Mémoire de fin d’études

L’intelligence artificielle appliquée aux personnages non-joueurs dans les jeux vidéo

Comment rendre l’environnement d’un jeu intelligent ? Dans quel but ?

Sébastien Gaulier - 5A IJV

Guillaume Ambrois - 5A IJV

Maître de mémoire : M. Alain Lioret



**5A 2015**

[1. Avant-propos 3](#_Toc439855928)

[1.1. Remerciements 3](#_Toc439855929)

[1.2. Résumé 4](#_Toc439855930)

[1.3. Abstract 4](#_Toc439855931)

[1.4. Mots clés 5](#_Toc439855932)

[2. Introduction 6](#_Toc439855933)

[3. Bref historique de l’IA appliquée aux jeux vidéo 7](#_Toc439855934)

[3.1. La naissance de l’IA 7](#_Toc439855935)

[3.2. Les premières IA pour le jeu vidéo 8](#_Toc439855936)

[3.3. Une IA évolutive 9](#_Toc439855937)

[4. Une nouvelle perspective pour l’IA 10](#_Toc439855938)

[5. Les avantages 11](#_Toc439855939)

[5.1. Une interaction avec l’environnement plus réaliste 11](#_Toc439855940)

[5.2. Une autonomie et une adaptabilité accrues 11](#_Toc439855941)

[5.3. Un gameplay plus vivant 11](#_Toc439855942)

[5.4. Une rejouabilité accrue 11](#_Toc439855943)

[6. Les inconvénients 12](#_Toc439855944)

[6.1. Temps de calcul 12](#_Toc439855945)

[6.2. Résultats plus imprévisibles 12](#_Toc439855946)

[6.3. Des résultats “presque” bien (Uncanney Valley) 12](#_Toc439855947)

[7. Les applications possibles prometteuses 13](#_Toc439855948)

[7.1. Le Machine Learning, l’IA qui apprend en même temps que le joueur 13](#_Toc439855949)

[7.1.1. Les réseaux de neurones 13](#_Toc439855950)

[7.1.2. Les arbres de décision avec l’apprentissage par renforcement 16](#_Toc439855951)

[7.2. Dynamic Game Difficulty Balancing, l’IA qui s’adapte au niveau du joueur 25](#_Toc439855952)

[7.3. Des jeux par les IA, pour les IA 27](#_Toc439855953)

[7.3.1. General Game Playing, l’IA qui apprend à jouer 27](#_Toc439855954)

[7.3.2. Angelina, l’IA qui développe des jeux 28](#_Toc439855955)

[8. De l’importance de l’IA pour l’immersion du joueur 29](#_Toc439855956)

[8.1. Suspension consentie de l'incrédulité et solipsisme 30](#_Toc439855957)

[8.2. Entre la sérendipité et l’apophénie 32](#_Toc439855958)

[8.2.1. Fable, un monde vivant 33](#_Toc439855959)

[8.2.2. Evie / Cleverbot 37](#_Toc439855960)

[9. Conclusion 39](#_Toc439855961)

[10. Références 40](#_Toc439855962)

# Avant-propos

## Remerciements

Nous tenons à remercier :

M. Alain LIORET, notre maître de mémoire, pour son soutien, sa passion, et son soutien dans notre travail.

L’équipe pédagogique

## Résumé

Les jeux vidéo sont de plus en plus beaux, que ce soit en approchant le photoréalisme, ou en ayant une patte graphique unique. Ils sont également de plus en plus profonds, avec une histoire, une narration et un univers toujours plus poussés. Le gameplay également n’a de cesse de s’améliorer, que ce soit à travers de nouvelles idées innovantes de game design, ou via de nouveaux périphériques pour renforcer l’interaction ainsi que l’immersion du joueur.

Mais le domaine dans lequel il reste le plus de progrès à faire pour proposer des expériences toujours plus intéressantes et immersives est l’intelligence artificielle (I.A.). Bien que celle-ci soit de plus en plus présente dans les jeux, et de plus en plus évoluée grâce à des algorithmes toujours plus performants, cela reste en général une I.A. possédant comportements prévues à l’avance en réponse d’évènements spécifiques.

En effet, bien souvent les personnages non joueurs (PNJ) sont des entités statiques du jeu ne possédant qu’un panel fixe d’actions ou de répliques de dialogues. Ces personnages sont ainsi associés à des éléments du décor au lieu d’occuper une place dans le scénario du jeu.

Dans ce mémoire, nous étudierons les différentes I.A. appliquées à des PNJs qui existent actuellement, ainsi que ce qu’elles apportent en terme d'expérience.

## Abstract

Video games are more and more beautiful, whether it be by approaching photorealism, or by using a unique graphic style. They’re also more and more deep, with a story, a narration and an universe always more advanced. The gameplay keeps getting better, either with innovative game design ideas, or with new device to enhance the interaction and immersion of the player.

But there is a field in which much progresses are yet to be done in order to offer ever more interesting and immersive experiences, and this field is artificial intelligence (AI). Although it’s more and more anchored in video games, and more and more efficient through algorithms ever more powerful, it remains, in general a scripted AI, with programmed behaviors which respond to specific events.

Indeed, often the non-player characters (NPCs) are static entities of the game with only a fixed panel of actions or dialogues replicas. These characters are well combined with decorative elements instead of a place in the scenario of the game.

In this thesis, we will study the state of the art of artificial intelligence applied on NPCs, and what they offer in term of experience.

## Mots clés

|  |  |
| --- | --- |
| Mots clés | Key words |
| Intelligence Artificielle  Jeu Vidéo  Personnage Non Joueur (P.N.J.)  Apprentissage Automatique | Artificial Intelligence  Video Game  Non Playable Character (N.P.C.)  Machine Learning |

# Introduction

Depuis leur invention, les jeux vidéo n’ont eu de cesse de s’améliorer, que ce soit du point de vue du gameplay, du scénario ou du matériel. Mais un aspect qui est reste encore aujourd’hui en retrait (en plus du fait que lorsqu’elle est bien réalisée, elle est discrète), c’est l’intelligence artificielle.

Un premier problème se situe au niveau de la crédibilité du comportement, empêchant le joueur de s'immerger totalement. C’est une problématique impactant principalement les jeux solo. En effet, le cœur des jeux vidéo se jouant seul est son scénario. Il est donc nécessaire que l’histoire ainsi que ses personnages paraissent le plus crédible possible aux yeux du joueur.

Un deuxième problème se situe au niveau du degré de compétences de l’IA. En effet, celui ci est souvent fixe, défini à l’avance, et impersonnel. Il existe plusieurs solutions pour pallier à ce problème, et si quelques jeux utilisent déjà ce genre d’IA un peu plus évoluée (notamment Left 4 Dead et son AI Director, dont nous parlerons plus en détail plus tard), trop peu sont les jeux à les utiliser.

Le constat est donc simple, pour apporter une meilleure expérience au joueur, il faut améliorer le comportement des agents en le rendant plus réaliste d’une part, et plus personnalisable, adaptable, ou encore évolutif.

Il existe beaucoup de types d’IA applicables pour les jeux vidéo, et beaucoup de façon de les réaliser. Nous présenterons dans ce mémoire les principales méthodes, notamment basées autour du Machine Learning, cependant nous restreindrons le champ d’application aux personnages non joueur.

# Bref historique de l’IA appliquée aux jeux vidéo

Dans les jeux vidéo, l’intelligence artificielle est utilisée pour donner vie aux PNJs. Les méthodes employées pour implémenter les algorithmes de ces PNJs s’appuient généralement sur les théories existantes du domaine plus général de l’intelligence artificielle. Cependant, contrairement à une IA classique comme celle d’un robot, l’objectif n’est pas nécessairement de reproduire le comportement d’un humain. Il est même fréquent de voir des PNJs tricher.

Dans la plupart des jeux, les PNJs ont par exemple connaissance de l’ensemble de la carte et de ce qui s’y déroule. Dans les jeux de type FPS (« First Person Shooter »), les IA sont souvent dotées d’une visée parfaite. Par conséquent, il est très souvent nécessaire de brider ces capacités hors normes pour donner au joueur une sensation d’équité.

L’histoire de l’IA dans ce domaine remonte sans doute au début des années 1940s avec le jeu de stratégie pure **Nim** (publié en 1942). L’IA était alors capable de gagner, même contre des joueurs chevronnés !



*Le jeu de Nim*

## **La naissance de l’IA**

En 1951, les premières IA pour les jeux de Dames et d’Echecs sont créées à partir de la machine Ferranti Mark 1[[1]](#footnote-1). Ces IA ont ensuite été améliorées jusqu’à leur point culminant : la défaite du joueur d’Echecs Garry Kasparov, vaincu par le super-ordinateur Deep Blue d’IBM.

Plus tard entre les années 1960 et le début des années 1970, les premiers jeux développés durant cette période utilisaient la logique discrète et, le plus souvent, n’intégraient pas d’IA car ils opposaient uniquement deux joueurs.



*Garry Kasparov contre Deep Blue d'IBM*

## **Les premières IA pour le jeu vidéo**

Dans les années 1970, les premiers jeux avec des modes 1 joueur apparaissent. Les plus remarquables sont Chasse au Wumpus et Star Trek en 1972 qui utilisaient des Stored Patterns[[2]](#footnote-2) pour implémenter les IA des ennemis. L’utilisation des microprocesseurs a permis une augmentation du nombre de mouvement.

Dans la fin des années 1970 et jusqu’au début des années 1990 : c’est l’âge d’or des jeux d’arcade. Les IA se popularisent. Nombre de jeux (comme Space Invaders en 1978 et Galaxian en 1979) intégraient maintenant une IA plus complexe avec différents niveaux de difficulté et de nombreux modèles de mouvement.

* **Pacman** (1980) a introduit un pattern IA pour les jeux de labyrinthe, avec des personnalités distinctes pour chaque ennemi.
* **Karate Champ** (1984) a quant à lui introduit le pattern pour les jeux de combat.
* **First Queen** (1988) était un RPG dans lequel des personnages contrôlés par l’IA devaient suivre un leader. Ce type d’IA ayant été amélioré par Dragon Quest IV (1990) avec le system « Tactics » (ajustement des routines utilisées par les IA) puis Secret of Mana (1993).

A partir des années 1990, l’émergence de nouveaux genres de jeux vidéo a conduit à l’utilisation d’outils plus sophistiqués tels que les automates finis. Les stratégies en temps réel requéraient la prise en compte de nombreuses contraintes : de nombreux objets, des informations incomplètes, la recherche de chemin, l’économie du nombre de calculs lors de la planification, etc.

Les premiers jeux rencontraient de nombreux problèmes (Herzog Zwei en 1989 avec une recherche de chemin basée sur un automate fini à trois états, ainsi que Dune 2 en 1992 qui utilisaient de nombreuses « astuces »).

## Une IA évolutive

Longtemps délaissée au profit des graphismes des jeux vidéo, cette branche constitue maintenant une part importante du travail dans un jeu vidéo. Les joueurs réclament de plus en plus des IA au comportement humain afin d'accroître le sentiment d’immersion. Certains jeux sont réputés pour être des précurseurs en matière d’IA, en voici quelques exemples :

- **Creatures** (1998) : ce jeu est célèbre pour être le premier à utiliser l’apprentissage automatique lors d’une simulation interactive. A l’aide des réseaux de neurones, les créatures (appelées « Norms ») apprennent divers comportements. Ils peuvent ainsi interagir avec leur environnement.

- **Halo : Combat Evolved** (2001) : ce jeu utilisait des arbres pour déterminer le comportement des PNJs (« Behavior Tree »), avec beaucoup d’attention portée sur le moindre détail du jeu. Ainsi, la gestion des groupes de PNJs était particulièrement bonne et a joué un rôle de précurseur.

- **F.E.A.R.** (2005) : l’IA utilise un planificateur (« Planner ») afin de générer des comportements sensibles au contexte, ce fut la première fois dans un jeu grand public. On ressent une grande habileté chez les PNJs. Ils sont en effet capables de trouver une couverture derrière des tables, basculer des étagères, ouvrir des portes, passer à travers les fenêtres, etc. Ce jeu constitue une référence en la matière.

- **Black & White** (2001) : le jeu propose au joueur d’incarner une divinité et de guider un peuple. Pour l’aider dans cette tâche, le joueur devra éduquer sa créature, un agent qu’il sera possible de récompenser après une action pour renforcer ce comportement, ou punir, afin de le diminuer.

# Une nouvelle perspective pour l’IA

Machine learning avec du Deep Neural Network

IA with procedural content

# Les avantages

## Une interaction avec l’environnement plus réaliste

Ouais toussa toussa

## Une autonomie et une adaptabilité accrues

En ce moment, la mode est au procédural. Or, il est très difficile, sinon impossible de prédire tous les cas de figures et donc de coder les comportements. Il suffit donc d’apprendre à un agent d’apprendre par lui-même pour lui permettre de se débrouiller

Détournement intelligent de “Donne un poisson à un homme, il mangera un jour. Apprend lui à pêcher et il mangera toute sa vie”, mais pour l’IA.

Teach an AI to XXX, it’ll XXX. Teach it to learn and it’ll take over the world.

## Un gameplay plus vivant

C’est quand même mieux qu’un gameplay mort.

C’est tout pour moi, merci.

## Une rejouabilité accrue

Comme avec le procédural, ça augmente la durée de vie et la rejouabilité.

# Les inconvénients

## Temps de calcul

Un des principaux freins de l’IA dans les jeux vidéo, c’est l’impact sur les performances. En effet.

## Résultats plus imprévisibles

L’utilisation de beaucoup d’algorithmes pour faire évoluer puis déterminer le comportement d’un agent à un moment donné fait qu’il est plus difficile de prédire ses actions.

## Des résultats “presque” bien (Uncanney Valley)

Même si c’est les résultats (de l’avancée de l’IA) ne vont qu’en s’améliorant, on arrive, comme dans le domaine de la robotique, dans l’Uncanney Valley.

# Les applications possibles prometteuses

## Le Machine Learning, l’IA qui apprend en même temps que le joueur

### Les réseaux de neurones

Cette section examine plusieurs techniques d'apprentissage machine qui peut être utilisé dans les jeux, et explique pourquoi neuroevolution (NE), à savoir les réseaux de neurones de l'évolution artificielle en utilisant une génétique algorithme, est la méthode idéale pour apprentissage en temps réel dans NERO. Parce que les agents dans Nero ont besoin d'apprendre en ligne comme Le jeu est joué, les objectifs de formation sont généralement prédéterminés ne sont pas disponibles, excluant les techniques supervisées telles que backpropagation (Rumelhart et al., 1986) et de l'arbre de décision apprentissage (Utgoff 1989). Apprentissage (RL) des techniques de renforcement traditionnelles telles comme Q-Learning (Watkins et Dayan 1992) et Sarsa (?) avec une fonction d'approximation de partir de cas (SARSA-CABA; Santamaria et al. 1998) peut s'adapter dans des domaines avec clairsemée rétroaction (. Kaelbling et al 1996; Sutton et Barto, 1998) et peut donc être appliquée aux jeux vidéo ainsi. Ces techniques apprennent à prédire la récompense à long terme pour prendre actions dans les différents Etats, en explorant l'espace de l'Etat et garder la trace des résultats. Cependant, les jeux vidéo ont plusieurs propriétés qui posent des défis importants à traditionnel RL:

1. Grand espace / d'action de l'Etat. Depuis les jeux ont généralement plusieurs différents types d'objets et personnages, et différentes actions possibles, l'état / espace d'action RL que doit explorer est de grande dimension. Pas seulement cela pose le problème habituel de codage d'un highdimensional espace (Sutton et Barto 1998), mais en un jeu en temps réel, il ya le défi supplémentaire de vérifier la valeur de toutes les actions possibles sur chaque jeu tick pour chaque agent dans le jeu.
2. comportements divers. Agents d'apprentissage simultanément ne devraient pas tous converger vers le même comportement parce une population homogène ferait le jeu ennuyeuse. Pourtant, depuis les techniques de RL sont basés sur la convergence garanties et ne maintiennent pas explicitement la diversité, un tel résultat est probable.
3. Conformément comportements individuels. RL dépend de prenant parfois une action aléatoire afin d'explorer de nouveaux comportements. Bien que cette stratégie fonctionne bien dans l'apprentissage hors ligne, les joueurs ne veulent pas voir un individu l'agent périodiquement faire inexplicable et idiosyncrasiques se déplace par rapport à son comportement habituel.
4. Adaptation rapide. Les joueurs ne veulent pas attendre des heures pour les agents d'adaptation. Pourtant, un état / représentation action complexe peut prendre un certain temps pour apprendre. De l'autre part, une représentation simple serait de limiter la capacité d'apprendre des comportements sophistiqués. Ainsi, en choisissant la un droit de représentation est difficile.
5. Mémoire des états passés. Si les agents se souviennent passé événements, ils peuvent réagir de façon plus convaincante à la Situation présente. Cependant, une telle mémoire exige garder la trace de plus de l'état actuel, au pouvoir méthodes traditionnelles de Markov sur.

Bien que ces propriétés font de l'application RL traditionnelle techniques difficiles, NE est une technique alternative qui RL peut répondre à chaque exigence: (1) NE fonctionne bien dans highdimensional espaces d'états (Gomez et Miikkulainen 2003), et ne produit qu'une seule action demandée sans vérifier les valeurs de plusieurs actions. (2) les populations diverses peut être explicitement maintenu (Stanley et Miikkulainen 2002b). (3) Le comportement d'un individu au cours de son durée de vie ne change pas. (4) Une représentation de la solution peut être évolué, permettant comportements simples à découvrir rapidement au début et plus tard complexifié (Stanley et Miikkulainen 2004). (5) les réseaux de neurones récurrents peut être évolué qui utilisent la mémoire (Gomez et Miikkulainen 1999). Ainsi, NE est un bon match pour les jeux vidéo. Le défi actuel est de parvenir à l'évolution en temps réel, que le jeu est joué. Si les agents pourraient être évolué dans un lisse cycle de remplacement, le joueur pourrait interagir avec l'évolution au cours du jeu et les nombreux avantages de NE serait disponible pour la communauté du jeu vidéo. Ce article introduit une telle technique de NE-temps réel, rtNEAT, qui est appliqué à l'état continu multi-agent NERO jeu vidéo. Dans Nero, les agents doivent maîtriser à la fois le contrôle moteur et la stratégie de niveau supérieur pour gagner le match. Le joueur agit comme un entraîneur, l'enseignement d'une équipe de robots les compétences dont ils ont besoin pour survivre. La section suivante examine la neuroevolution NEAT méthode, et comment il peut être amélioré pour produire rtNEAT.

La méthode est basée sur rtNEAT NEAT, une technique pour réseaux neuronaux évolution pour l'apprentissage par renforcement complexe tâches en utilisant un algorithme génétique (GA). Moissonneuses-batteuses NLFA la recherche d'habitude pour les poids de réseau appropriés avec complexification de la structure du réseau, ce qui permet le comportement des réseaux de neurones évolué pour devenir de plus en plus sophistiquée au fil des générations. Cette approche est très efficace: NEAT surpasse les autres neuroevolution (NE) par exemple les méthodes sur la tâche de référence d'équilibrage bipolaire (Stanley et Miikkulainen 2002a, b). En outre, parce que NLFA commence avec les réseaux simples et étend la recherche espace uniquement lorsque bénéfique, il est capable de trouver de manière significative contrôleurs plus complexes que l'évolution topologie fixe, que démontré dans un domaine de stratégie en apprentissage robotique (Stanley et Miikkulainen 2004). Ces propriétés en font un NEAT Procédé attractive pour faire évoluer les réseaux neuronaux dans complexe des tâches telles que les jeux vidéo. Comme la plupart des gaz, NEAT a été initialement conçu pour fonctionner déconnecté. Les individus sont évaluées une ou deux à la fois, et après toute la population a été évaluée, une nouvelle populationest conçu pour former la génération suivante. Autrement dit, dans un GA normale, il est impossible pour un être humain d'interagir avec les multiples agents évolution alors qu'ils évoluent. Cette section examine d'abord brièvement l'origine déconnecté NEAT méthode, et ensuite décrit comment il peut être modifié pour le rendre il possible pour les joueurs d'interagir avec des agents évoluant dans réel temps. Voir, par exemple Stanley et Miikkulainen (2002a, b, 2004) pour une description complète des NLFA. NEAT est basé sur trois idées clés. Tout d'abord, réseau évolutif structure nécessite un codage génétique flexible. Chaque génome comprend une liste de gènes de connexion, dont chacun se réfère à deux gènes de noeuds étant connectés. Chaque connexion spécifie le gène en noeud, l'out-noeud, la connexion poids, si oui ou non le gène de connexion est exprimé (le bit de validation), et un numéro d'innovation, ce qui permet de trouver gènes correspondant au cours de croisement. Mutation peut changer les poids de connexion et les structures de réseau. Connexion poids mutent comme dans tout système de NE, avec chaque connexion soit perturbé ou non. Mutations structurelles, qui permet d'augmenter la complexité, soit ajouter une nouvelle connexion ou un nouveau noeud au réseau. Par mutation, les génomes de différentes tailles sont créés, avec parfois complètement différent connexions spécifiées dans les mêmes positions. Chaque gène unique dans la population se voit attribuer un unique, numéro de l'innovation, et les chiffres sont héritées cours croisé. Numéros de l'innovation permettent d'effectuer NEAT croisement sans le besoin d'une analyse topologique cher. Génomes de différentes organisations et de tailles restent compatibles tout au long de l'évolution, et le problème de l'adaptation différentes topologies (Radcliffe 1993) est essentiellement évités. Deuxièmement, NEAT speciates la population, afin que les individus concurrence principalement au sein de leurs propres niches lieu de avec la population en général. De cette façon, les innovations topologiques sont protégés et avoir du temps pour optimiser leur structure avant la compétition avec d'autres niches dans la population. le mécanisme de reproduction pour NEAT est le partage de remise en forme explicite (Goldberg et Richardson, 1987), où les organismes présents dans les mêmes espèces doivent partager la condition physique de son créneau, la prévention une espèce est de prendre le contrôle de la population. Troisièmement, contrairement à d'autres systèmes qui évoluent topologies de réseau et le poids (Gruau et al 1996;. Yao 999) commence NEAT avec une population uniforme des réseaux simples sans noeuds cachés. Nouvelle structure est introduite progressivement à mesure que mutations structurelles se produisent, et seules les structures survivent qui sont jugées utiles par des évaluations de fitness. Ce Ainsi, par le biais de recherches NLFA un nombre minimal de poids dimensions et trouve le niveau de complexité approprié pour le problème. Dans des travaux antérieurs, chacune des trois composantes principales NEAT (c. marques historiques, spéciation et départ de structure minimale) a été soumise à une ablation dans expérimentalement afin de démontrer comment ils contribuent à la performance (Stanley et Miikkulainen 2002b). L'étude d'ablation ont démontré que les trois composants sont interdépendants et nécessaire pour rendre le travail de NEAT. La section suivante explique comment NEAT peut être amélioré pour travailler en temps réel.

### Les arbres de décision avec l’apprentissage par renforcement

Parmi les premiers algorithmes d'apprentissage par renforcement, on compte le TD-learning, proposé par Richard Sutton en 1988, et le Q-learning mis au point essentiellement lors d'une thèse soutenue par Chris Watkins en 1989 et publié réellement en 1992.

Toutefois, l'origine de l'apprentissage par renforcement est plus ancienne. Elle dérive de formalisations théoriques de méthodes de contrôle optimal, visant à mettre au point un contrôleur permettant de minimiser au cours du temps une mesure donnée du comportement d'un système dynamique. La version discrète et stochastique de ce problème est appelée un processus de décision markovien et fut introduite par Bellman en 1957.

D'autre part, la formalisation des problèmes d'apprentissage par renforcement s'est aussi beaucoup inspirée de théories de psychologie animale, comme celles analysant comment un animal peut apprendre par essais-erreurs à s'adapter à son environnement. Ces théories ont beaucoup inspiré le champ scientifique de l'intelligence artificielle et ont beaucoup contribué à l'émergence d'algorithmes d'apprentissage par renforcement au début des années 1980.

En retour, le raffinement actuel des algorithmes d'apprentissage par renforcement inspire les travaux des neurobiologistes et des psychologues pour la compréhension du fonctionnement du cerveau et du comportement animal. En effet, la collaboration entre neurobiologistes et chercheurs en intelligence artificielle a permis de découvrir qu'une partie du cerveau fonctionnait de façon très similaire aux algorithmes d'apprentissage par renforcement tels que le TD-learning. Il semblerait ainsi que la nature ait découvert, au fil de l'évolution, une façon semblable à celles trouvées par des chercheurs pour optimiser la façon dont un agent ou organisme peut apprendre par essais-erreurs. Ou plutôt, les chercheurs en intelligence artificielle ont redécouvert en partie ce que la nature avait mis des millions d'années à mettre en place. En effet, la zone du cerveau qui montre des analogies avec les algorithmes d'apprentissage par renforcement s'appelle les ganglions de la base, dont une sous-partie appelée la substance noire émet unneuromodulateur, la dopamine, qui renforce chimiquement les connexions synaptiques entre les neurones. Ce fonctionnement des ganglions de la base a été identifié comme existant chez l'ensemble des vertébrés, et on retrouve le même genre de résultats enimagerie médicale chez l'homme.

Enfin, la boucle d'échange scientifique entre neurobiologistes, psychologues et chercheurs en intelligence artificielle n'est pas terminée puisque actuellement, des chercheurs prennent inspiration du cerveau pour raffiner les algorithmes d'apprentissage par renforcement et essayer ainsi de mettre au point des [robots](https://fr.wikipedia.org/wiki/Robot) plus autonomes et adaptatifs que ceux existants. En effet, même si la nature et les chercheurs semblent avoir trouvé séparément une même solution pour résoudre certains types de problèmes tels que ceux décrits au paragraphe précédent, on se rend bien compte que l'intelligence des robots actuels est encore bien loin de celle de l'homme ou même de celle de nombreux animaux tels que les singes ou les rongeurs. Une voie prometteuse pour pallier cela est d'analyser plus en détail comment le cerveau biologique paramétrise et structure anatomiquement des processus tels que l'apprentissage par renforcement, et comment il intègre ces processus avec d'autres fonctions cognitives telles que la perception, l'orientation spatiale, la planification, la mémoire, et d'autres afin de reproduire cette intégration dans le cerveau artificiel d'un robot.

Une bonne façon de comprendre l'apprentissage de renforcement est d'examiner certains des exemples et des applications possibles qui ont guidé son développement.

* Un joueur maître d'échecs fait un mouvement. Le choix est informé à la fois par la planification - anticiper les réponses et counterreplies possibles - et par immédiats, jugements intuitifs de l'opportunité des positions et des mouvements particuliers.
* Dispositif de commande adaptatif ajuste les paramètres de fonctionnement de une raffinerie de pétrole en temps réel. Le contrôleur optimise le rendement / coût / qualité compromis sur la base des coûts marginaux spécifiées sans coller strictement aux consignes initialement proposées par les ingénieurs.
* Un veau gazelle se bat pour ses minutes de pieds après être né. Une demi-heure plus tard, il tourne à 20 miles par heure.
* Un robot mobile décide si elle doit entrer dans une nouvelle salle à la recherche de plus trash de collecter ou de commencer à essayer de trouver son chemin de retour à sa station de recharge des batteries. Il fonde sa décision sur la façon de rapidement et facilement, il a été en mesure de trouver le chargeur dans le passé.
* Phil prépare son petit déjeuner. Examiné de près, même cette activité apparemment banale révèle un réseau complexe de comportement conditionnel et de verrouillage relations but subgoal: la marche vers le placard, l'ouvrir, la sélection d'une boîte de céréales, puis pour atteindre, saisir, et la récupération de la boîte. Autres, à l'écoute, les séquences interactives complexes de comportement sont nécessaires pour obtenir un bol, cuillère, et pot à lait. Chaque étape implique une série de mouvements oculaires pour obtenir des informations et pour guider les atteindre et de la locomotion. Jugements rapides sont constamment faites sur la façon de porter les objets ou si il est préférable de transporter certains d'entre eux à la table de salle à manger avant d'obtenir d'autres. Chaque étape est guidé par des objectifs, comme saisir une cuillère ou d'entrer dans le réfrigérateur, et est en service d'autres objectifs, comme ayant la cuillère pour manger avec une fois la céréale est préparé et obtenir finalement la nourriture.

Ces exemples ont des caractéristiques qui sont si fondamentales qu'elles sont faciles à négliger. Toutes impliquent une interaction entre un agent de prise de décision active et son environnement, dans lequel l'agent cherche à atteindre un but malgré l'incertitude quant à son environnement. Les actions de l'agent sont autorisés à affecter l'état futur de l'environnement (par exemple, la position suivante d'échecs, le niveau des réservoirs de la raffinerie, le prochain emplacement du robot), affectant ainsi les options et les possibilités offertes à l'agent à des moments plus tard . Choix correct nécessite de prendre en compte, les conséquences indirectes d'actions différées, et donc peut exiger la prévision ou de planification.

Dans le même temps, dans tous ces exemples, les effets des actions ne peuvent être pleinement prédit; ainsi l'agent doit surveiller son environnement fréquemment et réagir de manière appropriée. Par exemple, Phil doit regarder le lait qu'il verse dans son bol de céréales pour l'empêcher de déborder. Tous ces exemples concernent des objectifs qui sont explicites en ce sens que l'agent peut juger des progrès vers son objectif basé sur ce qu'il peut détecter directement. Le joueur d'échecs sait si oui ou non il gagne, le contrôleur de la raffinerie sait combien le pétrole est produit, le robot mobile sait quand ses batteries sont déchargées, et Phil sait si oui ou non il connaît son petit déjeuner.

Dans tous ces exemples, l'agent peut utiliser son expérience pour améliorer ses performances au fil du temps. Le joueur d'échecs affine l'intuition qu'il utilise pour évaluer les postes, ce qui améliore son jeu; le veau gazelle améliore l'efficacité avec laquelle il peut fonctionner; Phil apprend à simplifier faisant son petit déjeuner. La connaissance de l'agent apporte à la tâche au début - soit à partir de l'expérience précédente avec des tâches ou intégrés par la conception ou l'évolution connexes - influences ce qui est utile ou facile à apprendre, mais l'interaction avec l'environnement est essentielle pour régler le comportement de exploiter les caractéristiques spécifiques de la tâche.

Apprentissage par renforcement est d'apprendre ce qu'il faut faire - comment mapper situations à des actions - afin de maximiser un signal de récompense numérique. L'apprenant est pas dit quelles actions à prendre, comme dans la plupart des formes d'apprentissage de la machine, mais doit découvrir quelles actions donnent le plus de récompense en les essayant. Dans les cas les plus intéressantes et stimulantes, les actions peuvent affecter non seulement la récompense immédiate, mais aussi la situation suivante et, à travers cela, toutes les récompenses suivantes. Ces deux caractéristiques - recherche par essais et erreur et récompense différée - sont les deux traits distinctifs les plus importants de l'apprentissage par renforcement.

Apprentissage par renforcement ne se définit pas par la caractérisation des méthodes d'apprentissage, mais en caractérisant un problème d'apprentissage. Toute méthode qui est bien adapté à la résolution de ce problème, que nous considérons comme une méthode d'apprentissage par renforcement. Une spécification complète du problème de l'apprentissage par renforcement en termes de contrôle optimal des processus de décision de Markov doit attendre jusqu'à ce que le chapitre 3, mais l'idée de base est tout simplement de saisir les aspects les plus importants de la vrai problème face à un agent de l'apprentissage en interaction avec son environnement pour atteindre un objectif. De toute évidence, un tel agent doit être capable de détecter l'état de l'environnement dans une certaine mesure et doit être en mesure de prendre des mesures qui affectent l'état. L'agent doit également avoir un objectif ou des objectifs relatifs à l'état de l'environnement. La formulation est destinée à comprendre que ces trois aspects - la sensation, l'action, et l'objectif - dans leurs formes les plus simples possibles sans banaliser l'un d'eux.

Apprentissage par renforcement est différente de l'apprentissage supervisé, le type d'apprentissage a étudié dans la plupart des recherches en cours dans l'apprentissage automatique, la reconnaissance des formes statistiques, et les réseaux de neurones artificiels. L'apprentissage supervisé est d'apprendre à partir d'exemples fournis par un superviseur externe compétent. Ceci est un type important de l'apprentissage, mais seule, elle ne suffit pas pour l'apprentissage de l'interaction. Dans problèmes interactifs, il est souvent impossible d'obtenir des exemples de comportement souhaité à la fois pertinente et représentative de toutes les situations dans lesquelles l'agent a à agir. En territoire inconnu - où l'on pourrait attendre à apprendre à être plus bénéfique - un agent doit être capable d'apprendre de sa propre expérience.

Un des défis qui se posent dans l'apprentissage par renforcement et pas dans d'autres types d'apprentissage est le compromis entre exploration et exploitation. Pour obtenir un lot de récompense, un agent de l'apprentissage par renforcement doit préférer les actions qu'il a essayé dans le passé et se sont révélées efficaces dans la production de récompense. Mais pour découvrir de telles actions, il doit essayer actions qu'il n'a pas sélectionnés avant. L'agent a d'exploiter ce qu'il connaît déjà pour obtenir la récompense, mais il a aussi à explorer afin de prendre de meilleures sélections d'action à l'avenir. Le dilemme est que ni exploration ni d'exploitation peuvent être poursuivis exclusivement sans faillir à la tâche. L'agent doit essayer une variété d'actions et de favoriser progressivement ceux qui semblent être meilleur. Sur une tâche stochastique, chaque action doit être essayé à plusieurs reprises d'obtenir une estimation fiable de sa récompense attendue. Le dilemme exploration-exploitation a été intensivement étudiée par les mathématiciens depuis des décennies (voir chapitre 2). Pour l'instant, nous notons simplement que toute la question de prospection et d'exploitation d'équilibrage ne se pose même pas dans l'apprentissage supervisé comme il est généralement défini.

Un autre élément clé de l'apprentissage par renforcement est qu'elle considère explicitement tout le problème d'un agent de goal-directed interagir avec un environnement incertain. Ceci est en contraste avec de nombreuses approches qui tiennent compte de sous-problèmes sans aborder la façon dont ils pourraient tenir dans une image plus grande. Par exemple, nous avons mentionné que beaucoup de recherche sur l'apprentissage de la machine est préoccupé par apprentissage supervisé sans préciser explicitement comment une telle capacité serait enfin utile. D'autres chercheurs ont développé des théories de la planification avec des objectifs généraux, mais sans prendre en considération le rôle de la planification en temps réel la prise de décision, ou de la question de savoir où les modèles prédictifs nécessaires à la planification viendraient. Bien que ces approches ont donné de nombreux résultats utiles, leur attention sur sous-problèmes isolés est une limitation importante.

Apprentissage par renforcement prend le chemin inverse, en commençant par une interactive, l'agent complète, but recherche. Tous les agents de renfort apprentissage ont des objectifs explicites, peut détecter les aspects de leur environnement, et peuvent choisir des actions pour influencer leurs environnements. En outre, il est généralement supposé dès le début que l'agent doit fonctionner malgré l'incertitude significative sur l'environnement, il doit faire face. Lorsque l'apprentissage par renforcement implique une planification, il doit faire face à l'interaction entre la planification et la sélection de l'action en temps réel, ainsi que la question de savoir comment les modèles de l'environnement sont acquis et amélioré. Lorsque l'apprentissage par renforcement implique l'apprentissage supervisé, il le fait pour des raisons spécifiques qui déterminent les capacités les plus importantes et qui ne sont pas. Pour apprendre la recherche de faire des progrès, sous-problèmes importants doivent être isolés et étudiés, mais ils devraient être sous-problèmes qui jouent des rôles clairs, des agents de but-recherche complets et interactifs, même si tous les détails de l'agent complète ne peuvent pas encore être remplis.

Une des tendances les plus vastes dont l'apprentissage par renforcement est une partie est que vers une plus grande contact entre l'intelligence artificielle et d'autres disciplines de l'ingénierie. Tout ce qui il ya pas longtemps, l'intelligence artificielle a été considérée comme presque entièrement séparée de la théorie du contrôle et de statistiques. Il avait à voir avec la logique et les symboles, pas des nombres. L'intelligence artificielle était de vastes programmes de Lisp, pas algèbre linéaire, équations différentielles, ou des statistiques. Au cours des dernières décennies, cette vue a progressivement érodé. Les chercheurs de l'intelligence artificielle modernes acceptent algorithmes statistiques et de contrôle, par exemple, que les méthodes concurrentes pertinentes ou simplement comme outils de leur métier. Les zones précédemment ignorées situées entre l'intelligence artificielle et de l'ingénierie classique sont maintenant parmi les plus actifs, y compris les nouveaux domaines tels que les réseaux de neurones, contrôle intelligent, et de notre sujet, l'apprentissage par renforcement. Dans l'apprentissage par renforcement, nous étendons les idées de la théorie du contrôle optimal et approximation stochastique pour répondre aux objectifs plus larges et plus ambitieux de l'intelligence artificielle.

Au-delà de l'agent et de l'environnement, on peut identifier quatre principaux sous-éléments d'un système d'apprentissage de renforcement: une politique, une fonction de récompense, une fonction de valeur, et, éventuellement, un modèle de l'environnement.

Une politique définit la façon de se comporter à un moment donné de l'agent de l'apprentissage. Grosso modo, une politique est une cartographie des états perçus de l'environnement pour les mesures à prendre lors de ces États. Il correspond à ce que la psychologie serait appelé un ensemble de règles ou d'associations stimulus-réponse. Dans certains cas, la politique peut être une simple table de fonction ou de recherche, alors que dans d'autres, il peut impliquer beaucoup de calculs comme un processus de recherche. La politique est au cœur d'un agent de l'apprentissage par renforcement dans le sens que lui seul est suffisant pour déterminer le comportement. En général, les politiques peuvent être stochastique.

Une fonction de récompense définit l'objectif dans un problème de l'apprentissage par renforcement. Grosso modo, il fait correspondre à chaque état perçu (ou état-action paire) de l'environnement à un seul numéro, une récompense, indiquant l'opportunité intrinsèque de cet état. Un seul objectif de l'apprentissage par renforcement agent est de maximiser la récompense totale qu'il reçoit dans le long terme. La fonction de récompense définit quels sont les bons et mauvais événements pour l'agent. Dans un système biologique, il ne serait pas approprié d'identifier les récompenses avec plaisir et la douleur. Ils sont les caractéristiques immédiates et définissant du problème rencontré par l'agent. En tant que tel, la fonction de récompense doit nécessairement être inaltérable par l'agent. Il peut cependant servir de base pour modifier la politique. Par exemple, si une action choisie par la politique est suivie par une faible récompense, alors la politique peut être changée pour sélectionner une autre action dans cette situation à l'avenir. En général, les fonctions de récompense peuvent être stochastique.

Tandis qu'une fonction de récompense indique ce qui est bon dans un sens immédiat, une fonction de valeur spécifie ce qui est bon dans le long terme. Grosso modo, la valeur d'un Etat est le montant total de la récompense d'un agent peut attendre d'accumuler sur l'avenir, à partir de cet état. Alors que les récompenses déterminer la désirabilité intrinsèque immédiate d'états environnementaux, les valeurs indiquent l'opportunité à long terme des Etats après prise en compte des États qui sont susceptibles de suivre, et les récompenses disponibles dans ces États. Par exemple, un Etat pourrait toujours obtenir une faible récompense immédiate, mais ont encore une grande valeur car il est régulièrement suivi par les autres Etats qui produisent de fortes récompenses. Ou l'inverse pourrait être vrai. Pour faire une analogie humaine, les récompenses sont comme le plaisir (si haute) et la douleur (si bas), tandis que les valeurs correspondent à un jugement plus raffiné et clairvoyante de la façon dont le plaisir ou déplaisir nous sommes que notre environnement est dans un état particulier. Exprimé de cette façon, nous espérons qu'il est clair que les fonctions de valeur formaliser une idée de base et familier.

Les récompenses sont dans un premier sens, tandis que les valeurs, comme des prédictions de récompenses, sont secondaires. Sans récompenses il pourrait y avoir aucune valeur, et le seul but de l'estimation des valeurs est de parvenir à plus de récompense. Néanmoins, ce sont les valeurs qui nous sont les plus concernés lors de la prise des décisions et l'évaluation. Choix d'action sont fondés sur les jugements de valeur. Nous cherchons des actions qui provoquent des états de plus grande valeur, pas plus haute récompense, parce que ces actions obtenir la plus grande quantité de récompense pour nous, sur le long terme. Dans la prise de décision et la planification, la quantité dérivée appelée valeur est celle avec laquelle nous sommes plus préoccupés. Malheureusement, il est beaucoup plus difficile de déterminer les valeurs qu'il est de déterminer récompenses. Les récompenses sont essentiellement donnés directement par l'environnement, mais les valeurs doivent être estimées et réestimation des séquences des observations d'un agent fait sur toute sa durée de vie. En fait, la composante la plus importante d'algorithmes d'apprentissage la quasi-totalité de renforcement est un procédé d'estimation de valeurs efficacement. Le rôle central de l'estimation de la valeur est sans doute la chose la plus importante que nous avons appris à propos de l'apprentissage par renforcement au cours des dernières décennies.

Bien que toutes les méthodes d'apprentissage de renforcement que nous considérons dans ce livre sont structurés autour de l'estimation des fonctions de valeur, il est pas strictement nécessaire de le faire pour résoudre les problèmes d'apprentissage de renfort. Par exemple, les méthodes de recherche tels que les algorithmes génétiques, programmation génétique, recuit simulé, et d'autres méthodes d'optimisation de la fonction ont été utilisées pour résoudre les problèmes d'apprentissage de renfort. Rechercher ces méthodes directement dans l'espace des politiques sans jamais faire appel à des fonctions de valeur. Nous appelons ces méthodes évolutives parce que leur fonctionnement est analogue à la façon dont l'évolution biologique produit organismes ayant des comportements qualifiés, même quand ils ne apprennent durant leur vie individuels. Si l'espace des politiques est suffisamment faible, ou peut être structuré de telle sorte que les bonnes politiques sont communes ou facile à trouver, puis méthodes évolutives peuvent être efficaces. En outre, les méthodes évolutives ont des avantages sur les problèmes dans lesquels l'agent d'apprentissage ne peut pas détecter avec précision l'état de son environnement.

Néanmoins, ce que nous entendons par apprentissage par renforcement implique l'apprentissage tout en interagissant avec l'environnement, les méthodes évolutives ne font pas. Il est de notre conviction que les méthodes capables de profiter des détails des interactions comportementales individuelles peuvent être beaucoup plus efficace que les méthodes évolutives dans de nombreux cas. Méthodes évolutionnistes ignorent beaucoup de la structure d'utilité de l'apprentissage par renforcement problème: ils ne utilisent pas le fait que la politique qu'ils recherchent est une fonction des états à des actions; ils ne remarquent pas ce qui indique un individu traverse pendant sa durée de vie, ou quelles actions il sélectionne. Dans certains cas, cette information peut prêter à confusion (par exemple lorsque les États sont mal perçus), mais le plus souvent il devrait permettre une recherche plus efficace. Bien que l'évolution et l'apprentissage partagent de nombreuses caractéristiques et peuvent naturellement travailler ensemble, comme ils le font dans la nature, nous ne considérons pas les méthodes évolutives par eux-mêmes à être particulièrement bien conçu pour armatures des problèmes d'apprentissage. Par souci de simplicité, dans ce livre, lorsque nous utilisons le terme «apprentissage par renforcement" nous ne comprennent pas les méthodes évolutives.

Le quatrième et dernier élément de certains systèmes d'apprentissage par renforcement est un modèle de l'environnement. Ceci est quelque chose qui imite le comportement de l'environnement. Par exemple, compte tenu de l'état et de l'action, le modèle pourrait prédire l'état suivant résultante et la prochaine récompense. Les modèles sont utilisés pour la planification, nous entendons par là un moyen de décider d'un plan d'action en tenant compte des situations futures possibles avant qu'ils soient réellement connu. L'incorporation de modèles et de la planification dans les systèmes d'apprentissage par renforcement est un phénomène relativement nouveau. Les systèmes d'apprentissage de renforcement précoce ont été explicitement apprenants d'essais et d'erreurs; ce qu'ils ont fait était considéré comme presque à l'opposé de la planification. Néanmoins, il est devenu peu à peu évident que les méthodes d'apprentissage de renfort sont étroitement liés aux méthodes de programmation dynamiques, qui font utiliser des modèles, et qu'à leur tour ils sont étroitement liés aux méthodes de planification espace-état. Dans le chapitre 9, nous explorons les systèmes d'apprentissage par renforcement qui apprennent simultanément par essais et erreurs, apprennent un modèle de l'environnement, et utilisent le modèle de planification. Moderne apprentissage par renforcement étend le spectre de bas niveau, d'essais et d'erreurs d'apprentissage à haut niveau, la planification délibérative.

## Dynamic Game Difficulty Balancing, l’IA qui s’adapte au niveau du joueur

Beaucoup d’exemples assez simple (on augmente automatiquement le niveau/nombre des ennemis, etc si on trouve que le joueur s’en sort bien, et inversement)

Les fameux items de Mario Kart (plus t’es loin, meilleurs sont tes objets)

Le meilleur exemple pour ce type d’IA se trouve du côté de Valve, et plus précisément dans leur FPS de survival horror, Left 4 Dead. Dans celui-ci, **là on explique brièvement le jeu, on sait jamais**. Et, contrairement à beaucoup de jeux dans lesquels la difficulté augmente de façon constante, ici un “AI Director” analyse constamment les performances individuelles et collectives des joueurs pour construire une expérience adaptée à tous.

The Director Directs, de Tim Buckley

## Des jeux par les IA, pour les IA

Bientôt, il n’y aura plus du tout besoin d’humains dans la boucle de création et consommation de jeu.

En effet, la prochaine IA dont nous allons parler est une IA capable d’apprendre à joueur à des jeux, tandis que la suivante, très jeune, est capable de développer des jeux (simples) complets.

<<< IMAGE >>>

Légende de l’image

### General Game Playing, l’IA qui apprend à jouer

C’est un domaine qui a gagné en popularité ces dernières années, notamment grâce à une IA qui a gagné le prix du public de la compétition annuelle organisée par l’AAAI (Association for the Advancement of Articifial Intelligence).

Cette IA est capable d’apprendre à jouer à un niveau de Mario Bros (un jeu de plateforme 2D) et de s’adapter aux différents niveaux qui le composent.

### Angelina, l’IA qui développe des jeux

Angelina est une IA qui, à partir d’une description en langage naturel, peut générer des jeux de plateforme 2D, avec une ambiance, des graphismes et un gameplay qui respectent (dans une certaine mesure) la description fournie.

C’est un sujet de recherche développé par Michael Cook, un chercheur britannique à l’université de Falmouth.

Exemple de jeu produit par Angelina

Pour arriver à ce résultat, elle analyse la description, **<<<DESCRIPTION>>>**

# De l’importance de l’IA pour l’immersion du joueur

Dans cette partie, nous parlerons de l’intelligence artificielle qui n’est pas flagrante, mais qui rend, inconsciemment pour le joueur, l'expérience de jeu plus réaliste. Cette intelligence artificielle est principalement importante dans les jeux solos, dans lesquels l’immersion est bien plus importante que dans les jeux multi-joueurs, dans lesquels il est dur de maintenir le joueur dans un univers fictif à cause de la présence d’autres joueurs dont le comportement est difficilement contrôlable et peut être perturbateurs.

Why RPG NPCs Always Repeat Themselves, de Dorkly

## Suspension consentie de l'incrédulité et solipsisme

Lorsqu’un PNJ répète sans cesse le même dialogue au joueur, celui ci est ramené à la réalité. Lorsqu’un PNJ rencontre un problème de pathfinding et n’arrive pas à se déplacer d’un point à un autre, le joueur est ramené à la réalité. 

Toutes ces actions cassent le 4ème mur, sans le vouloir, et mettent fin à ce que l’on appelle **Willing Suspension of Disbelief** (Suspension Consentie de l’Incrédulité en français). C’est comme un contrat que le joueur signe avec l’oeuvre de fiction (jeu, film ou livre), dans lequel le joueur accepte de mettre de coté son scepticisme le temps de sa consultation . Il peut ainsi s’immerger dans un univers avec des éléments imaginaires, pour peu que ceux-ci soient crédibles, et que les règles qui régissent l’oeuvre restent constantes.

Deadpool, personnage de l’univers Marvel

Lorsque ce contrat est rompu (par définition, il ne peut être rompu que par l’oeuvre de fiction), il est difficile de faire replonger un joueur dans l’univers du jeu, avec lequel il restera toujours un peu distant, et cela renforce le solipsisme du joueur.

<<<SOLIPSIME>>>

Dans ce domaine, uns des pires expériences qui puisse arriver au joueur, c’est rencontrer **l’Uncanny Valley** (Vallée Dérangeante en français).

<<<Uncanny Valley>>>

Ainsi, certains MMORPG (Jeu de Rôle Massivement Multijoueur) récents

Why RPG NPCs Always Repeat Themselves, de Dorkly



## Entre la sérendipité et l’apophénie

De par son côté de plus en plus évolutif et complexe, il est parfois difficile de prévoir tous les résultats possibles d’une interaction du joueur avec l’IA, ce qui donne parfois lieu à une sorte de gameplay émergent, avec des résultats imprévisibles même pour les créateurs.

Si dans certains jeux, ce n’est qu’une partie d’un tout plus vaste (Fable), d’autres reposent essentiellement sur ce principe (Evie, Cleverbot).

### Fable, un monde vivant

<<<Présentation rapide de Fable>>>

Voici les règles que les développeurs jugeaient vitales que le jeu suive, sans quoi il ne respecterait pas leur vision :

* Le personnage principal doit exprimer visuellement ce qu’il ressent.
* Le monde dans lequel le joueur évolue, ainsi que tous ses occupants doivent réagir de manière appropriée aux actions du joueur.

**Un héros expressif**

Plutôt que d’afficher plein de caractéristiques comme dans un RPG classique, il est plus intéressant de montrer par des mouvements et des expressions que le personnage réagit au monde qui l’entoure (plutôt que de rester planter là, sans émotions apparentes).

**Un monde simulé**

[ESSAYER DE FAIRE QUE TOUT SOIT INTERESSANT, PAS SCRIPTE]

Ils ont également remarqué que les joueurs étaient particulièrement sensibles aux émotions des enfants. Ils ont donc décidé de rendre le comportement des enfants très varié. Ainsi, certains sont de véritables petites pestes (comme certains le sont dans la réalité), et poursuivent et [FONT CHIER] les poules, tandis que d’autres vont à l’école bien en rangs, et rentrent gentiment à la maison à l’heure pour le goûter (comme certains le font dans la réalité).

Et, peut être le plus important, les enfants fixeront avec intensité ce qu’ils trouvent intriguant.

**Un héros expressif dans un monde simulé**

Quand on mélange les deux, voici ce que cela peut donner :

Imaginez vous la scène, une radieuse journée à Bowerstone (une ville du jeu). Les papillons papillonnent, le parfum des tulipes vous est porté par une brise. Au loin, on entend des enfants jouant et rigolant dans une cour de récréation, réticent à retourner en classe, malgré le son de cloche qui signifie la fin de la pause.

Bruits de pas. Cut to a grim face outlined against the town gates, staring into the distance, out toward the schoolhouse.

Le cours a repris, après quelques remontrances de l’institutrice. Un des enfants, S’ENNUYANT, regarde par la fenêtre. Y’A T’IL QUELQUE CHOSE D’INTERESSANT DEHORS ?

L’homme s’infiltre silencieusement dans l’école. L’institutrice lui tourne le dos. Elle ne se doute de rien.

Un par un, les enfants se mettent à fixer quelque chose derrière l’institutrice. La classe devient silencieuse. Finalement, l’institutrice entend un bruit derrière elle, se retourne et …

Image tirée de Fable

… tombe nez à nez avec un homme en caleçon, planté au milieu de sa salle de classe, souriant à pleines dents. Soudain, souriant de plus belle, il fait un doigt d’honneur. Tout le monde hurle …

**Ce qu’il s’est passé**

 Les enfants fixent quelque chose qui leur semble intéressant, et le doigt d’honneur est considéré comme un acte d’agression, d’où les cris.

Image tirée de Fable

### Evie / Cleverbot



Why RPG NPCs Always Repeat Themselves, de Dorkly

# Conclusion

Longtemps en retrait dans les jeux vidéo

# Références

* Auteur(s)/Site. nom de la référence, date.
* Auteur(s)/Site. nom de la référence, date.
* Auteur(s)/Site. nom de la référence, date.
* Auteur(s)/Site. nom de la référence, date.
* Auteur(s)/Site. nom de la référence, date.
* Auteur(s)/Site. nom de la référence, date.

1. Le Ferranti Mark I fut le premier ordinateur électronique généraliste commercialisé du monde.

   En novembre 1951, Dietrich Prinz écrivit un des premiers jeux vidéo, un programme d’échecs, pour le Ferranti [↑](#footnote-ref-1)
2. Les stored patterns sont des suites d’instructions répétées en boucle par l’agent. Hormis ce qui lui est dit de faire il ne fera rien d’autre [↑](#footnote-ref-2)