

Il nous est demandé d'établir dans un premier temps un profilage statique basé sur un automate à états finis, qui a pour but d'attribuer un unique comportement à chacun des joueurs pour toute la partie. Ce comportement sera attribué puis modifié après chaque nouvelle partie jouée contre un même joueur.

Par la suite, nous devrons mettre en place une nouvelle méthode de profilage afin de profiler de façon dynamique les joueurs. Dans cette seconde version, le profil d'un joueur sera mis à jour tout au long du déroulement d'une partie et

l'intelligence artificielle devra pouvoir adapter son comportement aux réactions du joueur adverse.

Tout au long du développement, des scénarios de tests seront mis en place afin de vérifier le bon fonctionnement de la catégorisation des joueurs.

1.4.3 Fonctionnalités optionnelles

Une fois les fonctionnalités obligatoires mises en place, il sera possible d'ajouter la possibilité d'avoir plus de deux joueurs dans une même partie, avec plusieurs intelligences artificielles ou plusieurs joueurs humains mais comprenant toujours au moins une intelligence artificielle.

On pourra également intégrer un système de dialogue entre les différents joueurs basé sur des phrases prédéfinies. Ainsi, chaque joueur, humain ou non, pourra envoyer des messages aux autres participants, et notre intelligence artificielle pourra utiliser ces messages pour mieux profiler ses opposants.

1.4.4 Spécifications techniques

Afin de faciliter les conditions de travail en groupe, il faudra utiliser un gestionnaire de version.

L'application devra être réalisée en utilisant le langage C++.

Le développement devra s'appuyer sur les méthodes agiles. Celles-ci sont basées sur une approche itérative dans un esprit collaboratif en prenant compte des besoins des utilisateurs et de leurs évolutions.

1.4.5 Organisation prévisionnelle

Comme nous pouvons le voir dans l'annexe A représentant le diagramme de Gantt prévisionnel que nous avons mis en place. Dans un premier temps, nous avons donc prévu une étude préalable du sujet consistant à nous familiariser avec le vocabulaire du poker. De ce fait, Nous avons donc rapidement mis en place un glossaire contenant des définitions des différents termes. Pendant cette période, nous avons également défini les besoins des utilisateurs avec entre-autres un diagramme de cas d'utilisations. Durant cette période, nous avons aussi défini comment nous allons nous organiser durant le déroulement du projet, en définissant la fréquence des rendez-vous ainsi que les modalités de partage des données.

Par la suite, nous avons prévu de continuer par une étude détaillée durant laquelle nous allons élaborer un diagramme de classe, rédiger le cahier des charges ainsi que commencer à lire des articles sur le sujet.

Ensuite, nous passerons à une première étude technique durant laquelle nous commencerons par établir les normes de programmation. Puis, nous établirons une intelligence artificielle simple permettant à un joueur de jouer. Nous établirons par la suite les différents algorithmes et méthodes permettant d'établir un profilage statique du joueur adverse. Et nous établirons des jeux de tests. En même temps que cette étude technique, nous commencerons la phase

de réalisation, en implémentant en parallèle l'interface graphique et l'intelligence artificielle simple puis, nous ajouterons la méthode de profilage statique et effectuerons les tests définis auparavant.

Nous effectuerons ensuite une seconde étude technique pendant laquelle nous définirons les méthodes et algorithmes qui seront utilisés afin d'implémenter la méthode de profilage dynamique. Dans ce même temps, nous implémenterons cette méthode de profilage dynamique, en effectuant les tests définis pendant la phase d'étude technique.

Enfin, la dernière partie sera réservée à la mise en place de la démonstration qui sera effectuée lors de la soutenance, ainsi qu'à la finalisation du rapport qui sera rédigé tout au long de la période du projet et à la préparation de la soutenance.

Chapitre 2

Organisation du projet

2.1 Organisation du travail

2.2 Choix des outils de travail

2.2.1 Gestionnaire de versions

2.2.2 Création de l'interface graphique

2.2.3 Rédaction et mise en page du rapport

2.2.4 Normes de programmation

Nous coderons en français. Par conséquent, les différents identifiants des attributs, noms de classes et fonctions/méthodes que nous implémenterons auront des noms français, les méthodes `toString` ainsi que pour les accesseurs et mutateurs que nous nommerons `getNomAttribut` et `setNomAttribut`, selon les conventions de codage établies.

Les noms des classes commenceront toujours par une majuscule suivie de minuscules. Si le nom de la classe est composé de plusieurs mots, alors on ajoutera une majuscule à chaque début de mot.

Les noms des méthodes, des variables et des attributs devront impérativement commencer par une minuscule avec une majuscule à chaque nouveau mot.

Pour l'indentation, il faudra mettre un maximum d'accolades, facilitant la reprise et l'ajout de code, même lorsque celles-ci ne sont pas obligatoires.

Des commentaires seront ajoutés pour chaque fonction et méthode, au dessus de la déclaration correspondante, dans les fichiers d'en-tête. Les commentaires des accesseurs et mutateurs ne sont pas obligatoires car explicites. Dans ces commentaires seront précisés l'action de la fonction/méthode, l'ensemble des paramètres, l'élément retourné s'il existe. Ces commentaires seront de la forme suivante :

```
/**
```

```
* @param
* @action
* @return
**/
```

Au début de chaque nouveau fichier créé, il faudra ajouter l'entête suivante:

```
/*=====
Nom: fichier.cpp          Auteur:
Maj: 27/03/2014          Creation: 01/02/2015
Projet: Profilage par essais et erreurs au poker
=====
Specification: Specifications du fichier
=====*/
```

De plus, chaque commit effectué sur le projet Git devra avoir une description explicite permettant de savoir, sans avoir à parcourir le code, les modifications qui ont été apportées.

2.2.5 Langage

Le langage utilisé pour la mise en place de l'application est le C/C++.

Chapitre 3

Analyse du projet

3.1 Système de jeu

3.2 Profilage

3.2.1 Choix d'enregistrement des données

Afin de profiler nos joueurs, nous avons choisi d'enregistrer un certain nombre de données dans des fichiers de sortie. Nous avons décidé de mettre en place un fichier par joueur. De ce fait, chaque fichier est représenté par le pseudo du joueur. Soit joueur ayant pour pseudo "Alain". Le fichier contenant les données de profilage de ce joueur sera donc "Alain.json". De ce fait, s'il se déconnecte et souhaite rejouer un autre jour, nous aurons toujours le précédent profilage effectué.

Comme on peut le constater, nous avons choisi d'enregistrer nos données en JSON. en effet, ce format de données a l'avantage d'être facilement exploitable. De plus, nous pouvons utiliser la librairie QJson pour exploiter les données du fichier.

3.3 Formule de calcul du taux de rationalité d'un joueur

Afin de calculer à combien de pourcentages un joueur a été rationnel pendant une partie, nous avons choisi de nous baser sur une courbe [ajouter un schéma de la courbe et l'expliquer] correspondant aux pourcentages théoriques de rationalité en fonction des chances de gagner. A partir de cette courbe, nous avons décidé de calculer la mise théorique que le joueur aurait dû miser en nous basant sur quatre paliers, comme on peut le voir dans la formule suivante.

Pour chaque palier, on regarde si les chances de gain du joueur sont comprises entre g1 et g2. Une fois le palier trouvé, on calcule la mise théorique, en sachant

que celle-ci sera comprise entre les m1 et m2 correspondant au palier.

Gain = chances de gain du joueur,

Ra = rationalité

M_{th} = Mise théorique.

$$M_{th} = \left((Gain - g1) * \left(\frac{m2 - m1}{g2 - g1} \right) \right) + m1 \begin{cases} Ra \in [m1 = 0 - m2 = 10] & \text{Si Gain} \in [g1 = 0 - g2 = 30] \\ Ra \in [m1 = 11 - m2 = 25] & \text{Si Gain} \in [g1 = 31 - g2 = 50] \\ Ra \in [m1 = 26 - m2 = 65] & \text{Si Gain} \in [g1 = 51 - g2 = 69] \\ Ra \in [m1 = 66 - m2 = 100] & \text{Si Gain} \in [g1 = 70 - g2 = 100] \end{cases}$$

Après avoir calculé la mise théorique que le joueur aurait dû faire, nous la comparons avec la mise réelle du joueur. On obtient alors la formule suivante pour calculer la rationalité :

$$Rationalité = 100 - abs(M_{th} - M_{réelle})$$

3.3.1 Formule de calcul du taux d'agressivité d'un joueur

Afin de calculer le pourcentage d'agressivité d'un joueur pendant une partie, de même que pour la rationalité, nous avons choisi de nous baser sur un système de paliers, en prenant en compte trois paramètres : le pourcentage de mises, le total des mises qui est exprimé en pourcentage de jetons par rapport aux jetons de départ du joueur et, la mise la plus haute du joueur, elle aussi exprimée en pourcentage de jetons par rapport aux jetons de base.

De même que précédemment, pour chaque palier, on va regarder si la mise la plus haute (mph) est comprise entre mph1 (mise la plus haute 1) et mph2. Une fois le palier trouvé, on calcule le pourcentage d'agressivité théorique qui sera compris entre ag1 et ag2.

$$Ag_{th} = \begin{cases} Ag_{th} \in [0 - 50] Simph \in [0 - 25] & \text{ratio1} = \frac{1}{2} \text{ ratio2} = \frac{1}{2} \\ Ag_{th} \in [51 - 80] Simph \in [26 - 50] & \text{ratio1} = \frac{2}{3} \text{ ratio2} = \frac{1}{3} \\ Ag_{th} \in [81 - 100] Simph \in [51 - 100] & \text{ratio1} = \frac{2}{3} \text{ ratio2} = \frac{1}{3} \end{cases}$$

Pour calculer le pourcentage théorique d'agressivité, on se base sur la formule suivante, en sachant que pour chaque palier, nous accordons des poids différents au total des mises et au nombre de mises.

$$\begin{aligned} 0 < y < mph2 - ag2 \\ x &= ratio1 * nbMises + ratio2 * totalMises \\ y &= \left(x * \frac{mph2 - ag2}{100} \right) \end{aligned}$$

Pour le dernier palier, nous ne nous basons pas sur la formule précédente mais sur la formule suivante :

$$y = \left(x * \frac{100 - mph}{100} \right)$$

3.3.2 Formule de calcul du taux de bluff d'un joueur

Étant donné le fait que nous partons du principe que le bluff est l'inverse de la rationalité, nous calculons le taux de bluff en utilisant la formule suivante :

$$bluff = 100 - rationalité$$

3.3.3 Formule de calcul du taux de passivité d'un joueur

Nous avons décidé que le taux de passivité d'un joueur se calcule en fonction du nombre de fois où il a suivi et du nombre de fois où il a checké. De ce fait, nous avons choisi d'établir la formule suivante pour calculer le taux de passivité d'un joueur :

$$passivité = tauxChecks + tauxSuivis$$

3.4 Profilage statique

3.5 Réutilisation des résultats du profilage

3.6 Profilage dynamique

3.7 Réutilisation des résultats du profilage

Chapitre 4

Mise en place des méthodes de profilage

4.1 Calibrage des intelligences artificielles

Par la suite nous avons revu le calibrage de l'agressivité qui n'était pas assez précis. En effet, tout comme pour calibrer la rationalité, nous laissons une part de hasard dans le calcul de l'action de l'intelligence artificielle. Ceci avait pour effet de produire des résultats trop imprévisibles, et parfois trop éloignés du calibrage de départ.

Pour pallier ce problème, nous avons calculé une action de mise correspondante au taux d'agressivité en fonction de la mise totale effectuée par l'intelligence artificielle au cours de la partie. Nous avons également adapté le calcul côté profilage afin que la mise totale devienne le critère principal, et plus la mise la plus haute.

A chaque tour, le résolveur calcule donc la mise théorique à effectuer selon les nouveaux paliers.

Ag. Mise totale

0 – 50 0 – 25 exclu 50 – 80 25 – 60 exclu 80 – 100 60 – 100 exclu 100 100

Selon le calibrage, le but est donc que l'intelligence artificielle effectue la mise prévue. Seulement, il n'est pas possible de déterminer la durée de la partie. Il faut donc que celle-ci effectue progressivement sa mise, sans terminer trop tôt où elle devra effectuer des actions check ou suivi pour terminer, ce qui diminuera le taux d'agressivité; et ni trop tard pour qu'elle puisse être considérée suffisamment agressive.

Pour ce faire, nous misons de façon croissante, depuis un départ minimum, jusqu'à atteindre la mise théorique totale calculée à partir du calibrage.

En prenant un rapport $\times 2$, on obtient donc les mises totales suivantes à chaque tour de jeu.

Ex : 5 – 10 – 20 – 40 – 80 avec 80 mise totale théorique.

Dans ce cas, cela revient donc à miser 5; 5; 10; 20 puis 40.

Afin d'obtenir ce résultat, nous différencions deux cas : celui où l'intelligence artificielle peut miser et celui où elle doit suivre/relancer ou se coucher.

Si elle peut miser, alors elle joue la mise théorique courante en faisant attention de ne pas dépasser la mise théorique totale. Dans le cas où on est au river (dernier tour), alors elle mise directement ce qu'il lui manque pour arriver à cette mise théorique

totale.

Dans le cas où l'adversaire a misé/relancé, alors l'intelligence artificielle va comparer la mise théorique courante avec la mise adverse et la valeur de relance minimum. Dans le cas où la mise théorique est inférieure à la valeur de suivi -10% alors on se couche, pour ne pas miser trop haut en suivant. Si celle-ci est comprise entre le taux de suivi $\pm 10\%$, on suit. Dans le cas où la relance est \leq à la valeur à miser +10% , on relance de la valeur la plus proche de la mise théorique. Sinon on suit.

A la suite de ces changements nous avons également revu le choix entre les actions d'agressivité et de rationalité, une fois ces deux dernières calculées.

Une fois que le calibrage nous a donné l'action agressive et l'action rationnelle, on effectue une fusion entre les deux. Dans les cas où les deux actions sont identiques, l'action est prise. Si ce sont toutes les deux des mises/relances, on prend la moitié des du nombre de jetons, proportionnellement aux pourcentages de calibrage. Enfin, il reste la possibilité que les actions soient différentes. Ici on distingue deux cas :

- on arrive à déterminer une action représentant un juste milieu, par exemple si les deux actions sont se coucher et relancer, on fait suivre l'intelligence artificielle. Dans le cas où on a checker / miser, on considère l'action checker comme miser 0 et on effectue le calcul de la mise moyenne expliqué ci-dessus.
- sinon on tire aléatoirement l'action parmi les deux, toujours en prenant en compte les pourcentages du calibrage.

4.2 Profilage statique

4.2.1 Scénarios de tests

4.2.2 Établissement des profils attendus

4.2.3 Réutilisation des résultats du profilage

4.3 Profilage dynamique

4.3.1 Scénarios de tests

4.3.2 Établissement des profils attendus

4.3.3 Réutilisation des résultats du profilage

Chapitre 5

Analyse des résultats obtenus

5.1 Profilage statique

5.1.1 Scénarios de tests

5.1.2 Analyse des gains de parties

5.2 Profilage dynamique

5.2.1 Analyse des gains de parties

Chapitre 6

Perspectives et conclusion

6.1 Perspectives d'amélioration du profilage

6.2 Profilage obtenu

Bibliography

- [1] Karl S. Brandt. Player profiling in texas holdem. 2004.
- [2] Maria de Lourdes Peña Castillo. *Probabilities and Simulations in Poker*. PhD thesis, Université d'Alberta, 1999.
- [3] Poker Listings. Poker listings - le guide de poker all-in. fr.pokerlistings.com.

Annexe A

Test annexe