

Rapport

Table des matières

1	Introduction	6
1.1	Généralités	6
1.2	Sujet	6
1.3	Rapide description des règles du Texas Hold'Em Poker	6
1.4	Cahier des charges	6
1.4.1	Problématique	6
1.4.2	Fonctionnalités obligatoires	6
1.4.3	Fonctionnalités optionnelles	7
1.4.4	Spécifications techniques	7
1.4.5	Organisation prévisionnelle	7
2	Organisation du projet	9
2.1	Organisation du travail	9
2.2	Choix des outils de travail	9
2.2.1	Gestionnaire de versions	9
2.2.2	Création de l'interface graphique	9
2.2.3	Rédaction et mise en page du rapport	9
2.2.4	Normes de programmation	9
2.2.5	Langage	10
3	Analyse du projet	11
3.1	Système de jeu	11
3.2	Profilage	11
3.2.1	Choix d'enregistrement des données	11
3.3	Formule de calcul du taux de rationalité d'un joueur	11
3.3.1	Formule de calcul du taux d'agressivité d'un joueur	12
3.3.2	Formule de calcul du taux de bluff d'un joueur	13
3.3.3	Formule de calcul du taux de passivité d'un joueur	13
3.4	Profilage statique	13
3.5	Réutilisation des résultats du profilage	13
3.6	Profilage dynamique	13
3.7	Réutilisation des résultats du profilage	13

4	Mise en place des méthodes de profilage	14
4.1	Calibrage des intelligences artificielles	14
4.2	Profilage statique	15
4.2.1	Scénarios de tests	15
4.2.2	Établissement des profils attendus	15
4.2.3	Réutilisation des résultats du profilage	15
4.3	Profilage dynamique	15
4.3.1	Scénarios de tests	15
4.3.2	Établissement des profils attendus	15
4.3.3	Réutilisation des résultats du profilage	15
5	Analyse des résultats obtenus	16
5.1	Profilage statique	16
5.1.1	Scénarios de tests	16
5.1.2	Analyse des gains de parties	16
6	Profilage dynamique	17
6.1	Analyse des gains de parties	17
7	Perspectives et conclusion	18
7.1	Perspectives d'amélioration du profilage	18
7.2	Profilage obtenu	18

Remerciements

Nous tenons tout d'abord à remercier Jimmy Lopez et Sébastien Beugnon pour nous avoir conseillés tout au long de notre projet, notamment lors de la phase de conception du projet.

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

Glossaire

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

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

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

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

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

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

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

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

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

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

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

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

Chapter 1

Introduction

1.1 Généralités

1.2 Sujet

1.3 Rapide description des règles du Texas Hold’Em Poker

1.4 Cahier des charges

1.4.1 Problématique

1.4.2 Fonctionnalités obligatoires

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

L’intelligence artificielle devra donc profiler ses adversaires en se basant sur différents critères comme la rationalité ou l’agressivité. Elle catégorisera 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 après plusieurs parties jouées contre un même joueur.

Par la suite, nous devons améliorer l’intelligence artificielle afin qu’elle soit en mesure de profiler dynamiquement chacun des joueurs. Dans cette seconde version, elle devra donc pouvoir adapter son comportement aux réactions des joueurs au fil de la partie. Le profilage dynamique se basera ici sur les réseaux bayésiens.

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

1.4.3 Fonctionnalités optionnelles

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

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

1.4.4 Spécifications techniques

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

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

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

1.4.5 Organisation prévisionnelle

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

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

Ensuite, nous passerons à une première étude technique durant laquelle nous commencerons par établir les normes de programmation. Puis, nous établirons une intelligence artificielle simple permettant à un joueur de jouer. Nous établirons par la suite les différents algorithmes et méthodes permettant d'établir un profilage statique du joueur adverse. Et nous établirons des jeux de tests. En même temps que cette étude technique, nous commencerons la phase de réalisation, en implémentant en parallèle l'interface graphique et l'intelligence artificielle simple puis, nous ajouterons la méthode de profilage statique et effectuerons les tests définis auparavant.

Nous effectuerons ensuite une seconde étude technique pendant laquelle nous définirons les méthodes et algorithmes qui seront utilisés afin d'implémenter la

méthode de profilage dynamique. Dans ce même temps, nous implémenterons cette méthode de profilage dynamique, en effectuant les tests définis pendant la phase d'étude technique.

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

Chapter 2

Organisation du projet

2.1 Organisation du travail

2.2 Choix des outils de travail

2.2.1 Gestionnaire de versions

2.2.2 Création de l'interface graphique

2.2.3 Rédaction et mise en page du rapport

2.2.4 Normes de programmation

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

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

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

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

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

```
/**
```

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

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

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

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

2.2.5 Langage

Chapter 3

Analyse du projet

3.1 Système de jeu

3.2 Profilage

3.2.1 Choix d'enregistrement des données

Afin de profiler nos joueurs, nous avons choisi d'enregistrer un certain nombre de données dans des fichiers de sortie. Nous avons décidé de mettre en place un fichier par joueur. De ce fait, chaque fichier est représenté par le pseudo du joueur. Soit joueur ayant pour pseudo "Alain". Le fichier contenant les données de profilage de ce joueur sera donc "Alain.json". De ce fait, s'il se déconnecte et souhaite rejouer un autre jour, nous aurons toujours le précédent profilage effectué. Comme on peut le constater, nous avons choisi d'enregistrer nos données en JSON. en effet, ce format de données a l'avantage d'être facilement exploitable. De plus, nous pouvons utiliser la librairie QJson pour exploiter les données du fichier.

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

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

Pour chaque palier, on regarde si les chances de gain du joueur sont comprises entre $g1$ et $g2$. Une fois le palier trouvé, on calcule la mise théorique, en sachant que celle-ci sera comprise entre les $m1$ et $m2$ correspondant au palier.

Gain = chances de gain du joueur,

Ra = rationalité

M_{th} = Mise théorique.

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

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

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

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

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

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

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

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

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

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

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

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

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

$$bluff = 100 - rationalité$$

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

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

$$passivité = tauxChecks + tauxSuivis$$

3.4 Profilage statique

3.5 Réutilisation des résultats du profilage

3.6 Profilage dynamique

3.7 Réutilisation des résultats du profilage

Chapter 4

Mise en place des méthodes de profilage

4.1 Calibrage des intelligences artificielles

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

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

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

Ag. Mise totale

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

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

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

En prenant un rapport $\times 2$, on obtient donc les mises totales suivantes à chaque tour de jeu. Ex : 5 – 10 – 20 – 40 – 80 avec 80 mise totale théorique.

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

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

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

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

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

Une fois que le calibrage nous a donné l'action agressive et l'action rationnelle, on effectue une fusion entre les deux. Dans les cas où les deux actions sont identiques, l'action est prise. Si ce sont toutes les deux des mises/relances, on prend la moitié des du nombre de jetons, proportionnellement aux pourcentages de calibrage. Enfin, il reste la possibilité que les actions soient différentes. Ici on distingue deux cas : - on arrive à déterminer une action représentant un juste milieu, par exemple si les deux actions sont se coucher et relancer, on fait suivre l'intelligence artificielle. Dans le cas où on a checker / miser, on considère l'action checker comme miser 0 et on effectue le calcul de la mise moyenne expliqué ci-dessus. - sinon on tire aléatoirement l'action parmi les deux, toujours en prenant en compte les pourcentages du calibrage.

4.2 Profilage statique

4.2.1 Scénarios de tests

4.2.2 Établissement des profils attendus

4.2.3 Réutilisation des résultats du profilage

4.3 Profilage dynamique

4.3.1 Scénarios de tests

4.3.2 Établissement des profils attendus

4.3.3 Réutilisation des résultats du profilage

Chapter 5

Analyse des résultats obtenus

5.1 Profilage statique

5.1.1 Scénarios de tests

5.1.2 Analyse des gains de parties

Chapter 6

Profilage dynamique

6.1 Analyse des gains de parties

Chapter 7

Perspectives et conclusion

7.1 Perspectives d'amélioration du profilage

7.2 Profilage obtenu