

Rapport

Table des matières

1	Introduction	3
1.1	Généralités	3
1.2	Sujet	3
2	Panification prévisionnelle	5
2.1	analyse	5
2.2	méthode	6
3	Planification réelle	7
3.1	Comparaison entre le réel et le prévisionnel	7
4	Données quantitatives	8
4.1	Utilisation d'un gestionnaire de version	8
4.1.1	Mise en place de la méthode git-flow	8
4.2	Brain Storming	8
5	Conclusion	9

Chapitre 1

Introduction

1.1 Généralités

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

1.2 Sujet

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

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

et irrationnel ?

Chapitre 2

Panification prévisionnelle

2.1 analyse

Comme on peut le voir dans l'annexe A représentant le diagramme de Gantt prévisionnel du déroulement du projet, un premier jour sera accordé à une étude préalable durant laquelle les membres du groupe devront se familiariser avec les règles du poker et les termes techniques. De plus, durant cette période, les membres du groupe devront établir ensemble les normes de programmation qui seront respectées par la suite tout au long du projet.

Par la suite, notre projet sera divisé en quatre sous-projets. Le sous-projet correspondant à l'interface graphique commencera tout d'abord pendant une étude détaillée durant laquelle il faudra décider entre-autres des technologies à utiliser pour son développement. L'implémentation de l'interface graphique commencera ensuite, suivie par une série de tests visant à vérifier que l'interface graphique est bien fonctionnelle et répond aux besoins.

Dans le même temps, le sous-projet correspondant à la mise en place de l'intelligence artificielle basique et le jeu commencera. Celui-ci débutera d'abord par une étude préalable durant laquelle il faudra établir de façon claire et précise les différentes règles du Texas Hold'Em Poker. Une fois ces règles établies, une courte étude détaillée aura lieu, afin de discuter des différents points techniques de la mise en place du jeu et des différentes technologies qui pourraient être utilisées. Le jeu et l'intelligence artificielle seront alors développés puis testés.

Une fois les deux premiers sous-projets finis, le sous-projet "profilage statique" débutera. Le but de ce sous-projet sera de mettre en place des méthodes permettant de profiler de manière statique un joueur. Il commencera tout d'abord par une étude préalable durant laquelle une analyse de l'existant sera effectuée notamment par la lecture d'articles sur le sujet. Puis, une petite étude détaillée sera effectuée afin de bien définir les calculs qui seront à implémenter. La tâche de développement des algorithmes à implémenter commencera alors, avec l'écriture formelle des différents algorithmes de calculs qui devront être présents dans l'application. Enfin, l'implémentation des algorithmes dans l'application

commencera, suivie par la phase de test des données ajoutées à l'application.

Enfin, le sous-projet "profilage dynamique" débutera. De même que pour le sous-projet "profilage statique", il débutera par une phase d'analyse de l'existant avec lecture d'articles traitant du sujet. Puis, sera suivi par une étude détaillée. Par la suite, une phase d'étude technique commencera avec le développement des algorithmes à implémenter. Enfin, les phase d'implémentation des algorithmes et de tests suivront.

En parallèle à toutes ses tâches aura lieu la rédaction du rapport et, à la fin du temps imparti, la préparation de la soutenance et de la démonstration.

2.2 méthode

2.2.1 Utilisation d'un gestionnaire de version

Afin de permettre un partage efficace des données, le gestionnaire de version Git sera utilisé. Morgane Vidal aura le répertoire principal et Manuel Chataigner et Benjamin Commandre devront mettre leurs ajouts sur leurs propres répertoires puis demander à inclure leurs modifications sur le répertoire principal.

Chapitre 3

Planification réelle

3.1 Comparaison entre le réel et le prévisionnel

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.

Chapitre 4

Données quantitatives

4.1 Mise en place de la méthode git-flow

Après avoir assisté au cour durant lequel les workflow nous étaient présentés, nous avons choisi de passer en git-flow, donc de mettre en place une branche develop sur laquelle seraient ajoutées les modifications et qui seraient par la suite ajoutées à la branche master une fois celles-ci correctes.

4.2 Brain Storming

Chapitre 5

Conclusion