



UNIVERSITÉ
DE MONTPELLIER



Rapport de projet

Profilage par essais et erreurs au poker

Réalisé par
Manuel Chataigner, Benjamin Commandré,
et Morgane Vidal

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

Nous remercions aussi Jimmy Lopez et Sébastien Beugnon pour nous avoir donné quelques conseils tout au long de notre projet.

Table des matières

1	Introduction	10
1.1	Généralités	10
1.2	Sujet	10
1.3	Rapide description des règles du Texas Hold'Em Poker	11
1.3.1	Mise en place du jeu	11
1.3.2	Préflop	11
1.3.3	Flop	11
1.3.4	Turn	12
1.3.5	River	12
1.3.6	Les combinaisons	12
1.4	Cahier des charges	15
1.4.1	Problématique	15
1.4.2	Fonctionnalités obligatoires	15
1.4.3	Fonctionnalités optionnelles	16
1.4.4	Spécifications techniques	16
1.4.5	Organisation prévisionnelle	16
2	Organisation du projet	18
2.1	Organisation du travail	18
2.1.1	Réunions	18
2.1.2	Mise en commun du travail	18
2.1.3	Planification	19
2.2	Méthodes et outils de travail	20
2.2.1	Gestionnaire de versions	20
2.2.2	Normes de programmation	21
3	Analyse du projet	23
3.1	Analyse de l'existant	23
3.2	Système de jeu	24
3.2.1	Deux joueurs	24
3.2.2	Intelligence artificielle contre intelligence artificielle	25
3.2.3	Choix d'enregistrement des données	25
3.3	Profilage	26
3.4	Phase de jeu	27

4	Mise en place du jeu et de l'interface graphique	28
4.1	Jeu	28
4.2	Interface graphique	29
4.2.1	Paramétrer le jeu	29
4.2.2	Intelligence artificielle face à un joueur humain	30
4.2.3	Intelligence artificielle face à une autre intelligence artificielle	31
5	Calcul du profil d'un joueur	33
5.1	Rationalité	33
5.2	Agressivité	34
5.3	Bluff	35
5.4	Passivité	35
6	Intelligence artificielle	36
6.1	Résolveur	36
6.1.1	Résolveur basique	36
6.1.2	Résolveur avec calibrage	36
6.2	Estimation des chances de gain	40
7	Méthodes de profilage	42
7.1	Scénarios de tests	42
7.1.1	Mise en place des scénarios	42
7.1.2	Analyse des résultats	44
7.2	Établissement des profils attendus	47
7.3	Vérification du profil établi	48
8	Analyse des gains de parties	49
8.1	Un premier calibrage inefficace	49
8.1.1	Mise en place du calibrage	49
8.1.2	Analyse des résultats	50
8.2	Recherche du calibrage optimal	52
8.3	Recherche du meilleur type de joueur	55
8.4	Résultats	57
9	Perspectives et conclusion	58
9.1	Perspectives d'amélioration du profilage	58
9.1.1	Profilage dynamique	58
9.1.2	Profilage avec plus de deux joueurs	58
9.2	Profilage obtenu	59
A	Planification prévisionnelle	61
B	Planification réelle	62

Liste des figures

1.1	Quinte Flush Royale	12
1.2	Quinte Flush	13
1.3	Carré	13
1.4	Full	13
1.5	Couleur	14
1.6	Suite	14
1.7	Brelan	14
1.8	Double Paire	14
1.9	Paire	15
1.10	Carte Haute	15
2.1	Schéma de fonctionnement de Git	21
4.1	Diagramme de classe	28
4.2	Fenêtre d'options de l'application	30
4.3	Interface humain contre intelligence artificielle	31
4.4	Interface intelligence artificielle contre intelligence artificielle	32
5.1	Mise théorique en fonction de la force du jeu	33
6.1	Paliers d'agressivité et mise totale en fonction	38
6.2	Choix de l'action à effectuer en fonction des actions d'agressivité* et de rationalité	39
6.3	Chances de gains obtenues en fonction du nombre de tests effectués	41
7.1	Actions attendues	43
7.2	Taux de similarité du profilage en fonction du nombre de partie	45
7.3	Profil établi en fonction du nombre de parties	46
7.4	Taux de similarité des profils établis en fonction des paliers	46
8.1	Calibrage de l'intelligence artificielle en fonction du profil adverse	49
8.2	Données situées à la fin d'un fichier contenant le profil établi d'un joueur	50
8.3	Nombre de parties gagnées pendant les scénarios de tests et après	51
8.4	Nombre de jetons gagnés pendant les scénarios de tests et après	52

8.5	Gains en fonction du calibrage face à un joueur 80% agressif et 80% rationnel	53
8.6	Calibrage final de l'intelligence artificielle en fonction du profil adverse	54
8.7	Nombre de parties gagnées en fonction du premier calibrage établi et du second calibrage calculé	54
8.8	Nombre de parties gagnées en fonction des deux calibrages	55
8.9	Résultats sur 100 parties face à un comportement 80% agressif et 80% rationnel	56
8.10	Gains de l'intelligence artificielle en fonction de la valeur de x . .	56

Glossaire

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

Agressivité : 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.

Blinde : Terme correspondant aux mises obligatoires pour les deux joueurs situés à gauche du dealer.

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.

Bouton : Le bouton représente le dealer qui distribue les cartes.

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

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.

Coup : Correspond à la distribution courante d'un jeu.

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

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

IA : Intelligence artificielle.

Mise : Montant placé sur la table par un joueur à un instant donné.

Parler : Effectuer une action.

- Préflop :** Instant du jeu où le joueur possède deux cartes en main avant que les cartes communes n'aient été révélées.
- Profilage :** Méthode consistant à établir le comportement d'un joueur, sa façon de jouer.
- 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.
- 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.
- Rationalité :** La rationalité appliquée au poker stipule que toutes les actions d'un joueur ont une logique basée sur les probabilités de gain par rapport à la force d'une main.
- Relancer :** Une relance correspond au moment où un joueur va miser plus que ce que ses adversaires viennent de miser.
- River :** Cinquième carte commune.
- Se coucher :** Correspond au moment où le joueur abandonne le coup. Ses mises sont alors perdues.
- Turn :** Quatrième carte commune.

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, celui-ci 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 la main qu'il possède de manière certaine. C'est pourquoi, il est dur de profiler un joueur correctement. 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 efficacement 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. Pour la présentation des règles suivantes, nous nous sommes inspirés du site Poker Listings [5].

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, à savoir, deux cartes par joueur.

Le jeu se déroule alors en quatre tours d'enchères, qui sont le préflop*, le flop*, le turn*, et le river*.

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*. Le tour d'enchères se termine lorsque chacun des joueurs a eu l'opportunité d'agir et que tous les joueurs qui n'ont pas abandonné ont misé le même montant d'argent dans le pot. 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. Il est identique au flop*. Une fois celui-ci terminé, 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 cætera jusqu'à la combinaison la plus faible.

Les deux premières combinaisons sont en réalité deux sous-espèces d'une seule et même combinaison, la quinte flush, qui correspond à une suite de cartes de la même couleur.

La plus forte des deux, la quinte flush royale est une suite d'une seule couleur, allant de l'as au dix. Par exemple, la main suivante représente un exemple de cette combinaison.



Figure 1.1: Quinte Flush Royale

La seconde combinaison est composée de toutes les autres suites d'une même couleur, formant une quinte flush, comme on peut le voir dans la figure suivante.

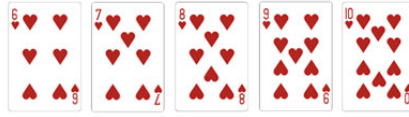


Figure 1.2: Quinte Flush

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

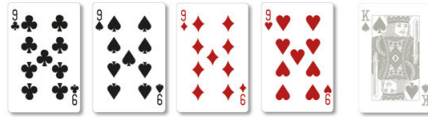


Figure 1.3: Carré

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.



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, à savoir trois cartes identiques, on compte uniquement ces 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.

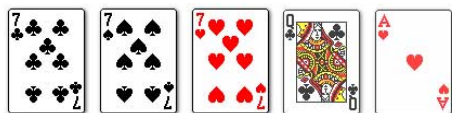


Figure 1.7: Brelan

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



Figure 1.8: Double Paire

Comme la combinaison précédente, lorsqu'un joueur a une paire, s'il y a égalité entre deux joueurs pour la paire, on va comparer les cartes suivantes.



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 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.

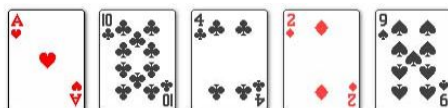


Figure 1.10: Carte Haute

1.4 Cahier des charges

1.4.1 Problématique

Durant les dix 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 statique et éventuellement de façon dynamique.

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 de 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, s'il reste du temps, 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 adversaire 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++ et l'interface sera développée en utilisant le framework Qt.

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 mis en place, nous avons prévu dans un premier temps une étude préalable du sujet consistant à nous familiariser avec le vocabulaire du poker. De ce fait, nous avons rapidement mis en place un glossaire contenant les définitions des différents termes. Pendant cette période, nous avons également défini les besoins des utilisateurs avec entre-autres un diagramme de cas d'utilisations. Nous avons aussi défini comment nous allions nous organiser durant le déroulement du projet, en spécifiant 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 développerons une intelligence artificielle simple capable de jouer. Nous réfléchirons par la suite aux différents algorithmes et méthodes permettant d'établir un bon profilage* statique. Par la suite, nous établirons des jeux de tests. Dans le même temps, 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 réaliserons 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 la 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 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

2.1.1 Réunions

Comme prévu au départ, nos tuteurs et nous-même avons choisi de nous réunir chaque semaine afin de discuter de nos avancées, des éventuels problèmes rencontrés et des méthodes choisies pour le profilage*. De ce fait, nous garantissons que ce que nous faisons était toujours en accord avec les attentes.

De notre côté, nous avons choisi de nous réunir chaque mercredi et vendredi matin afin de pouvoir réfléchir ensemble sur les différentes méthodes de calculs. Par exemple, comment déterminer le taux d'agressivité* d'un joueur ? Nous répartissions ensuite le contenu à implémenter entre les différents membres du groupe.

Étant donné le fait que notre groupe n'était composé que de trois membres, nous avons choisi de ne pas élire de chef de projet mais plutôt de travailler tous sur un pied d'égalité. De ce fait, il n'y a pas eu de spécialisation, chaque membre du groupe a participé à toutes les parties du projet. En effet, notre projet étant basé en particulier sur la réflexion et sur la mise en place de différents algorithmes, il était plus intéressant de travailler à plusieurs.

2.1.2 Mise en commun du travail

Nous avons choisi d'utiliser un git workflow, https://github.com/Gouga34/TERM1_Poker, pour la gestion de notre projet. Cette méthode consiste à mettre en place une branche master qui contient le code de production et sur laquelle rien n'est développé. Elle est donc réservée aux versions fonctionnelles et abouties. Nous avons ensuite une branche develop sur laquelle sont ajoutées l'ensemble des modifications effectuées au cours du développement et le code qui sera ajouté pour la nouvelle release. Sur cette branche, on peut corriger ou

encore améliorer des fonctionnalités.

C'est donc une fois l'application dans une version finalisée que sont ajoutés à la branche master l'ensemble des commits effectués sur le develop.

Pour chaque nouvelle fonctionnalité ou correction, nous ajoutons une nouvelle branche en local dans nos répertoires sur laquelle les modifications étaient effectuées. Elles étaient ensuite intégrées à la branche develop en local, puis envoyées sur le répertoire public correspondant. Enfin, elles étaient intégrées au répertoire de référence.

2.1.3 Planification

Le diagramme de Gantt de l'annexe B correspond à la planification réelle de notre projet. Comme on peut vite le constater, nous avons eu du retard et n'avons pas pu finir toutes les tâches prévues au départ. De même, on peut constater que quelques tâches absentes au départ ont été ajoutées.

On peut tout d'abord voir que la mise en place de l'interface graphique a pris plus de temps que prévu. En effet, nous n'avions pas prévu que la bibliothèque que nous souhaitions utiliser s'avérerait difficile à mettre en place et que par conséquent il nous faudrait partir de zéro. Étant donné le retard pris, nous avons aussi eu quelques jours de retard sur l'implémentation du jeu de poker et de l'intelligence artificielle basique.

Cependant, on peut constater que nous avons démarré l'analyse de l'existant pour le profilage statique* bien en avance. Malgré cela, nous avons ajouté à la liste des articles à lire quelques articles supplémentaires dont une thèse qui nous aura pris plus de temps à lire. De ce fait, nous avons pris du retard sur la phase d'étude détaillée et sur la phase de développement des algorithmes. De plus, lorsque nous avons voulu passer à la phase d'implémentation des algorithmes, nous devions au départ récupérer une calculatrice de probabilités permettant de calculer les chances de gain d'un joueur. Cependant, cette calculatrice n'ayant pas été retrouvée, nous avons ajouté de nouvelles tâches : l'étude détaillée, l'implémentation de la calculatrice de probabilité et l'implémentation d'un évaluateur de mains. Par conséquent, nous avons mis en pause le développement des algorithmes afin que tous les membres du groupe puissent discuter ensemble de l'étude détaillée de la calculatrice de probabilités. De ce fait, nous avons pris encore plus de retard dans le déroulement de notre projet.

Comme on peut le constater, à mi-parcours du projet, du 31 mars au 10 avril, nous nous sommes rendus compte que la partie concernant le jeu et l'intelligence artificielle était difficilement réutilisable. En effet, nous découvrions régulièrement des bogues qui n'avaient pas été détectés au moment des tests. Cette partie étant difficilement débogable et nous faisant perdre beaucoup de temps, nous avons par conséquent choisi de reprendre complètement cette partie. Nous avons

perdu beaucoup de temps à ce moment-là et nous nous sommes rendus compte à quel point la découverte tardive d'une bogue fait prendre beaucoup de retard sur le développement de l'application. Par conséquent, alors que la période d'implémentation des algorithmes liés au profilage statique* aurait dû être terminée, elle a été retardée, ce qui a fait prendre du retard aux tâches suivantes.

Enfin, nous pensions au départ que nous ne devrions que profiler les joueurs adverses alors que nous devions aussi faire jouer l'intelligence artificielle en fonction du profil établi afin de faire en sorte de gagner le plus d'argent possible. De ce fait, nous avons ajouté les tâches en rapport à la mise en place du jeu de l'intelligence artificielle en fonction des profils établis.

Nous pouvons donc en conclure que nous avons pris beaucoup de retard, sûrement à cause du fait que nous n'avions pas bien compris l'ampleur des tâches qui seraient à effectuer et à cause d'une mauvaise conception au départ de la partie correspondant au jeu et à l'intelligence artificielle basique. Finalement, étant donné que l'on s'est rendu compte du retard pris sur le projet, plutôt que terminer précipitamment la partie sur le profilage statique* pour commencer le profilage dynamique* prévu, nous avons préféré ne pas nous précipiter et terminer correctement le profilage statique* et obtenir de bons résultats, quitte à ne pas faire la partie dynamique.

2.2 Méthodes et outils de travail

2.2.1 Gestionnaire de versions

Notre choix s'est porté sur le gestionnaire de version Git, car contrairement à d'autres tels que Subversion, Git est décentralisé. Il ne repose pas sur un seul et même dépôt en ligne sur lequel chaque nouvelle version d'un fichier envoyée remplace la précédente et affecte donc l'ensemble du groupe. Git met en place, en plus du dépôt de référence commun à tous, un dépôt en ligne et un dépôt local pour chacun des utilisateurs, comme on peut le voir dans le schéma suivant.

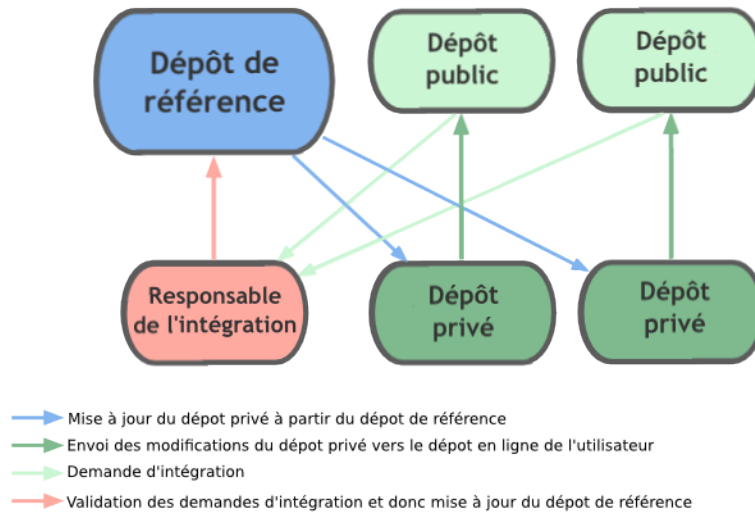


Figure 2.1: Schéma de fonctionnement de Git

L'utilisation de dépôts publics, en vert clair sur le schéma, permet d'éviter les conflits, notamment lorsque deux personnes travaillent sur un même fichier, puis souhaitent l'envoyer sur le dépôt commun simultanément. Ici, les fichiers sont envoyés sur les dépôts en ligne personnels, puis intégrés sur le dépôt de référence, représenté en bleu, par la personne en charge lorsque les utilisateurs effectuent une demande d'intégration auprès du responsable (integration manager).

Les conflits n'ont lieu dans ce cas, que lorsqu'une même ligne est modifiée plusieurs fois, auquel cas, c'est le responsable d'intégration qui le verra et donc n'acceptera pas les modifications proposées.

2.2.2 Normes de programmation

Concernant la réalisation du projet, l'application sera développée 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, on ajoutera alors 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'en-tête 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.

Chapitre 3

Analyse du projet

3.1 Analyse de l'existant

Avant de commencer à réfléchir aux différentes méthodes de profilage, nous avons choisi de lire différents articles sur le sujet pour nous faire une idée des données à prendre en compte et des différentes méthodes que nous pourrions utiliser.

Malgré la lecture des articles, nous avons choisi de développer nos propres méthodes de profilage, sans utiliser exactement les calculs décrits dans les différents articles. Cependant, nous avons quand même choisi de nous inspirer des articles, notamment pour le choix des paramètres importants à prendre en compte, par exemple pour le calcul du profil du joueur, à savoir, le taux d'agressivité, de rationalité, de passivité et de bluff.

Ainsi, grâce à l'article "Poker as a Testbed for AI Research" [1], expliquant entre autres les paramètres à prendre en compte pour avoir une base de jeu solide, nous avons isolé de façon plus rapide les paramètres importants pour établir les profils des joueurs. De ce fait, nous en avons déduit que le paramètre à prendre en compte en priorité est la force de la main et qu'il faudra bien entendu la recalculer à chaque nouvelle étape puisque suite à l'apparition de nouvelles cartes, les chances de gain d'un joueur peuvent complètement changer. L'article en question décrit également les critères de profilage comme l'agressivité. De même, nous nous sommes basés sur l'article pour la façon dont notre intelligence artificielle allait jouer pour gagner. En effet, dans cet article, il est expliqué à quel point il est important d'avoir un jeu imprévisible, c'est à dire de ne pas effectuer toujours les mêmes actions.

Cependant, nous ne nous sommes pas seulement inspirés de l'article présenté précédemment. En effet, cet article ne se concentre que sur la force de la main, le potentiel d'une main ainsi que la stratégie de jeu et ne traite pas de façon

précise des paramètres à prendre en compte pour profiler un joueur.

Pour choisir les paramètres à prendre en compte pour le profilage d'un joueur, nous nous sommes inspirés de la thèse "Probabilities and Simulations in Poker" [4]. En effet, dans cette thèse, Maria de Lourdes Peña Castillo explique que pour profiler de manière efficace un joueur, il faut se baser sur le triplet suivant, le taux de suivis, de mises et de checks. Nous retrouvons une idée similaire dans l'article "Improved Opponent Modeling in Poker" [3], qui propose d'utiliser une table de fréquence de paris comptant également à chaque fois qu'un joueur check, mise ou se couche dans chaque contexte, c'est à dire chaque étape de jeu.

Par la suite, si nous avons lu beaucoup d'articles traitant de la mise en place d'une table de poids pour calculer la possible main d'un adversaire en fonction de ses actions, notamment la thèse "Probabilities and Simulations in Poker" [4], "Player Profiling in Texas Hold'em Poker" [2] ou encore l'article "Improved Opponent Modeling in Poker" [3]. Cependant, au vu de la complexité de la mise en place de ce calcul, nous avons choisi de mettre de côté l'implémentation de cette fonctionnalité et de la reprendre éventuellement si nous avons du temps à la fin.

3.2 Système de jeu

3.2.1 Deux joueurs

Afin de pouvoir effectuer le profilage* d'un joueur, nous avons mis en place un jeu de poker à deux joueurs. Dans un premier temps, il s'agissait d'un joueur humain contre une intelligence artificielle ayant pour but de le profiler. Pour pouvoir jouer, une interface graphique a été mise en place afin de permettre à la fois le suivi de la partie et le choix des actions.

Au départ, l'intelligence artificielle reposait sur un algorithme basique lui permettant de jouer une partie. Puis nous avons rapidement mis en place un calibrage de l'intelligence artificielle, afin de pouvoir lui attribuer un style de jeu (agressif, rationnel...). Lors de l'ajout de ce calibrage, nous avons complexifié le jeu de l'intelligence artificielle en prenant en compte le calibrage donné mais aussi en y ajoutant un choix d'action plus varié.

Afin de différencier les joueurs profilés, nous avons mis en place un système de pseudos. Avant le démarrage du jeu, il est alors possible de renseigner un pseudo, nouveau ou déjà existant, qui permet à l'intelligence artificielle de collecter et rassembler les données de profilage* d'un même joueur. De ce fait, s'il se déconnecte et souhaite rejouer un autre jour, l'intelligence artificielle disposera toujours du précédent profilage* effectué.

3.2.2 Intelligence artificielle contre intelligence artificielle

Par la suite, afin de permettre l'automatisation du profilage*, et le lancement d'une série de parties, il nous a été demandé de revoir le système de jeu afin que l'intelligence artificielle existante puisse jouer contre une deuxième intelligence artificielle ne profilant pas. Dans cette version, l'interface n'est plus prise en compte et les deux intelligences jouent simultanément jusqu'à la fin de la partie. Dans ce cas, un calibrage est donné pour cette deuxième intelligence artificielle, et celui-ci reste fixe pour la suite de parties. C'est l'intelligence artificielle qui profile qui va tenter de retrouver les valeurs de ce calibrage.

Pour permettre ces séries de tests, nous avons donc ajouté la possibilité de choisir le nombre de parties à lancer, afin de tester de façon efficace le profilage* mis en place et déterminer la façon optimale d'établir le profil du joueur adverse.

Par la suite, nous avons de même permis le lancement de profilages* sur une suite d'adversaires ayant un comportement déterminé aléatoirement.

3.2.3 Choix d'enregistrement des données

Afin de profiler les joueurs, nous avons choisi d'enregistrer un certain nombre de données dans des fichiers de sortie. Selon le système de pseudos développé, nous avons décidé de mettre en place un fichier par joueur, représenté par le pseudo du joueur ou bien son calibrage dans le cas d'une intelligence artificielle. Dans le but d'une réutilisation des données enregistrées par l'application, nous avons au départ choisi le format de données Json car il a l'avantage d'être facilement exploitable. De plus, nous avons pu utiliser la librairie QJson pour traiter les données des fichiers.

Nous avons ensuite déterminé les données à prendre en compte au cours d'une partie, qui permettront le profilage* du joueur adverse, et qui seront enregistrées dans les fichiers json. Chaque valeur est représentée sous forme de taux, en pourcentage.

Au cours d'une partie, l'intelligence artificielle va enregistrer les chances de gain du joueur, calculées en fonction des cartes du joueur adverse en fin de partie.

Elle va également calculer les différentes actions effectuées. Ici, nous avons choisi de classer les différentes actions en trois catégories, les suivis, les checks, et les mises qui comprennent aussi les relances. Chacune de ces valeurs représente donc le pourcentage de réalisation de cette action par le joueur en fonction du nombre de tours de jeu, par exemple si le joueur a suivi une fois sur deux, on aura 50% de suivis. Si le joueur ne se couche pas, le total de ces trois valeurs est donc 100%. Enfin, un booléen indique si le joueur s'est couché ou non au cours de la partie.

Concernant les mises, afin d'être plus précis sur l'agressivité* du joueur, nous mémorisons également la mise* la plus haute effectuée en une fois par le joueur, ainsi que le total du montant misé sur l'ensemble des tours.

Finalement, le fichier contient tous les types de comportements possibles que l'on peut attribuer au joueur profilé. Nous avons choisi de catégoriser le joueur selon quatre critères : l'agressivité*, c'est à dire sa capacité à miser, la rationalité*, le bluff* et la passivité*. Ces quatre valeurs sont exprimées en pourcentage. Pour chacune d'elle nous avons déterminé une formule qui, à partir des données enregistrées dans le fichier et présentées plus haut, va calculer les taux associés.

Enfin, dans le but d'effectuer un profilage* plus précis des joueurs, nous avons distingué chaque étape de jeu, et donc enregistré l'ensemble de ces valeurs pour les quatre étapes d'une partie : préflop*, flop*, turn* et river*. Un profil global de la partie est ensuite écrit avec les quatre valeurs de comportement. Puis un booléen contient le résultat de la partie.

Finalement, nous nous sommes rendu compte qu'enregistrer les différentes valeurs utile au profilage* en fonction de chaque étape n'était pas optimal. En effet, un profil calculé globalement donnait des résultats plus cohérents qu'un profil calculé à chaque étape d'une partie. Nous avons donc choisi d'enregistrer uniquement les données de manière globale.

Par la suite, nous avons besoin de créer des graphes avec les données enregistrées dans les fichiers de sortie. Nous nous sommes alors rendus compte que l'utilisation de fichiers de type csv était plus adaptée. De ce fait, nous avons décidé de changer le type de nos fichiers de sortie en csv.

Nous avons ensuite ajouté le gagnant de la partie, ainsi que les gains ou pertes d'argent de l'intelligence artificielle au cours de la partie. De cette manière, il nous serait facilement possible d'observer les gains cumulés de l'intelligence artificielle au bout de plusieurs parties et en faire un graphique.

3.3 Profilage

Pour établir le profil d'un joueur, nous nous basons sur quatre taux qui seront établis en fonction du comportement du joueur adverse.

Nous prenons d'abord en compte la rationalité* du joueur, c'est à dire si le montant de ses mises ou bien ses actions sont en accord avec son jeu. Pour calculer ce taux, nous nous basons donc sur les chances de gain du joueur ainsi que sur le montant total de ses mises au cours de la partie. Comme on ne connaît pas durant la partie les cartes du joueur adverse, le taux de rationalité* est calculé uniquement à la fin de la partie, après avoir déterminé le gagnant,

une fois les cartes du joueur profilé visibles.

Le second critère pris en compte est l'agressivité*, correspondant à la capacité d'un joueur à prendre l'initiative et à continuer à miser sans forcément prendre en compte son jeu, dans le but d'intimider son adversaire et ainsi prendre l'avantage. Les données prises en compte sont l'ensemble des mises du joueur. C'est à dire, le total des mises, la mise* la plus haute ainsi que le nombre de mises effectuées durant la partie.

Il faut aussi prendre en compte la passivité d'un joueur, c'est à dire, si le joueur a tendance à suivre son adversaire plutôt qu'à prendre les devants. Pour cela, nous observons le nombre de suivis et le nombre de checks effectués tout au long de la partie par le joueur profilé.

Enfin, le critère du bluff* est à prendre en compte. Un joueur bluffeur est un joueur qui va effectuer des actions qui ne sont pas en accord avec ses chances de gain. On part donc du principe qu'un joueur bluffeur est un joueur irrationnel.

Ces quatre valeurs vont nous permettre d'établir le profil d'un joueur au cours d'une partie. Elles seront mises à jour à chaque partie pour un profilage statique* et à chaque tour de jeu pour un profilage dynamique*.

Pour établir le profil global d'un joueur, avoir les données sur une partie n'est pas suffisant. Il faudra donc que l'intelligence artificielle joue plusieurs parties contre le joueur afin d'en déduire le profil global. Nous lancerons donc, pour chaque nouvel adversaire plusieurs parties pour établir leur profil puis, nous pourrons commencer à jouer de manière efficace une fois ce profil connu.

Dans le cadre du TER, nous nous sommes concentrés sur le profilage statique* afin qu'il soit le plus juste possible. Les parties suivantes traiteront donc uniquement du profilage statique*.

3.4 Phase de jeu

Le but de l'intelligence artificielle qui profile est d'établir un comportement pour pouvoir ensuite l'utiliser comme référence afin de tenter de gagner un maximum de parties et de jetons. Après la phase de profilage*, il y a donc la phase de jeu. Durant cette phase, l'intelligence artificielle ne profile plus le joueur adverse, mais tente d'augmenter ses gains. Durant cette phase, si l'intelligence artificielle commence à trop perdre, la phase de profilage* est relancée. En effet, un trop grand nombre de pertes peut être dû à un mauvais profil établi.

Chapitre 4

Mise en place du jeu et de l'interface graphique

4.1 Jeu

La première étape de développement a consisté à mettre en place le jeu de poker pour permettre le lancement de séries de parties pour le profilage*.

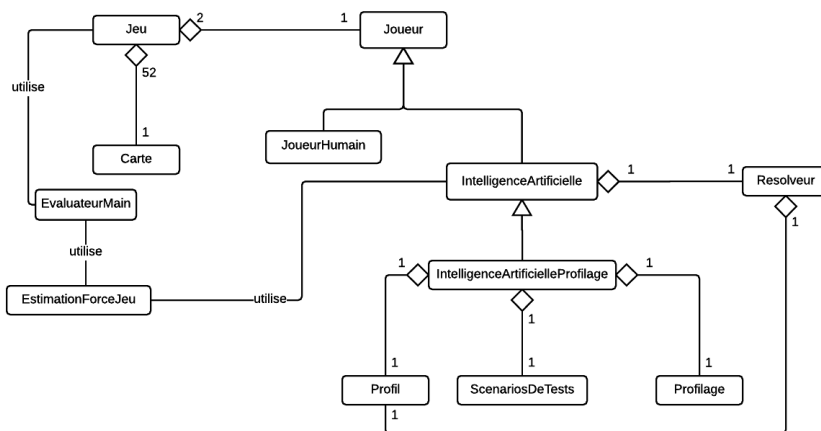


Figure 4.1: Diagramme de classe

La gestion globale du jeu de poker est composée des classes représentées de façon minimaliste dans la figure ci-dessus. Nous avons donc créé une classe `Jeu` gérant l'ensemble d'une partie. Puis nous avons créé différents types de joueurs, à savoir un joueur humain, une intelligence artificielle et une intelligence artificielle profilant son adversaire. Chacune des intelligences artificielles possède un résolveur chargé du calcul de la prochaine action à effectuer en fonction de la situation courante et de son calibrage.

Une classe `EstimationProba` permet à chaque intelligence artificielle de calculer ses chances de gain, et la classe `EvaluateurMain` permet de déterminer le gagnant à partir des cartes en jeu. Cet évaluateur est utilisé à chaque fin de partie, ainsi que dans l'estimateur qui lance des séries de parties tests.

Enfin, pour permettre le profilage*, plusieurs classes sont utilisées par l'intelligence artificielle correspondante. Il s'agit de la classe `Profilage` qui contient toutes les données à enregistrer dans les fichiers, et de la classe `ScenarioDeTests` qui s'occupe de la gestion des anciens profilages* effectués afin d'affiner au mieux la recherche du profil adverse.

4.2 Interface graphique

4.2.1 Paramétrer le jeu

Lors du lancement de l'application, une première boîte de dialogue permet à l'utilisateur d'indiquer l'ensemble des paramètres de la partie. La figure suivante donne un aperçu des données à paramétrer.

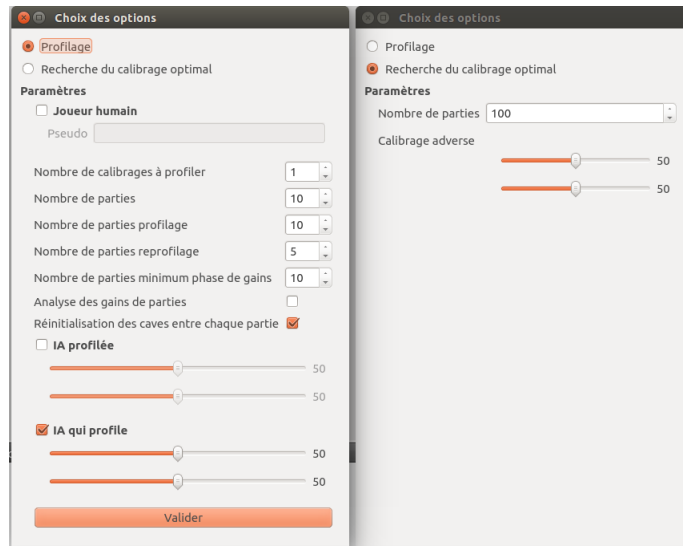


Figure 4.2: Fenêtre d'options de l'application

Comme on peut le voir, l'utilisateur peut lancer un profilage*, à savoir la recherche du comportement d'un adversaire, ou bien rechercher un calibrage optimal face à un adversaire donné.

Lorsque l'utilisateur sélectionne le mode profilage*, plusieurs données sont à paramétrer comme par exemple si c'est un humain qui joue ou non face à l'intelligence artificielle qui profile, le nombre de parties, ou encore les calibrages des différentes intelligences artificielles si besoin.

Concernant la recherche du calibrage optimal, il faut choisir le nombre de parties ainsi que le calibrage à battre.

4.2.2 Intelligence artificielle face à un joueur humain

Lors du démarrage de la phase de développement, nous avons conçu parallèlement au jeu, l'interface graphique afin de permettre à un joueur humain de pouvoir jouer face à une intelligence artificielle et de suivre le déroulement d'une partie.

Comme on peut le voir dans la figure suivante, la fenêtre contient l'ensemble des cartes du jeu, avec celles du joueur adverse, face cachées pendant la partie et dévoilées à la fin. L'ensemble des boutons* d'action sont mis à disposition du joueur avec une boîte permettant de donner le montant des mises. Les boutons*

sont désactivés lorsque c'est à l'adversaire de jouer. Une boîte de texte contient également l'ensemble des logs de la partie.

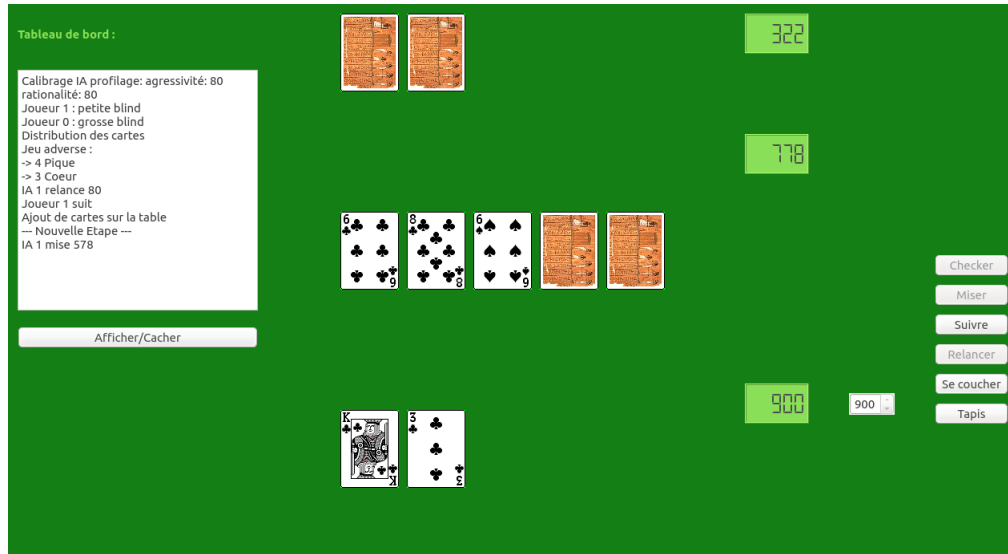


Figure 4.3: Interface humain contre intelligence artificielle

4.2.3 Intelligence artificielle face à une autre intelligence artificielle

Étant donné qu'il faut aussi pouvoir lancer un grand nombre de parties en faisant jouer deux intelligences artificielles l'une contre l'autre, il n'était pas utile que l'interface affiche le déroulement de l'ensemble des parties. C'est pourquoi nous avons conçu une deuxième interface, présentée dans la figure suivante, qui n'affiche alors que les données résultant de chaque partie dans un tableau. À la fin de chaque partie, une ligne est ajoutée au tableau indiquant le taux de profilage*, et les gains de l'intelligence artificielle sur la partie.

Parties en cours									
Calibrage IA profilée : 42% d'agressivité et 73% de rationalité									
	Scenarios en cours	Agressivité IA qui profile	Rationalité IA qui profile	Agressivité déduite	Rationalité déduite	Taux de similarité	Jeu agressif	Gains	Gain IA qui profile
1	1	84	50	64.8333	50	77.0833	0	499	1
2	1	18	50	71.4446	50	73.7777	0	100	1
3	1	65	50	47.6297	50	85.6851	0	100	1
4	1	100	50	51.1412	55.429	86.6439	0	40	1
5	1	39	50	41.073	54.3432	90.2081	0	-246	0
6	1	59	50	45.4525	53.6193	88.5834	0	-941	0
7	1	93	50	47.7701	56.2045	88.7172	0	0	0
8	1	21	50	45.4853	60.9823	92.2485	0	-411	0
9	1	54	50	48.7647	59.762	89.9987	0	-141	0
10	1	86	50	50.0558	60.9574	89.9508	0	400	1
11	0	80	80	50.0558	60.9574	89.9508	0	-605	0
12	0	80	80	50.0558	60.9574	89.9508	0	-400	0
13	0	80	80	50.0558	60.9574	89.9508	0	-80	0
14	0	80	80	50.0558	60.9574	89.9508	0	1207	1
15	0	80	80	50.0558	60.9574	89.9508	0	-780	0
	Agressivité déduite	Rationalité déduite	Taux de similarité	Nombre de jeux agressifs	Total gains	Nombre de parties gagnées			
1	50.0558	60.9574	89.9508	0	-1258	6			

Figure 4.4: Interface intelligence artificielle contre intelligence artificielle

Chapitre 5

Calcul du profil d'un joueur

Dans le but d'implémenter le profilage* d'un joueur, nous avons établi les formules de calcul de chacun des comportements.

5.1 Rationalité

Afin de calculer le taux de rationalité* d'un joueur pendant une partie, nous avons choisi de nous baser sur une courbe correspondant aux pourcentages théoriques de mises en fonction des chances de gains.

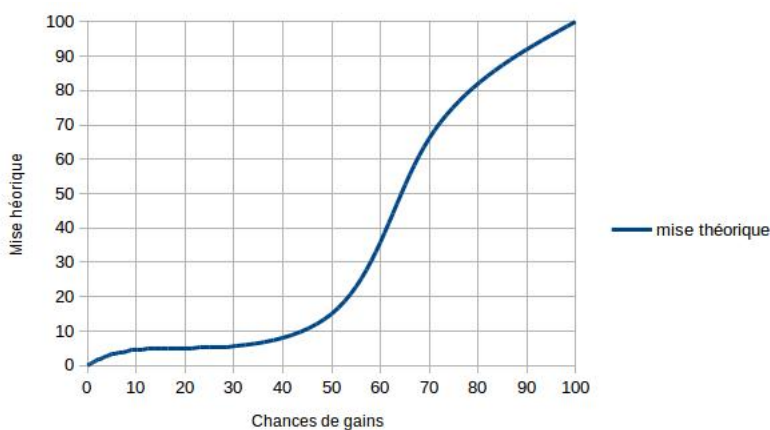


Figure 5.1: Mise théorique en fonction de la force du jeu

La figure ci-dessus représente la courbe sur laquelle nous nous basons pour calculer le pourcentage de jetons qu'un joueur devrait théoriquement miser par rapport à la cave* de départ, en fonction de ses chances de gain. Ainsi, on peut

voir qu'au départ, un joueur rationnel ne doit presque rien miser, à savoir pas plus de 15% de ses jetons, puis, à partir du moment où ses chances de gain sont supérieures à 50%, il doit commencer à miser beaucoup, de façon proportionnelle.

A partir de cette courbe, nous avons décidé de calculer la mise* que le joueur aurait dû théoriquement mettre en jeu en nous basant sur quatre paliers, comme nous pouvons le voir dans la formule suivante.

$$\begin{aligned} \text{Gain} &= \text{chances de gain du joueur,} \\ \text{Ra} &= \text{rationalité} \\ M_{th} &= \text{Mise théorique.} \end{aligned}$$

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

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.

Dans chacun de ses paliers, on calcule donc la mise* théorique correspondante, de façon proportionnelle aux chances de gain du joueur. Par exemple, 15% de chances de gain vont donner 5% de mise* théorique.

Après avoir calculé la mise* théorique, nous la comparons avec la mise* réelle du joueur. La rationalité* s'obtient alors avec la formule suivante.

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

5.2 Agressivité

Pour calculer le taux d'agressivité* d'un joueur, nous nous sommes basés sur le même système de paliers que pour la rationalité*. Nous prenons en compte trois paramètres, qui sont le taux de mises, le total des mises et la mise* la plus haute, exprimés en pourcentage de jetons par rapport à la cave* de départ. Ici, nous allons donc établir une formule à partir des trois paramètres listés précédemment.

Au départ, nous avons décidé que le critère le plus important serait la mise* la plus haute. Cependant, nous nous sommes vite aperçus que ce n'était pas le critère le plus représentatif de l'agressivité*. Par conséquent, la formule met maintenant en avant le total des mises effectuées.

Chaque palier est donc déterminé par le total des mises du joueur. De ce fait, le palier correspondant est celui pour lequel la mise* totale est comprise entre les bornes

inférieures et supérieures. Le taux d'agressivité* théorique est alors calculé en fonction du palier correspondant et des deux autres paramètres, nombre de mises et mise* la plus haute, qui ont pour ratios, respectivement ratio1 et ratio2.

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

Le calcul de la mise* théorique d'agressivité* se base sur la mise* totale du joueur à laquelle on ajoute les deux autres critères avec les poids correspondant aux ratios présentés précédemment.

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

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

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

Finalement, l'agressivité* est calculée comme suit.

$$y = (mtot + y)$$

5.3 Bluff

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 :

$$bluff = 100 - rationalité$$

5.4 Passivité

Nous avons décidé que le taux de passivité d'un joueur se calcule en fonction du nombre de ses actions check et suivre.

$$passivité = tauxChecks + tauxSuivis$$

Chapitre 6

Intelligence artificielle

6.1 Résolveur

6.1.1 Résolveur basique

Comme nous devons rapidement avoir une simple intelligence artificielle pour commencer à profiler le joueur adverse, nous avons commencé par implémenter une intelligence artificielle, ne se basant que sur ses chances de gain au préflop*.

Nous regardions donc tout d'abord si les chances de gain étaient supérieures à 75%. Si c'était le cas, l'intelligence artificielle se mettait alors à jouer de façon très agressive, en misant, relançant ou parfois même en faisant tapis dès que possible.

Si la force du jeu de l'intelligence artificielle était supérieure à 50%, elle suivait ou misait selon l'action possible.

Enfin, si les chances de gain étaient inférieures à 50%, l'intelligence artificielle avait le choix entre les actions suivre, checker* ou encore se coucher* si jamais le montant nécessaire pour rester en jeu était trop important. Notre intelligence artificielle était donc rationnelle mais n'était pas paramétrable, ce qui par la suite devait être corrigé. C'est pourquoi nous avons ensuite modifié son implémentation afin de pouvoir lui associer un comportement.

6.1.2 Résolveur avec calibrage

Afin de pouvoir tester facilement notre profilage*, et pouvoir faire varier les actions des intelligences artificielles, nous devons pouvoir choisir le comportement de chaque intelligence artificielle. Pour cela, nous avons permis le calibrage de l'intelligence artificielle, à partir d'un taux d'agressivité* et d'un taux de rationalité* donné.

Pour cela, nous devons donc faire en sorte que les actions choisies par le résolveur soient en accord avec les taux d'agressivité* et de rationalité* donnés. Nous avons donc choisi de mettre en place une liste d'actions possibles pour une action agressive et de même pour une action rationnelle. Après avoir calculé ces deux actions, nous

faisons par la suite un choix entre les deux actions, afin d'avoir des actions cohérentes par rapport aux deux taux. Nous détaillerons par la suite la façon dont nous avons implémenté le choix des différentes actions et la méthode de fusion.

Pour notre première version du calibrage, nous avons choisi de calculer une action pour chaque paramètre du calibrage, donc une action d'agressivité* et une action de rationalité*. Puis nous avons mis en place une méthode de sélection à partir des deux actions calculées et des taux de calibrage correspondants.

Pour le choix de l'action agressive, le calcul se base sur une mise* théorique agressive en fonction des mêmes paliers que pour le profilage*. Afin de respecter le pourcentage d'agressivité* donné à l'intelligence artificielle, nous avons dans un premier temps pensé à alterner entre action agressive et action passive en fonction du calibrage. En effet, le profilage* de l'agressivité* s'effectue selon trois critères : la fréquence de mises, la mise* la plus haute et le montant total des mises. De ce fait, afin que notre intelligence artificielle ne mise* pas de façon constante, nous avons pensé à générer un nombre aléatoire entre un et cent et le comparer au taux d'agressivité* donné pour choisir l'action. Par exemple, si le calibrage d'agressivité* vaut 70%, alors si le nombre aléatoire est compris entre 0 et 70, on effectue une action de mise* cohérente selon la mise* théorique, une mise*, une relance ou un suivi selon la situation, et si au contraire le nombre est compris entre 70 et 100, on choisit une action passive.

Le choix de l'action rationnelle était basé sur le même principe. La mise* théorique de rationalité* est calculée en fonction des chances de gain de l'intelligence artificielle, puis de la même manière que pour l'agressivité*, un nombre aléatoire détermine si on effectue une action rationnelle ou non. Si on a une valeur de rationalité* entre zéro et le taux, alors on choisit une action rationnelle, qui correspond à la mise* théorique et qui varie selon la situation actuelle du jeu. Sinon, on liste l'ensemble des actions irrationnelles, comme une mise* en dessous ou en dessus de la mise* théorique, puis pour renforcer l'irrationalité, et donc l'imprévisibilité du joueur, la sélection de l'action parmi toutes celles considérées comme irrationnelles se fait de façon aléatoire.

Enfin, une fois les deux actions calculées, une sélection aléatoire est faite entre ces deux actions, avec les chances de sélection proportionnelles aux valeurs de calibrage.

Par la suite nous avons revu le calibrage de l'agressivité* qui n'était pas assez précis. En effet, 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 parfois trop éloignés du calibrage de départ.

Pour pallier ce problème, nous avons mis en place le calcul d'une action de mise* correspondant 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, à la place de la mise* la plus haute. A chaque tour, le résolveur calcule donc la mise* théorique à effectuer selon les nouveaux paliers.

Agressivité	Mise totale
0 – 50	0 – 25 exclu
50 – 80	25 – 60 exclu
80 – 100	60 – 100 exclu
100	100

Figure 6.1: Paliers d'agressivité et mise totale en fonction

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.

Exemple : 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*, alors elle mise* directement ce qui 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. Quant la relance est inférieure ou égale 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 la sélection 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. Lorsque les deux actions sont identiques, l'action est prise. Si ce sont toutes les deux des mises/relances, on prend la moitié du nombre de jetons, proportionnellement aux pourcentages de calibrage.

Si l'on considère une intelligence artificielle ayant un taux d'agressivité* égal à 70% et un taux de rationalité* égal à 20%, et des actions de mises choisies dans

chacun des cas, avec une mise* de 300 jetons pour l'action correspondant au taux d'agressivité* et une mise* de 50 jetons pour l'action de rationalité*. Nous prenons donc une mise* comprise entre le minimum et le maximum. Cette mise* aura donc plus de chances d'être proche de la mise* du taux le plus fort, c'est à dire ici, proche de 300. Après avoir effectué le calcul, on obtient une mise* égale à 244 jetons. Comme dit précédemment, ce résultat est proche de 300 puisque l'intelligence artificielle est calibrée à 70% d'agressivité* et à seulement 20% de rationalité*.

Le tableau suivant présente des exemples de combinaisons possibles, avec par conséquent des actions agressives et rationnelles différentes ainsi que l'action finale déduite.

Action agressive	Action rationnelle	Action finale
Checker	Se coucher	Checker
Checker	Suivre	Tirage aléatoire
Checker	Miser	Miser (fusion)
Suivre	Relancer	Si montant \geq relance minimum, Relancer Sinon, Suivre
Suivre	Se coucher	Tirage aléatoire
Relancer	Se coucher	Suivre
Miser	Se coucher	Miser (fusion)
Tapis	Miser	Miser (fusion)
Tapis	Relancer	Relancer (fusion)
Tapis	Se coucher	Si peut suivre, Suivre Sinon, Miser cave/2
Tapis	Checker	Si peut suivre, Suivre Sinon, Miser cave/2

Figure 6.2: Choix de l'action à effectuer en fonction des actions d'agressivité* et de rationalité

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 1 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.

En reprenant les valeurs de calibrage de l'exemple précédent, à savoir 70% d'agressivité* et 20% de rationalité*, et les actions suivantes, suivre pour l'action agressive et se coucher* pour l'action rationnelle. Dans ce cas, l'action à effectuer sera choisie aléatoirement entre les deux. Pour cela, un nombre aléatoire est généré dans l'intervalle $[0 - \text{somme}]$, où somme correspond à l'addition des taux d'agressivité* et de rationalité*. Dans ce cas, il s'agira d'un nombre entre 0

et 90. Si le nombre obtenu est compris entre 0 et la valeur du taux d'agressivité*, alors on choisit l'action agressive, sinon on prend l'action rationnelle.

6.2 Estimation des chances de gain

Pour calculer les chances de gain d'un joueur au préflop*, nous avons récupéré une calculette de probabilité permettant de connaître la force de la main pour chaque paire de cartes, lors d'une partie à deux joueurs. Nous avons donc utilisé cette calculette afin de déterminer au préflop* les chances de gain d'un joueur. Par la suite, nous devons aussi pouvoir calculer les chances de gain aux étapes suivantes, en fonction des cartes découvertes sur la table.

Au départ, il était prévu que nous récupérions une calculette de probabilités afin de calculer les chances de gain d'un joueur aux étapes flop*, turn* et river*. Finalement, nous avons dû développer notre propre calculette de probabilités.

Nous avons alors choisi de nous baser sur une estimation des chances de gain d'une main en effectuant X tirages aléatoires de mains. Après chaque tirage, nous évaluons les deux mains pour savoir quelle serait la gagnante. De ce fait, nous obtenons les chances de gain de l'intelligence artificielle. Lorsqu'une confrontation entre deux mains nous donnait comme résultat une égalité, nous avons choisi de ne pas la compter comme un gain.

Nous avons donc effectué plusieurs tests avec plusieurs nombres de tirages différents. Nous comparions les résultats avec des calculettes de probabilités en ligne [6], ce qui nous permettait de vérifier l'exactitude des résultats.

Comme on peut le voir dans l'exemple suivant, nous nous sommes vite rendus compte que 10 000 tirages aléatoires permettaient d'avoir des chances de gain précises à plus ou moins 5%. Si nous augmentions encore le nombre de tirage, les chances de gains étaient à peine plus précises et cela augmentait de façon significative le temps de calcul.

La figure ci-dessous montre les différentes chances de gains obtenues en fonction du nombre de tests lancés, sur un même instant d'une partie, à savoir, la main composée d'un as de pique et d'un valet de trèfle, à l'étape flop*, avec les cartes dix de trèfle, deux de cœur et trois de pique découvertes. Les chances de gain que l'on cherche à atteindre sont de 52,35%. On peut constater que lorsque nous lançons cent tests, le résultat obtenu est encore trop éloigné du résultat attendu. Cependant, lorsque nous lançons mille tests, ce résultat est déjà plus proche de la vérité. De même, lorsque l'on lance cinq mille tests, on se rapproche encore du résultat attendu. Nous avons ensuite choisi de lancer dix mille tests et nous nous sommes rendus compte que les résultats obtenus étaient en moyenne très précis, à plus ou moins 2%. Nous avons ensuite choisi de lancer les tests sur cinquante mille parties afin de savoir si l'on se rapprochait encore plus du résultat attendu. Nous nous sommes alors rendus compte que la différence n'était pas énorme et ne dépassait pratiquement jamais les 1%. Par conséquent, nous avons opté pour le lancement de dix mille tests afin d'obtenir de bons résultats.

Nombre de tests	Chances de gain obtenues
50000	51.616%
10000	51.17%
5000	50.38%
1000	50%
100	48%
Calculatrice de probabilités en ligne	52,35%

Figure 6.3: Chances de gains obtenues en fonction du nombre de tests effectués

L'implémentation de cette fonctionnalité ayant considérablement ralenti l'application, nous avons choisi de paralléliser l'estimation des chances de gain. En effet, à chaque nouvelle étape, 10 000 tirages étaient effectués, avec 10 000 comparaisons entre la main effective et celle tirée aléatoirement. De plus, lorsque nous faisons jouer deux intelligences artificielles l'une contre l'autre, le temps d'attente pour avoir le résultat de la partie était multiplié par deux. Comme notre but final n'était pas de lancer une partie mais beaucoup plus, nous avons rapidement choisi de rendre cette exécution plus rapide en parallélisant les calculs.

Chapitre 7

Méthodes de profilage

Afin d'étudier le comportement d'un joueur adverse, il nous a été conseillé de mettre en place des scénarios de tests. Lorsque nous souhaitons établir le profil du joueur, l'intelligence artificielle se base sur des scénarios prédéfinis au départ, puis mis à jour au fur et à mesure du profilage*.

7.1 Scénarios de tests

7.1.1 Mise en place des scénarios

Afin de pouvoir évaluer le profilage* effectué par l'intelligence artificielle, il nous a été conseillé d'étudier la différence entre l'action attendue du joueur adverse, et son action réelle. De ce fait, il est possible de déterminer facilement l'efficacité du profilage*, en analysant la différence entre l'attendu et le réel.

Pour que l'intelligence artificielle qui effectue le profil puisse connaître le comportement attendu d'un joueur à partir d'une situation donnée, nous avons établi l'ensemble des scénarios possibles. Chaque scénario se base sur l'agressivité* de l'intelligence artificielle qui profile, ainsi que sur ses chances de gain. Pour chacune de ces situations, le comportement attendu rationnellement par le joueur adverse a été établi. Il est exprimé sous la forme d'un taux d'agressivité*. La différence entre l'action attendue et réelle peut alors être calculée.

Agressivité IA	Chances de gain IA	Agressivité attendue	Rationalité attendue
75-100	75-100	0	50
75-100	50-75	0-25	50
75-100	35-50	50-75	50
75-100	0-35	75-99	50
50-75	75-100	0	50
50-75	50-75	25-50	50
50-75	35-50	50-75	50
50-75	0-35	75-99	50
35-50	75-100	0-25	50
35-50	50-75	25-50	50
35-50	35-50	50-75	50
35-50	0-35	99-100	50
0-35	75-100	25-50	50
0-35	50-75	50-75	50
0-35	35-50	75-99	50
0-35	0-35	99-100	50

Figure 7.1: Actions attendues

Le tableau ci-dessus représente les actions attendues de base, sans que l'on ait encore commencé à profiler le joueur. Comme on peut le voir, nous avons choisi de mettre par défaut la rationalité* attendue à 50%. Pour le taux d'agressivité*, nous le calculons en fonction des chances de gain de l'intelligence artificielle qui profile ainsi qu'en fonction de son taux d'agressivité*. Prenons par exemple une intelligence artificielle calibrée à 80% agressive et ayant 80% de chances de gagner, on s'attend dans ce cas là à ce que le joueur adverse se couche. En effet, si l'intelligence artificielle qui profile a 80% de chances de gagner, il y a de fortes chances que le joueur adverse ait un jeu moins bon donc, il y a de fortes chances que le joueur adverse se couche.

Dans une première version des scénarios de tests, nous avons choisi de tester les réactions du joueur adverse face à tous les calibrages possibles, et de lancer un nombre de parties prédéfini pour chacun de ces calibrages. Ainsi, l'intelligence artificielle commençait avec une agressivité* et une rationalité* égales à 100% puis, on décrémençait chacun des paramètres du calibrage de 25% pour obtenir toutes les combinaisons possibles.

Nous avons choisi de lancer cinq parties par calibrage. Cependant, nous nous sommes vite rendus compte que cela voulait dire lancer 125 parties uniquement pour la première phase de profilage*, ce qui était beaucoup trop long. En effet, pour effectuer un bon profilage*, il est primordial de déterminer rapidement le comportement de l'adversaire afin d'exploiter les résultats le plus tôt possible.

Finalement, pour analyser la réaction du joueur adverse face à différents types de comportements et différentes situations de jeu, le calibrage de l'intelligence artificielle qui profile est tiré aléatoirement à chaque nouvelle partie. De ce fait, nous obtenons un profil global représentant le joueur profilé. Nous calculons ensuite une moyenne de

toutes les parties qui donnera le profil global déduit du joueur.

7.1.2 Analyse des résultats

Lors du lancement des parties de profilage*, une ligne de résultats est ajoutée dans un fichier csv correspondant. Cette ligne contient la situation du jeu, calibrage de l'intelligence artificielle et chances de gain, les taux d'agressivité* et de rationalité* attendus, les taux réels et la distance entre ces deux. Enfin une moyenne du profil attendu est mise à jour à chaque ajout, c'est elle qui va donner le profilage* obtenu au bout de X parties.

Cela va donc permettre le lancement de plusieurs parties intelligence artificielle contre intelligence artificielle, pour obtenir un profilage* rapidement. Suite à ces résultats, il est alors possible de réaliser des courbes permettant d'observer la convergence entre le profil établi, et le calibrage réel de l'intelligence artificielle profilée. La distance doit alors être de plus en plus proche de zéro.

Nous pouvons donc ajuster le nombre optimal de parties de la phase de profilage* en fonction des taux de similarités écrits dans le fichier csv présenté précédemment.

Suite à plusieurs tests, nous avons observé qu'au bout de 10 ou 15 parties, nous arrivions à obtenir un profilage* satisfaisant, et que celui-ci ne tendait pas à s'améliorer, même en jouant plus de parties. En effet, si l'on regarde la courbe suivante représentant le taux de similarité entre le profil réel et le profil établi en fonction du nombre de parties, on constate que le taux de similarité a tendance à augmenter jusqu'à atteindre un certain niveau, à savoir pratiquement 100% de similarité dans l'exemple suivant. À partir du moment où ce taux est atteint, le taux de similarité reste constant. Par conséquent, lancer plus de dix parties ne permet pas d'obtenir un meilleur profil.

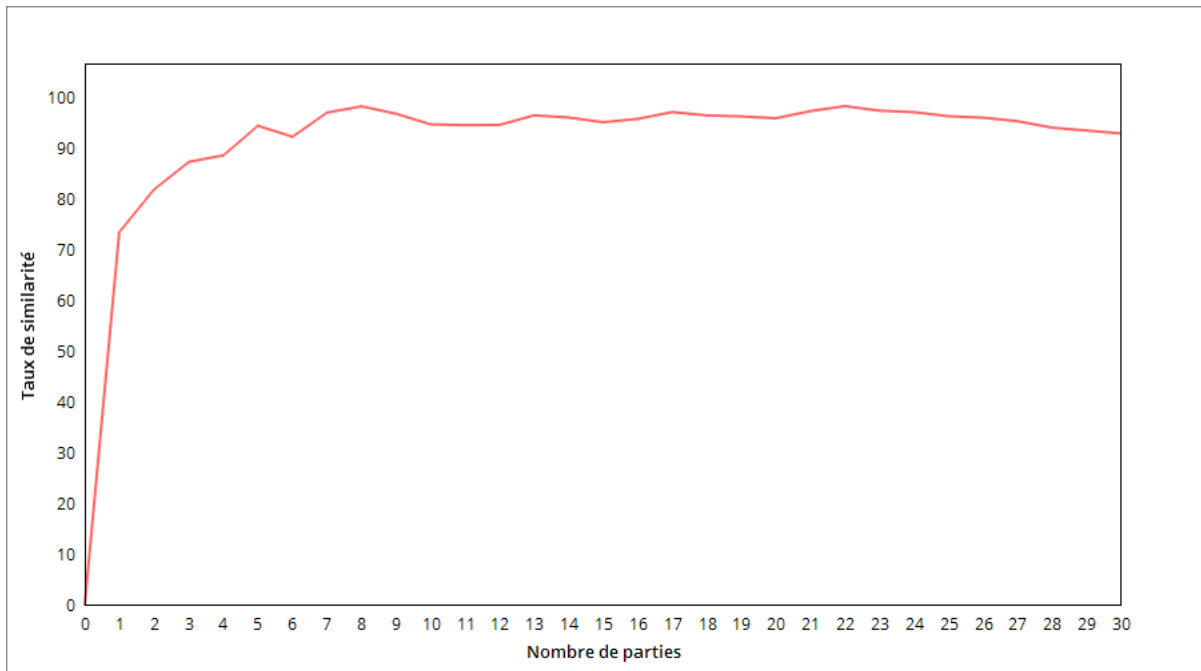


Figure 7.2: Taux de similarité du profilage en fonction du nombre de partie

Afin de pouvoir obtenir des résultats facilement observables et pouvoir les étudier, nous avons également produit des graphes comparant le profilage* établi et le calibrage à déterminer du joueur en fonction du nombre de parties. La figure suivante correspond à un exemple de graphe dans lequel un adversaire agressif à 52% et rationnel à 64% est profilé. Comme nous pouvons le voir, les courbes d'agressivité* et de rationalité* correspondant au profil global établi au fur et à mesure des parties ont tendance à converger vers le profil réel.

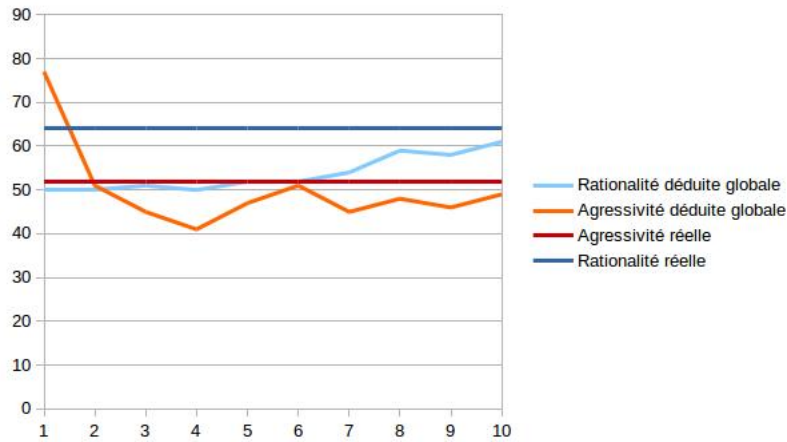


Figure 7.3: Profil établi en fonction du nombre de parties

Finalement, après avoir lancé un nombre important de parties, nous avons conçu le tableau suivant représentant le taux de similarité moyen obtenu pour chaque palier.

Palier d'agressivité	Palier de rationalité	Taux de similarité moyen
0-40	0-40	69%
0-40	40-60	85%
0-40	60-100	82%
40-60	0-40	76%
40-60	40-60	89%
40-60	60-100	87%
60-100	0-40	70%
60-100	40-60	85%
60-100	60-100	78%

Figure 7.4: Taux de similarité des profils établis en fonction des paliers

Comme on peut le constater, on profile assez bien tous les types de joueurs. Les joueurs que nous arrivons le moins à profiler sont ceux ayant un taux de rationalité* et un taux d'agressivité* compris entre 0% et 40%. En effet, il est difficile de profiler un joueur qui aura tendance à jouer de manière irrationnelle, donc de manière très imprévisible. Ce type de joueur pourra alors miser fortement lorsqu'il n'a pas un bon jeu et miser peu lorsqu'il a un bon jeu. De plus, ce joueur étant à la fois irrationnel et peu agressif, nous obtenons des résultats pas forcément en accord avec les deux paramètres. Si le joueur a un mauvais jeu et qu'il joue de manière irrationnelle, il va miser fortement cependant, il n'est pas censé être agressif donc son taux d'agressivité* s'en trouvera augmenté.

7.2 Établissement des profils attendus

Pour calculer une action attendue, c'est à dire un taux d'agressivité* et de rationalité* attendu pour le joueur, l'intelligence artificielle utilise lors des premières parties les scénarios de tests établis.

Une fois le profilage* lancé, et les premiers résultats obtenus, la méthode de profilage* peut alors commencer à se servir des situations précédentes pour calculer l'action attendue d'un joueur de façon plus précise. En effet, selon la situation, il est possible de déterminer quelle va être l'action du joueur en se référant aux scénarios de tests conçus, qui sont communs à tous les joueurs, mais également en se référant aux parties précédentes, si l'on a déjà testé sa réaction précédemment dans une situation équivalente. Les scénarios de tests sont alors mis à jour de façon dynamique.

Pour reconnaître une situation équivalente, nous considérons le taux d'agressivité* du joueur adverse, donc l'intelligence artificielle qui profile, et des chances de gain du joueur. A partir de cette méthode, la valeur d'une action réelle dans une situation va permettre de donner la valeur d'une action attendue dans une prochaine partie. Cette méthode va ici nous permettre de profiler le joueur plus rapidement, en réutilisant les résultats précédents.

Pour effectuer cette réutilisation, plusieurs méthodes ont été considérées. Tout d'abord, celle qui cherche la situation la plus proche, selon l'agressivité* du profileur et les chances de gain, sans considérer le taux de similarité entre les deux situations. C'est à dire que dans cette version, les valeurs de l'action attendue correspondent aux valeurs des scénarios de tests uniquement pour la première partie. Par la suite, c'est la valeur précédente la plus proche qui est prise en compte, même si la situation n'est pas forcément similaire.

Nous nous sommes ensuite rendus compte qu'il n'était pas optimal de reprendre une valeur précédente lorsque les paramètres de la partie étaient trop éloignés. Une seconde méthode a donc été de limiter la réutilisation des actions réelles comme actions attendues en considérant l'équivalence entre les parties. Toujours en prenant en compte les valeurs d'agressivité* et des chances de gain, les pourcentages doivent alors ne varier que de 20% maximum par exemple, pour que l'on puisse considérer la situation comme équivalente. Ici, on prend toujours la situation la plus proche, mais seulement parmi les parties dites équivalentes. Par conséquent, dans le cas où les valeurs sont trop éloignées, on se réfère aux scénarios de tests de départ.

Nous avons également fait des tests en prenant en compte la situation dans laquelle la distance est la plus petite possible, c'est à dire prendre une situation dans laquelle l'intelligence artificielle a réussi à profiler correctement le joueur. Nous nous sommes rendus compte que la différence entre les résultats n'était pas significative, et qu'il valait mieux se baser sur les paramètres de la partie, à savoir le calibrage et les chances de gain de l'intelligence artificielle qui profile.

7.3 Vérification du profil établi

Une fois le profil du joueur adverse établi, l'intelligence artificielle a pour but de jouer de façon à gagner le plus de jetons possible. Cependant, si le joueur adverse n'a pas été bien profilé, avec par exemple un profil à 40% seulement, l'intelligence artificielle va jouer en imaginant faire face à un type d'adversaire qui en fait n'est pas le bon. Ainsi, nous avons choisi de mettre en place la possibilité de reprendre le profil afin de l'affiner si besoin.

Par conséquent, si lors de la phase de jeu et à partir de dix parties jouées, l'intelligence artificielle perd des jetons ou bien gagne moins de jetons que lors de la phase de scénarios de tests, la phase de profilage* du joueur reprend. Cette phase de reprise du profilage* dure cinq parties. En effet, après avoir testé avec plusieurs nombres de parties différents, nous nous sommes rendus compte que cinq parties était largement suffisantes pour corriger le comportement attribué à l'adversaire. Par la suite, l'intelligence artificielle reprend sa phase de jeu pour gagner et, si au bout de dix parties, les gains ne sont toujours pas bons, l'intelligence artificielle recommence à profiler le joueur, et cætera.

Chapitre 8

Analyse des gains de parties

Une fois un profil établi, nous pouvons alors faire jouer l'intelligence artificielle en fonction des résultats obtenus, dans le but de gagner le plus de jetons possibles. Lors de cette étape, le profilage* est interrompu, et on utilise le comportement déduit final pour calibrer l'intelligence artificielle de façon à ce qu'elle puisse gagner face à ce type d'adversaire.

8.1 Un premier calibrage inefficace

8.1.1 Mise en place du calibrage

Pour une première version, nous avons choisi de seulement prendre en compte le profil final établi et d'attribuer un calibrage à l'intelligence artificielle qui profile selon divers paliers, pour lui permettre de prendre le dessus. Ici on prend en compte trois paliers pour l'agressivité* ainsi que pour la rationalité*. De 0 à 40% pour le premier palier, 40 à 60% pour une valeur moyenne puis de 60 à 100% pour le palier haut. Nous avons alors déduit les calibrages correspondants à chaque palier du joueur profilé. Les résultats sont présentés dans le tableau suivant.

Agressivité déduite	Rationalité déduite	Agressivité IA	Rationalité IA
0-40	0-40	70-100	50
0-40	40-60	50-70	50
0-40	60-100	70-100	30-50
40-60	0-40	70-100	50
40-60	40-60	70-100	30-50
40-60	60-100	50-70	70-90
60-100	0-40	50-70	70-90
60-100	40-60	70-100	50-70
60-100	60-100	50	90-100

Figure 8.1: Calibrage de l'intelligence artificielle en fonction du profil adverse

Le but de notre intelligence artificielle étant, une fois un profilage* correctement établi, de pouvoir augmenter son gain de parties mais aussi son gain d'argent. Nous avons donc renseigné dans l'application le nombre de parties au bout desquelles l'intelligence artificielle va pouvoir arrêter le profilage*, et donc adapter elle-même son calibrage pour pouvoir gagner un maximum de parties.

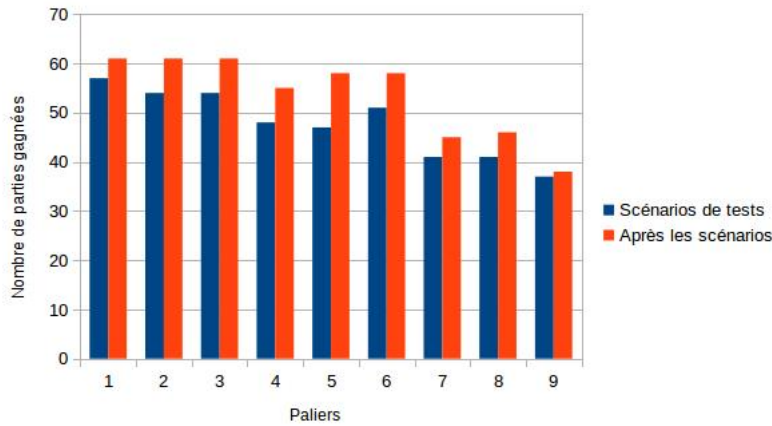
8.1.2 Analyse des résultats

Afin de pouvoir étudier les gains de l'intelligence artificielle avant et après le profilage*, nous avons alors ajouté dans les fichiers résultats le gain de parties et d'argent pour chacune des parties. Ceci nous a permis de pouvoir obtenir un ratio du nombre de parties gagnées sur le nombre de parties jouées pendant le profilage*, puis pendant la phase de jeu. De même pour le nombre de jetons obtenus. Nous avons donc ajouté les lignes suivantes à la fin de chaque fichier contenant le profil établi. Ces données sont mises à jour à chaque nouvelle partie.

Nombre de parties:	30
Nombre de parties gagnées par l'IA pendant profilage :	6 sur 10
Nombre de parties gagnées par l'IA pendant jeu :	15 sur 20
Gains de l'IA pendant profilage :	-924
Gains de l'IA pendant jeu :	3530

Figure 8.2: Données situées à la fin d'un fichier contenant le profil établi d'un joueur

La figure suivante représente le nombre de parties gagnées pendant les scénarios de tests et le nombre de parties gagnées après, pour chaque type d'adversaire. Comme on peut le constater, le nombre de parties gagnées après le déroulement des scénarios de tests, donc lorsque l'intelligence artificielle qui profile joue en fonction des résultats obtenus, est meilleur. En effet, celle-ci gagne souvent plus de parties que lors de la période de scénarios de tests.



Palier	Agressivité	Rationalité
1	0-40	0-40
2	0-40	40-60
3	0-40	60-100
4	40-60	0-40
5	40-60	40-60
6	40-60	60-100
7	60-100	0-40
8	60-100	40-60
9	60-100	60-100

Figure 8.3: Nombre de parties gagnées pendant les scénarios de tests et après

Même si nous gagnons plus de parties que lors de la phase de scénarios de tests, nous avons remarqué que les gains de jetons sont nettement moins bons. En effet, comme on peut le voir dans la figure suivante représentant le nombre de jetons gagnés en moyenne pour chaque partie, les gains sont moins bons lors de la phase de jeu. L'intelligence artificielle perd même des jetons dans la majorité des paliers. Ces mauvais résultats peuvent s'expliquer par le fait que nous avons mis en place des taux d'agressivité* élevés en réponse aux différents profils établis. De ce fait, lorsque l'intelligence artificielle gagne des jetons, elle n'en gagne pas énormément car le joueur adverse va avoir tendance à se coucher* rapidement étant donné qu'il sera effrayé par le fort taux d'agressivité*. Cependant, lorsque l'intelligence artificielle perd des parties, étant agressive, elle aura auparavant mis en jeu beaucoup de jetons. Par conséquent, elle aura tendance à perdre plus de jetons qu'elle n'en gagne.

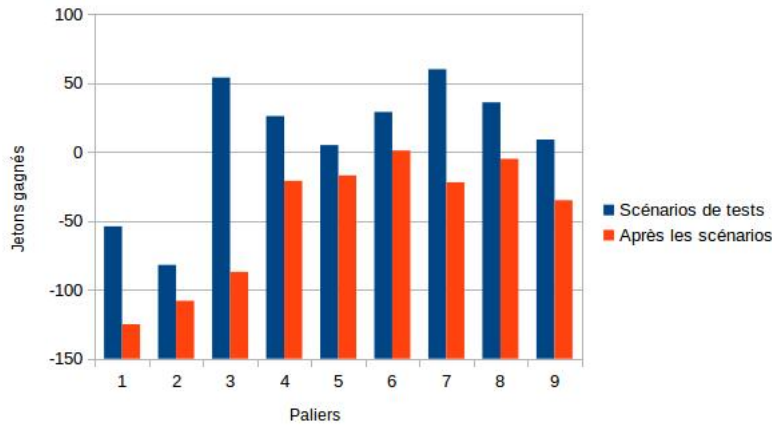


Figure 8.4: Nombre de jetons gagnés pendant les scénarios de tests et après

8.2 Recherche du calibrage optimal

Puisque les premières réponses face à un profilage* établi pour gagner le plus de parties et de jetons possibles n'étaient pas concluantes, nous avons établi des tests permettant de déterminer un calibrage optimal.

Pour cela, nous avons lancé une série de parties face à un profil donné en faisant varier le calibrage de l'intelligence artificielle qui profile. Nous avons fait varier chacun des paramètres, agressivité* et rationalité*, de 15 en 15 de la façon suivante.

15 - 30 - 45 - 60 - 75 - 90 - 100

Nous avons ensuite effectué le produit cartésien entre les deux listes pour obtenir tous les calibrages possibles, par exemple 15-15, 15-30, et cætera. Puis pour chaque calibrage, nous avons lancé 100 parties.

Nous récupérons à la fin des 100 parties le taux de parties gagnées ainsi que le nombre de jetons gagnés ou perdus.

La figure suivante correspond au résultat de la recherche du calibrage optimal face à un joueur 80% agressif et 80% rationnel. Ainsi, la courbe bleue correspond au taux de parties gagnées alors que la seconde courbe correspond au nombre de jetons gagnés au total pendant les cent parties lancées par calibrage. L'abscisse correspond à l'ensemble des calibrages testés. Ainsi, le calibrage numéro un correspond au calibrage 15-15 alors que le dernier calibrage, à savoir le numéro 49, correspond au calibrage 100-100.

Comme on peut le voir, la courbe du nombre de parties gagnées a tendance à croître de manière significative à partir du moment où on arrive aux calibrages ayant

75% d'agressivité* et plus. En effet, le joueur adverse est calibré à 80% d'agressivité* et 80% de rationalité*, par conséquent, les joueurs étant moins agressifs gagnent moins souvent. Concernant les jetons gagnés, on constate que la courbe dans son ensemble ne semble pas avoir un comportement défini.

On peut voir que les points les plus intéressants sont le point correspondant au calibrage numéro six, à savoir, un comportement 15% agressif et 90% rationnel, et le point correspondant au calibrage numéro 46, à savoir, un comportement 100% agressif et 60% rationnel. Le calibrage correspondant au point numéro six permet à l'intelligence artificielle de gagner environ 45% des parties et 15 000 jetons. Cependant le point numéro 46 semble plus intéressant. En effet, avec ce calibrage, l'intelligence artificielle gagne, lors de sa phase de jeu, un nombre important de parties, à savoir, plus de 65% de parties mais aussi un nombre important de jetons, à savoir, plus de 13 000 jetons. Finalement, nous avons choisi de calibrer notre intelligence artificielle à 100% d'agressivité* et 60% de rationalité* face à un joueur profilé avec un taux d'agressivité* compris entre 60% et 100% et un taux de rationalité* compris entre 60% et 100%. Cela correspond donc au calibrage numéro 46 décrit précédemment.

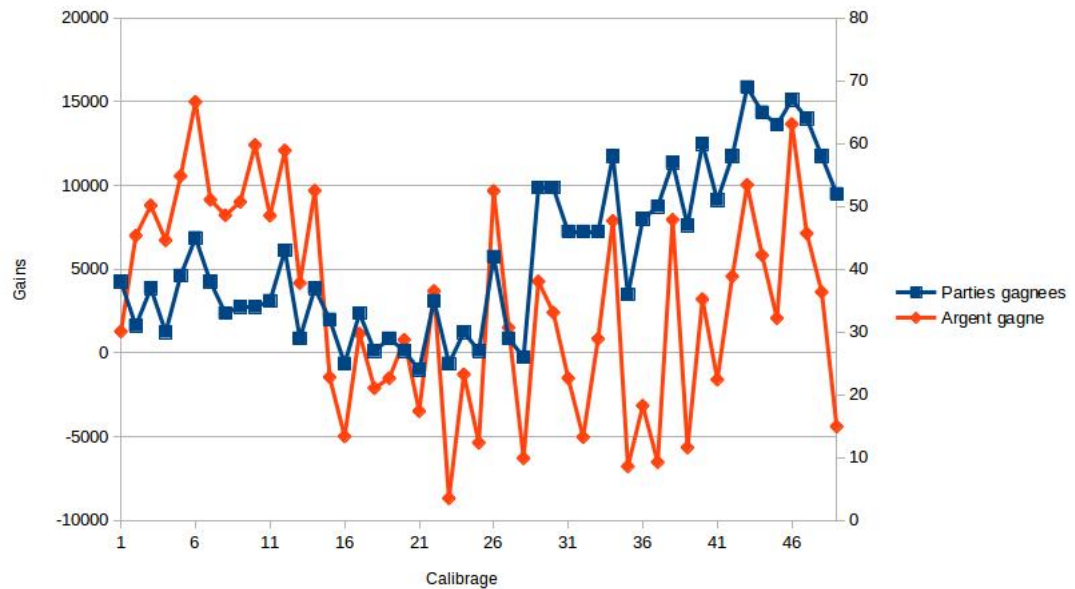


Figure 8.5: Gains en fonction du calibrage face à un joueur 80% agressif et 80% rationnel

Après avoir effectué une première recherche concluante du calibrage optimal pour un palier donné, nous avons choisi de lancer cette recherche pour chacun des autres paliers, afin de permettre à notre intelligence artificielle d'avoir le meilleur jeu possible dans n'importe quelle situation.

Nous avons finalement mis à jour notre tableau et choisi les calibrages suivants pour les différents paliers présentés ci-dessous.

Agressivité déduite	Rationalité déduite	Agressivité IA	Rationalité IA
0-40	0-40	30	30
0-40	40-60	30	90-100
0-40	60-100	15	90-100
40-60	0-40	30	90
40-60	40-60	70-80	70-80
40-60	60-100	90	30
60-100	0-40	30	90-100
60-100	40-60	15	75-100
60-100	60-100	100	60

Figure 8.6: Calibrage final de l'intelligence artificielle en fonction du profil adverse

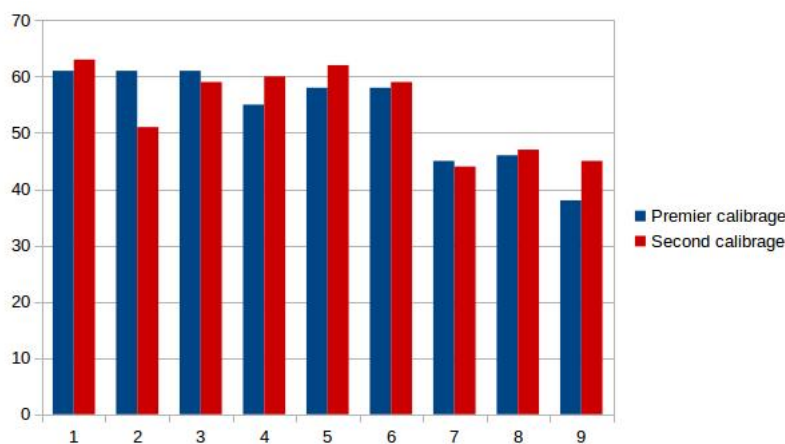


Figure 8.7: Nombre de parties gagnées en fonction du premier calibrage établi et du second calibrage calculé

Comme on peut le voir dans la figure ci-dessus, le changement de gains de parties par rapport aux résultats précédents n'est pas énorme voire moins bon dans certains paliers. En effet, les données bleues et rouges, respectivement le premier calibrage établi inefficace et le second calibrage résultant de la recherche du calibrage optimal sont sensiblement égales. Nous avons choisi de nous concentrer davantage sur les gains de jetons. Ainsi, même si l'intelligence artificielle qui profile ne gagne pas plus de parties, elle gagne plus de jetons qu'auparavant.

Le diagramme suivant représente les jetons gagnés en fonction du premier calibrage établi, en bleu, et du second calibrage établi en fonction des données collectées pendant la recherche du calibrage optimal, en rouge. On constate rapidement que le nombre de jetons gagnés en moyenne sur une partie est nettement meilleur qu'auparavant. Notamment dans le premier palier, correspondant à un jeu face à une intelligence artificielle profilée avec une agressivité* et une rationalité* comprises entre 0% et 40%, où le nombre de jetons gagnés passe d'une perte de plus de 120 jetons à un gain d'une trentaine de jetons par partie.

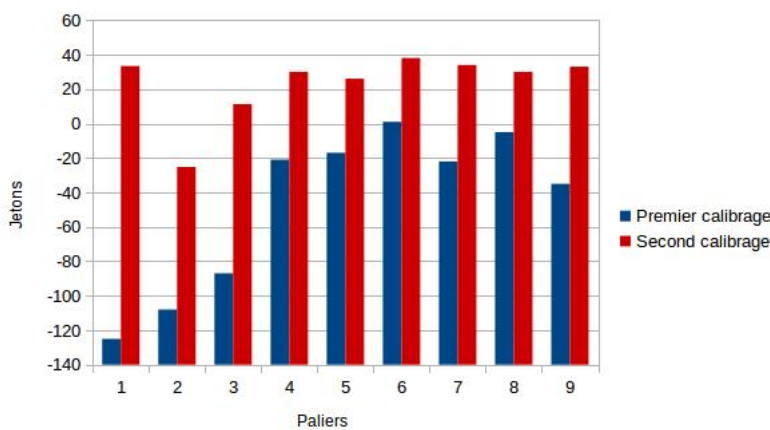


Figure 8.8: Nombre de parties gagnées en fonction des deux calibrages

La recherche de calibrages optimaux pour chacun des palier nous a donc permis d'améliorer de façon significative le nombre de jetons gagnés par partie après établissement du profil de l'adversaire.

8.3 Recherche du meilleur type de joueur

Après plusieurs tests, nous avons cherché à déterminer le meilleur type de joueur, c'est à dire le calibrage qui va réussir à gagner face à un maximum de comportements différents.

Ainsi, nous nous sommes rendus compte que ce meilleur calibrage correspond à un comportement 80% agressif et 80% rationnel. En effet, comme on peut le voir dans la figure suivante représentant les gains de ce type de comportement face à différents autres joueurs, ce calibrage permet de gagner dans la majorité des cas. Cependant, on constate que face à un joueur très agressif et parfois irrationnel, on a tendance à perdre des jetons. En effet, ce type de joueur, de part son irrationnalité un peu plus importante, aura tendance à aller au bout de certaines parties, donc à prendre des risques, dans lesquelles un joueur très rationnel va se coucher*. Ainsi, dans les cas où le joueur plus irrationnel a un jeu moyen, il va tenter de gagner en allant le plus

loin possible dans la partie. Par conséquent, ce joueur plus irrationnel possède cette part de risque qui lui permet de gagner parfois plus alors qu'un joueur rationnel aurait abandonné.

Comportement adverse (agressivité - rationalité)	Gains sur 100 parties (jetons)	Parties gagnées
20-20	18 550	76%
20-50	11 130	70%
20-80	20 841	75%
50-20	16 241	77%
50-50	24 336	84%
50-80	8 054	69%
80-20	3 180	47%
80-50	-609	46%
80-80	± 0	50%

Figure 8.9: Résultats sur 100 parties face à un comportement 80% agressif et 80% rationnel

Suite à cette observation, il est intéressant de chercher ce degré d'irrationnalité supplémentaire permettant à un joueur très agressif et très rationnel de battre d'autres joueurs du même type.

Ce degré d'irrationnalité correspond à une probabilité d'aller au bout d'une partie avec un jeu moyen qu'il n'est pas possible de prédire. Ainsi, le fait d'amener l'adversaire à la fin doit être effectué toutes les x parties lors desquelles l'intelligence artificielle a un jeu moyen. Par conséquent, nous avons cherché à découvrir la valeur optimale de ce x . Le tableau présente résultats moyens obtenus sur des séries de 100 parties selon différentes valeurs pour x .

x	Moyenne Jetons gagnés	Moyenne parties gagnées
0	± 0	50
3	-4118	40
5	-678	44
10	-2014	41
15	-3249	51
20	700	47
25	179	45
30	720	51
40	844	50
45	-1947	49

Figure 8.10: Gains de l'intelligence artificielle en fonction de la valeur de x

Comme on peut le constater, il semble que lorsque x est compris entre 20 et 40,

on a tendance à gagner plus de jetons que l'intelligence artificielle adverse ayant le même calibrage. Ces résultats s'expliquent par le fait que le degré d'irrationalité qu'il faut avoir ne doit pas être trop important. Lorsque cette intelligence artificielle possède un jeu moyen, il ne faut pas qu'elle aille au bout de la partie de manière trop fréquente. En effet, le tableau montre que pour des x petits, donc des fréquences d'agressivité* importantes, comme par exemple 1 partie sur 3, ne produisent pas des résultats satisfaisants.

8.4 Résultats

Suite à ces résultats, nous avons donc pu produire des courbes de gain en fonction du nombre de parties jouées, et aussi en fonction du mode de jeu de l'intelligence artificielle, profilage* ou jeu pour gagner. Nous avons également pu afficher dans un même graphe entre les courbes de profilage* et courbe de gain.

Dans le but de rendre plus significative la courbe de profilage*, et donc de mieux voir la corrélation entre les deux courbes, nous avons modifié l'affichage de la courbe de profilage*. En effet, selon le calibrage à déterminer, la variation de la courbe n'est pas la même. Nous avons donc fait en sorte que celle-ci corresponde à la différence entre le profilage* établi et le calibrage réel. Ainsi, plus cette courbe diminue, plus le profilage* établi est précis. Et donc plus cette courbe diminue, plus la courbe de gain est supposée augmenter.

Chapitre 9

Perspectives et conclusion

9.1 Perspectives d'amélioration du profilage

9.1.1 Profilage dynamique

Même si nous n'avons pas eu le temps d'implémenter la méthode de profilage dynamique*, nous avons pu réfléchir un peu à la façon de profiler dynamiquement un adversaire.

Premièrement, lors de la phase de scénarios de tests, nous n'établirions pas le profil uniquement à la fin de la partie mais plutôt au fur et à mesure, en se basant sur les actions qu'il a effectué en réponse à nos actions. De même, lors de la phase de scénarios de tests, plutôt que de définir au départ de la partie un calibrage fixe pour l'intelligence artificielle qui profile, nous pourrions choisir de modifier le calibrage en fonction des actions effectuées par le joueur pendant la partie. En effet, si l'on se rend vite compte que le joueur adverse a tendance à être agressif face à un joueur peu agressif, nous pourrions augmenter l'agressivité* de l'intelligence artificielle qui profile afin de voir s'il répond toujours ou bien s'il a tendance à suivre. De ce fait, nous pourrions définir des paliers et rapidement savoir dans quel palier d'agressivité* et de rationalité* se trouve l'adversaire.

De plus, après les scénarios de tests, lors de la phase de gains, nous pourrions décider de faire en sorte que l'intelligence artificielle modifie son calibrage à chaque tour de jeu, en fonction de l'action jouée par le joueur adverse et de la situation du jeu, contrairement à la phase de gains actuelle, pour laquelle l'intelligence artificielle ne change pas de profil.

9.1.2 Profilage avec plus de deux joueurs

Notre méthode de profilage* étant actuellement basée sur un jeu à uniquement deux joueurs, on peut se poser la question d'un profilage* sur un jeu avec plusieurs adversaires. En effet, les paramètres à prendre en compte pour établir le profil d'un joueur adverse ne seraient pas les mêmes. Il faudrait par exemple ajouter dans ces paramètres le nombre de joueurs adverses, qui influence les comportements. Si un

joueur a peu de chances de gain et un nombre important de joueurs adverses, il sera sûrement bien plus rationnel et aura sûrement plus tendance à se coucher* que s'il avait été contre un seul adversaire. En effet, il y a plus de chances que ses joueurs adverses aient un meilleur jeu que lui. De même, un joueur ayant un jeu moyen et jouant de manière prudente n'aura sûrement pas le même comportement s'il est face à un ou plusieurs joueurs. De même, on peut penser que face à plusieurs adversaires, un joueur va prendre plus de risques pour rester dans le jeu et donc, être moins rationnel et plus agressif. De ce fait, dans un jeu à plusieurs joueurs, le nombre de joueurs serait un critère très important pour le calcul des taux d'agressivité* et de rationalité* d'un adversaire.

Dans le cas d'un profilage* avec plusieurs adversaires, lors de la phase de scénarios de tests, l'intelligence artificielle n'aurait pas forcément besoin d'être aussi agressive puisqu'il y aurait plus de joueurs donc possiblement des joueurs qui pourraient jouer de manière agressive pour elle, dans le but de continuer la partie. Par conséquent, l'intelligence artificielle pourrait faire en sorte de perdre moins de jetons pendant les parties de profilage* tout en continuant à profiler ces adversaires.

9.2 Profilage obtenu

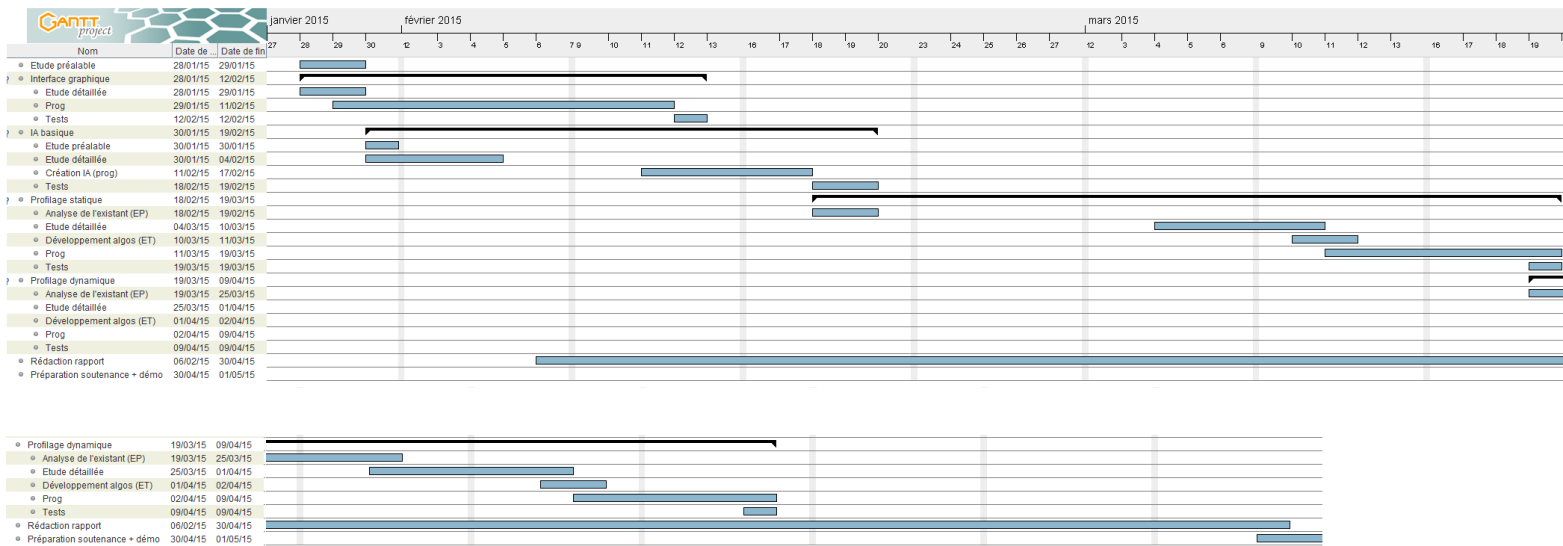
Au terme du projet, nous avons finalement obtenu un bon profilage statique*. En effet, nous arrivons à profiler un joueur à 80% près en moyenne au bout de quelques parties. Après avoir établi le comportement, l'intelligence artificielle est capable de gagner plus de jetons que durant la phase de profilage*. Cependant, on a pu constater que face à un joueur très irrationnel, il n'est pas facile pour nous de gagner beaucoup de jetons. En effet, il est très difficile de prédire les actions qui seront effectuées par un joueur irrationnel. Même si face à ce type de joueurs nous n'obtenons pas de résultats satisfaisants, nous obtenons dans l'ensemble des résultats très corrects pour la plupart des types de joueurs adverses.

Bibliographie

- [1] Darse Billings, Jonathan Schaeffer, Duane Szafron, and Denis Papp. Poker as a testbed for ai research. *Department of Computing Science, University of Alberta*.
- [2] Karl S. Brandt. Player profiling in texas holdem. 2004.
- [3] Aaron Davidson, Darse Billings, Jonathan Schaeffer, and Duane Szafron. Improved opponent modeling in poker. *Department of Computing Science, University of Alberta*.
- [4] Maria de Lourdes Peña Castillo. *Probabilities and Simulations in Poker*. PhD thesis, Université d'Alberta, 1999.
- [5] Poker Listings. Poker listings - le guide de poker all-in. fr.pokerlistings.com.
- [6] Pokernews. Pokernews. fr.pokernews.com/poker-tools/poker-odds-calculator.htm.

Annexe A

Planification prévisionnelle



Annexe B

Planification réelle

