

**BERNARD Arthur**

**Compte-rendu de projet de Simulation**  
**Multi-agents :**  
**Le jeu du Texas Hold'Em**

# Introduction

Ce rapport a pour but de présenter le résultat du projet de Simulation Multi-Agents. Le sujet choisi pour ce projet a été la conception d'une simulation d'une partie de Poker, selon la variante **Texas Hold'Em**. Le poker est un jeu pratiqué depuis de très nombreuses années et qui connaît aujourd'hui une notoriété ainsi qu'une exposition médiatique extrêmement importante, notamment le Texas Hold'Em. On pourrait s'interroger sur un tel succès pour ce jeu mais une partie de la réponse est sans doute dans le fait que c'est un jeu qui mêle stratégie, chance, bluff et surtout une part très importante d'argent. C'est donc un jeu pour lequel la modélisation de comportements humains peut être très intéressante à étudier.

Le but de ce projet a alors été de coder, sur la plateforme **GAMA**, un modèle permettant à la fois de fournir une simulation à la fois complète mais aussi modulable d'une partie de Poker. En effet, parce que l'on peut imaginer énormément de types d'agents différents jouant au Poker, avec tous une certaine façon de décider, il a paru intéressant de fournir une base permettant d'ajouter facilement de nouveaux agents. Ceci présente un intérêt notamment pour tester des agents entre eux afin de comparer différentes implémentations d'agents avant par exemple de les essayer contre des joueurs humains, l'avantage étant de pouvoir multiplier les simulations et le nombre de données collectées.

En plus de cette simulation, des agents ont bien évidemment été développés afin de tester le fonctionnement du modèle et surtout de réfléchir au problème intéressant de la création d'agents pouvant jouer à un jeu aussi varié dans les stratégies que le Poker. Ces agents n'ont ni la prétention de battre des joueurs humains de haut niveau ni de simuler de manière réaliste des comportements humains mais sont une première réflexion quant à ce que pourraient être des agents qui jouent au Poker.

Enfin, il est à noter que cette simulation dispose aussi d'une interface graphique, certes simple, mais qui permet de montrer de manière rapide et (je l'espère) claire le déroulement d'une partie.

Dans un premier temps, nous présenterons le jeu du Texas Hold'Em en lui-même, afin de bien mettre en valeur les problématiques qui ont pu surgir lors du développement de cette simulation. Certains points de règles importants seront donc ainsi explicités afin de permettre à quelqu'un n'ayant peu ou pas de connaissance de ce jeu de pouvoir comprendre les points importants de la simulation.

Puis nous nous intéresserons alors à faire une description du modèle développé, en présentant tout d'abord le fonctionnement de général de l'outil ainsi que les points importants qui régissent le fonctionnement de l'agent monde, qui a le rôle de croupier dans notre simulation. Nous pourrons alors nous intéresser plus en détail aux agents joueurs développés dans le cadre du projet.

Enfin, nous terminerons par une partie contenant un certain nombre d'expériences réalisées afin de tester et d'essayer d'améliorer les performances des différents agents. Cette partie a principalement pour but de mettre en avant les qualités et défauts des stratégies développées ainsi que d'étudier les phénomènes qui peuvent avoir une influence sur le fonctionnement de ces agents.

# I. Présentation du jeu

## A. Le Poker

Le poker est un jeu de cartes compétitif qui se joue (normalement) avec un deck de base de 52 cartes (du 2 à l'As et quatre couleurs). Son intérêt principal réside dans le fait que chaque joueur possède une main de cartes qui n'est entièrement connue des autres joueurs qu'à la fin du jeu et que chaque joueur doit être capable de miser en fonction de cette information imparfaite.

Une partie de poker est remportée ou bien par le joueur possédant la main la plus forte (en terme de combinaison de cartes) ou bien lorsqu'un seul joueur est encore en jeu, les autres s'étant retirés (couchés), ce qui ouvre la voie à un jeu qui peut être partiellement basé sur le bluff.

Le déroulement d'une partie de poker est caractérisé par un *pot*, qui contient l'ensemble des mises des joueurs et d'une mise à pourvoir par chaque joueur pour pouvoir continuer le jeu. A chaque fois que c'est son tour, un joueur a le droit à trois actions différents :

- **Se coucher** : le joueur considère que sa main de carte n'est pas assez forte pour qu'il puisse continuer le jeu et battre les autres joueurs. Tout ce qu'il a misé reste dans le pot et il ne peut plus participer avant la prochaine partie.
- **Suivre** : ceci veut dire que le joueur choisit de rajouter autant d'argent dans sa mise que nécessaire pour égaler la mise demandée. Si jamais il n'y a pas de mise demandée et que le joueur n'a donc pas rajouter d'argent, on dit que ce joueur effectue une *parole*, c'est-à-dire qu'il souhaite simplement voir le jeu continuer sans miser.
- **Relancer** : le joueur qui choisit de faire une relance, mise l'argent qu'il manque pour atteindre la mise demandée et rajoute en plus de l'argent, augmentant donc la mise demandée aux autres joueurs.

Cette phase durant laquelle les joueurs peuvent miser s'appelle un *tour d'enchères*. Il s'arrête lorsque tous les joueurs qui ne se sont pas couchés ont **suivi** la mise demandée. Pour éviter que le jeu ne s'éternise, nous empêchons les joueurs de relancer plus qu'un certain nombre de fois à chaque tour d'enchères. Le joueur débutant le tour d'enchères est généralement celui assis à gauche du **donneur**. Le donneur est un rôle donné à un joueur qui change à chaque partie.

## B. Les combinaisons

Au poker, les mains de chaque joueur sont évaluées afin de pouvoir leur donner une "puissance" et donc qu'elles puissent être comparées les unes aux autres. Une main valide est toujours formée d'une combinaison de 5 cartes (que le joueur possède en effet ces 5 cartes ou non). Les combinaisons au poker sont les suivantes, dans l'ordre de la moins forte à la plus forte :

### 1. La carte haute :

Cette combinaison correspond en fait au fait que le joueur ne possède pas de "combinaison particulière", c'est-à-dire que c'est un ensemble de 5 cartes ne formant aucune combinaison. Pour comparer deux mains à carte haute, on compare les cartes les plus fortes une à une, l'As étant la plus forte carte.

### 2. La pair :

Cette combinaison correspond à la présence d'une pair de deux cartes identiques (en valeur) dans la main et d'autres cartes à côté. Les pairs sont d'abord comparées sur la valeur de la pair en elle-même puis sur les trois cartes l'accompagnant de la même manière que pour la carte haute.

**3. La double pair :**

La double pair signifie que l'on possède deux pairs différentes dans notre main. Deux doubles pairs sont comparées d'abord selon leur pair la plus forte, puis leur pair la plus faible puis enfin la carte les accompagnant.

**4. Le brelan :**

Le brelan correspond à la présence de trois cartes de valeurs identiques dans la main et pas de cartes particulière l'accompagnant. La comparaison de deux mains se fait cette fois sur la valeur des deux brelans.

**5. La suite :**

Avoir une suite signifie que les 5 cartes du joueur ont des valeurs qui se suivent. Il est à noter un point important que, pour le cas de la suite et, selon les variantes, l'As peut aussi bien être la carte la plus faible (avant le 2) ou la plus forte (après le Roi). Ceci signifie que les deux suites "As 2 3 4 5" et "10 Valet Dame Roi As" sont correctes. C'est cette variante que nous utiliserons dans notre simulation. Deux suites sont comparées selon leur carte haute uniquement, c'est-à-dire la carte à la valeur la plus grande.

**6. La couleur :**

La couleur, aussi appelée *flush*, correspond à la présence de 5 cartes de la même couleur dans la main du joueur. Il est à noter que le terme couleur ne correspond pas à Rouge/Noir mais à Trèfle/Carreau/Pique/Coeur. Deux couleurs sont comparées selon les valeurs des 5 cartes exactement comme pour une combinaison carte haute.

**7. Le Full :**

Le Full (pour Full House), ou main pleine, correspond au fait d'avoir une pair et un brelan en même temps dans la main. Deux Fulls sont comparés selon leur brelan.

**8. Le carré :**

Le carré est la présence des 4 cartes de valeur identique dans la main du joueur. Deux carrés sont donc comparés selon leur valeur de carré.

**9. La Quinte flush :**

Cette combinaison est la combinaison la plus forte et correspond à une suite dont toutes les cartes ont la même couleur. Le terme particulier de Quinte flush royale correspond au fait d'avoir une Quinte flush à l'As ("10 Valet Dame Roi As") qui est donc la combinaison la plus puissante du jeu. Deux Quintes flush (ce qui est très improbable), sont comparées comme des suites normales.

## **C. Les blinds**

Dans la variante que nous allons utiliser, il est mis en place un système de *blinds*. Les blinds sont en fait des mises obligatoires qui permettent d'éviter que des joueurs se couchent systématiquement à chaque partie jusqu'à ce qu'ils obtiennent la main de départ qui leur convienne. Ceci permet donc un jeu plus nerveux et plus intéressant.

A chaque partie, on définit un joueur *petite blind*, qui est le joueur directement à gauche du

dealer et un joueur *grosse blind*, qui est celui à gauche de la petite blind. Une valeur de blind est aussi définie pour la partie et, au début de chaque partie, le joueur petite blind doit commencer par miser la moitié de la valeur de la blind, après quoi le joueur big blind doit miser la valeur totale de la blind. Le premier joueur à véritablement commencer le tour d'enchères est donc le joueur suivante le joueur grosse blind.

## **D. Le tapis**

Le tapis est une règle particulière permettant aux joueurs n'ayant plus assez d'argents de continuer à jouer. Ainsi, si un n'importe quel moment de la partie, après avoir misé, un joueur n'a plus d'argent, on dit qu'il a fait tapis. Il n'est pas éliminé pour autant et peut continuer à suivre la partie, même s'il ne mise évidemment plus du tout pendant les étapes d'enchères.

Ainsi, si un joueur est tapis et même si son tapis ne suffit pas à atteindre la mise demandée et que d'autres joueurs continuent à effectuer des relances, on continue qu'il suit toutes les mises jusqu'à la fin de la partie : il n'est pas considéré comme un joueur couché. Pour les autres joueurs si jamais un tapis modifie la mise demandée (c'est donc une relance), il leur est nécessaire de mettre comme normalement la mise demandée ou de se coucher ou de faire eux-mêmes tapis si c'est leur seul moyen de suivre.

Le tapis représente donc un moyen d'effectuer un coup de pression sur les autres joueurs (que ce soit du bluff) ou non, en avançant une très forte somme d'argent ou alors de permettre à soi-même de tout miser sur un dernier coup qui pourrait nous sauver.

Néanmoins, pour que le tapis ne soit pas trop puissant, un joueur ayant fait tapis qui remporterait la partie, ne gagne pas forcément tout le pot mais seulement à hauteur de son tapis pour chaque joueur. En effet, le tapis d'un joueur provoque la création d'un pot parallèle au pot principal. Les autres qui ne sont pas en tapis, participent alors à remplir le pot parallèle, qui ne peut être gagné que par eux et pas par le joueur en tapis.

Pour mieux comprendre, prenons l'exemple de deux quatre joueurs A, B, C et D. Imaginons que la mise demandée (finale) soit de 200, qu'elle a été suivie par C et D mais que les joueurs A et B ont tous les deux fait tapis, respectivement avec une valeur de 50 et de 150.

Le tapis de joueur A ne laisse dans le pot principal qu'une valeur égale à la hauteur de son tapis pour chaque joueur. C'est-à-dire que le pot principale contient  $50 + 50 + 50 + 50 = 200$ . Le pot parallèle créé contient donc  $600 - 200 = 400$ .

Mais le tapis du joueur B crée un second pot parallèle. On ne laisse dans le premier pot parallèle, qu'une valeur égale à la hauteur du tapis de B moins la valeur que B a laissé dans le pot principale, encore une fois pour chaque joueur pouvant payer. Comme ceci peut paraître peu clair, imaginons que le joueur A n'existe pas et que seul B ait donc fait tapis, il laisserait dans le pot principal une valeur de  $150 + 150 + 150 = 450$ . Mais dans notre cas, chaque joueur a déjà laissé une partie de sa mise, plus exactement 50, dans le pot principal écarté lors du tapis de A. Le pot parallèle de B est donc constitué  $100 + 100 + 100 = 300$ . Le second parallèle correspond donc à ce que seuls C et D peuvent gagner, soit  $400 - 300 = 100$ .

Au bout du compte, chaque joueur peut prétendre au maximum aux gains suivants :

- Joueur A : *pot principal* = 200.
- Joueur B : *pot principal* + *premier pot parallèle* =  $200 + 300 = 500$ .
- Joueurs C et D : *tous les pots* = 600.

Il est donc nécessaire à la fin de garder un classement des joueurs afin de bien redistribuer les pots. Imaginons par exemple que le classement soit le suivant :  
*Joueur A – Joueur B à égalité avec Joueur C – Joueur D*

on aurait le partage des gains suivants :

- *Le joueur A prend l'entièreté de son pot -> 200*
- *Le joueur B partage le premier pot parallèle avec C (égalité) et le joueur C prend l'entièreté du secon pot parallèle -> 150 pour B et 250 pour C*
- *Il ne reste rien pour le joueur D -> 0*

## **E. Le Texas Hold'Em**

Le Texas Hold'Em est aujourd'hui une des variantes de Poker les plus jouées notamment en compétition, entre autres parce qu'elle est considérée comme étant une des plus stratégiques. En effet, elle se base sur la présence de cartes révélées partagées et de cartes inconnues propres à chaque joueur, leur permettant de piocher dans les unes ou les autres pour faire leur combinaison de cartes.

Le fonctionnement est le suivant :

1. Au début du jeu, chaque joueur se voit distribué 2 cartes qui constituent sa main et qu'il garde bien sûr secrètes. On démarre alors le premier tour d'enchères (par le joueur à gauche de la grosse blind dans le cas où on joue avec des blinds), qui dure jusqu'à ce que tout le monde ait suivi et ou qu'il ne reste plus qu'un seul qui ne se soit pas couché. C'est la phase de **Pré-Flop**.
2. Lorsque les enchères sont terminées, on passe à la phase de **Flop**. On brûle la première carte du paquet (on la retire du jeu sans la montrer) puis les trois cartes suivantes sont révélées. Ces cartes sont communes à tous les joueurs et forment ce qu'on appelle le *tableau*. Les joueurs peuvent alors créer des combinaisons de 5 cartes et on relance un tour d'enchères, qui commence cette fois par le joueur à la gauche du dealer (il en sera de même pour tous les tours suivants).
3. Une fois ce tour d'enchères terminé, la phase suivante est la phase de **Turn**. On brûle de nouveau la première carte et on révèle la suivante. Les joueurs peuvent donc faire des combinaisons de 5 cartes en choisissant parmi 6. On relance un tour d'enchères.
4. Ces enchères terminées, on arrive à la phase de **River**. Encore une fois, on brûle une carte, on en révèle une autre et on lance le dernier tour d'enchères. Les joueurs forment des combinaisons de 5 cartes parmi 7.
5. A l'issue de ces dernières enchères et s'il ne reste pas qu'un seul joueur en course, on passe à la phase de **Showdown**, l'abattage. Tous les joueurs qui sont encore en jeu révèlent leurs cartes et on obtient alors le classement, ce qui permet de partager les gains.

## II. Modélisation

### A. Modèle général

Maintenant que nous avons présenté les règles du jeu, nous sommes en mesure d'expliquer comment la situation a été implémentée. Vous trouverez sur la page suivante une capture d'écran de l'interface lors du déroulement d'une partie.

Sur cette interface sont représentés l'ensemble des agents utilisés dans la simulation : le *monde* et les *Joueurs*. Le monde est uniquement représenté par les informations centrales, que sont les valeurs courantes du pot et de la mise demandée ainsi que par les cartes présente au tableau.

Les agents Joueur eux, sont représentés par des ronds répartis autour de la table de jeu et dont le code couleur permet d'avoir des informations sur leur état :

- Un rond vert est un joueur qui est encore en jeu, sans été particulier.
- Un rond rouge correspond au joueur en jeu dont c'est le tour de jouer.
- Un rond bleu indique que le joueur est en jeu mais qu'il s'est mis en tapis.
- Un rond gris indique que le joueur s'est couché pour cette partie.
- Enfin, un petit rond noir, indique que le joueur est "out", c'est-à-dire qu'il a quitté le jeu (ou plutôt a été forcé de quitter le jeu) parce qu'il n'avait plus d'argents.

D'autres informations importantes sont indiquées sur les agents, notamment leur mise et leur argent, leur statut (dealer, small blind et big blind) et enfin leur nom, qui est simplement un identifiant composé de "J" et d'un indice ainsi que leur espèce, qui indique donc le type de joueur qu'il est. Nous voyons dans l'exemple que nous n'avons que des *JoueurBluffer* dont nous parlerons un peu plus tard.

### B. L'agent monde

L'agent monde, créé de base avec GAMA comme étant l'agent global, est celui qui va se charger de centraliser l'ensemble du jeu. En effet, de part le type de jeu, il a paru bien plus simple et logique d'appliquer une simulation centralisée au tour de ce qui fait office de croupier qui se charge de veiller à ce que le jeu se déroule bien.

C'est donc cet agent qui est chargé de la mise à jour de toute les variables "centrales" du jeu, telles que le pot, la mise demandée ou encore les cartes du tableau. Le fonctionnement principal de cet agent est basé sur un certain nombre de *reflex* dont le déclenchement est fonction de l'état du jeu. Ainsi, lors d'un tour d'enchères (ce qui arrive dès le début), le reflex qui va s'activer est le reflex *joueur\_suivant* dès que le monde a le **jeton**. En fait, ce jeton est ce qui permet de faire jouer les joueurs les uns après les autres et pas tous en même temps dans le chaos.

A chaque fois que le monde récupère le jeton et si les enchères ne sont pas finies, il va chercher à qui passer ce jeton. Pour cela, partant du joueur courant, il va chercher le prochain joueur à jouer, qui est donc le joueur suivant dans la liste qui ne soit ni out, ni couché ni en tapis (les joueurs en tapis ne misent plus). Si jamais ce joueur suivant est ce qui est stocké dans le monde en tant que *premier joueur*, alors ça veut dire qu'on a fait un tour complet d'enchères et que celles-ci sont finies. Sinon, il passe le jeton à ce joueur pour lui indiquer que c'est à lui de jouer.

Le premier joueur est en fait déterminé, au début du tour d'enchères évidemment par le premier joueur à commencer les enchères mais après, on considère que le premier joueur est le

Argent : 421  
Mise : 20  
J 4 : JoueurBluffer



Dealer  
Argent : 18  
Mise : 0  
J 0 : JoueurBluffer



Small blind  
Argent : 0  
Mise : 1  
J 1 : JoueurBluffer



Pot : 41  
Mise courante : 20



Big blind  
Argent : 20  
Mise : 20  
J 3 : JoueurBluffer



Argent : 0  
Mise : 0  
J 2 : JoueurBluffer



dernier joueur à avoir relancé. Ainsi, ceci laisse la possibilité à tous les joueurs après lui de se coucher, de suivre voir de relancer.

Si le monde a le jeton et que les enchères sont terminées, c'est un autre reflex qui se déclenche : *etape\_suivante*. Ce reflex permet de lancer les différentes étapes du jeu les unes après les autres : flop, turn, river et showdown. Il permet donc de tirer les nouvelles cartes du tableau à chaque étape puis de relancer un tour d'enchères de la même manière qu'expliquée précédemment. L'exception étant bien évidemment au moment du showdown où le but du monde est uniquement à ce moment-là de déterminer le classement et de partager les gains.

Il est à noter que le monde a aussi un rôle de validateur, dans le sens que, pour garder son rôle de croupier, il n'y a que lui qui modifie le pot et la mise demandée. Lorsqu'un joueur mise, il appelle l'action *valider\_miser* du monde et c'est dans cette action que le monde va déterminer si la mise est correcte. Si ce n'est pas le cas, par exemple que le joueur ne mise pas autant que demandé sans pour autant qu'il soit en tapis, le monde se charge de passer le joueur en état couché en tant que punition. Si la mise est correcte, le monde se charge de l'ajouter au pot et de faire tous les traitements qui en découlent avant de laisser jouer le joueur suivant. A noter qu'on considère qu'un joueur qui se couche mise une valeur de **-1**, ce qui permet au monde de traiter ceci au même endroit.

Le monde valide aussi quand un joueur est out, ce joueur appelant l'action *valider\_out*, ce qui permet au monde de prendre les mesures qui s'imposent et de supprimer ce joueur de la liste des joueurs actifs.

Enfin, c'est le monde qui se charge de distribuer les cartes mais c'est aussi lui qui se charge d'évaluer les combinaisons présentes dans les mains des joueurs. Ceci a été fait afin de centraliser cette opération et les joueurs doivent demander au monde s'ils souhaitent connaître la meilleure combinaison dans leur main. Ceci est, comme expliqué plus tôt dans les règles, en effet capital pour déterminer la meilleure main et il peut être utile (voir capitale pour des joueurs intelligents) aux joueurs de se faire une idée de la main qu'ils possèdent avant l'abattage.

Pour évaluer une main, on crée toutes les combinaisons possibles réalisables par le joueur avec sa main de 2 cartes et le tableau de 3 à 5 cartes. Ce qui veut dire qu'au flop, il n'y a qu'une main possible, 6 mains possibles au turn (5 parmi 6) et enfin 21 mains possibles à partir du river (5 parmi 7). Pour chacune des combinaisons possibles, en partant de la plus forte (Quinte flush), on cherche dans chaque main si celle-ci répond à cette combinaison et on la compare à une autre main si il y en a une autre répondrait à la même combinaison afin de trouver la meilleure. Ceci permet de trouver tout de suite la meilleure combinaison sans avoir besoin de tester toutes les combinaisons possibles (sauf dans le pire des cas, ce qui arrive très souvent...). Néanmoins, tout cela est tout de même heureusement très rapide.

## **C. L'agent Joueur**

L'agent Joueur est un type d'agent permettant de servir de base fonctionnelle à tous les agents implémentés dans le modèle. Il possède tous les différents reflex et actions servant au fonctionnement d'un joueur, ce qui permet, un créant un agent qui hérite de Joueur, de ne rajouter que son "intelligence" propre à notre agent, rendant le modèle modulable. C'est dans ce but qu'a été créé l'ensemble du modèle.

De ce fait, l'espèce Joueur possède toutes les actions de bases nécessaires à son fonctionnement, notamment les actions *miser* et *se\_coucher*, dont les effets sont évidents. Pour effectuer une relance, un agent a juste à miser une valeur supérieur à celle nécessaire pour atteindre la mise demandée et pour suivre, il mise juste assez. Il est à noter que le tour d'un agent ne se terminer pas tant qu'il n'a pas effectué une de ces deux actions. Ce sont en effet ces deux actions qui permettent de faire valider la décision de l'agent par le monde (comme vu avant) ainsi que de lui

rendre le jeton pour que le déroulement de la partie puisse continuer. Ce sont aussi ces actions (plus spécifiquement *miser*) qui vérifient d'elles-mêmes si l'agent a le droit de faire ce qu'il fait ou si sa mise l'amène dans une situation de tapis. Ces vérifications sont donc faites au nouveau de l'agent père, ce qui évite quiconque voudrait implémenter un agent d'avoir à les faire.

Les informations propres au fait que l'agent soit dealer, petite blind ou grosse blind sont bien évidemment elles aussi stockées au niveau de Joueur vu qu'elles ne relèvent que du fonctionnement "mécanique" de l'agent. De la même façon, c'est une action particulière *mise\_obligatoire* appelée par le monde qui se charge de collecter les mises de petite et grosse blind, ceci n'est pas à mettre dans le comportement de l'agent.

Il n'y a en fait qu'une seule chose à implémenter lors de création d'un agent dérivant de Joueur, c'est le reflex abstrait *choisir\_action*. C'est dans ce reflex, déclenché à chaque fois que l'agent possède le jeton, que toute "l'intelligence" et le processus de décision doit être mis. Comme expliqué dans les règles, un joueur a le choix à son tour entre les trois actions *se coucher*, *suivre* et *relancer* (les deux dernières étant liées). Une fois que toute la réflexion a été faite et la décision prise, il est nécessaire, comme dit précédemment d'appeler une des deux actions *miser* ou *se coucher* afin que cette action soit validée par le monde et qu'il passe la main au joueur suivant. Il n'y a, dans l'état actuel, pas de restriction mise sur le temps de jeu d'un joueur et donc sur son temps de réflexion.

## **D. Agents implémentés**

Nous allons parler ici des différents types d'agents Joueur qui ont été implémentés dans la simulation. Ceux-ci sont au nombre de 4.

### ***1. Le JoueurZISuiveur***

Premier type d'agent implémenté et premier type d'agent Zéro Intelligence présenté, le JoueurZISuiveur fonctionne selon un principe très simple : jouer le plus longtemps possible. C'est donc un joueur sans grande intelligence qui se contente simplement de suivre autant que possible la mise demandée. S'il a assez, il mise donc juste assez pour suivre et sinon il mise tout son argent, effectuant ainsi un tapis.

On voit tout de suite que ce joueur ne peut survivre dans la durée de longues parties pour la simple raison que l'aléatoire devrait faire qu'il perdra plus qu'il ne gagnera, en tout cas en présence de plusieurs joueurs. Néanmoins on peut lui trouver la qualité d'être totalement insensible à la variation de la mise par les relances et donc au coup de bluff. Le suiveur est déterminé à jouer (il est là pour passer un bon moment).

### ***2. Le JoueurZIAléatoire***

Deuxième agent Zéro Intelligence et comme son nom l'indique, ce joueur totalement aléatoire. Plus exactement il choisit ses actions selon des probabilités plus ou moins fortes qui sont déterminées au départ de la simulation. Ainsi, ce joueur est paramétré selon trois variables : *proba\_coucher*, *proba\_suivre* et *proba\_relancer*. Il est bien évidemment naturel de faire sommer ces probabilités à 100 (si ce n'est pas le cas et que l'agent aléatoire ne sait pas quoi faire, il se couche).

Pour effectuer sa décision le joueur choisit aléatoirement un nombre entre 1 et 100 et le compare à ses probabilités. S'il est inférieur à *proba\_coucher*, il se couche. S'il est supérieur à *proba\_coucher* mais inférieur à *proba\_coucher* + *proba\_suivre*, il suit. Enfin, s'il est supérieur à

$proba\_coucher + proba\_suivre$  et inférieur à  $proba\_coucher + proba\_suivre + proba\_relance$ , il choisit de relancer. Dans le cas où il choisit de relancer, il choisit aléatoirement la valeur de sa relance entre 0 et l'argent qu'il lui reste après avoir payé la mise pour atteindre la mise demandée.

Il est à noter que on regarde avant même de tirer le nombre aléatoire, si l'agent est capable de relancer (si son argent est strictement supérieur à la mise qu'il doit payer pour suivre). Si ce n'est pas le cas, on répartit sa probabilité de relance entre sa probabilité de se coucher et celle de suivre.

### 3. Le Joueur Prudent

Ce joueur possède un comportement plus intelligent que les joueurs Zéro Intelligence puisqu'il va chercher à évaluer sa **confiance** avant de jouer. Pour cela, la première chose qu'il fait et qui est très différente du comportement des ZI, c'est qu'il évalue la meilleure combinaison qu'il possède avec sa main et les cartes du tableau (s'il y en a). En effet, la confiance du joueur prudent est déterminé uniquement selon deux critères : la **force** de sa main et le **risque** sur la mise. Ce n'est donc pas un joueur qui va chercher à savoir ce que vont faire les autres et réfléchir à ses probabilités de victoire mais uniquement choisir en fonction de ce qu'il sait, d'où son nom (même s'il est tout à fait possible d'en faire un joueur casse-cou selon les paramètres qu'on lui met).

Comme dit juste avant, ce joueur prend sa décision en fonction de sa confiance, qui représente en fait la confiance qu'il a sur le fait de continuer le jeu. Il est donc nécessaire dans un premier temps d'évaluer cette confiance. Pour cela on procède comme suit :

1. On évalue tout d'abord la force de la main du joueur. Cette force est bien évidemment composée à la fois de la force de la combinaison (pair, brelan etc...) mais aussi de la force des cartes composantes cette combinaison elle-même. Elle doit donc prendre en compte ces deux facteurs. Pour cela, on détermine donc un nombre à trois chiffres XXX tel que le chiffre des centaines corresponde à la force de la combinaison (de 0 pour la carte haute à 8 pour la quinte flush) et les deux derniers chiffres à la moyenne des valeurs des cartes présentes dans la main (l'As est alors compté comme 14). Ainsi, à combinaison égale, la main la plus forte a une valeur moyenne de cartes plus importante.

Par exemple, la force d'une main constituée d'un brelan de 3, d'un Roi (valeur 13) et d'un 10 est de **300** (valeur de la combinaison) + **6,4** (moyenne des valeurs) = **306,4**.

Pour plus de simplicité, on ramène cette force entre 0 et 100 en la divisant par la force maximale possible pour une main et en multipliant par 100. Cette force maximale est différente lors de l'étape de pré-flop, où l'on a que deux cartes et les étapes suivantes où on en a 5. En pré-flop, la force maximale est celle d'une paire d'As, soit **114** et dans les autres étapes, c'est celle d'une quinte flush royale, soit **812**.

2. Une fois la force évaluée, on évalue le risque sur la mise. Ce facteur est là pour prendre en compte le risque par rapport à l'argent qu'il possède que prend l'agent en misant la valeur minimale demandée. C'est donc aussi un moyen de prendre en compte les relances et mises des autres joueurs. Plus les joueurs vont relancer haut, plus l'agent à miser va devenir important et le risque aussi. Ceci permet donc aussi de rendre le joueur un peu moins "aveugle" à ce qui se passe autour de lui. Ce risque est calculé très simplement de la façon suivante :  $risque = \left( \frac{\text{valeur minimale à miser}}{\text{argent}} \right) * 100$ . le risque est donc lui aussi ramené entre 0 et 100. Dans le cas où la mise minimale implique un tapis, le risque est directement mis à 100.

3. On peut alors finalement évaluer la confiance du joueur. Pour cela, la formule est simplement :  $confiance = \frac{force - risque + 100}{2}$ . Ceci nous permet, là encore d'avoir un nombre entre 0 et 100.

Une fois qu'on connaît la confiance de l'agent, on peut alors prendre la décision. Pour cela, l'agent est paramétré selon deux seuils : *seuils\_suivre* et *seuils\_relance*. Si sa confiance est supérieure à son seuil de relance, l'agent relance. Sinon, si elle est supérieure à son seuil pour suivre, l'agent suit. Sinon l'agent se couche.

Dans le cas où l'agent relance, il est aussi nécessaire d'évaluer de combien il relance. Pour cela, un autre paramètre est définie : *ratio\_max\_relance*, qui permet de savoir jusqu'à quel ratio maximal de son argent, le joueur va relancer. La mise de relance dépend aussi bien évidemment de la force de la main, ce qui nous permet d'arriver à la formule suivante :

$$relance = \frac{force}{100} * ratio\ max * (argent - valeur\ minimale\ à\ miser)$$

#### 4. Le JoueurBluffer

Le JoueurBluffer part sur la même base que le JoueurPrudent, c'est-à-dire l'évaluation de sa confiance dans son processus de décision mais avec la possibilité de pouvoir bluffer. Dans l'idée, et en tout cas dans la façon dont c'est abordé par ce joueur, le bluff est le fait de faire croire aux autres joueurs que notre main est puissante pour les pousser à abandonner le jeu. Il se pratique donc par l'utilisation de relances assez fortes même sans qu'on n'ait aucune combinaison pour inciter les joueurs à ne pas continuer le jeu et peut s'avérer extrêmement efficace contre des joueurs humains. Il est évident que ceci n'aura pas le même effet contre les agents implémentés mais il a paru intéressant de rajouter cette facette très importante du poker.

Comme expliqué précédemment, le JoueurBluffer utilise le même calcul de confiance que le JoueurPrudent pour décider (mais les valeurs des seuils peuvent être différents). Néanmoins, si jamais sa confiance n'est pas suffisante pour suivre ou tenter une relance, le bluffer ne se couche pas forcément mais à une certaine probabilité de rentrer dans un état de bluff dans lequel il va rester jusqu'à la fin de la partie (bluffer pour une seule relance ne sert pas à grand chose).

Pour estimer cette probabilité on part du principe que, plus les autres joueurs sont prudents et ne tentent pas de relance, plus on peut considérer que cela veut dire que leur main n'est pas si forte que ça et qu'ils risquent donc de ne pas suivre une relance provoquée par un coup de bluff. Par conséquent, on estime un *seuil\_bluff* de la façon suivante :

$$seuil = \frac{nombre\ de\ raises}{nombre\ de\ raises\ maximal} * 100$$

Plus ce nombre est proche de 100, et plus il y a eu de raises à ce tour-ci. On tire ensuite un nombre entre 1 et 100 et, si ce nombre est supérieur au seuil, alors le joueur rentre dans un état de bluff.

Pour estimer ensuite la valeur de la relance, on estime que, parce que le but du joueur est le coup de bluff, il faut définir une valeur minimale de relance. Un petit bluff n'est pas vraiment un bluff et a plus de chances de faire perdre inutilement de l'argent. Le JoueurBluffer est donc défini avec un paramètre *ratio\_min\_relance* qui décrit le ratio minimal de l'argent du joueur dont il va se servir pour relancer en cas de bluff. On tire alors la valeur de relance aléatoirement entre la valeur

minimale  $ratio\_min\_relance * (argent - mise\ minimale)$  et  $argent - mise\ minimale$ .

## 5. Autre agent possible

Un autre type d'agent qu'il aurait été très intéressant d'implémenter est le joueur calculateur. L'idée derrière ce joueur est qu'il va travailler à partir de probabilité de victoire. Pour cela, il devrait être capable d'évaluer la probabilité que sa main a de gagner, en fonction des cartes qu'il peut encore tirer et des cartes que pourraient avoir les autres joueurs. C'est donc le comportement le plus proche de ce que fait un joueur humain (de haut niveau)

Néanmoins, et même s'il existe des techniques pour implémenter cela, comme le **2 + 2 Poker Evaluator**, qui se base notamment sur un gigantesque tableau de plus de 32 millions d'entrées pour évaluer la force de la main, tout ceci aurait demandé beaucoup de temps et il s'est avéré, après recherches sur internet, qu'il était difficile de pouvoir bénéficier facilement de ces différents algorithmes.

## E. Paramètres du modèle

Ici nous allons parler des différents paramètres du modèles qui sont définis à chaque simulation et permettent de modifier cette simulation. Ces paramètres sont les suivants :

### **Paramètres monde :**

- *nb\_joueurs* : il s'agit du nombre total de joueurs présents dans la partie. Il peut varier de 2 à 20.
- *argent\_init* : c'est la somme d'argent que possède chaque joueur au début du jeu et qui peut aller de 10 à n'importe quelle somme.
- *max\_raises* : ceci représente le nombre de relances maximal que peut faire un joueur à chaque tour d'enchères, permettant ainsi de limiter la durée des parties. Cette valeur peut aller de 1 à 10.
- *blind* : la valeur de la blind que doivent payer les joueurs à chaque partie pour pouvoir jouer. Certaines variantes font varier la blind au cours du jeu mais nous avons choisi dans notre modèle d'avoir une blind constante.

### **Paramètres agents :**

- *partsAgents* : map permettant de représenter les parts des différents agents que l'on souhaite avoir dans notre simulation. Dans l'idée, cette map aurait dû être sous la forme d'une liste de paires (Agent::part) mais ceci provoquait une erreur dans GAMA et est donc sous la forme de paires (indice::part). L'indice est unique pour chaque espèce d'agents et est résolu grâce à une autre map, qui n'est pas un paramètre et qui est composée de paires (indice::agent).
- *proba\_coucher\_param*, *proba\_relance\_param*, *proba\_suivre\_param* : ceux-ci sont les trois paramètres du joueur ZIAleatoire, permettant de définir ses probabilités d'action comme expliqué précédemment.
- *seuil\_suivre\_param\_prudent*, *seuil\_relance\_param\_prudent*, *max\_relance\_param\_prudent* : cette fois ce sont les paramètres de seuil de confiance utilisés par les JoueurPrudents dans

leur décision ainsi que la valeur de ratio max de la relance du joueur.

- *seuil\_suivre\_param\_bluffer*, *seuil\_relance\_param\_bluffer*, *max\_relance\_param\_bluffer*, *min\_relance\_param* : ce sont les mêmes genres de paramètres que pour le JoueurPrudent mais qui ont été dédoublés afin de pouvoir créer deux agents différents. Il est à noter la présence en plus du paramètre *min\_relance\_param* qui permet de définir le ratio minimum de relance lors d'un coup de bluff.

# III. Expériences et analyse

## A. Calibration

Comme nous l'avons vu juste avant, il y a un certain nombre de paramètres à définir pour le fonctionnement des agents et qu'il est nécessaire de calibrer. Comme nous n'avons pas cherché à définir un modèle d'agent représentant de manière réaliste un comportement humain, cette calibration n'est pas basée sur des données provenant de véritables parties de poker mais est faite "à la main", en se plaçant comme expert.

Le but de cette calibration est donc surtout d'obtenir des agents exhibant des comportements que l'on attend en les créant et que l'on considère comme plus adéquats. Il est aussi à noter que certains des paramètres seront modifiés durant les expériences afin de pouvoir tester si modification de certains paramètres par rapport à notre calibration ne permet pas d'obtenir de meilleurs résultats (ce qui est donc proche d'un objectif **normatif**).

### *Agents ZIAleatoires :*

Les premiers paramètres d'agents à définir sont ceux des agents ZIAleatoires. De part le principe même de ces agents, il est évident qu'on ne peut pas trouver de "bons" paramètres puisque leur comportement est déjà à la base non idéal. Nous fixons donc leurs paramètres en fonction de la façon dont on aimerait qu'ils se comportent.

De ce fait, et afin d'avoir des simulations intéressants, il est important que leur probabilité de se coucher ne soit pas trop forte, à la fois pour éviter que les parties ne s'éternisent et aussi pour ne pas que les agents ne se couchent pas trop souvent en fin de partie alors qu'ils ont déjà beaucoup misé dans une partie.

De plus la probabilité de suivre une mise doit être la plus importante puisque c'est au final l'action qui présente le moins de risque, surtout que suivre peut parfois se résumer à simplement effectuer une "parole" (c'est-à-dire suivre une mise de 0). Nous en arrivons donc aux paramètres suivants :

Paramètre	Valeur
proba_coucher	5,00%
proba_suivre	75,00%
proba_relance	20,00%

### *Agents Prudents :*

Pour les agents prudents, la calibration peut déjà être un peu plus "réfléchie", dans le sens qu'on a à faire à des comportements plus "humains". Comme leur nom l'indique, on veut partir sur des agents qui vont jouer prudemment et qui vont donc avoir besoin d'une confiance assez élevée avant de pouvoir jouer.

La relance est une action risquée mais qui peut portée ses fruits. Néanmoins, dans le cas d'un comportement prudent, on s'attend à ce que le joueur soit assez passif par rapport aux enchères et se contente principalement de suivre sans prendre de risque tant que les enjeux ne deviennent pas trop important par rapport à son argent.

On veut de plus éviter d'en faire un agent peureux qui se coucheraient tout le temps. Il a notamment été remarqué que, lors des mises de pré-flop, la majorité des agents évaluaient une confiance un peu au-dessus de 40 avec les blinds et l'argent utilisés lors des tests, de part le fait que leur main est encore assez faible à ce niveau-là du jeu. Pourtant, ça serait une stratégie non optimale

de se coucher pendant ce tour, alors même que toutes les cartes n'ont pas été distribuées.

Enfin, pour ce qui est du ratio maximal d'argent que le joueur prudent va mettre dans une relance, de part le fait qu'il joue justement prudemment, il a été estimé qu'il ne devrait pas mettre plus de 10% de son argent dans une relance. Voici donc les paramètres choisis :

Paramètre	Valeur
seuil_suivre	40,00%
seuil_relance	80,00%
ratio_max_relance	0,1

### *Agents Bluffers :*

Pour ces agents, et parce qu'une partie de leur comportement est très proche de celui implémenté dans les agents prudents, il a paru plus intéressant de choisir leurs paramètres de manière à ce qu'ils aient un jeu plus basé sur le risque que les joueurs prudents, ce qui est déjà d'ailleurs un peu pris en compte dans le fait qu'ils tentent le bluff. On choisit donc d'abaisser les deux seuils et d'augmenter le ratio de relance pour les agents bluffers.

Pour ce qui est de leur quatrième paramètre, qui est le minimum de relance lors d'un coup de bluff, il paraît nécessaire de choisir un ratio assez élevé de manière à ce que le coup de bluff ait un effet. Voici donc les paramètres qualifiés pour les agents bluffers :

Paramètre	Valeur
seuil_suivre	30,00%
seuil_relance	70,00%
ratio_max_relance	0,25
ration_min_relance	0,25

## **B. Expériences**

En utilisant le mode Batch de GAMA, plusieurs expériences ont été utilisées afin de tester l'efficacité relative de nos agents. L'objectif principal est de faire s'affronter les différents types de joueurs pour essayer d'en extraire une stratégie et donc un type d'agent meilleur que les autres et ceci selon différentes situations de jeux. Différentes expériences ont donc été conduites et analysées. L'ensemble de ces expériences sont présentes dans le modèle mais mises en commentaire afin de ne pas surcharger l'interface. Il est donc tout à fait possible de les reproduire.

### ***1. Expérience 1 :***

#### ***Présentation :***

Cette expérience (ou plutôt série d'expériences) a pour but d'essayer de voir, avec les paramètres d'agents tels que précédemment, que type d'agent semble être le meilleur. Pour cela, les paramètres globaux fixés sont les suivants :



Paramètre	Valeur
nb_joueurs	12
argent_init	1000
blind	10
max_raises	3

Les paramètres qui varient dans cette expérience sont les parts des différents agents, toutes les combinaisons étant réalisées de manière à voir les différents cas possibles. Dans chaque situation définie par les parts des agents, 100 simulations sont réalisées et on relève l'agent ayant comptabilisé le plus de victoires. Une simulation s'arrête quand on a trouvé un vainqueur (tous les autres joueurs n'ont plus d'argent) ou au bout de 20000 cycles pour éviter de voir des simulations trop traîner. Les résultats sont les suivants :

ZIS	ZIA	P	B	Victoires ZISuiveurs	Victoires ZIAleatoires	Victoires Prudents	Victoires Bluffers	Egalités
50%	50%	0%	0%	55/10	45/1	-	-	11/11
50%	0%	50%	0%	0	-	63/37	-	37/37
50%	0%	0%	50%	0	-	-	100	0
0%	50%	50%	0%	-	17	83/22	-	22/22
0%	50%	0%	50%	-	21	-	79	0
0%	0%	50%	50%	-	-	11/3	41	51/3
33%	33%	33%	0%	0	14	86/30	-	30
33%	33%	0%	33%	22	17/2	-	61	2/2
33%	0%	33%	33%	0	-	16/7	40	51/7
0%	33%	33%	33%	-	7	43/12	19	43/12
25%	25%	25%	25%	0	3	68/24	15	36/24
<b>Total</b>				77	124	370	355	283

On présente à chaque fois dans ce tableau le nombre de victoires de chaque agent sur les 100 simulations ainsi que le nombre d'égalités. Certaines égalités sont parfois uniquement entre des agents de la même espèce et donc dans ce cas, on peut considérer que c'est une victoire pour l'espèce de cet agent. On compte donc cette victoire mais on le signale dans le tableau de la manière suivante : X/Y, ce qui veut dire : "X victoires, dont Y dans les égalités". De même, dans la colonne des égalités, on indique X/Y où X est le nombre d'égalités totales et Y le nombre d'égalités "résolues" comme expliqué précédemment. On a relevé en bleu l'agent totalisant le plus de victoires dans chaque cas.

## 2. Analyse :

Que peut-on tirer de ces résultats ? Tout d'abord, à la simple vue du nombre de victoires, on peut en déduire que les joueurs prudents et bluffers ont des résultats très semblables et, surtout, qu'ils écrasent les joueurs suiveurs et aléatoires (ce qui est rassurant).

Pour ce qui est du fait que le joueur prudent semble légèrement meilleur que le joueur bluffer, il y a deux choses à prendre en compte. La première est l'écart de seulement 15 victoires entre les deux, ce qui est évidemment très peu et à cause duquel il est très difficile de réellement décider

si l'un est meilleur que l'autre. La deuxième chose, c'est que dans toutes les situations où l'on a à la fois le joueur prudent et le joueur bluffer en même temps, la majorité des égalités non résolues, en plus d'être entre des joueurs de ces deux espèces, étaient à l'avantage du joueur bluffer. En effet, à part dans le cas (25%, 25%, 25%, 25%), dans la très grosse majorité de ces égalités, le joueur bluffer possédait plus de 10 fois la somme d'argent du joueur prudent, ce qui laisse à penser que, si on avait attendu encore un peu, c'est le joueur bluffer qui aurait gagné. Quand on voit le nombre d'égalités que cela semble représenter (notamment dans le cas (0, 0, 50%, 50%)), c'est un nombre de victoires en plus non négligeable pour le joueur bluffer.

On peut aussi voir que la stratégie de suivre coûte que coûte semble être moins performante qu'une stratégie purement aléatoire. Ceci provient du fait que le joueur aléatoire, en se couchant aléatoirement, va éviter des situations où il aurait perdu beaucoup d'argents et ses relances peuvent en plus provoquer des coups de bluffs et l'aider à gagner dans des situations dans lesquels il n'aurait normalement pas pu.

Un autre point à noter est le nombre très fort d'égalités, surtout dans les situations dans lesquelles le joueur prudent est présent (c'est quand il est là qu'on observe les égalités les plus fortes). Ces égalités ne sont pas des égalités strictes dans le sens qu'elles résultent simplement du fait que le nombre de cycles d'exécution n'était pas assez important pour les résoudre. Ceci semble donc vouloir dire que la présence d'un joueur prudent va avoir tendance à ralentir le jeu. C'est en effet une stratégie basée sur des relances assez faibles et sur le moindre risque, ce qui entraîne évidemment un allongement de la durée du jeu.

Une autre donnée intéressante, est le fait que le joueur bluffer semble gêné par le joueur aléatoire. En effet, bien qu'il gagne haut la main contre ce dernier et qu'il arrive à gagner significativement (ne pas oublier que la majorité des égalités sont à l'avantage du bluffer) contre le joueur prudent quand il n'y a pas de joueurs aléatoires, dès que celui-ci est présent dans une partie, le joueur bluffer perd contre le joueur prudent. La raison que je vois est que le joueur aléatoire détruit complètement la stratégie de bluff du joueur bluffer puisqu'il y est, par définition, insensible, alors que le bluff est une stratégie qui peut très bien fonctionner contre un joueur prudent seul. On aurait pu s'attendre à ce que le joueur suiveur gêne plus que ça le joueur bluffer, étant lui aussi insensible au bluff, mais ça ne semble pas être le cas, sans doute à cause du fait que les joueurs suiveurs, de par le fait qu'ils ne se couchent jamais, perdent très rapidement.

## 2. *Expérience 2 :*

### *Présentation :*

Le but de cette deuxième série d'expériences est de tenter d'améliorer les agents prudents et bluffers. Pour cela, on fixe les paramètres globaux suivants :

Paramètre	Valeur
nb_joueurs	12
argent_init	1000
blind	10
max_raises	3
partsAgents	(25%, 25%, 25%, 25%)

Le choix a été fait de mettre une configuration avec  $\frac{1}{4}$  de chaque agent afin de mettre la configuration la plus "réaliste" dans le sens que, dans une partie, les joueurs sont sensés tous jouer de manière très différente. On lance encore des séries de 100 simulations donc les conditions

d'arrête sont la victoire d'un joueur ou une durée dépassant 20000 cycles et on fait varier les différents seuils de chaque agent séparément, ce qui veut dire qu'avec chaque expérience on cherche le meilleur agent indépendamment. Les paramètres que l'on fait évoluer sont les deux valeurs de seuil. `seuil_suivre` varie parmi [20, 30, 40, 60] et `seuil_relance` parmi [60, 70, 80, 90]. Les meilleures configurations pour les deux agents que nous avons trouvées sont alors :

Type d'agent	seuil_suivre	seuil_relance
Prudent	40	60
Bluffer	30	60

La deuxième partie de cette expérience est alors de comparer ces deux agents, maintenant qu'on a trouvé leur meilleure configuration, afin de savoir si une stratégie semble vraiment meilleure qu'une autre entre les deux. Pour cela, on lance simplement une série de simulation avec les mêmes paramètres globaux et les configurations d'agents trouvées précédemment. On garde les mêmes parts d'agents, puisque c'est dans ce contexte qu'on a trouvé ces configurations optimales. Il est à noter que, puisque les paramètres ne vont pas varier pendant les simulations, on peut se permettre d'augmenter le nombre de simulation à 1000 et la durée maximale d'une simulation à 30 000. On obtient alors les résultats suivants :

Type d'agent	Nombre victoires
Prudent	716
Bluffer	243

On a donc une victoire très nette du joueur prudent sur le joueur bluffer. Il est néanmoins à noter que les autres paramètres de ces deux agents (c'est-à-dire les ratios de relance) n'ont pas été modifiés et qu'il aurait peut-être pu être intéressant de voir s'ils ont un impact sur le résultat de ces agents. Dans tous les cas, le meilleur comportement semble être celui de la prudence mais avec un seuil de suivi relativement bas et un seuil de relance lui aussi finalement plus bas que ce qui était pensé au début.

### 3. Expérience 3 :

#### *Présentation :*

La troisième et dernière expérience, a pour but de tester l'influence de l'argent et des blinds sur les résultats des différents agents. Pour cela, on part avec les paramètres globaux suivants :

Paramètre	Valeur
<code>nb_joueurs</code>	12
<code>max_raises</code>	3
<code>partsAgents</code>	(25%, 25%, 25%, 25%)

Pour ce qui est des paramètres d'agents, on utilise les mêmes paramètres du JoueurZIAleatoire qu'avant et on utilise les meilleures configurations trouvées précédemment pour les JoueurPrudents et JoueurBluffers. L'idée est de comparer le nombre de victoires de chaque agent en fonction de la variation de paramètres globaux.

Pour cela, on repart sur une durée limite de 20000 cycles et 100 répétitions de chaque simulation. L'argent varie dans l'ensemble [1000, 2000, 5000, 10000], tandis que les blinds varient

dans l'ensemble [10, 20, 50, 100]. Les résultats obtenus sont alors les suivants :

Argent	Blind	Victoires ZISuiveur	Victoires ZIAleatoire	Victoires Prudent	Victoires Bluffer	Egalités
1000	10	1	4	68	27	0
1000	20	0	3	70	27	0
1000	50	0	9	75	16	0
1000	100	7	4	58	30	1
2000	10	0	2	71	27	0
2000	20	0	2	75	23	0
2000	50	0	4	73	23	0
2000	100	2	6	65	26	1
5000	10	0	5	71	23	1
5000	20	0	6	73	21	0
5000	50	0	3	78	19	0
5000	100	1	4	66	29	0
10000	10	0	1	78	21	0
10000	20	0	3	83	14	0
10000	50	0	4	77	19	0
10000	100	0	3	72	25	0

Cette expérience ne semble pas vraiment apporter beaucoup d'informations à la vue de ce tableau. En effet, les variations apportées par les modifications de l'argent initial et de la blind ne semblent pas réellement se refléter dans les victoires des différents types d'agents.

Les seules variations qui semblent être vraiment "marquantes" sont les victoires des agents prudents et bluffers quand les blinds valent 100. C'est pour cette valeur que l'agent prudent a le moins de victoire et l'agent bluffer le plus de victoires, les deux étant liées puisque ce sont ces deux agents qui se battent vraiment pour les victoires. Ceci peut peut-être s'expliquer par le fait que l'agent prudent va être gêné par une blind élevée puisqu'elle va diminuer sa valeur de confiance et donc le pousser à se coucher plus souvent. Le joueur bluffer, lui, va tenter des coups de bluffs là où le joueur prudent se couche, ce qui explique que le bluffer peut profiter de ce moment de faiblesse du joueur prudent.

On peut globalement voir d'ailleurs que le joueur prudent semble être meilleure à mesure que la somme d'argent initial. Ceci est lié à l'explication précédente puisque, plus l'argent initial augmente, plus le risque sur la mise diminue, ce qui augmente donc la confiance du joueur prudent et le pousser à jouer.

## Conclusion

Nous avons présenté tout au long de ce rapport, l'implémentation en GAMA d'un modèle de simulation multi-agents pour le jeu de Poker Texas Hold'em. Ce jeu jouit en effet aujourd'hui d'une très grande popularité et possède de nombreuses qualités stratégiques, mêlant chance, bluff et calcul, qui rendent la création d'une simulation l'ayant pour sujet très intéressante. L'objectif principal a été de fournir un modèle permettant l'ajout relativement simple d'agents sans avoir à se soucier des spécificités techniques du modèle. Ainsi, toute la réflexion peut être placée dans la conception de l'intelligence-même de l'agent afin de chercher à développer des bons agents joueurs de Poker.

Il est à noter que ce modèle n'est pas fait, sans modification, pour permettre à un joueur humain de participer à une partie de Poker et de se mesurer aux agents créés. L'idée principale qui a guidé la conception de cette simulation est surtout, parce que c'est une simulation, de permettre de tester différents types d'agents entre eux et donc de pouvoir aussi effectuer un benchmark des performances de différents agents.

Durant le développement de ce modèle, quatre types d'agents ont été codés. L'agent JoueurZISuiveur, agent zéro intelligence, avait pour but de simuler un comportement très simple et idiot de suivre constamment pour essayer de rester le plus longtemps possible en jeu. Le deuxième type d'agents, un zéro intelligence aussi, est un agent purement aléatoire qui, d'après un certain nombre de probabilités qui le définissent, va jouer en fonction d'un nombre aléatoire qu'il tire. Ces deux agents se sont bien évidemment montrés très inefficace et ne simule en rien un comportement réaliste de joueur de Poker. Même s'il est tout à fait imaginable de trouver des joueurs qui jouent ainsi.

Les deux autres types d'agents ont été des tentatives d'avoir des agents plus intelligents en en faisant des agents cognitifs. Le JoueurPrudent fonctionne selon l'évaluation d'une valeur de confiance, basée uniquement sur la force de sa main et sur le risque qu'il prend en misant par rapport à l'argent qu'il lui reste, qui lui sert à rendre sa décision. C'est donc un joueur très égo-centré qui ne joue qu'en fonction de ce qu'il a sous les yeux. Mais, le bluff étant une part importante du Poker, un dernier agent, le JoueurBluffer, créé sur une base de JoueurPrudent, a été créé afin d'essayer de toucher à cette stratégie. Ce dernier joue comme un JoueurPrudent, sauf qu'il peut tenter, si sa main est trop mauvaise des coups de bluffs. Étonnamment, les performances du joueur bluffers se sont avérées très en dessous de celles du JoueurPrudent mais son fonctionnement est encore très basique et il serait sans doute tout à fait possible de beaucoup l'améliorer et l'amener à battre (souvent) un JoueurPrudent.

En conclusion, on pourrait parler de ce qui pourrait encore être fait dans cette simulation, notamment sur les agents à développer. En effet les quatre agents implémentés restent extrêmement simple et il aurait été intéressant d'avoir au moins un agent capable de calcul probabiliste, comme sont sensés le faire les meilleurs joueurs de Poker. Ce type d'agent existe dans les simulations de Poker mais s'est avéré malheureusement très difficile à implémenter. Néanmoins, le modèle est tout à fait capable d'accueillir de nombreux nouveaux agents différents, ce qui pourrait être intéressant dans le futur.