

Problématiques d'intelligence artificielle  
dans le domaine des jeux sur ordinateur

Damien Clauzel

15 décembre 2003

*A strange game.  
The only winning move is  
not to play.*  
– Joshua, Wargames

# Table des matières

<b>1</b>	<b>Des jeux intelligents ?</b>	<b>3</b>
1.1	Historique . . . . .	3
1.2	Attraits de l'intelligence artificielle . . . . .	4
1.3	Quelques repères . . . . .	5
<b>2</b>	<b>Les modes fondamentaux</b>	<b>6</b>
2.1	Tours de jeu . . . . .	6
2.1.1	Tour par tour . . . . .	6
2.1.2	Déroulement continu (temps réel) . . . . .	7
2.2	Raisonnements . . . . .	8
2.2.1	Algorithmique . . . . .	8
2.2.2	Expertise . . . . .	8
2.2.3	Neuronique . . . . .	9
<b>3</b>	<b>Domaines concernés</b>	<b>10</b>
3.1	Adversaire informatique . . . . .	10
3.2	Relations vis à vis de l'environnement . . . . .	11
3.3	Acquisition et analyse d'informations . . . . .	12
<b>4</b>	<b>Projection dans l'avenir</b>	<b>15</b>
4.1	Support matériel . . . . .	15
4.2	Gestion de profils . . . . .	16
<b>5</b>	<b>Conclusion</b>	<b>17</b>

# Chapitre 1

## Des jeux intelligents ?

### 1.1 Historique

Au début était l'informaticien. Et cet informaticien, venant du monde des mathématiques, se mit à rêver. Rêver que ces nouveaux ordinateurs, capable d'effectuer plusieurs milliers d'opérations par secondes, seraient un jour en mesure de résoudre des problèmes complexes. Un autre de ses espoirs était de donner vie à la machine, ou tout du moins l'intelligence dans un premier temps, cela lui suffirait largement.

Au fur et à mesure que les années passaient, que les ordinateurs gagnaient en puissance, que les algorithmes s'amélioraient et que la connaissance globale du domaine s'accroissait, les problèmes abordés devinrent de plus en plus complexes. Déjà, le morpion laissait la place aux échecs, aux dames et au go[1]. Des voyageurs de commerce parcouraient inlassablement les routes à la recherche d'un chemin optimal sur des cartes coloriées au plus juste. On assistait à la définition de catégories de problèmes, et à l'émergence de nouvelles approches. Cela se passait des années 60 à 90. De l'apparition des premiers ordinateurs dans les universités jusqu'au débarquement massif de l'informatique individuelle.

Lorsque John MacCarthy affirmait que "toute activité intellectuelle peut être décrite avec suffisamment de précision pour être simulée par une machine", il ne songeait probablement pas aux jeux actuels sur ordinateur. Depuis une dizaine d'années, les jeux informatiques ne se contentent plus de reproduire des principes existants, mais en inventent d'autres. À côté des

jeux traditionnels comme les échecs se trouvent les wargames dont les figurines jadis de plombs s'animent désormais au son du clairon virtuel, mais aussi des simulations de courses de véhicules ou des immersions à la première personne dans un univers tridimensionnel rempli d'adversaires hostiles.

## 1.2 Attraites de l'intelligence artificielle

Les principaux avantages d'un adversaire informatique sont qu'il est toujours disponible et d'humeur égale, même quand il perd. D'autres qualités sont sa capacité d'adaptation de niveau, sa logique et sa faculté à suivre des raisonnements prédéfinis. Cela permet d'essayer des variantes ou encore d'affiner des stratégies en les reprenant à un certain point. Au-delà de ces éléments, des logiciels sachant jouer sont de nos jours considérés comme des adversaires à part entière. Il n'est pas rare de voir des titres de rang attribués à des programmes, les joueurs pouvant ainsi se comparer à eux. Cet aspect culturel est entré dans le monde du jeu depuis de nombreuses années et devrait y rester. Quel joueur d'échecs pourrait aujourd'hui affirmer que Deeper Blue n'est pas un vrai joueur d'échecs, capable non seulement d'analyse, mais également de prise de décision stratégique et d'improvisation ? La complexité de son raisonnement est devenue telle que ses développeurs ne sont plus capables de prédire avec certitude ses actions à venir, mais juste de donner des indications sur son travail en cours. Trop complexe pour être rangé dans la catégorie des programmes pousseurs de pions, mais pas encore assez pour prétendre pouvoir se passer de ses concepteurs afin de continuer à s'améliorer. Si son cas n'est pas représentatif de l'intelligence artificielle en elle-même, il en est certainement un symbole fort.

Un intérêt indirect de l'intelligence artificielle se cache dans la maniabilité d'un jeu. Si l'ordinateur arrive à repérer des habitudes chez un joueur, il peut lui proposer de prendre en charge lui-même ces tâches, afin que l'être humain puisse se consacrer à autre chose. Par exemple, en sélectionnant un groupe d'unités, si le joueur en laisse une de côté car manifestement son action était trop malhabile, l'ordinateur pourra juger s'il est pertinent d'inclure ou non cette unité. Cette catégorie, plus orientée sur les interactions homme-machine que sur la réflexion pure, connaît actuellement une forte croissance. Les jeux étant de plus en plus riches, il devient nécessaire de

fournir aux joueurs des outils en adéquations avec la complexité des interactions disponibles.

### 1.3 Quelques repères

Une poignée de grands jeux, connus de tous et faisant l'objet d'études depuis de nombreuses années, font office d'échelle de comparaison dans le domaine du jeu sur ordinateur. Voici les principaux d'entre eux[2]. Chacun caractérise une façon d'aborder la question du jeu, mais quasiment tous emploient la même méthode, la force brute. Des approches différentes comme la reconnaissance de formes pour le Go ou les réseaux de neurones pour othello sont parfois utilisés, mais cette approche reste très minoritaire malgré les succès rencontrés.

Nb possibilités	Jeu	Nb emplacements	Nb pièces
$10^{12}$	Awélé	12 trous	48 graines
$10^{32}$	Dames	100 cases	2 x 20 pions noirs et blancs
$10^{60}$	Othello	64 cases	64 pions bicolores
$10^{128}$	Échecs	64 cases	2 x 16 pièces noires et blanches
$10^{172}$	Go	361 points	180 pierres blanches et 181 pierres noires

État actuel :

- \* **Bridge** : les programmes ne dépassent pas le niveau d'un débutant
- \* **Awélé** : 889 063 398 406 parties possibles, jeu maîtrisé
- \* **Dames** : les IA ont dépassé les champions humains
- \* **Othello** : victoire sans appel de l'ordinateur en 1997
- \* **Échecs** : uniquement une poignée d'humain arrivent à tenir tête aux ordinateurs
- \* **Go** : l'ordinateur a le niveau d'un faible joueur en club

## Chapitre 2

# Les modes fondamentaux

### 2.1 Tours de jeu

Nous distinguerons deux grands modes de fonctionnement des jeux : le tour par tour et le continu, souvent nommé à tort "temps réel".

#### 2.1.1 Tour par tour

Nous évoluons ici dans le domaine des jeux dit "classiques". Morpion, échecs, dames, go ou backgammon, ces jeux ont en commun plusieurs caractéristiques[3] : les joueurs jouent à tour de rôle, à chaque tour ils choisissent une action parmi un nombre fermé et petit de possibilités et on peut prédire l'effet d'une action sur le jeu. Ce type de fonctionnement laisse un temps de réflexion élevé à l'ordinateur. Celui-ci peut en effet exploiter non seulement le temps qui lui est imparti durant son tour, mais également durant celui de son (ou ses) adversaire(s). De plus, la gestion du jeu à proprement dit est souvent extrêmement simple, requérant donc peu de ressources. L'essentiel des calculs se porte par conséquent sur la réflexion en elle-même.

Traditionnellement, les problèmes posés dans cette catégorie de jeu sont résolus par la force brute. La représentation des situations passant par des arbres, on se retrouve à employer des algorithmes bien connus, sur lesquels sont apportées des améliorations mineures. Les heuristiques[4] sont les fers de lance de ce type de résolution car intervenant aux endroits critiques de ces algorithmes, à savoir comment juger quelles branches de l'arbre il est intéressant de développer.

Les problèmes rencontrés dans ces cas sont principalement la taille des arbres à explorer<sup>1</sup>. Plus la fonction d'évaluation est élaborée, plus il faudra de temps pour la calculer, ce qui limite la profondeur de l'arbre étudié. En revanche, une fonction simple donnera une réponse rapide, mais potentiellement d'une qualité faible par rapport à ce qui aurait pu être trouvé. Tout l'art des heuristiques repose sur le subtil compromis entre l'efficacité et la vitesse.

### 2.1.2 Déroulement continu (temps réel)

Les jeux à déroulement continu constituent une gageure pour l'ordinateur. Non seulement doit-il gérer le jeu en lui-même, planifier ses actions et tenir compte des agissements des autres participants (humains ou informatiques), mais également le faire rapidement, et de façon continue. On imagine mal les joueurs d'une équipe de football s'arrêtant toutes les 3 minutes afin de décider des actions à suivre. Certains jeux n'existent que par cette notion de continuité, afin de mettre à l'épreuve les réflexes et capacités de raisonnement instantané de l'être humain.

Dans cette approche, la question de la réactivité est primordiale. Ne pas agir est souvent aussi lourd de conséquences que de mal agir. Les méthodes de prise de décision ont donc une très forte contrainte temporelle. Ainsi, dans les jeux d'échecs on pourra s'astreindre de rendre une réponse en un temps donné. L'accent est par conséquent mis sur la segmentation d'un problème en un ensemble pouvant être traité plus simplement, et rapidement. Une pratique courante est d'affecter un facteur de priorité à chacune de ces sous-tâches, et d'effectuer une juste répartition du temps de calcul disponible selon des critères relatifs à la situation actuelle dans le jeu.

Ici aussi, les heuristiques jouent un rôle primordial dans la détermination des actions à effectuer en priorité. On les retrouve à différents niveaux, mais celles suscitant actuellement le plus d'intérêt sont celles en haut de la pyramide de décision, car ce sont elles qui conditionnent la stratégie globale de l'ordinateur. Mieux vaut-il attaquer que se défendre? Sur quelle partie

---

<sup>1</sup>Un joueur d'échecs a en moyenne 35 possibilités de jeu à chaque coup, une partie tenant habituellement lieu sur 30 demi-coups, la taille de l'espace d'états est de l'ordre de  $35^{30}$



du jeu dois-je fournir un effort ? Les meilleurs techniques de jeu ne servent à rien au niveau local si elles ne sont pas décidées d'un point de vue global.

## 2.2 Raisonnements

Il existe 3 grandes façon de raisonner pour l'ordinateur[5].

### 2.2.1 Algorithmique

La résolution du problème est obtenue par la mécanisation. Il s'agit de trouver une méthode systématique conduisant au résultat. Tout le chemin à parcourir est décrit pas à pas pour atteindre le résultat désiré. Il s'agit là d'une procédure d'automate totalement planifiée et présentée selon une séquence très précise à respecter. Il faut étudier complètement le problème et codifier exhaustivement la solution. Il est nécessaire de bien décortiquer celle-ci pour la mettre dans un automate rapide qui va la dérouler beaucoup plus rapidement que le cerveau. On peut y ajouter un peu de réflexion et optimiser le calcul, mais cela restera une procédure imaginée par l'homme.

Les jeux mettant en oeuvre cette méthode sont les plus courants. En effet, dans la grande majorité des problèmes, la définition d'un algorithme suffit à obtenir une réponse à une question donnée. Cependant, on se prive de la capacité d'apprentissage.

### 2.2.2 Expertise

On effectue une mécanisation du raisonnement pour obtenir des déductions et des conclusions. Grâce à la connaissance du domaine, préalablement et consciencieusement communiquée par un expert, on décrit les règles à suivre. Il s'agit ensuite de les enchaîner d'une façon générale en employant les résultats fournis au fur et à mesure. Cela revient à transcrire tout le savoir et la maîtrise d'un expert d'un jeu précis dans une base de connaissance, puis de confronter le système au problème.

Cette méthode est très rarement employée pour des raisons de performances. C'est un système dont on ne sait pas prédire l'arrêt, et dont les étapes intermédiaires sont insatisfaisantes. Il nécessite une compréhension

aigu des mécanismes du jeu et n'est (pratiquement) pas capable d'apprendre à partir de situations inconnues.

### **2.2.3 Neuronique**

Le but de cette méthode est l'apprentissage par la pratique, en tentant de mémoriser les différences entre les parties. Aucune expertise externe n'est apportée, si ce n'est lors de la création du réseau et par mimétisme des opposants. Un programme de sélection récompense les meilleures méthodes et élimine les plus faibles. Ce procédé est gourmand en temps, et est généralement réalisé en deux phases. La première consiste à laisser la machine apprendre et à développer sa propre logique de résolution du problème. La 2<sup>ème</sup> est l'exploitation proprement dite, où l'on confronte la méthode de la machine à la situation afin de la résoudre.

Cette façon d'aborder la résolution de problème est considéré comme la plus élégante et la plus efficace car l'ordinateur décide seul de la manière de résoudre la situation qui lui est exposé, sans aide extérieur et en s'adaptant. Malheureusement, elle demeure fort difficile à exploiter à cause de la longueur d'apprentissage, et de la complexité inhérente à la gestion des neurones.

## Chapitre 3

# Domaines concernés

### 3.1 Adversaire informatique

Une façon naturelle d'aborder la question des adversaires informatiques est d'envisager la simulation des raisonnements humains. Or, les réalisations de ces dix dernières années ont montrés que les performances n'étaient pas à la hauteur des espérances. Outre le fait d'être trop lourde et complexe à mettre en oeuvre pour être efficace, cette méthode implique aussi une connaissance suffisante de l'intelligence humaine telle que nous l'employons au quotidien. Cette voie est donc laissée de côté en faveur d'une approche opposée, à savoir créer une forme de raisonnement propre à la machine.

Une des questions revenant fréquemment en avant est de savoir s'il est souhaitable pour l'utilisateur de pouvoir différencier un adversaire humain d'un opposant informatique, en se basant uniquement sur ses réactions dans le jeu. On voit déjà le test de Turing[6] poindre son nez à l'horizon. Si l'on interroge les principaux intéressés, à savoir les joueurs eux-mêmes, il ressort globalement que ceux-ci préfèrent affronter des adversaires ayant un comportement semblable à celui d'un humain plutôt qu'un "stupide programme". Le critère de qualité, fort subjectif au demeurant, semble être la capacité d'un programme à s'inspirer de comportement humain afin d'assurer une bonne efficacité. Même si cela peut en réalité imposé une façon de jouer plus faible à l'ordinateur, car la stratégie humaine n'est pas toujours la plus pertinente. Ainsi, là où l'ordinateur pourrait déterminer que la méthode gagnante est d'effectuer toujours le même parcours entre différents points du jeu afin de réaliser des objectifs mineurs, le joueur humain emploiera un

autre comportement en argumentant que oui, peut-être que cette façon de jouer est efficace en terme de victoire, mais elle ne l'est pas du point de vue ludique.

Un individu humain joue principalement pour le plaisir[7], alors que l'ordinateur joue pour gagner. Il y a là une différence fondamentale entre nos deux protagonistes. à ma connaissance, il n'existe pas de travaux actuels prenant en compte l'aspect qu'un opposant informatique doit avant tout *distraindre* l'humain avant de lui opposer une résistance farouche.

### 3.2 Relations vis à vis de l'environnement

La tendance actuelle dans le domaine des jeux sur ordinateur est l'agrandissement massif des zones exploitables. Les cartes deviennent de plus en plus vastes, avec toujours plus de complexité. Reliefs, moyens de déplacement, perspectives, éléments évolutifs, tout ceci contribue à offrir à l'utilisateur encore plus d'interactions, et donc d'amusement. Là où les choses deviennent difficiles, c'est que notre ami l'ordinateur, lui, s'arrache les puces. Comment tenir compte de tous ces paramètres[8]? Quels sont les critères permettant de calculer le déplacement optimum de ses unités? Et s'il doit en outre gérer des groupes d'éléments aux caractéristiques hétérogènes (vitesse de déplacement, puissance de feu pour forcer le passage, etc), il fini par ne plus s'en sortir. Il est toutefois possible de décomposer une situation complexe en plusieurs sous-problèmes. Ainsi, le déplacement coordonné de plusieurs groupes d'unité (niveau macro) pourra être géré en employant une méthode, tandis que la gestion d'un groupe sera confié à un autre, pour finir au niveau micro, l'unité en elle-même, qui ne se déplacera que dans le cadre restreint du groupe. Selon les cas, on peut employer algorithmes comme A\* ou des techniques multi-agents communicant via un blackboard. Mais cette dernière méthode est peu usitée, car difficilement employable dans des logiciels à la durée de vie courte.

La capacité pour l'ordinateur à tirer profit de l'environnement est aujourd'hui un des éléments clé en matière de stratégie[9]. Non seulement cela lui permet de faire la différence entre les bons et les mauvais endroits ("hum, ceci est de la lave, et je sais que m'y immerger n'est pas une action souhaitable pour ma survie"), mais permet également d'associer des notes à un

emplacement ("je suis mort ici durant les 3 dernières parties, donc je ne vais pas retourner dans ce tunnel sans en savoir plus")

Dans cette catégorie de problématique se trouve également des cas plus traditionnels, comme la question de déterminer le chemin optimal entre 2 points. Aux difficultés de la NP-complétude s'ajoutent les paramètres évoqués ci-dessus. Tout ce petit monde évoluant naturellement constamment dans le temps.

Il existe deux façons d'appréhender la gestion du terrain dans un jeu sur ordinateur : l'approche générique consistant à définir des mécanismes abstraits de perception et d'analyse de cartes, et l'approche spécifique où l'ordinateur se base sur des informations construites lors de la création de la carte[10]. S'il est communément admis que l'idéal semble être de développer des solutions de haut niveau afin de pouvoir les réemployer partout, comme le ferait un être humain, dans la pratique les développeurs ajoutent souvent aux cartes des métadonnées relatives à l'environnement. Comment atteindre rapidement un point difficile, dans quelle zone se trouve tel objet, etc. La simplification des calculs qui en résulte permet à l'ordinateur de se consacrer à d'autres tâches, permettant ainsi de ne pas s'engluer dans des détails secondaires. L'inconvénient de cette tactique est que si la carte est modifiée après la suite, ces métadonnées risquent fort de devenir obsolètes. La complexité du développement est donc plus importante. La conception de cartes demandant énormément de temps, et nécessitant différents experts, on comprend le besoin d'amélioration de l'approche générique. Cependant, celle-ci semble de nos jours encore loin d'être réellement exploitable.

La majorité de ces problèmes sont de nos jours peu traités, ou bien naïvement. Des pistes sont étudiées du côté des systèmes d'information géographique pour en retirer de l'aide, mais les liaisons ne sont pas évidentes. Le plus souvent, des heuristiques larges sont employées, avec des probabilités de succès fortement liés à un contexte précis.

### **3.3 Acquisition et analyse d'informations**

La prise de position repose sur la connaissance d'informations. Elles peuvent être perçues ou déduites, complètes ou partielles, certaines, imprécises voir

même erronées, mais toutes participent à l'élaboration d'une représentation de la situation par l'ordinateur qui lui servira à décider de ses prochaines actions.

Tout comme le joueur, l'ordinateur commence par rassembler des informations sur son environnement. L'être humain le fait de façon intuitive et effectue énormément d'assomptions sur des implications de sens commun : les murs sont solides, le feu brûle, une porte doit pouvoir s'ouvrir, etc. L'ordinateur, lui, part de zéro. Le problème est donc de fournir à l'ordinateur ces renseignements, de les tester, de le remettre en cause et d'en apprendre de nouveaux.

Une autre facette des connaissances que doit acquérir l'ordinateur est que chaque joueur, humain ou informatique, influe sur l'environnement : une porte que l'on a laissée fermée pourra être retrouvée ouverte, un objet que l'on a laissé pourra avoir disparu. Ces connaissances ne sont pas figées, elles évoluent en fonction des événements. Par conséquent, il faut trouver une méthode pour attribuer un score de véracité à chaque information, avec un affaiblissement en fonction du temps : si on laisse de l'argent au milieu d'une rue fréquentée au temps  $t$ , il sera très probablement encore là au temps  $t+1$  ; mais à  $t+1000$ , ce n'est plus aussi certain...

Il peut être judicieux de faire la distinction entre 2 types de connaissances. Les premières sont celles du monde, fixes, universelles et absolues. Ce sont par exemple la gravité, le fait que la pierre est plus résistante que le bois, que l'on se noie dans l'eau, etc. Ces informations sont habituellement fournies par le développeur, l'ordinateur n'ayant pas ici à redécouvrir la roue, ce qui lui permet de tenir tête à l'humain. L'intérêt de ce type de connaissance réside dans leur généralité. Si les éléments ont un comportement identique d'un terrain à l'autre, d'un jeu à l'autre, il est possible de définir des collections de connaissances pouvant servir de base à de futurs développements. De plus, la définition commune permet d'assurer une cohérence vis à vis du joueur : dans un certain type de jeux, il est logique que des éléments semblables adoptent des réactions similaires face à un stimulus répété.

Le second type d'information englobe le reste. Susceptibles d'évoluer, elles devront être remises en question. L'être humain les perçoit avec ses sens et les

traduit en schémas mentaux ; l'ordinateur, lui, n'a pas d'action particulières à effectuer. La position d'un élément, le niveau d'un autre, qu'importe, il lui suffit de consulter sa mémoire. Une situation inacceptable pour le joueur humain qui se sent ici lésé car il ne peut pas savoir ce qui se trouve derrière le mur, lui. Cette question d'éthique est en fait souvent ignorée afin d'améliorer les performances de l'ordinateur. Au lieu de passer de nombreux cycles de calculs à déterminer ce que peut "voir" un personnage contrôlé par l'ordinateur, il est plus rapide de prendre directement les informations dans le référentiel et de les exploiter. Le temps ici gagné sera employé à d'autres tâches.

Dans le cas où l'on dispose des ressources informatiques nécessaires à la résolution de ces calculs de perceptions, on rentre dans une optique d'apprentissage. L'ordinateur doit donc posséder des capacités à comparer des éléments, à reconnaître des schémas plus ou moins similaires et à les associer. Les réseaux de neurones peuvent ici être employés, mais les contraintes de réactivité font que souvent on leur préfère des solutions plus rapides comme les classiques parcours d'arbres. Ces derniers font d'ailleurs des merveilles dans la gestion de petits nombres d'éléments, ce qui tend à démontrer que dans de très nombreux cas des heuristiques suffisent à résoudre des problèmes semblant nécessiter de "l'intelligence".

La plupart du temps, l'ordinateur se basera donc sur les informations du monde pour prendre ses décisions, et la petite partie d'apprentissage ne servira qu'à moduler légèrement ses décisions. Ce problème profite des avancées de l'intelligence artificielle en la matière, et trouve ici des domaines d'applications concrets.

La limitation de l'acquisition des connaissances dans le domaine du jeu provient surtout du matériel, et non pas des méthodes de calculs. D'avantage de puissance de calcul permettrait de réaliser des progrès sans remettre en question les méthodes de raisonnement actuelles.

## Chapitre 4

# Projection dans l'avenir

### 4.1 Support matériel

Un des goulets d'étranglement actuel en matière d'intelligence artificielle étant la puissance de calcul disponible, on pourrait imaginer dans un futur proche le développement de supports matériels dédiés. Cette pratique a en effet permis aux technologies de l'imagerie de réaliser des progrès spectaculaires il y a quelques années grâce à des produits comme la 3Dfx<sup>1</sup>.

Des supports tels que la NNPC-12<sup>2</sup> commencent à apparaître, mais dans des environnements spécialisés. Je pense toutefois qu'il y a un avenir certain pour ces supports matériels. La sous-traitance des calculs répétitifs que sont les parcours d'arbres ou les itérations dans un réseau de neurones pouvant être confié à des processeurs spécialisés, il serait ainsi possible d'améliorer les capacités d'analyse du système.

Cependant, cela reviendrait à figer des méthodes de résolutions, et les jeux voulant exploiter ces supports devraient le faire avec des contraintes immuables. Il y aurait donc un risque de perdre le côté évolutif de cette solution.

---

<sup>1</sup><http://www.nvidia.com>

<sup>2</sup><http://www.modulusfe.com/nnpc12/default.asp>



## 4.2 Gestion de profils

Plusieurs jeux commencent à intégrer la notion de profil d'utilisateur. Au fil ses parties, l'ordinateur en apprend de plus en plus sur celui-ci, lui permettant d'adapter ainsi sa façon d'agir pour toujours renouveler l'intérêt du jeu. Plus l'utilisateur progresse, plus l'ordinateur est capable de soutenir son niveau. Toutefois, un apprentissage est spécifique à un jeu, pour un utilisateur donné.

Il pourrait être intéressant d'étendre ce principe à un ensemble de jeux, toujours pour un même utilisateur. On aura donc un profil générique de lui que pourraient interpréter différents logiciels, afin de s'adapter dès le début à ses caractéristique. Un utilisateur mettant du temps à progresser se verrait ainsi opposé dès le début des adversaires informatiques adaptés à son niveau, au lieu de se baser sur un modèle relativement abstrait du joueur débutant.

Je vois apparaitre dans cette solution des questions d'éthiques : que se passera-t'il si l'on essaye de déduire les facultés intellectuelles d'une personne à partir de ces informations ? Y'a-t'il des risques de dérives possibles en extrapolant depuis une activité ludique, donc non productive, des capacités de travail ? Mais si l'on tient compte de ces concidérations, pourrait-on profiter de cette masse d'information afin d'accroître la connaissance sur l'être humain ?

## Chapitre 5

# Conclusion

Edward Fredkin disait que *l'intelligence, c'est d'avoir un problème et de savoir le résoudre*<sup>1</sup>. S'il est vrai que désormais on comprend mieux les problématiques qui se posent en matière de jeux sur ordinateurs, on sait également que l'intelligence artificielle n'est pas la panacée que l'on croyait. Les méthodes de résolutions classiques que sont les algorithmes de parcours de graphes montrent aujourd'hui qu'elles ne sont pas prêtes de partir. Et d'ailleurs, pourquoi le feraient-elles ? Elles s'acquittent parfaitement des tâches qui leur sont confiées.

Face à la complexité croissante des jeux, la définition de catégories a permis la segmentation de problèmes complexes en sous-ensemble de domaines réduits, ce qui a contribué à la réalisation d'approches plus fines en matières de résolution. Chaque catégories peut en effet être vue comme une application concrète des grandes problématiques d'intelligence artificielle. Recherche de parcours, prises de décision, apprentissage, tout est repris et exploité. Les jeux commerciaux profitent très rapidement des avancées des domaines, et en sont même parfois les moteurs. Des logiciels comme *Creatures*<sup>2</sup> ou plus récemment *Black and White*<sup>3</sup> ont surpris bon nombre de joueurs grâce à leurs réseaux de neurones. Mais après une phase de découverte, le temps pour l'être d'humain d'apprendre comment la machine réagissait, ces jeux ont rapidement déclinés ; l'homme s'adaptant en effet plus vite que la machine. Trop pauvres pour être réellement amusants, leur intérêt

---

<sup>1</sup>*Sciences*, 17 octobre 1980

<sup>2</sup>de *Creatures Labs*

<sup>3</sup>de *Lionhead Studios*

résidait surtout dans leur capacités à apprendre de l'humain. Il faut peut-être également envisager le fait que le battage médiatique autour d'eux a fini par convaincre les joueurs qu'ils étaient révolutionnaires. Ils l'étaient sans doute, mais pas dans une façon qui soit perceptible par le grand public.

De la même façon que les avions volent sans battre des ailes, les ordinateurs peuvent gagner sans suivre une logique humaine. Si DeeperBlue se débrouille mieux aux échecs que 99.99% de la population humaine, cela fait de lui un joueur d'échecs, malgré qu'il soit fait de silice et non de carbone. Le piège concisterait donc à mélanger le but à atteindre et la façon d'y parvenir. La question du discernement entre ce qui nécessite réellement de l'intelligence, et ce qui peut se résoudre d'une manière algorithmique est par conséquent cruciale à mes yeux. Ce qui nous renvoie à l'éternelle question : au fond, qu'est-ce que l'intelligence, et une machine peut-elle l'être ? John Searl a clairement statué que le cerveau est le siège de la pensée et qu'il peut-être totalement décrit par des mécanismes chimiques et physiques[11], donnant l'espoir au futur de l'intelligence artificielle forte.

Actuellement, il est difficile d'avoir du recul sur l'efficacité de l'intelligence artificielle dans les jeux sur ordinateurs, car son apparition est encore trop récente. Mais l'identification des problèmes à résoudre ayant été établie par la recherche depuis de nombreuses années, la difficulté réside surtout dans le fait d'employer ces technologies dans des logiciels grands publics.

# Bibliographie

- [1] Daniel Crevier. *À la recherche de l'intelligence artificielle*. Champs/Flammarion, 1999. Origine et histoire de l'intelligence artificielle.
- [2] John H. Conway. L'échec de l'intelligence artificielle. *Sciences et vie*, décembre 2002.
- [3] John H. Conway. *On Numbers and Games*. Academic Press, 1976.
- [4] Levy and Bealjournal. Heuristic programming in artificial intelligence : The first computer olympiad. 1989.
- [5] Gérard Villemin. Intelligence artificielle. 2003. <http://membres.lycos.fr/villemingerard/Logique/IAintro.htm>.
- [6] A.M. Turing, C. Strachey, M.A. Bates, and B.V. Bowden. Digital computers applied to games. In B.V. Bowden, editor, *Faster Than Thought*, pages 286–310. Pitman, London, 1953.
- [7] James P. Carse. *Finite and Infinite Games*. Ballantine Books, 1987. Réflexions philosophiques sur les jeux et la vie en général.
- [8] B. Girard and G. Robert. Intelligence artificielle située et jeux vidéo. In *Actes de Conférence du IVème colloque Jeunes Chercheurs en Science Cognitives (CJC4)*, pages 157–160. ISC Lyon, 2001.
- [9] William van der Sterren. *Terrain Reasoning for 3D Action Games*. Proceedings of the 2001 Game Developer Conference, 2001.
- [10] J.M.P. van Waveren. *The Quake 3 Arena Bot*. PhD thesis, University of technology, Delft, 2001. Thèse portant sur la conception d'un adversaire informatique pour le jeu Quake 3.
- [11] John R. Searle. *Du cerveau au savoir*. Hermann, 1985.