# Vivarium: vulgarisation des mécanismes de l'Intelligence Artificielle par l'exemple

Défense de

**Jérémy Morel**

moreljb0@etu.unige.ch

Supervisée par

**Stéphane Marchand-Maillet**

Stephane.Marchand-Maillet@unige.ch

Dans le cadre de l'obtention du

Master en Sciences Informatiques

(Soutenu le 8 Février 2018)

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

© 2018

Jérémy Morel

Tous droits réservés

# Abstract

Depuis ses débuts dans les années 1940, l'intelligence artificielle a pris une place prépondérante dans la vie quotidienne. Non seuleument dans le paysage culturel mais aussi dans les usages quotidiens par la démocratisation des technologies. Toutefois, de par sa nature et son mimétisme des capacités humaines, l'intelligence artificielle suscite beaucoup de fantasmes et d'incompréhension. Puisqu'elle nécessite des connaissances mathématiques et informatiques assez poussées, la vulgarisation de l'intelligence artificielle peut se montrer difficile à aborder. Ce travail de master propose donc un essai de vulgarisation par l'exemple. Vivarium est un jeu simulant un environnement en vase clos dans lequel évoluent des créatures munies d'un réseau de neurones s'améliorant grâce à un algorithme génétique. Le joueur est amené à observer le développement du comportement des créatures jusqu'à voir émerger une intelligence. Après une présentation des motivations de ce travail de master et d'un historique des utilisations existantes du *Machine Learning*, cette défense présentera un *post-mortem* de la conception de Vivarium avec une description des choix de conception puis du développement. Des tests utilisateurs ont été effectués sur un public cible (enfants de 10 à 14 ans), les retours (quoi que positifs) ont soulevés des questions au niveau de l'interactivité du jeu, un atelier annexe a été mis en place pour encadrer l'utilisation de Vivarium. Le déroulé de cet atelier a comblé certaines lacunes de Vivarium et a donné de bons résultats mais a aussi ouvert de nouvelles questions et pistes de réflexion quant à la vulgarisation de l'intelligence artificielle auprès du grand public. Ces questions sont traitées à la fin de cette défense.

# Sommaire

**Abstract** **iii**

**1 Introduction** **1**

**2 Le Machine Learning** **3**

2.1 Etat des lieux . . . . . 3

2.1.1 Historique . . . . . 3

2.1.2 Évolutions récentes . . . . . 4

2.2 Vulgariser le Machine Learning . . . . . 4

**3 La vie artificielle comme outil de démonstration** **7**

3.1 Le choix de la vie artificielle . . . . . 7

3.2 Etat de l'art: Algorithmes génétiques et vie artificielle . . . . . 7

**4 Vivarium** **20**

4.1 Concept . . . . . 20

4.2 Post-mortem . . . . . 22

**5 Résultats** **30**

5.1 Expériences . . . . . 30

5.2 Présentation des retours utilisateurs . . . . . 33

**6 Conclusion** **38**

6.1 Pistes pour de futurs travaux . . . . . 39

**Bibliography** **41**

## CHAPTER 1

# Introduction

Des automates cellulaires à *AlphaGo*, l'intelligence artificielle a parcouru une long chemin depuis ses débuts. Envisagée dès les premières heures de l'informatique, elle démontre son efficacité dans plsuieurs tâches (classification, prise de décision, jeux, etc) mais c'est son potentiel qui fascine. De ce fait, le domaine sera activement exploré tout au long de ces dernières décenies et, parallèlement à ces recherches, occupera une grande place dans le paysage culturel.

Si la conception des automates cellulaires de von Neumann date des années 1940, c'est dans le courant des années 1960 que diverses recherches mènent à un modèle mathématique. Les besoins en informatique commencent à émerger auprès de l'industrie et la course à l'espace entre les États-Unis et l'URSS stimulent l'imaginaire. La science-ficiton s'impose dans la culture populaire mais on note dès lors une dualité dans le traitement de l'intelligence artificielle, deux facettes qui sont intrinsèquement liées.

D'un côté c'est le côté fascinant de l'IA qui est exploré: beaucoup d'oeuvres d'anticipation traitent de l'émergence d'une IA super-intelligente et de notre rapport à celle-ci. L'IA est souvent associée à la thématique des robots parce qu'elles partagent plusieurs problématiques: le rapport au corps, à la place de l'être humain dans une société où des entités artificelles sont plus performantes ou le rapport à une autre forme de vie intelligente. On peut citer des oeuvres comme *Les Robots* d'Isaac Asimov [1], *Astro* d'Osamu Tezuka [2] ou encore *2001, l'odyssée de l'espace* de Stanley Kubrick [3]. Toutes ces oeuvres traitent de notre rapport à une vie intelligente artificielle alors même que l'IA n'en est qu'à ses balbutiements. Mais la plupart du temps notre vision des rapports hommes-machines sont pessimistes. Dans ses oeuvres, Issac Asimov introduit le *Frankenstein complex*: la peur d'être dépassé par sa propre création [4]. Outre les conjectures des oeuvres de science-fiction, cette peur se trouve un écho dans une problématique très contemporaine lorsque les premières machines remplacent le travail humain dans les usines. Petit à petit, un climat de méfiance s'est donc installé envers les IA.

L'état actuel et les évolutions de ces représentations peuvent s'expliquer par deux facteurs techniques: premièrement, la tendance générale des premières IA se limite au systèmes experts. Les décisions sont alors prises à parti de règles connues (ou déduites par inférence à partir de règles connues). L'IA est alors constituée de faits, de règles et d'un moteur d'inférence permettant de

déduire de nouveaux faits à partir des règles. Bien qu'ayant toujours une utilité aujourd'hui dans divers domaines, la tendance actuelle est plutôt vers l'apprentissage automatique. Nous sommes donc passés d'un système où l'IA agissait dans un cadre limité grâce à des règles dictées par son concepteur à une mimique d'une capacité de réflexion: l'apprentissage. Bien que fonctionnant différement du notre, l'apprentissage était jusqu'alors considéré (par le public non-spécialiste) comme une barrière fondamentale entre nous et l'IA.

La deuxième barrière à tomber est celle de la performance. En effet, au fur et à mesure de ses évolutions, l'IA s'est mise à surpasser les capacités humaines dans de nombreux domaines. Le cas le plus populaire est la défaite de Gary Kasparov face à *Deep Blue* aux échecs en 1997. Mais l'IA nous surpasse dans de nombreux domaines que l'on aurait pu penser hors de portée d'une IA: analyse d'image, diagnostic médical, détection d'émotions humaines ou dé-pixelisation d'images. Cette compétence croissante dans de nombreux domaines a rendu l'IA omniprésente dans l'accomplissemnt de certaines tâches: l'assemblage industriel en premier lieu mais aussi aujourd'hui pour des tâches de plus en plus complexes comme la conduite.

Paradoxalement, ces phénomènes ont coduit à en même temps craindre les dérives de l'IA mais aussi à lui accorder notre confiance en lui déleguant un nombre croissant de prises de décisions dans notre quotidien. Ainsi, si on craint la disparition de son emploi ou qu'on refuse que la décision d'une IA prévale sur celle d'un humain; on confie aujourd'hui à une IA le choix des programmes que l'on va visionner, l'ordonancement des résultats de recherche sur internet ou des articles susceptibles de nous intéresser.

Nous sommes donc confrontés quotidiennement à l'IA, que ce soit dans le paysage culturel, mais surtout dans son utilisation et en comprendre les enjeux nécessite d'en comprendre les mécanismes. De plus, comme précisé plus haut, nous accordons de plus en plus (consciement ou non) notre confiance à des systèmes dirigés par une IA et comprendre la manière dont ces systèmes fonctionne est donc essentiel. L'IA est donc un domaine qui nécessite un effort de vulgarisation considérable et c'est dans cette démarche que s'inscrit ce travail de master.

CHAPTER 2

# Le Machine Learning

## 2.1 Etat des lieux

Bien que le sujet principal consiste à vulgariser l'IA, j'ai choisi de resserer le propos sur le *Machine Learning* (apprentissage automatique en français, abbrgégé ML dans la suite de ce rapport), de par son quasi-monopole dans les interactions avec les utilisateurs non-spécialistes mais pour la manière unique dont il est perçue par le grand public comme illustré en introduction. Dans le but de contextualiser la conception de ce travail de master, cette section présentera un bref historique des recherches portant sur le ML jusqu'à aboutir à la situation actuelle qui sera rapidement analysé pour identifier la manière dont est perçue le ML afin de mieux cerner les objectifs que doit remplir un bon outil de vulgarisation.

### 2.1.1 Historique

Le *Machine Learning* est une branche de l'intelligence artificielle qui date presque du début de celle-ci, en 1950 Alan Turing publie*Computing machinery and intelligence* [5] et évoque la possibilité d'ordinateurs capable d'apprendre et de devenir "artificiellement intelligent". L'année suivante, à partir de modèles précédents (notament celui de McCulloch et Pitts dans [6]), Minsky et Edmonds publient le premier modèle de réseau de neurones artificiels [7]. Plusieurs modèles suivent, notament en 1958 avec le Perceptron, un classifieur fonctionnant par apprentissage supervisé qui prend $n$ entrées qui sont les caractéristiques de l'objet à classifié et une sortie qui correspond à la classe de l'objet (le perceptron est un classifieur binaire), calculée à partir des poids reliant les entrées à la sortie (des modèles multi-couches existent et fonctionnent sur le même principe. L'apprentissage se fait par correction des poids par rapport à la sortie attendue et nécessite donc des exemples éttiquetés pour fonctionner [8]. Diverses améliorations des modèles existants voient le jour dans les années 1960 et les algorithmes de ML entrent dans le monde du jeu: des IA de jeu de dames ou de *Tic-Tac-Toe* apparaissent. Le potentiel des réseaux de neurones créé de fortes émulations dans la recherche (voir [9]).

Toutefois ces ardeurs seront vite tempérées, les limites des réseaux de neurones apparaissent et plusieurs échecs du ML mettent un frein aux recherches. En conséquence les fonds pour la

recherche de l'IA s'effondrent et mènent à un *AI Winter* jusqu'aux années 1980 pendant lequel la recherche stagne. Suite à cette période, plusieurs avancées sortent le domaine de sa torpeur: le *neocognitron* en 1980 donne de bons résultats en reconaissance d'écriture [10], les réseaux de Hopfiled font leur apparition en 1982 [11], le *Q-learning* en 1989 [12] et les *Support Vector Machine* (SVM) en 1995 [13]. Les progrès sont significatifs mais les systèmes ne sont ni assez rapides ni assez performants pour être utilisés en situation et les systèmes experts restent prépondérants dans l'industrie.

### 2.1.2 Évolutions récentes

À partir des années 2000, diverses avancées dans la recherche, mais surtout les énormes bases d'informations disponibles vont permettre des avancées significatives dans le domaine. La popularisation du *Deep Learning* dans la reconaissance d'image illustre parfaitement ce tournant. La reconaissance d'image est un problème compliqué qui est pourtant facilement résolu par un humain, toutefois c'est un problème extrêmement difficile à modéliser et est donc hors de portée des systèmes experts. Le *Deep Learning* [14] existe depuis les années 1990 mais dans les années 2000 les avancées combinées dans les réseaux de neurones (notamenent les réseaux de neurones convolutifs), l'exploitation des GPU dans la reconaissance d'image (voir [15]) et surtout la masse importante d'images étiquetées disponibles, ont permis de rendre les IA extrêmement performantes à cet exercice.

Une autre avancée, retenue surtout pour son ampleur médiatique, a été la publication *AlphaGo* en 2015 [16]. Encore une fois, il s'agit d'un problème difficile à modéliser car dans le jeu de go il est difficile d'évaluer la pertinence d'un coup, le comptage des points est abstrait tant que la partie n'est pas terminée. Il s'agit donc d'un des jeux antiques qui résistaient aux IA, cependant en 2015, *AlplhaGo* devient la permière machine à battre un joueur professionel de go et l'année suivante elle bat le champion du monde dans des parties extrêmement médiatisées. Les réactions à cette victoire cristallisent les sentiments décrits tout au long de l'introduction de cet article: beaucoup de questionnements émergent sur les limites et le potentiel des IA chez le public non-spécialiste.

## 2.2 Vulgariser le Machine Learning

La vulgarisation scientifique dans son ensemble est un exercice délicat, surtout lorsque le sujet à vulgariser demande des connaissances préalables indispensables: il faut à la fois s'assurer que le public a toutes les clés en main pour comprendre les mécanismes exposés, mais aussi ne pas surcharger sa présentation avec un excès d'information. Qui plus est pour un sujet aussi ancré dans son domaine que l'est l'IA en informatique, il est facile d'oublier toutes les connaissances qui paraissent triviales à un spécialiste: la modélisation de problème, l'algorithmie ou l'algèbre sont autant de compétences qui doivent être soit introduites de manière limpide (elles ne sont pas le sujet de l'exposé) ou mises de côté. Dans le deuxième cas, il faut alors s'assurer que ces compétences

ne sont pas nécessaires à la compréhension du reste de l'exposé. De plus, lorsque le sujet est tentaculaire comme celui de l'IA, qui possède de nombreux modèles, il faut encore une fois faire un choix sur les notions qui seront introduites.

Ce sont ces raisons qui m'ont poussé à restreindre mon sujet à une branche de l'IA seuleument, plusieurs raisons m'ont amené à choisir le *Machine Learning*. La première est que le ML est la suite logique de la situation décrite en introduction: le ML produit des algorithmes qui dépassent les capacités humaines dans des exercices que l'ont pensait hors de portée des machines: reconnaissance d'image, analyse d'émotion ou diagnostic médical de certaines pathologies. Or lorsqu'une machine donne des résultats différents d'un humain dans ce genre de problème, nous avons du mal à accepter les résultats de la machine. Vulgariser les mécanismes de fonctionnement du ML pourrait selon moi réduire ce biais. Mais la raison la plus importante selon moi est de dissiper l'incompréhension que génère le ML, de par sa nature, le ML utilise des algorithmes inspéirés du vivant et de capacités de réflexion. Par conséquent, on assiste à une humanisation de l'IA basée sur la méconnaissance des mécanismes et des algorithmes: on prête une intention, une volonté et même une personalité à des alorithmes qui peuvent pourtant être réduits à des fonctions mathématiques assez simples. En 2015, *Google* présente *DeepDream* [17], il s'agit d'une modification de son IA de reconnaissance d'image par réseau de neurones convolutif, les images voient leurs *features* renforcées. Les images produites sont psychédéliques et produisent un grand impact auprès du public, on prête alors à *DeepDream* un propos, une intention artistique et une vision du monde. Les images sont qualitifées de "subconscient" ou de "rêve" de la machine, une démarche encouragée par le nom de l'algorithme lui-même.

**FIGURE 2.1** "La nuit étoilée" de Vincent Van Gogh après traitement par *DeepDream*.

C'est un comportement courant d'associer à un comportement pourtant dénué de sens des mécanismes et des émotions humaines. L'étude [18] de Heider et Simmel décrit ce phénomène. Sur le court film étudié, des formes géométriques se déplacent dans un décors rudimentaire, chaque

spectateur y associe pourtant une histoire et est capable d'attribuer un rôle et même une personalité à chacune des formes.

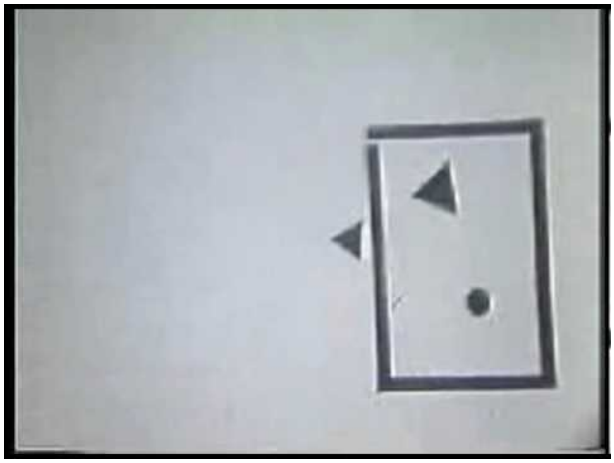


**FIGURE 2.2** Un extrait du film de Heider et Simmel.

Le but de ce projet n'est pas de réduire l'effet de comportement, mais comprendre les mécanismes du ML permet de mieux comprendre la teneur réelle de l'apprentissage du réseaux de neurones articiels sans lui prêter des vertus qu'il ne possède pas.

Pour résumer, le *Machine Learning* a été choisi pour sa capacité à mimer des comportement du monde vivant, car par ce fait il créé des attentes et des analyses déraisonnables sur ses capacités actuelles. Comprendre comment ces comportement sont construits pourrait permettre de prendre du recul et d'analyser de façon plus juste l'état actuel de la recherche en matière d'IA, ce projet s'inscrit comme un premier pas pour expliquer la manière dont fonctionne le ML à un public non-spécialiste.

CHAPTER 3

# La vie artificielle comme outil de démonstration

## 3.1 Le choix de la vie artificielle

Dans un processus de vulgarisation, il est primordial de restreindre le plus possible le champ de connaissances nécessaires à la compréhension de l'exposé, de plus donner au public des points d'accroche pour qu'il soit (au moins en partie) en terrain connu, l'aidera à assimiler. Un outil combinant vie artificielle et algorithme génétique répondait à tout ces critères. En effet, les algorithmes génétiques miment un mécanisme de la vie avec lequel la plupart des gens son familiers: l'évolution. Cela permet donc de créer un parallèle entre ce que l'on cherche à vulgariser et des conaissances que le public possède déjà.

De plus, la mimique de comportement du vivant par des algorithmes (réseaux de neurones, algorithmes génétiques, etc) sont des concepts qui nourissent beaucoup d'incompréhension puisqu'on à tendance à pousser la métaphore trop loin et à tirer des conclusions trop hâtives sur le potentiel de ces techniques. Il me paraît donc important d'appliquer ces techniques à un exemple simple pour dissiper les confusions.

Malgré leur caractère désuet, les algorithmes génétiques restent des fonctions d'optimisation qui fonctionnent bien sur des exemples simples et les performances ne seront pas un problème. Je pense donc que c'est un outil tout à fait adapté à la vulgarisation.

## 3.2 Etat de l'art: Algorithmes génétiques et vie artificielle

Ce chapitre présente un historique des différentes applications des algorithmes génétiques dans des systèmes de vie artificielle. Le but étant d'apporter un éclairage sur ce que peuvent apporter des algorithmes d'optimisation pour développer et faire émerger des comportements complexes à des entités artificielles évoluant dans un environnement fermé. Ces travaux ont servis de base et d'inspiration pour Vivarium.

# État de l'art: Algorithmes génétiques & Vie artificielle

Jérémy Morel

Université de Genève
moreljb0@etu.unige.ch

2017

**Abstract**

**Les algorithmes génétiques constituent une pierre angulaire des métaheuristiques. Leur utilisation a joué un rôle majeur dans le développement de la vie artificielle et l'étude de l'apprentissage et de l'évolution. Cet état de l'art se propose d'étudier l'application de différents algorithmes génétiques dans ce cadre à travers de nombreux exemples d'applications retraçant l'histoire et l'évolution de la vie artificielle.**

## I. Introduction

Le développement de la vie artificielle constitue un domaine majeur, très actif, en sciences informatiques. De par ses multiples définitions et la précocité de son apparition, il est difficile de retracer précisement l'origine de l'idée de vie artificielle. Toutefois, le concept d'automate de Jon von Neumann, présenté dans [33], correspond déjà aux aspects que nous étudierons ici. Nous utiliserons en effet la définition large de la vie artificielle par C. Taylor et D. Jefferson dans [47]: "*Artificial systems that serve as models of living systems for the investigation of open questions in biology*". Il est donc important de noter que les applications de la vie artificielle ne se basent pas sur une définition de la vie mais sur une théorie de la vie pour chercher à mettre en évidence des phénomènes supposés ou tester la possibilité d'émergence de certains comportements.

Les algorithmes génétiques sont quant à eux des algorithmes probabilistes d'optimisation introduits dans les années 1960. La conception des algorithmes génétiques "moderne" et encore utilisée aujourd'hui est développée ces mêmes années par J. H. Holland jusqu'à la publication de [20] en 1975 qui achève d'imposer les algorithmes génétiques tels que nous les connaissons aujourd'hui. *A Field Guide to Genetic Programming* ([13]) résume son fonctionnement en 5 étapes:

- Étape 1: générer aléatoirement une population de chromosomes
- Étape 2: exécuter le programme et calculer la *fitness* de chaque chromosome
- Étape 3: Sélectionner les meilleurs chromosomes et appliquer les opérations génétiques afin de générer une nouvelle population
- Étape 4: répéter depuis l'étape 2 jusqu'à trouver une solution acceptable au problème
- Étape 5: retourner le meilleur individu généré jusque là

Pour compléter cette définition, il convient de revenir sur la terminologie utilisée, les algorithmes génétiques constituent une analogie partielle de la théorie de l'évolution et de la sélection naturelle, par conséquent des termes repris de la biologie se retrouvent utilisés ici avec un sens altéré. En effet, dans le cas des algorithmes génétiques, on parle de *chromosome*

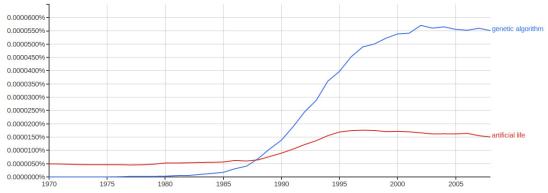
pour désigner un candidat potentiel comme solution au problème posé, il sera donc composé d'une série de bits que l'on appelera les *gènes*.

La *sélection* désigne le processus par lequel on va choisir les chromosomes utilisés pour générer une nouvelle population, différentes méthodes sont utilisées (conserver les 50% meilleurs chromosomes, choisir chaque parent avec une probabilité dépendante de ses performances, etc [31]) et seront étudiées dans les sections suivantes.

Les opérations génétiques désignent le *crossover* (c'est-à-dire le mélange des gènes des parents) et la *mutation* (soit l'altération de gênes aléatoires dans le génome du chromosome), encore une fois plusieurs méthodes existent pour chacune de ces opérations et seront discutées plus bas.

Enfin, chaque itération de l'algorithme constitue une *génération*. La *fitness* quant à elle est un terme purement informatique et désigne la fonction qui va chiffrer la performance du chromosome lors de sa génération. Le but de l'algorithme est de trouver le chromosome qui maximise (ou minimise selon le problème) la fitness, l'*espace de solution* désigne l'ensemble des chromosomes possibles à la résolution du problème.

Cet état de l'art se propose de donner un aperçu des applications des algorithmes génétiques dans le domaine de la vie artificielle. Toutefois, afin d'exposer le contexte ainsi que les questions qui ont conduit à ces applications, une première partie présentera exclusivement la vie artificielle, ses enjeux et différents domaines d'application.




*Utilisation des expressions "genetic algorithm" et "artificial life" dans la littérature selon [37].*

Mentionnons enfin que cet état de l'art n'a pas pour vocation de traiter de manière exhaustive un sujet aussi vaste, aussi nous concentrerons nous sur certaines publications clés dans l'histoire de la vie artificielle, tout en mentionnant de manière plus supperficielle les travaux importants et représentatifs afin de donner un aperçu le plus complet possible sur les évolutions des algorithmes génétiques associés à la vie artificielle.

## II. Historique de la vie artificielle

Étymologiquement, la vie artificielle regroupe, par opposition à la vie naturelle, "toute vie créée par l'Homme plutôt que par la Nature" ([29]). Toutefois, cette définition, en plus de couvrir un champ de recherche extrêmement vaste chevauchant plusieurs disciplines, est dépendante de la définition même de la vie en biologie, définition qui n'est définie qu'empiriquement. C'est pourquoi, dans le cadre de cette étude, nous nous imposerons deux critères: premièrement, afin de se concentrer sur les sciences informatiques, nous ne considèrerons ici que les solutions logicielles. Deuxièmement, comme précisé en introduction et en accord avec [47], nous considèrerons une définition *de facto* de projet de vie artificielle: si un projet se déclare vouloir servir de modèle dans l'étude de la vie, alors il rentre dans le cadre de cet état de l'art.

Ce choix peut se montrer pertinent, puisque le critère qui nous intéresse présentement, n'est pas la capacité d'un système à imiter le vie naturelle, mais bien la volonté d'apporter grâce au système en question une piste de réflexion ou d'illustrer une hypothèse. Nous étudierons donc ici la vie artificielle comme un moyen de suppléer à l'étude classique de la vie par la biologie. Cette vision correspond à celle de C. Langton dans [29]: "*A-life complements the analytic approach of traditional biology with a synthetic approach: rather than studying biological phenomena by taking living organisms apart to see how they work, we attempt to put to-*

*gether systems that behave like living organisms."* [1].

Les premiers signes de développement de vie artificielle peuvent remonter extrêmement loin, puisque les automates mécaniques, remontant de l'Antiquité jusqu'aux automates de Jacques de Vaucanson au XVIIIe siècle, rentrent tout à fait dans ce cadre. Toutefois, ne répondant pas à notre premier critère (celui de ne considérer que les solutions logicielles), nous n'en parlerons pas plus.

Les automates "informatiques" sont les premières apparition d'une vie artificielle logicielle. En effet, après avoir imaginé une machine auto-reproductrice avec son *kinematic model* en 1949 ([34]), J. von Neumann va chercher à s'affranchir des contraintes physiques. Avec S. Ulam, ils développent le concept d'automate cellulaire ou *cellular automata* ([42]) au début des années 1950.

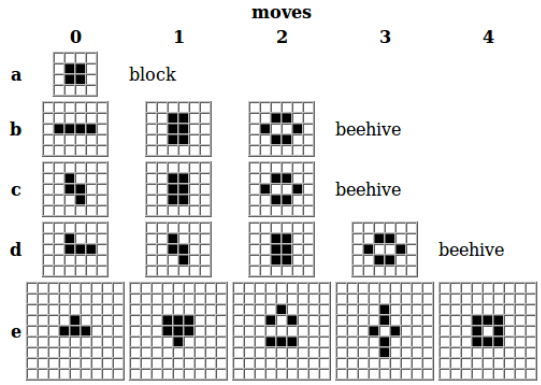
C. Langton définit un automate cellulaire[2] comme un "trellis d'automates finis", c'est-à-dire un ensemble d'automates avec un ensemble d'états et de transitions fini pour lesquels l'ensemble des états des automates au temps *t* constitue l'input du temps *t+1*. Tous les automates partagent la même table de transition.

De nombreux automates vont voir le jour: en 1952 J. von Neumann imagine un automate de 200 000 cellules capable de reproduire une copie de la configuraiton initiale ([53]). Les années 1960 verront naître de nombreux automates cellulaires, notamment ceux de E. Fredkin en 1961 ou de S. Ulam en 1967 ([53], [25]), mais c'est l'automate cellulaire à 2 dimensions de J. Conway le *Game of Life* qui popularisera cet outil et qui révèlera son potentiel en matière de vie artificielle en 1970 ([11]).

En effet, cet automate met en évidence le fait que des règles simples peuvent donner lieu à des comportements extrêmement variés et complexes. L'automate n'est rien d'autre qu'une grille à 2 dimensions dont chaque case constitue une cellule de l'automate, chacune peut se trouver dans l'état mort ou vivant, 4 règles régissent le changement d'état:

- Une cellule meurt par sous-population si moins de 2 de ses voisins sont vivants
- Une cellule meurt par surpopulation si plus de 3 de ses voisins sont vivants
- Une cellule revient à la vie par reproduction si exactement 3 de ses voisins sont vivants
- Si une cellule vivante est entourée de 2 ou 3 voisins vivants, alors elle reste en vie

L'automate possède des propriétés remarquables: des structures périodiques, qui se déplacent ou même qui se répliquent (découverts en 2010 [14]). Le jeu est par ailleurs Turing-Complet.




*Structures menant à des structures périodiques dans le Game of Life [11].*

C'est à cette période que le développement de la vie artificielle va prendre son envol: les systèmes vont mettre l'accent sur une intelligence plus distribuée ([28]) et les systèmes vont s'inspirer des mécanismes du vivant observés pour tenter de les comprendre. En effet dans [29], C. Langton fournit une analogie pertinente entre la physique (à travers les équations de Navier-Stokes) et la biologie pour expliquer la différence entre la prédiction et la compréhension. En effet, ces équations nous permettent de prédire les mouvements des fluides, pourtant aucun de ces fluides ne calcule d'équation pour se déplacer, ce déplacement fait intervenir de

[1] extrait de [29] page 1
[2] [29] section 2.6

nombreux mécanismes que ces équations ne nous permettent pas de comprendre. La vie artificielle va donc tenter de reproduire les modèles de prédiction de la biologie pour les observer et tenter de les comprendre.

De nombreux chercheurs, comme Doyne ([9]) ou Wolfram ([54]) vont se pencher sur les automates cellulaires, d'autres modèles vont voir le jour: Reynolds développe les *Boids* en 1986 ([40]). Langton donne pour la première le nom d'*Artificial Life* à cette discipline à l'occasion d'*Artificial Life I*, la première conférence dédiée à la vie artificielle, et en donne une des premières définitions: *"Artificial life is the study of artificial systems that exhibit behavior characteristic of natural living systems. It is the quest to explain life in any of its possible manifestations, without restriction to the particular examples that have evolved on earth. This includes biological and chemical experiments, computer simulations, and purely theoretical endeavors. Processes occurring on molecular, social, and evolutionary scales are subject to investigation. The ultimate goal is to extract the logical form of living systems."*[3]

## III. Introduction des algorithmes génétiques

C'est dans ce contexte qu'apparaissent les algorithmes génétiques: bien que certains algorithmes évolutionnaires commencent à apparaître dans les années 1960, ce n'est qu'en 1975 que J. H. Holland formalise et publie dans [20] les algorithmes génétiques tels que nous les connaissons. Ils vont founir un terrain très fertile et qui sera exploité pour le développement de la vie artificielle dans les années qui suivent.

Les *learning classifier systems* sont les premières applications à apparaître mêlant algorithmes génétiques et vie artificielle. Ils commencent à se développer au début des années 1980, sous l'impulsion des travaux de J. Holland sur les algorithmes génétiques ([20]), K. de Jong et S. Smith modélisent un automate probabiliste basé sur la prise de décision à partir de règles dans [46] en 1980. Plusieurs travaux ont vu le jour, notamment ceux de Holland et Reitman en 1978 ([23]), de Booker en 1982 ([5]) ou de Riolo en 1991 ([41]), mais nous détaillerons ici les travaux de S. Wilson, publiés dans [52] en 1986.

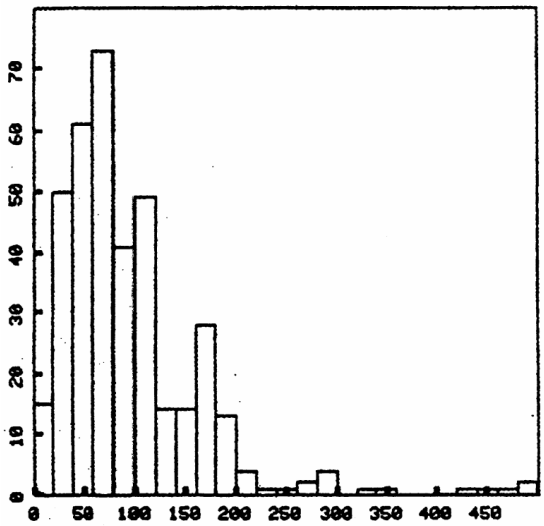
L'expérience consiste à utiliser un classifieur pour permettre à des animaux de trouver de la nourriture, l'environnement est une grille périodique de 18x58, dans laquelle se trouve un animal, des obstacles (notés *T* pour *Tree*) et de la nourriture (notée *F* pour *Food*). L'animal a une perception limitée à ses 8 voisins représenté par un vecteur de 8 caractères, chacun étant codé sur 2 bits: 00 pour un espace vide, 01 pour un obstacle et 11 pour de la nourriture. Une règle du classifieur est donc constituée de 16 bits représentant la situation dans la règle est applicable, une réponse (la direction dans laquelle l'animal va se déplacer si la règle est choisie) et d'un poids qui détermine la probabilité de choisir cette règle si elle entre en concurrence avec d'autres règles. Chaque bit de la règle peut donc prendre la valeur 0, 1 ou # qui agit comme un "joker" et valide la règle quel que soit le symbole à cette position.

À chaque pas, l'animal observe son envionnement puis détermine toutes les règles qui sont applicables, l'une d'elle est choisit au hasard (les poids des règles déterminent leur probabilité d'être choisie), le poids de la règle est affaiblie (si elle mène à se déplacer sur une case *Tree*) ou renforcée (si elle mène sur une case *Food*). Si aucune règle ne correspond au vecteur alors une règle est créée (en remplaçant aléatoirement des bits par #) en choisissant un poids et une réaction au hasard.

Une fois que l'environnement ne contient plus de nourriture, une nouvelle génération est créée à partir des meilleurs éléments (on applique un crossover classique pour générer de nouveaux classifieurs). On mesure les performances des individus en décomptant leur période, c'est à dire le nombre de pas nécessaire en moyenne pour ramasser une unité de nouriture, voici les résultats obtenus

[3] Citation extraite de [29].

par S. Wilson:




*Résultats présentés dans [52], en abscisse la période de l'individu et en ordonnées le nombre de classifieurs.*

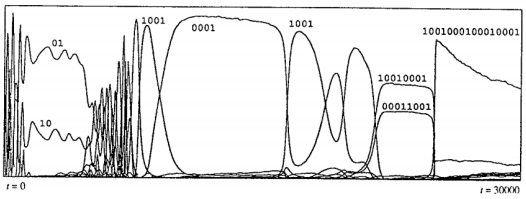
On peut voir que l'animal développe une certaine intelligence dans le cadre de la définition donnée par Wilson: il devient performant à satisfaire ses besoins, la période diminue considérablement au fur et à mesure de l'évolution. Wilson discute des différents paramètres de la simulation et propose des extensions de son modèle.

K. Lindgren publie en 1991 dans [30] une application des algorithmes génétiques sous une forme assez simple. Le but de l'expérience est d'observer l'émergence d'une stratégie coopérative malgré l'appât des gains immédiats. Pour cela K. Lindgren prend pour base de l'environnement le *problème du Prisonnier* dans lequel deux agents ont le choix entre coopérer ou se trahir, chacun des cas de figure détermine la somme gagnée par chacun des agents:

| | Coopérer | Trahir |
|---|---|---|
| Coopérer | (3,3) | (0,5) |
| Trahir | (5,0) | (1,1) |

Le but étant de maximiser ses gains, la solution immédiate consiste à trahir son partenaire, quelle que soit sa décision, toutefois si les deux agents choisissent cette solution alors le gain sera minime. L'idée est donc que pour maximiser le gain, une stratégie collective est préférable. Pour cela, chaque agent est doté d'une mémoire qui lui permet de se souvenir des $n$ derniers coups de son adversaire, pour une mémoire d'un seul coup (mémoire-1) l'agent ne tient compte que du dernier coup de son adversaire, 4 stratégies existent alors: toujours coopérer, toujours trahir, jouer le même coup que son adversaire ou le coup opposé. On a donc une mémoire sur 2 bits, de manière générale on double le nombre de bits en considérant un coup de plus, par conséquent une mémoire-n demande $2n$ bits, ces bits de mémoire constituent le patrimoine génétique des agents.

Le principe consiste donc à faire jouer à chaque génération les agents les uns contre les autres, à chaque reproduction, l'agent peut muter, ce qui revient à ajouter un coup à sa mémoire. Dans ce cas la mutation est neutre, cela signifie qu'on double son patrimoine sans changer la prise de décision (le comportement `01` qui signifie "Trahir après une trahison", deviendra `0101` ce qui revient aussi à trahir après une trahison). Après 30 000 générations, Lindgren obtient le résultat suivant:




*Représentation des popularités des différentes stratégies [30].*

On observe qu'après une période de chaos pendant laquelle se succèdent et cohabitent plusieurs stratégies, l'environnement se stabilise: une coopération globale se met en place et une stratégie domine largement, jusqu'à ce que les mutations donnent naissance a une stratégie qui profite de cette stratégie globale, la majorité adopte alors une nouvelle stratégie

globale. L'expérience de Lindgren illustre parfaitement l'émergence d'écosystème.

On peut citer d'autres travaux portant sur l'étude de l'émergence d'un comportement coopératif. En 1987, R. Axelrod (déjà auteur de [2] sur l'étude de la coopération) publie dans [3] une approche du dilemme du prisonnier. Les travaux de 1991 de Collins et Jefferson sur *Antfarm* ([7]) étudient l'émergence d'une ogranisation dans une foumilière artificielle. La même année Werner et Dyer étudient le rôle et l'évolution des communications en vie artificielle ([51]).

En 1992, J. Holland publie dans [21] une description du modèle *Echo*, un environement faisant intervenir des agents dotés de patrimoine génétique et destinés à se reproduire. Un agent est défini par son chromosome, une chaîne de caractères composée à partir d'un alphabet de 4 lettres *{a, b, c, d}*, qui sont les ressources de l'environement. Un agent est capable de stocker des ressources jusqu'à obtenir toutes les lettres qui composent son patrimoine, moment auquel il va se cloner (avec une faible chance de mutation). Il n'y donc pas de principe de sélection puisque celui-ci est intrinsèque aux règles de la simulation: les agents les plus performants pour se reproduire répanderont plus leur patrimoine.

Dans le modèle le plus simple[4], le chromosome d'un agent est composé de plusieurs sous-chaînes qui définit son comportement: 3 tags (combat, défense et accouplement), 3 conditions (combat, échange et accouplement). En effet, lorsqu'un agent en rencontre un autre, 3 interactions sont possibles:

- Le combat: si la condition de combat de l'agent 1 correspond au tag offensif de l'agent 2, alors l'agent 1 est attaqué, le tag offensif de l'agent 2 ets alors comparé au tag offensif de l'agent 1 et le perdant cède ses ressources au vainqueur.
- Le commerce, s'ils ne se battent pas les agents comparent leur condition d'échange (une seule lettre), si les deux conditions sont satisfaites alors elle transfèrent l'excès de ressources correspondant à leur condition de commerce.
- L'accouplement, s'ils ne se battent pas les agents comparent leur condition d'accouplement avec le tag d'accouplement, si les conditions sont satisfaitent alors elles échangent une partie de leur chromosome par *crossover*.

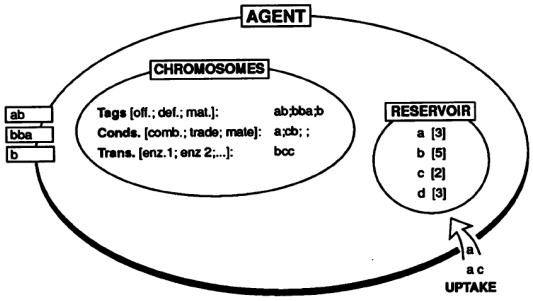



*Schéma d'un agent de Echo.*

Malgré ces règles simples, *Echo* a donné des résultats étonnamment complexes, avec l'apparition d'écosystèmes dotés de proies, de prédateurs ou de commerçants. J. Holland présente ([21]) différents modèles, comportant des entités plus complexes composées de plusieurs agents ou des fontaines de ressources dans le territoire. Toutefois la diversité des comportements n'augmente pas de manière espérée en complexifiant les interactions ([45]).

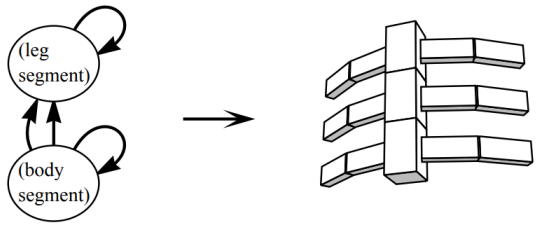
À la même période, on peut citer les travaux de T. Ray sur *Tierra* [39] en 1991. Dans lequel des programmes auto-générés à partir d'un *ancêtre* codé par Ray se partagent un environnement virtuel composé d'un espace mémoire ainsi qu'un processeur unique. Le but des programmes est donc de s'accaparer le plus de temps de calcul possible afin de se reproduire. Au départ seul l'ancêtre est présent et est capable de se reproduire, au fil des mutations il va donner naissance à de nouveaux programmes de plus en plus différents. L'expérience a vu naître des

[4]Différentes variantes du modèles existent qui modifient les tags, séparent le tag de commerce de la ressource à échanger, les résolutions des combats ou encore les mutations ([24]).

programmes au comportement sophistiqué comme des parasites qui utilisent les autres programmes pour se reproduire et des super parasites s'immunisant contre ces derniers. La question des parasites sera notamment étudiée par W. D. Hillis, pour leur impact sur les algorithmes génétiques afin d'éviter les optimum locaux ([19]). En 1993, l'expérience *Avida* (C. Ofria, C. Adami et C. T. Brown dans [1]), s'inspire du fonctionnement de *Tierra*. Les individus possède ici leur propre mémoire et unité de calcul et se partagent un environnement dans lequel ils doivent se reproduire.

D'autres travaux ont avancé diverses manières d'appliquer les algorithmes génétiques à la vie artificielle: pour faire évoluer les poids des arcs d'un réseau de neuronnes (Garis dans [12]), pour générer des relation *stimulus-réponse* (Ngo et Marks dans [36]) ou pour optimiser des réseaux de senseurs (van de Panne et Fiume dans [48]). Nous développerons ici les travaux de 1994 de K. Sims dans [44] et [43] qui consistent à générer des créatures en 3 dimensions grâce à un algorithme génétique. Les créatures sont composées de parallélépipèdes de dimensions diverses reliées entre elles par des articulations, elles évoluent dans une arène dans laquelle elles affrontent une autre créature pour la possession d'un cube lors d'un match de 8 secondes.

Le génotype d'une créature est composé d'un graphe orienté qui décrit sa morphologie: les noeuds correspondent aux parties du corps en 3D et les arcs les articulations.




*Exemple de créature à côté de sa représentation en graphe orienté [43].*

Chaque partie du corps contient donc diverses informations: ses dimensions, un ensemble de neurones qui seront discutés plus tard et les différentes articulations qu'elle possède, ainsi que les degrés de liberté de ces articulations. Les articulations quant à elles contiennent leur orientation, position et échelle.

Chaque créature est aussi équipée d'un "cerveau" ils s'agit d'un ensemble de senseurs (pour capter la lumière, l'état des articulations et les contacts avec d'autres objets) reliés aux outputs (bouger une articulation) par des neurones pouvant effectuer des opérations mathématiques simples (somme, différence, sin, cos, etc) ou des opérations logiques (>, if, etc) ou même stocker des données en mémoire.

Les créatures s'affrontent ensuite dans un match de 8 secondes, placées à égale distance d'un cube dans une arène complètement plate, leur but est de minimiser la distance qui les séparent du cube tout en maximisant celle de leur adversaire. On attribue alors un score allant de 0 à 2 à chaque créature (la somme des deux scores étant égale à 2). Plusieurs types de tournois sont proposés par Sims mais c'est le *all vs. best* qui est conservé, c'est-à-dire que toutes les créatures de l'espèce A affrontent le meilleur représentant de l'espèce B (et vice-versa).

Les créatures dont le score est inférieur au seuil de survie sont supprimées, les meilleures se reproduisent (un crossover et des mutations interviennent au niveau du graphes et des informations contenues dans les parties du corps et articulations).

Au final, les créatures parviennent à un comportement optimal au bout d'une centaine de générations, de nombreuses espèces sont générées avec une grande variété de techniques pour s'approprier le cube: certaines développent de long bras fins, d'autres coincent le cubes, certaines s'en prennent directement à leur adversaire. Sims note que les résultats les plus intéressants surviennent lorsque les deux espèces conccurentes possèdent une technique pour s'approprier le cube et cherchent alors à contrer la technique de leur adversaire ([43]).

Plusieurs travaux ont étudiés l'émergence d'un organisme complexe grâce aux algorithmes génétiques, on peut citer ceux de Bagley et Farmer en 1991 dans [4] sur l'émergence d'un métabolisme à travers celle d'une fonction de fitness.

## IV. État actuel

À partir du début des années 2000, après plusieurs années de recherches intensives sur un domaine presque neuf, beaucoup d'algorithmes dédiés au départ à la vie artificielle se démocratisent, s'exportent vers d'autres domaines et font l'objets d'études sur leur fonctionnement et leurs résultats. En 2005, en préface de [6] les auteurs écrivent: "*The Artificial Life term appeared more than 20 years ago* [...]. *Since then the area has developed dramatically, many researchers joining enthusuastically and research groups sprouting everywhere.* [...] *We are now reaching a stage that one may describe as maturer: with more rigour, more benchmarks, more results, more stringent, acceptance criteria, more applications, in brief, more sound science.*".

Plusieurs autres travaux tentent d'appliquer les méthodes d'auto-organisation à des cas pratiques, le plus souvent avec des robots ou des senseurs. On peut citer les articles de I. Harvey en 1998 ([16]), de S. Nolfi et D. Floreano en 2000 ([35]) ou encore de R. Watson, S. Ficici et J. Pollack en 2002 ([49]).

De nombreux travaux voient le jour dans des domaines variés à la même période. En 2007, D. H. Kim, A. Abraham et J. H. Cho exploitent les algorithmes génétiques développés afin de mimer la collecte de nourriture de la bactérie *E.coli* pour résoudre des problèmes d'optimisation ([27]). R. Morris et T. Watson publient en 2009 dans [32] leurs travaux sur la *Frequency-Dependent Selection*, c'est-à-dire à la dépendance de certaines fonctions de fitness à la fréquence de certains génotypes, cas de figure courant en vie artificielle. En effet, souvent certains comportements n'ont de sens que s'ils apparaissent fréquement dans une population, ou au contraire s'ils ne sont pas trop présents. Ces travaux formalisent et rejoignent les travaux sur les comportements collaboratifs déjà traités comme ceux de Lindgren ([30]).

En 2009, Cunningham et O'Riordan expérimentent dans [8] l'application d'algorithmes géntiques développés pour la vie artificielle dans le problème d'optimisation des gains dans le cas d'investissement dans un marché commun. Dans cette situation, chaque joueur (8 dans ce cas-ci) se voit distribuer des ressources sous forme de jetons (entre 10 et 25), a le choix d'investir ses jetons dans un des deux marchés:

- Le premier marché propose un retour sur investissement fixe pour chaque jeton investi.
- Le second marché (le marché commun) propose un retour sur investissement dépendant du nombre de jetons investis par l'ensemble des joueurs. Le retour est faible s'il ya peu de jetons investis et augmente au fur et à mesure, mais cette croissance n'est pas monotonne: le retour décroit lorsque trop de jetons sont investis, jusqu'à devenir moins rentable que le retour du marché 1.

Dans le problème présenté dans [8], l'équilibre de Nash est atteint lorsque 8 jetons par joueur sont investis dans le marché commun. On fait donc évoluer les joueurs de manière à atteindre cet équilibre, les joueurs évolutifs ont donc accès à l'historique des parties précédentes et jouent contre des joueurs naifs appliquant une stratégie pré-implémentée, puis on fait jouer huit joueurs évolutifs (ayant évolué séparément) entre eux. On observe que l'équilibre de Nash est atteint rapidement avec une moyenne de 8.55 jetons investis par joueur, toutefois en faisant jouer les joueurs évolutifs avec des joueurs investissant une quatité fixe de jetons par partie alors les joueurs évolutifs ont tendance à combler eux-mêmes le déficit en jetons du

marché commun s'ils ont un grand nombre de jetons à disposition, ce qui est exploité par les joueurs jouant un nombre de jetons légèrement en dessous de l'équilibre de Nash.

On remarque que ces articles partagent l'optimisation comme sujet de fond. En effet, comme précisé en introduction de cette section, on assiste désormais à une application des algorithmes de vie artificielle beaucoup plus dans le cadre des sciences informatiques (en tout cas au sens traditionnel du terme). Cela rejoint la prédiction de Langton à ce sujet en 1986: *"It is quite likely that we will learn more about computation by studying life than we will learn about life by studying computation."*[5].

Toutefois on constate, une perte de vitesse de l'utilisation des algorithmes génétiques au cours des années 2000. On peut expliquer cela par le développement des métaheuristiques qui fournissent de nouvelles méthodes, parfois plus efficaces qui dépassent certaines limites des algorithmes génétiques. De plus, malgré le regain d'intérêt pour les réseaux de neurones, on constate plus une effervescence autour de l'optimisation de fonction ou la résolution de problème d'intelligence artificielle (dans le cadre du *machine learning*).

Toutefois, on remarque une forte augmentation du nombre d'applications ludiques et pédagogiques des algorithmes de vie artificielle développés au cours des décennies précédentes (et prséentés plus haut). En effet, les fortes ressemblances entre ces algorithmes et des concepts communs et connus (au moins en partie) comme l'évolution, la génétique ou la sélection naturelle, permettent à ces méthodes d'être attractives même auprès de non spécialistes, ce qui offre une porte d'entrée à une utilisation plus large de ces algorithmes.

On peut citer de nombreux exemples dans des domaines variés: J. Issacs, R. Watkins & S. Foo publient en 2002 leur algorithme pour simuler une fourmilière artificielle ([26]. En 2009, suite à leurs travaux sur l'utilisation d'algorithmes génétiques pour générer des particules et des animations graphiques ([17]), E. Hastings, R. Guha & SK. Stanley produisent un jeu vidéo (basé sur ces principes) donc le contenu se génère grâce à des algorithmes développés pour la vie artificielle: *Galactic Arms Race* ([18]).

En 2011, R. Weber, un étudiant de l'Université de Californie à Santa Cruz, met en ligne une application 2D dont le but est de créer une voiture devant rouler le plus loin possible sur une route chaotique. La voiture designée par l'utilisateur n'est en fait qu'une ébauche qui servira de base à un algorithme génétique pour faire évoluer la voiture de manière à compléter le circuit.

Des projets commerciaux verront même le jour implémentant des algorithmes de vie artificielle: la série des "Creatures" (4 jeux sdéveloppés entre 1996 et 2011 par *Creatures Lab* iamginée par S. Grand ([15]). Le jeu "Black & White" développé par *Lionhead Studios* en 2001 utilise les travaux de vie artificielle pour générer en cours de partie les évolutions de la "Créature" du joueur (bien qu'ici les perceptrons soient la base de l'algorithme [10]. On peut encore citer le jeu "F.E.A.R", développé en 2005 par *Monolith Productions*, qui utilise des algorithmes d'auto-organisation pour contrôler l'intelligence artificielle des adversaires ([38]).

## V. Conclusion

Tout au long de cet état de l'art, nous avons pu observer l'étendue du sujet de la vie artificielle, ainsi que l'impact qu'ont eu les algorithmes génétiques sur cette discipline. Nous avons pu étudier un nombre conséquent d'articles aux sujets très diversifiés et apportant des contributions dans des domaines variés qui ont permis de nombreuses avancées grâce à un engouement très rapide dans un domaine où tout restait à inventer.

Toutefois, le nombre conséquent de publications sur le sujet sur une période très courte (principalement les années 1990) rend la synthèse de ces publications assez délicate, d'autant plus avec un sujet aux contours aussi

[5]Citation extraite de la section 2.7 de [28]

flous (tant au niveau de sa définition qu'au niveau des disciplines abordées). En conclusion de leur exposé de 1994 ([31]), M. Mitchell et S. Forrest discutent de la difficulté de séparer les "*cute simulations*"[6] des systèmes qui apportent des éléments importants sur la conception de vie artificielle, l'étude de la vie ou la manière de construire de meilleurs algorithmes.

C'est une problématique formulée en plein coeur de cette recherche frénétique sur la question de la vie artificielle, mais qui est non seulement nécessaire pour regrouper les différents travaux effectués dans un seul article (qui est le point soulevé par Mitchell et Forrest) mais aussi, je pense, toujours actuelle dans une discipline qui attire toujours plus de travaux s'éloignant du cadre de la recherche pour apporter un cadre ludique et pédagogique, que ce soit dans le cadre des sciences informatiques ou de la biologie.

## References


[1] Adami C., Brown C. T. "Evolutionary Learning in the 2D Artificial Life Systems Avida". *Artificial Life IV*. MIT Press. 1994. p. 377-381.

[2] Axelrod R. "The evolution of cooperation". Basic Book. 1984.

[3] Axelrod R. "The evolution of strategies in the iterated Prisoner's Dilemma". *Genetic Algorithms and Simulated Annealing*. L.D Davis editor. 1987.

[4] Bagley R. J. & Farmer J. D. "Spontaneous Emergence of a Metabolism". *Artificial life II*. 1991. p. 93-139.

[5] Booker L. "Intelligent Behavior as an Adaptarion of the Task Environment". Ph.D. Dissertation (Computer and Communication Sciences. University of Michigan. 1982.

[6] Capcarrere M., Freitas A. A., Bentley P. J., Johnson C. G. & Timmis J. *Advances in Artificial Life: 8th European Conference, ECAL 2005, Canterbury, UK, Spetember 5-9 2005*. Vol. 3630. Springer. 2005.

[7] Collins R. J. & Jefferson D. R. "AntFarm: Towards Simulated Evolution". *Artificial life II*. 1991. p. 579-601.

[8] Cunningham A. & O'Riordan C. "A genetic programming approach to an apparopriation common pool game". *European Conference on Artificial Life*. Springer Berlin. Heidelberg. 2009. p. 167-174.

[9] Farmer D., Doyne J., Toffoli T. & Wolfram S. "Cellular Automata: Proceedings of Interdisciplinarity Workshop". Los Alamos. New Mexico. March 7-11. 1983.

[10] Evans R. "AI Game Programming Wisdom". Charles River Media. 2002. Capter 11.2: Varieties of Learning. p. 567-578.

[11] Gardner M. "Mathematical Games: The Fantastic combinations of John Conway's new solitaire game "life" ". *Scientific American* (223), p. 120-123

[12] Garis H. "Genetic Promgramming: Building Artificial Nervous Using Genetically Programmed Neural Network Modules". *Proceedings of the 7th International Conference on Machine Learning*. 1990. p. 132-139.

[13] R. Poli, W. B. Langdon, and N. F. McPhee. "A field guide to genetic programming". Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008. (With contributions by J. R. Koza).

[14] Goucher A. P. "Oblique Life spaceship created". *Game of Life News*. 2010.

[15] Grand S. "Creation: Life and How to Make It". Harvard University Press. 2003.

[16] Harvey I. "Artificial evolution and real robots". *Artificial Life and Robotics 1*. 1997. p. 35-38.


[6] Citation extraite du texte

[17] Hastings E. J., Guha R. K. & Stanley K. O. "Interactive evolution of particle systems for computer graphics and animation". *IEEE Transactions on Evolutionary Computation 13(2)*. p. 418-432.

[18] Hastings E. J., Guha R. K. & Stanley K. O. "Evolving content in the Galactic Arms Race video game". *IEEE Symposium on Computational Intelligence and Games*. p. 241-248.

[19] Hillis W. D. "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure". *Artificial life II*. 1991. p. 313-324.

[20] Holland J. H. "Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence." *Ann Arbor*, 1975, MI: University of Michigan Press.

[21] Holland J. H. "Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence." 1992, MIT Press.

[22] Hollad J. H., Holyoak K. J., Nisbett R. E., Thagard P. R. "Induction: Process of Inference, Leaning and Discovery". The MIT Press. 1986.

[23] Holland J. H. & Reitman J. S. "Cognitive systems based on adaptative algorithms". *Pattern-Directed Inference Systems*. New York: Academic Press. 1978.

[24] Hraber P. T., Jones T. & Forrest S. "The ecology of Echo". *Artificial Life* 3.3. 1997. p. 165-190.

[25] Ilachinski A. "Cellular automata: a discrete universe" *World Scientific Publishing Co Inc*. 2001.

[26] Issacs J. C., Watkins R. K., & Foo S. Y. "Evolving ant colony systems in hardware for random number generation". *In Proceedings of the 2002 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Neural Networks Council. Vol. 2. p. 1450-1455.

[27] Kim D. H., Abraham A. & Cho J. H. "A hybrid genetic algorithm and bacterial foraging approach for global optimisation". *Information Sciences*. 2007. p. 3918-3937.

[28] Langton C. G. "Studying artificial life with cellular automata". *Ann Arbor*. Elsevier Science Publishers. 1986.

[29] Langton C. G. "Artificial Life". 1989

[30] Lindgren K. "Evolutionary phenomena in simple dynamics". *Artificial life II*. 1991. p. 295-312.

[31] Mitchell M. "An introduction to genetic algorithms". *Cambridge, Massachusetts, England, Fifth printing* 3. 1999. p.62-75.

[31] Mitchell M. & Forrest S. "Genetic Algorithms and Artificial Life". *Artificial Life I*. MIT Press. 1994. p. 267-289.

[32] Morris R. & Watson T. "Positively versus Negatively Frequency-Dependent Selection". *European Conference on Artificial Life*. Springer Berlin. Heidelberg. 2009. p. 77-84.

[33] von Neumann J. "The General and Logical Theory of Automata". *The World of Mathematics*, 1956, J. R. Newman Vol.4, p. 2070-2098

[34] von Neumann J. "The Theory of Self-reproducing Automata". A. Burks, ed. 1966. Urbana, IL: Univ. of Illinois Press

[35] Nolfi S. & Floreano D. "Evolutionary robotics: The biology, intelligence and technology of self-organizing machines". Cambridge. MIT Press. 2000.

[36] Ngo J. T. & Marks J. "Spacetime Constraints Revisited". *Computer Graphics*. Annual Conference Series. 1993. p. 343-350.

[37] Google Ngram Viwer. Web. `https://books.google.com/ngrams`.

[38] Orkin J. "Three States and a Plan: The A.I of F.E.A.R" *Game Developpers Conference 2006*. 2006.

[39] Ray T. S. "Evolution and optimization of digital organisms". *Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers*. 1991.

[40] Reynolds C. "Flocks, herds and schools: A distribued behavioral model". *SIGGRAPH 87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery. 1987. p. 25-34.

[41] Riolo R. "Lookahead planning and latent learning in classifier system". In J.A. Meyer and S. W. Wilson editors. *From animals to animats: Prceedings of the first international conference on simulation of adaptative behavior*. MIT Press. 1991. p. 316-326.

[42] Schiff J. L. "Cellular automata: a discrete view of the world". Vol.45. John Wiley & Sons. 2011.

[43] Sims K. "Evolving 3D Morphology and Behavior by Competition". *Artificial Life IV Proceedings*. ed. by Brooks & Maes. MIT Press. 1994. p. 28-39.

[44] Sims K. "Evolving Virtual Creatures". *Computer Graphics*. Annual Conference Series. 1994. p. 15-22.

[45] Smith R. M. & Bedau M. A. "Is Echo a complex adaptative system?". *Evolutionary computation* 8.4. 2000. p. 419-442.

[46] Smith S. F. "A learning system based on genetic adaptative algorithms". University of Pittsburg. 1980.

[47] Taylor C. and Jefferson D. "Artificial life as a tool for biological inquiry". *Artificial Life*, Fall 1993/Winter 1994, MIT Press Cambridge. Volume 1 Issue 1-2, p. 1-13.

[48] van de Panne M. & Fiume E. "Sensor-Actuator Networks". *Computer Graphics*. Annual Conference Series. 1993. p. 335-342.

[49] Watson R. A., Ficici S. G. & Pollack J. B. "Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems 39*. 2002. p. 1-18.

[50] Weber R. "BoxCar2D". `boxcard2d.com`

[51] Werner G. M. & Dyer M. G. "Evolution of communication in artificial organisms". *Artificial life II*. 1991. p. 659-687.

[52] Wilson S. W. "Knowledge growth in an artificial animal". *Adaptative and Learning Systems*. Springer US. 1986. p. 255-264.

[53] Wolfram S. "A new kind of science" Vol. 5. *Champaign: Wolfram media*. 2002.

[54] Wolfram S. "Statistical mechanics of cellular automata". *Reviews of Modern Physics*. 1983.

CHAPTER 4

# Vivarium

## 4.1 Concept

Vivarium est un outil de vulgarisation, développé en collaboration avec Camille Rattoni pour son Master d'Arts et Design à la HEAD, dans lequel l'utilisateur observe un environement clos dans lequel évoluent deux types de créatures: un groupe d'herbivores et un groupe de carnivores. Le but de chaque créature est de survivre le plus longtemps en se nourrissant. Chaque créature démarre avec une espérance de vie fixe, celle-ci étant allongée à chaque repas. Les herbivores se nourrissent des plantes présentes en quantité fixe sur la carte (à chaque plante dévorée une nouvelle est générée à des coordonées aléatoires) et les carnivores se nourrissent des herbivores.

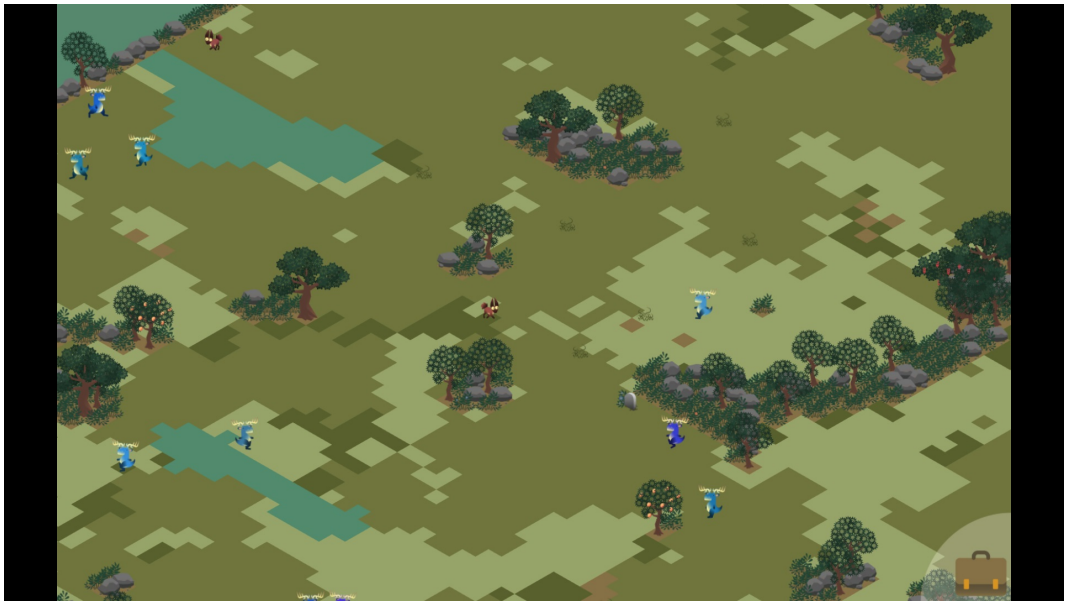


FIGURE 4.1 Capture d'écran de Vivarium.

Chaque génération démarre avec un nombre fixe d'individus de chaque espèce, une fois que tout les individus sont morts, une nouvelle génération d'individus est générée et la simulation reprend. Le terrain est une surface plane, carrée et garnie d'obstacles générés aléatoirement en ilôts. Chauqe individu est doté d'un réseau de neurones qui dicte sont comportement. Les créatures

avancent en permanence et dévorent leur nourriture lorsqu'elles rentrent en contact avec elle (en s'arrêtant quelques secondes au passage), les sortties du réseau de neurones indiquent si l'individu doit changer de direction. Les réseaux de neurones s'améliorent génération après génération par algorithmes génétiques à partir des individus les plus performants, la *fitness* étant la durée de vie de la créature, le but est que les individus deviennent de plus en plus performants pour atteindre leur nourriture.

Le principe consiste à suivre le développement des créatures au comportement aléatoire au départ jusqu'à voir émerger un comportement intelligent, pour cela, l'utilisateur possède différents outils qui lui permette d'identifer et de différencier les créatures entre elles. Un carnet fournit les informations sur chaque individus et sur la génaration en cours et permet de prendre en note ses observations. Un sablier permet d'accélérer ou de ralentir la vitesse de la simulation pour arriver à un stade plus avancé ou pour observer les interactions entre les créatures. Une paire de jumelles permet de retirer tout les éléments graphiques pour avoir un vision plus claire de l'environnement. Il est aussi possible de mettre fin immédiatement à la génération en cours ou de sauvegarder l'état en cours pour y revenir plus tard. La souris permet de déplacer la caméra dans les limites de l'environnement et de zoomer, un clic sur une créature permet de garder le focus de la caméra sur celle-ci.

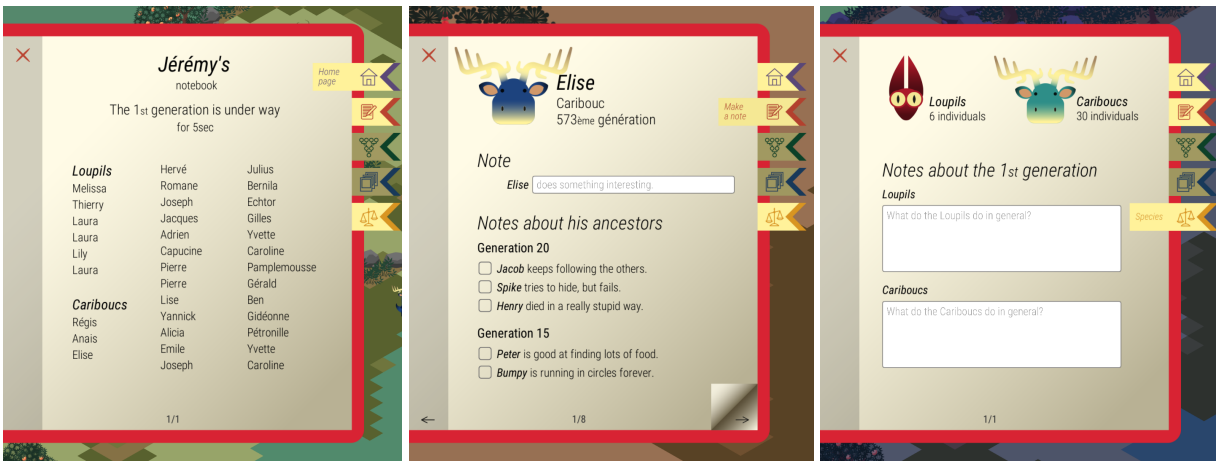



**FIGURE 4.2** Le carnet de Vivarium.

Plusieurs éléments visuels n'ont pas d'impact sur la simulation: la couleur des individus et celle de l'herbe par exemple. Un cycle jour/nuit est aussi présent mais n'affecte pas les créatures. Les différents obstacles (les arbres, les rochers ou les points d'eau) ont le même impact: ils bloquent le déplacement et la vision des créatures. Les créatures n'ont pas besoin de boire, l'eau n'a aucune autre utilité que celle de servir d'obstacle et les créatures n'ont pas d'autre besoin que celui de se nourrir, être dévoré par un carnivore est la seule action pouvant réduire l'espérance de vie d'un individu. Les carnivores ne se dévorent pas entre eux et les créatures ne constituent pas un obstacle pour les autres. Les créatures ne se reproduisent pas en cours de génération, de la même manière les réseaux de neurones ne s'améliorent pas en cours de génération. Les créatures de la même

espèce possèdent les même caractéristiques (mis à part leur réseau de neurones), les créatures se déplacent à la même vitesse et prennent le même temps pour se nourrir quelle que soit son espèce. Les carnivores sont présents en moins grand nombre que les herbivores mais leurs repas allongent plus leur durée de vie que ceux des herbivores. Lorsqu'une créature rencontre un obstacle elle repart automatique dans la direction opposée.

## 4.2 Post-mortem

Le concept de Vivarium est assez simple mais il a demandé beaucoup de réflexion, il fallait en effet un algorithme mimant le vivant, qui soit amusant à observer et qui permette un comportement relativement complexe. Parmi les méthodes retenues, un concept de jeu présentant le *Particle Swarm* a longtemps été envisagé. Il s'agissait de faire s'entraîner un essaim à accomplir une tâche (un jeu vidéo dans les premiers croquis) et de pouvoir faire varier les paramètres de la simulation de manière ludique en personalisant son essaim.

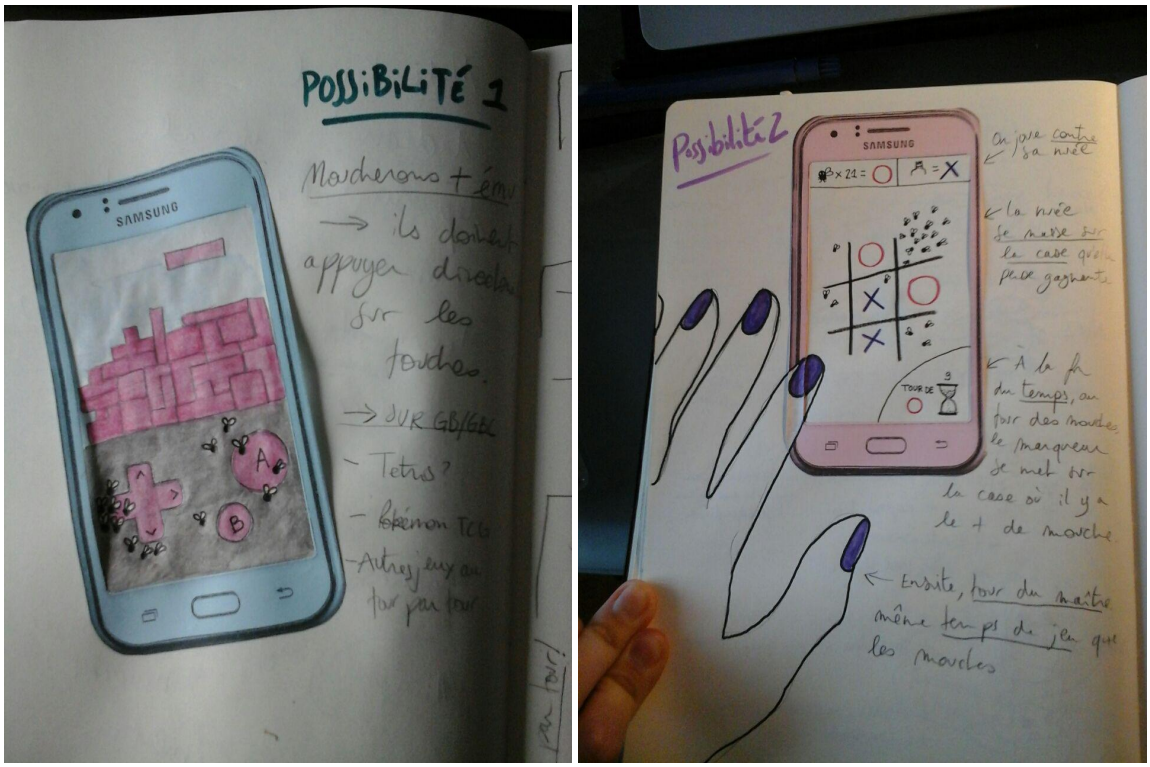



**FIGURE 4.3** Croquis d'un concept de jeu vulgarisant le *Paricle Swarm*.

Plusieurs raisons ont amené à abandonner cette idée:

- Il a été difficile de trouver une tâche que pourrait résoudre un algorithme de *particle swarm* et qui soit amusant à observer. Le but est de passer les itérations à un rythme compréhensible pour un regard humain et donc passer l'algorithme au ralenti aurait perturbé n'importe quelle tâche se déroulant en temps réel.

- Inclure trop d'éléments de jeu nous paraissait contre-productif par rapport à l'objectif initial: en "déguisant" les éléments de vulgarisation le danger est qu'au final l'utilisateur n'arrive plus à faire le lien avec l'IA et n'arrive donc pas à transposer ce qu'il a appris lorsqu'il sera confronté à du *Machine Learning*.
- Le *particle swarm* n'est pas très populaire parmi les algorithmes actuels (contrairement aux réséaux de neurones qui ont été retenus) et fonctionne assez différement des algorithmes couramment utilisés, beaucoup de connaissance transmises par le jeu aurait donc été inutiles.
- L'apprentissage du *particle swarm* n'est pas très explicite visuellement et le jeu aurait manqué d'impact si l'utilisateur était passé à côté du côté "automatique" de l'apprentissage.

Nous nous sommes alors tourné vers des concepts se rapprochant beaucoup plus de la vie artificielle, plusiuers exemples peuvent être trouvés, et ceux représentant diverses créatures vouées à effectuer une tâche ont retenu notre attention.

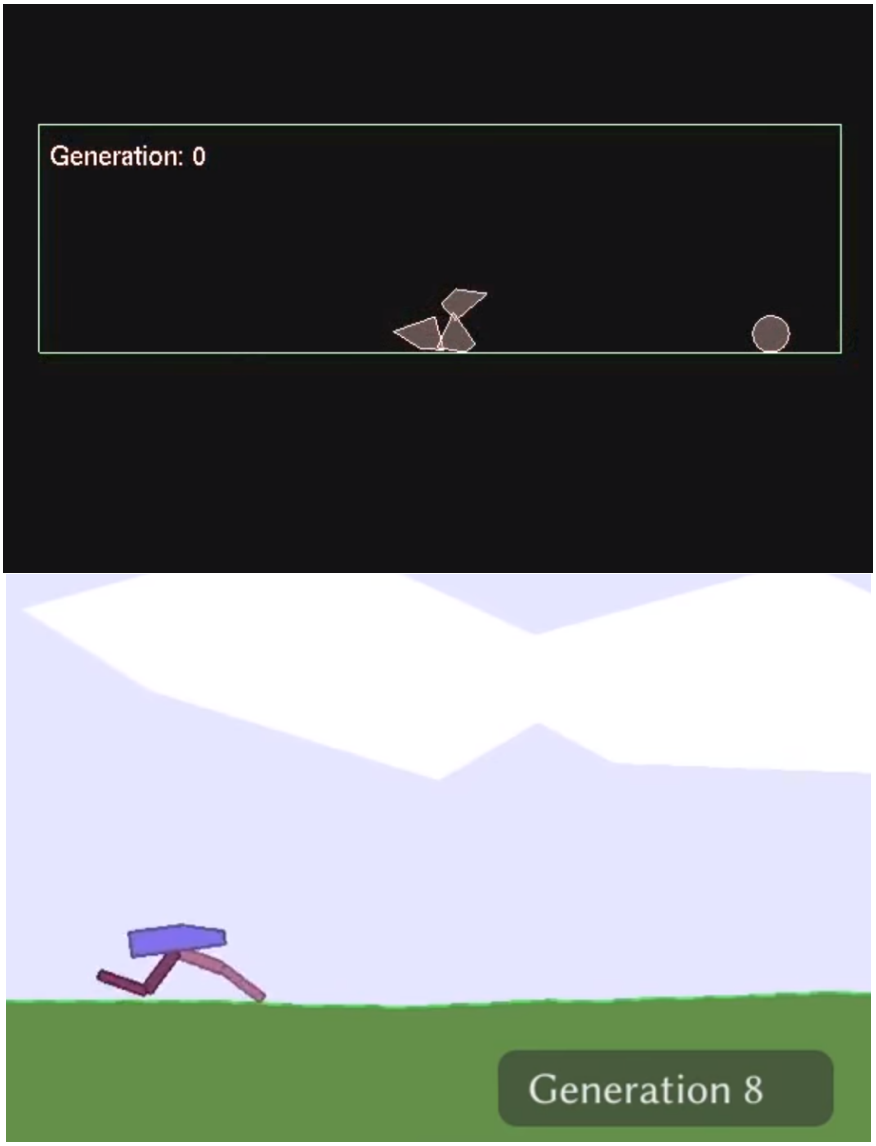



**FIGURE 4.4** Exemple d'application d'algorithmes génétiques ( [19] et [20]).

Nous avons donc choisi un outil beaucoup plus passif basé sur l'observation. L'idée d'inclure deux types de créatures fut assez naturelle, elle permet de montrer les différences d'apprentissage (la tâche des carnivores étant plus compliquée que celle des herbivores) et donne lieu à des situations plus ludiques à suivre pour un spectateur. La première version de Vivarium est entièrement en Java, le monde est représentée sous forme d'une grille, les individus se déplacent de manière continue sur cette grille. Cette version ne comporte qu'une seule espèce qui se dirige aléatoirement dans l'environnement. Les obstacles sont générés avec un algorithme de *Cellular automata* afin de créer des ilôts irréguliers qui donnent un rendu crédible pour un environnement naturel (les couleurs bleu et gris sont uniquement esthétiques et n'affectent pas la simulation). Les plantes qui servent de nourriture aux herbivores est déjà présente, les crétures la ramassent instantanément lorsqu'elles les rencontrent. Nous sommes encore loin de la simulation (et encore plus loin d'une simulation contrôlée par une IA) mais les bases de l'environnement sont là. Il est intéressant de noter que bien que les "créatures" soient réduites à des formes géométriques et qu'elles se déplacent de manière aléatoire, on ressent le phénomène d'humanisation décrit par Simmel et Heider [18], on se prend d'affection et on imagine une intention derrière les actions des créatures.

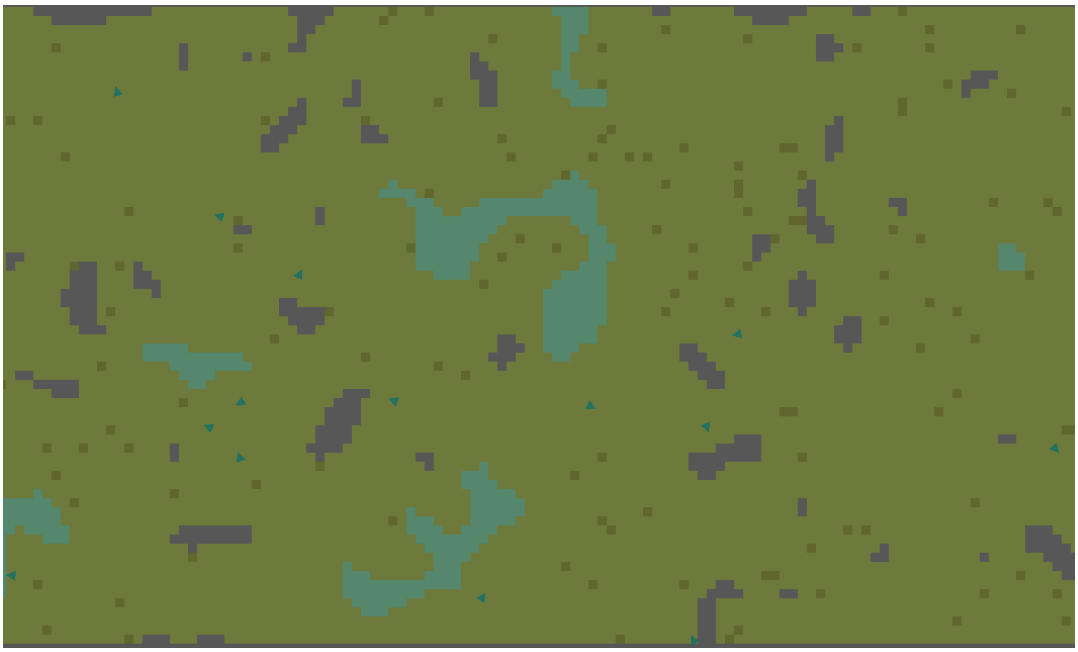


**FIGURE 4.5** Première version de Vivarium.

La tâche suivante consistait à doter les créatures d'un véritable comportement, pour cela j'ai commencé par créer un champ de vision. Chaque créature possède une vision à 360 degrés de n cases de rayon, on effectue ensuite un *raycasting* sur chaque case du cercle extérieur champ de vision pour déterminer les cases dont la vision est bloquée par un obstacle. J'ai ensuite codé une IA simpliste pour les créatures: elle se dirige vers la source de nourriture la plus proche dans leur champ de vision. J'ai aussi rajouté une deuxième espèce, identique à la première, si ce n'est qu'elle considère les herbivores comme sa source de nourriture. On approche alors d'une simulation et on

entre dans le vif du sujet puisque c'est la partie intelligence qui sera la prochaine étape.

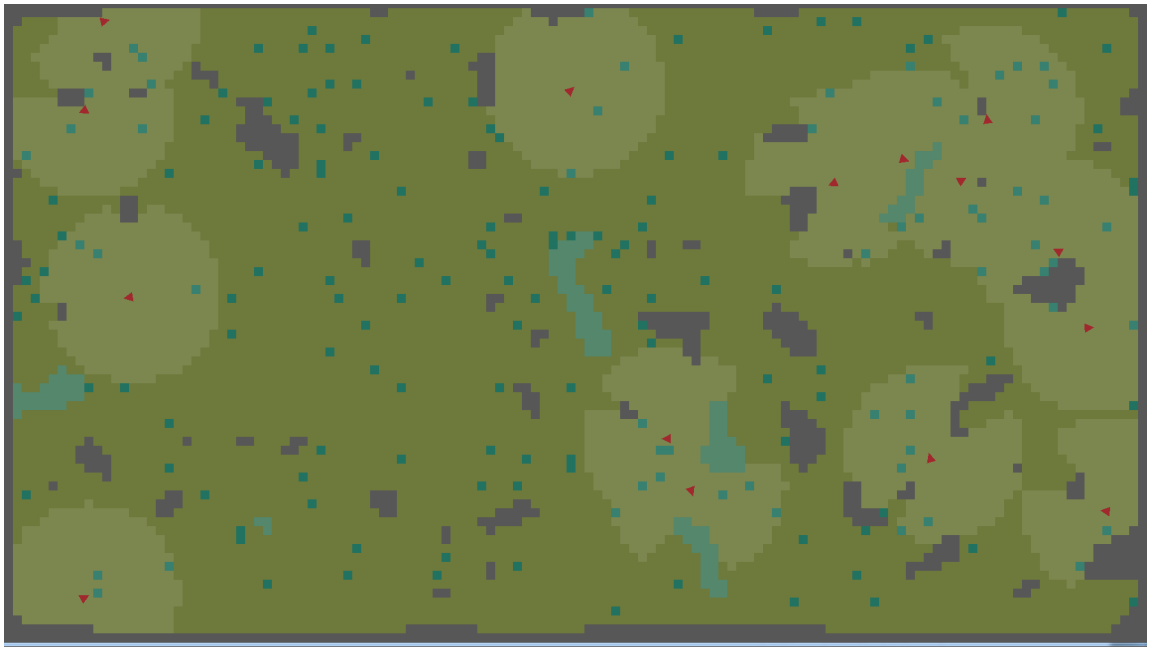


**FIGURE 4.6** Deuxième version du prototype de Vivarium avec les champs de vision dessinés à l'écran.

La première intention de Vivarium était de faire un réseau sophistiqué où les créatures apprendraient à distinguer leurs prédateurs de leur nourriture qu'avec la couleur, les entrées seraient donc les composantes RGB de l'entité la plus proche. L'apprentissage aurait permis de distinguer la nature des entités et le réseau aurait dû composer avec les légères variations de couleur entre les individus. Cette méthode a cependant posé plusieurs problèmes. Tout d'abord le temps d'apprentissage aurait été assez long et cela aurait donc nuit à l'expérience. De plus, cela aurait impliqué que les créatures n'aurait pas souvent une source de nourriture à portée de vue et cela aurait parasité l'apprentissage, pire encore le fait de se retrouver alternativement dans deux situations différentes aurait rendu difficile de séparer les entraînement des deux compétences (trouver de la nourriture et échapper à ses prédateurs). Enfin les composants RGB ne sont pas forcement connus du public cible et par conséquent cela aurait rajouté des notions à expliquer.

Je m'en suis donc tenu a un réseau plus simple à trois couches, 4 neurones d'entrée, 4 neurones dans la couche cachée et 2 neurones de sortie. Un herbivore perçoit deux choses: sa source de nourriture la plus proche (s'il y en une à portée de vue) et son prédateur le plus proche, pour chacune de ses deux entités il perçoit la distance qui les sépare et l'angle avec sa direction actuelle. La sortie contient des neurones qui s'activent pour savoir si la créature doit tourner à gauche ou à droite. Les poids des arcs du réseau sont initialisés aléatoirement et le seuil d'activation des neurones est de 0.5. Après chaque génération, on classe les individus selon leur perfomance (la *fitness* étant simplement le temps qu'ils ont survécus) et on créé des paires d'individus, la probabilité qu'un individu soit choisi est proportionnel à sa *fitness* par rapport à la moyenne du groupe. On applique ensuite un *crossover* entre les deux génomes des individus choisi, on mute aléatoirement le génome obtenu

pour obtenir un nouveau génome qui sera celui du nouvel individu. On répète le processus pour obtenir une population complète et la simulation peut alors reprendre, il s'agit de la seule opération qui modifie le réseau de neurones des créatures, il n'est pas altéré pendant la simulation.

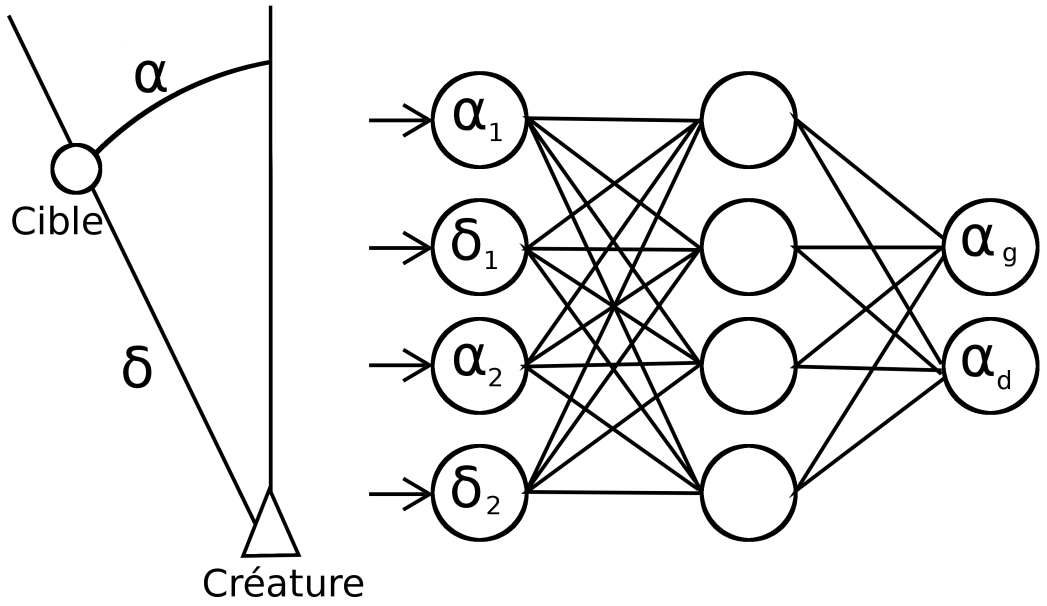



**FIGURE 4.7** Réseau de neurones d'une créature de Vivarium.

À ce stade, faire fonctionner correctement la simulation a demandé de nombreux ajustements de plusieurs paramètres. Tout d'abord l'espérance de vie intiale des individus, en étant trop courte elle rend quasiment impossible l'apprentissage des premières génération, trop longue elle rend difficile de trier les bons individus et allonge inutilement les générations. La vitesse de course des individus a aussi été déterminante: au départ les carnivores étaient plus rapides que les herbivores, cela me semblait un avantage nécessaire pour pallier au fait que les carnivores doivent se diriger vers une source de nourriture mobile. L'idée était que si un carnivore se dirigeait correctement vers sa cible il devait être récompensé, ce qui ne serait pas le cas s'il se faisait distancer. Toutefois, il s'est avéré que cet avantage était beaucoup trop conséquent et la population d'herbivore se faisait alors éradiquer rapidement, finalement en courant à la même vitesse que leur proie, les carnivores ne sont pas désavantagés puisqu'ils peuvent profiter du moment où les herbivores s'arrêtent pour manger. Ce fut une découverte imprévue qui donne lieu à des comportements intéressants à observer. Concernant les herbivores, la quantité de plantes comestibles disponible sur la carte et la durée rajoutée à l'espérance de vie à chaque repas ont été des paramètres importants et délicats à ajuster. En effet, si les plantes sont trop nombreuses ou qu'elles fournissent trop d'énergie, alors les herbivores entrent rapidement dans un cycle où ils ne meurent jamais et la *fitness* augmente alors de manière disproportionnée. Mais si les plantes sont trop peu nombreuses, alors il arrive trop souvent que les herbivores n'aient aucune plante à portée de vue et cela fausse l'apprentissage. La disposition du terrain joue aussi un rôle important, en soi les obstacles freinent l'apprentissage mais sont nécessaires pour apporter un peu de vie et de vraissemblance au paysage, il s'agissait

donc de créer un paysage esthétique sans trop impacter la simulation. Pour cela j'ai choisi d'éviter les obstacles isolés et de fonctionner par îlots, toutefois il fallait que ces îlots soient irréguliers mais qu'ils ne comportent pas trop de recoin susceptibles de coincer les créatures. Enfin, la quantité d'individus dans une population par rapport à la suface de terrain était certainement l'élément le plus déterminant. En effet, si la population d'herbivore n'est pas très importante (mis à part pour la lisibilité de la simulation), la quatité de carnivores par rapport aux herbivores est critique. En effet ma prmeière idée était de placer autant de carnivores que 'herbivores dans le vivarium mais il s'est avéré que cela entrainait une extinction rapide des herbivores sans aucun apprentissage puis qu'ils tombaient sur leurs proies rapidement par hasard. J'ai dû donc réduire le nombre de carnivores et cela ralenti considérablement l'apprentissage (qui est déjà plus lent que celui des herbivore à cause de la complexité de leur tâche), il a donc fallut trouver un équilibre.

L'étape suivante consiste à porter le jeu sur le moteur de jeu Unity, la version actuelle très épurée a été très pratique pour effectuer rapidement des tests mais une version plus esthétique permet de toucher un public plus large. Unity fonctionnant en C#, qui est très proche de Java, le portage n'a pas posé problème, toutefois l'affichage se fait maintenant en isométrique, j'ai donc codé un *renderer* qui affiche la grille représentant le terrain en isométrique, les obstacles et les créatures sont alors représentés par des sprites en 2D.

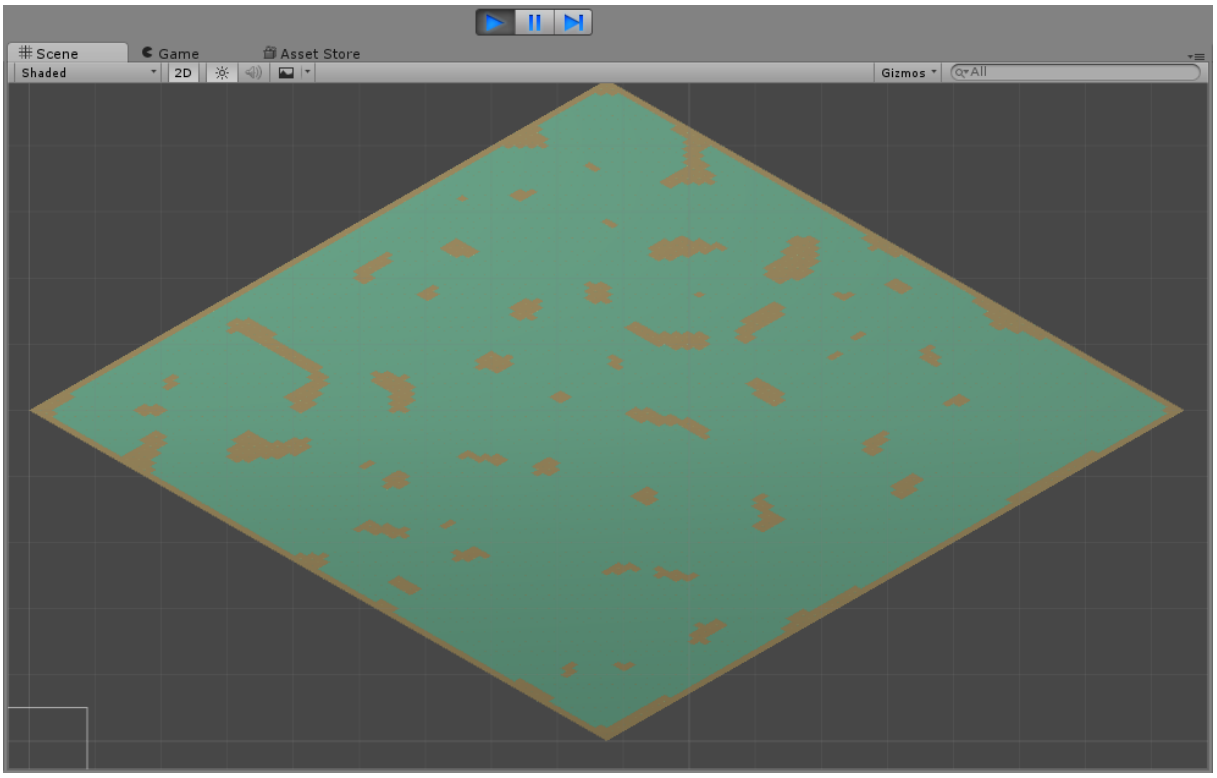



**FIGURE 4.8** L'affichage du terrain en vue isométrique.

J'ai aussi rajouté les animations qui se jouent lors de différentes actions (manger, mourir et marcher) ainsi qu'un menu d'accueil. Dans le processus, j'ai aussi abandonné le modèle client/serveur et porté tout le moteur du jeu sous Unity. Ce modèle a été choisi à la base dans

l'optique de faire tourner une immense simulation côté serveur et de pouvoir s'y connecter depuis un client sur une machine modeste mais finalement la taille de l'environement n'est pas très importante et ce modèle complique un peu l'utilisation.

**FIGURE 4.9** Quelques sprites de Vivarium par Camille Rattoni.

**FIGURE 4.10** L'écran d'accueil de Vivarium.

L'étape finale consiste à rajouter les outils de Vivarium, il s'agit d'une étape assez longue mais peu intéressante à détailler ici puisqu'il s'agit d'outils simples à implémenter. Les outils sont présentés sous forme de roulette en bas de l'écran, disponible lorsque l'on clique sur la saccoche. C'est l'outil "jumelles" qui a demandé le plus de travail puisque j'ai dû coder un nouveau renderer où les animations ne sont pas jouées, les couleurs du terrain sont modifiées et les éléments de la simulation ont des sprites alternatifs. Ce mode est toutefois un peu sous-exploité, l'objectif initial était de créer un mode dans lequel l'utilisateur aurait accès à toutes les variables de la simulation et pourrait voir les réseaux de neurones s'activer en temps-réel. Le but était de fournin assez

d'éléments à l'utilisateur pour qu'il puisse se passer d'explications mais a été abandonné suite aus premiers retours utilisateur. Ces *features* pourraient faire l'objet de travaux futurs mais ces points seront discutés dans la suite de ce rapport.

**FIGURE 4.11** La vue simplifiée grâce aux jumelles.

Après de longs mois de réflexion et de développement, Vivarium est désormais terminé. Le résultat est assez proche de ce que nous avions imaginé à la base et la simulation donne de bons résultats, les comportements qui émergent sont parfaits pour une démonstration: après plusieurs générations les herbivores commencent à être performants. Les carnivores prennent plus de temps mais finissent par être des prédateurs hors pair, ce qui constitue un excellent point pour montrer les différences d'apprentissage selon la complexité de la tâche. On observe même quelques simili-stratégies innatendues en cours d'apprentissage pour trouver de la nourriture (surtout de la part des carnivores qui passent par différents stades) ce qui donne un côté très ludique et donne des point d'accroche pour des explications lors de la démonstration. Malgré les quelques *features* abandonnées, d'un point de vue développement Vivarium est un succès.

CHAPTER 5

# Résultats

## 5.1 Expériences

Plusieurs tests ont été effectués pendant la période de développement de Vivarium, dès la fin du portage sous Unity, nous avons en effet été invité à participer au stand de jeux indépendants suisses de *SwissGames* à la Gamescom 2016 de Cologne, une exposition majeure dans le monde du jeu vidéo, avec d'autres projets étudiants. Vivarium n'étant pas encore terminé à cette période, nous avons recueilli les retours de manière informelle pour aiguiller nos choix. Le public était majoritairement composé de journalistes (spécialisés pour la plupart mais pas seuleument) ainsi que des joueurs adultes, une minorité était composé d'un public hétéroclite (la Gamescom 2016 ayant accueilli plus de 345 000 visiteurs). Peu de temps après nous avons pu exposer Vivarium à d'autres évènements, notamment la G4LAN de Genève où le public était ici composé exclusivement de jeunes adultes joueurs.

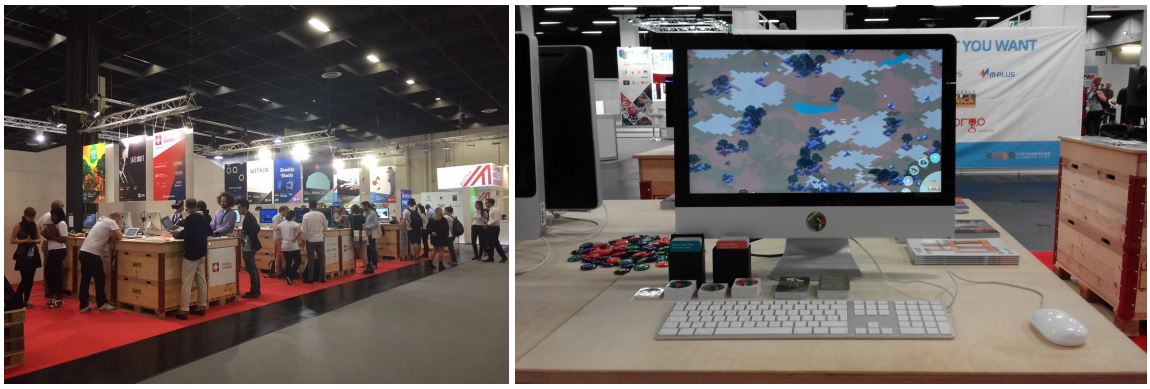



FIGURE 5.1 Le stand de Vivarium à la Gamescom.

La Gamescom est un salon international annuel prenant place à Cologne pendant 5 jours. Il réunit les plus grand acteurs du jeu-vidéo: constructeurs, studios, éditeurs et journalistes pour des annonces et exposition des projets à venir pour l'année. Le salon est divisé en deux parties: un secteur *buisness* (réservé aux journalistes, dans laquelle Vivarium était exposé) et un secteur convention. L'éddition 2016 a rassemblé plus de 700 exposants de 88 pays différents. La G4LAN quant à elle est une convention de jeux en réseau de deux jours au Palladium de Genève, la pre-

mière éddition en 2016 (à laquelle a participé Vivarium) a rassemblé près de 350 joueurs. En marge des tournois et sessions de jeux, le salon encourage les acteurs locaux du jeu vidéo à présenter leurs jeux, c'est à cette occasion que Vivarium a été présenté. Dans les deux cas le thème des expositions était le jeu vidéo et non la vulgarisation scientifique, le public a donc des attentes différentes mais ces expositions n'étant pas ouvertes (uniquement sur inscription et payante dans le cas de la Gamescom), on suppose que les personnes présentes sont intéressées par le jeu en général, qu'il soit scientifique ou non.

C'est donc à ces deux occasions que j'ai pu collecter les premiers retours sur le potentiel de vulgarisation de Vivarium. Comme précisé plus haut, ces retours ont été recueillis de manière informelle et ne feront donc pas l'objet d'une analyse chiffrée et détaillée, il s'agit d'un retour sur expérience qui a aiguillé mes choix sur la suite du projet. Vivarium a été testé par plusieurs dizaines de personnes, pendant des durées variables (allant d'une minute à un quart d'heure) et deux constats se sont imposés pendant ces démonstrations:

- Plusieurs critiques ont été émises au niveau de l'interactivité avec le jeu, en effet le carnet d'observation est finalement peu utile vu la durée du jeu et la durée de vie de chaque individu. La prise de note se révèle fastidieuse et supperflue et est vite abandonnée par l'utilisateur, ce qui le rend assez passif dans son observation de l'écosystème. Ces observations doivent cependant être remises dans leur contexte: puisqu'il s'agissait d'expositions dédiées au jeu-vidéo, les attentes du public étaient fortement orientées vers une expérience de jeu et moins d'observation. Toutefois, la simulation en elle-même a généré beacoup d'enthousiasme: la plupart des utilisateurs étaient curieux de constater l'émergence d'un comportement intelligent sont restés pour observer le déroulement de la simulation et ont été très curieux d'en apprendre plus les mécanismes.
- Ce qui nous amène au deuxième constat: pour que l'expérience fonctionne correctement, la présence d'un accompagnateur capable d'expliquer tout les fonctionnements de la simulation s'avère indispensable pour comprendre les tenants et les aboutissants de Vivarium et donc de profiter de l'expérience. Sans cette présence, Vivarium se révèle être une expérience amusante mais rapidement consommée et perd tout son potentiel de vulgarisation. Cela nous a permit de nous rendre compte que les *features* prévues pour l'outil "jumelles" seraient insuffisantes pour qu'un utilisateur puisse se débrouiller seul et réaliser le potentiel de vulgarisation voulu par Vivarium.

Après avoir terminé le développement de Vivarium, il a donc fallut revoir certains axes pour mieux coller aux objectifs fixés. C'est à l'occasion de l'évènement "Élargis tes Horizons" 2017 auquel sera présent Vivarium que j'ai décidé de transformer un peu l'expérience et de faire de Vivarium le coeur d'un atelier dédié à la vulgarisation du *Machine Learning*. "Élargis tes Horizons" est un évènement de sensibilisation des métiers scientifiques pour les jeunes filles qui s'est tenu en novembre 2017 à l'Univseristé de Genève. 450 inscrites ont pu participer à des ateliers destinés à les sensibiliser aux carrières scientifiques, parmi ces ateliers plusieurs activités

en lien avec l'informatique ont été présentées et Vivarium en faisait parti. J'ai donc dû concevoir un atelier de 35 minutes dédié à un public jeune, sans connaissances requises, avec pour but de vulgariser le ML. 15 à 20 minutes sont nécessaires pour observer correctement la simulation de Vivarium, j'ai donc prévu de l'inclure dans un atelier pratique destiné à introduire toutes les notions nécessaires pour comprendre les mécanismes de Vivarium.

L'atelier était divisé en 4 parties:

- Une première partie dont le but est d'introduire la notion d'algorithme et d'intelligence artificielle. L'angle choisi a été de faire jouer à un jeu à deux enfants: l'un qui joue normalement et l'autre qui suit une stratégie écrite. J'ai choisi le Puissance 4 car il s'agit d'un jeu aux règles simples que tout le monde connait mais pour lequel il est assez aisé de trouver une stratégie forte mais exploitable ce qui permet de mettre en exergue les limites d'une IA sans apprentissage et donc la nécessité du ML.
- Une description d'un problème basique que l'on résout avec un algorithme de ML pour expliquer ce qu'on entend par apprentissage lorsque l'on parle de machines.
- Introduction très courte aux mécanismes de Vivarium: les règles de la simulation, le but, le fonctionnement des créatures, cela permet de ne pas passer trop de temps à expliquer le jeu. Une fois lancé, les enfants peuvent prendre en main le jeu et savent déjà quoi observer, cette introduction hors du jeu permet de les laisser un instant seul avec le jeu dans le but de se familiariser avec l'interface.
- Vivarium en lui-même, comme indiqué au-dessus, la simulation dure une quinzaine de minutes, avec un accompagnateur par groupe de 3 ou 4. En effet, la simulation était plutôt passive, c'est un excellent moment pour les questions avec Vivarium en tant qu'appui visuel. Ce quart d'heure permet une vraie discussion (même avec des groupes un peu plus importants) avec ce qui se passe dans la simulation comme base. De plus, le côté esthétique de Vivarium retient bien l'attention, surtout chez des enfants.

**FIGURE 5.2** Déroulement de l'atelier Vivarium à "Élargis tes horizons".

## 5.2 Présentation des retours utilisateurs

J'ai donc présenté 3 fois cet atelier à trois groupes différents, soit 31 filles de 11 à 14 ans auxquelles j'ai soumis un questionnaire dans le but de compulser les retours. Mis à part l'âge toutes les réponses demandaient une réponse sur une échelle de 0 à 3: "pas du tout", "un peu", "moyennement" ou "beaucoup". 3 questions portaient sur le passif des enfants pour classifier les réponses: l'âge, les connaissances en informatiques et les connaissances sur le *Machine Learning*.

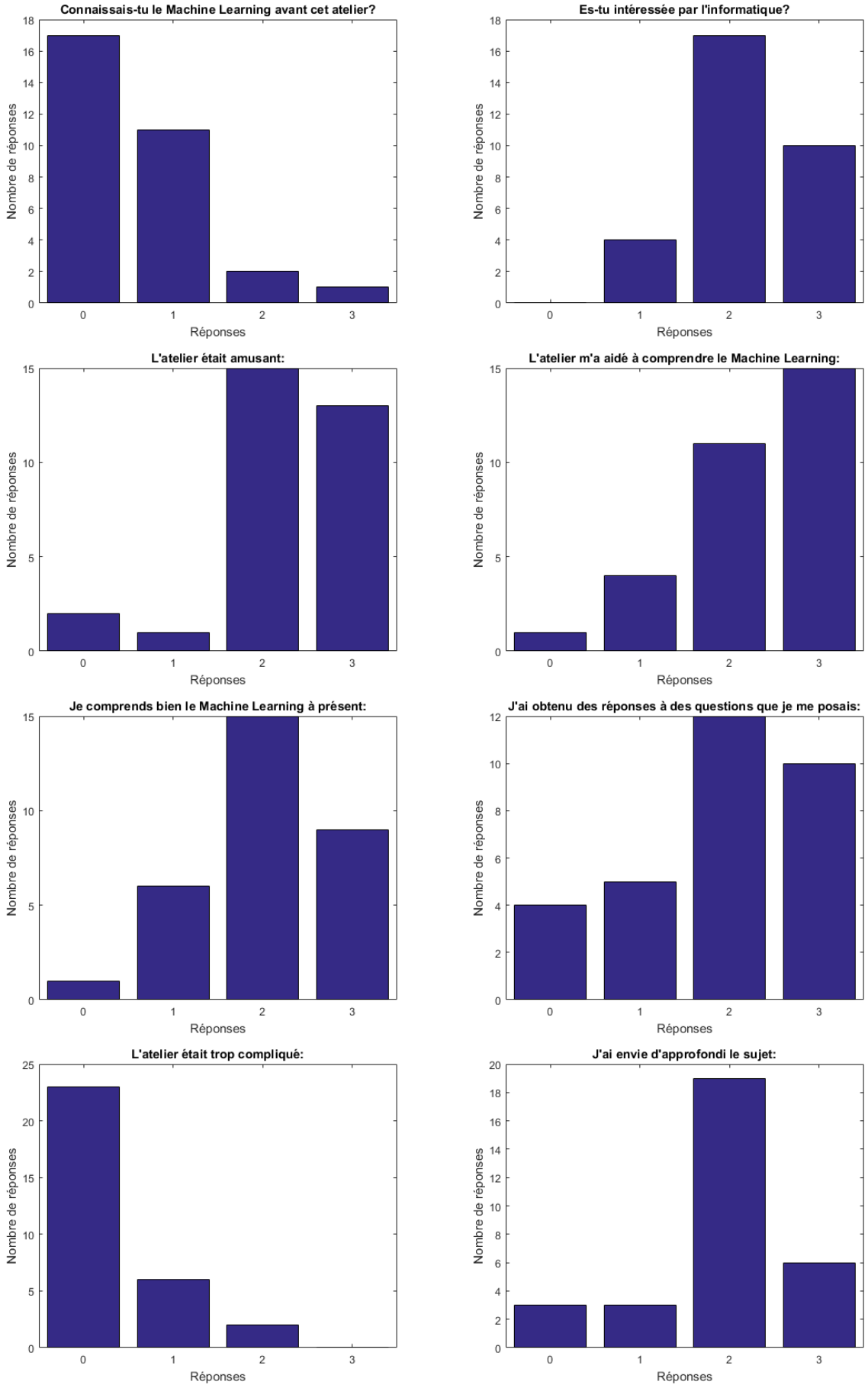



**FIGURE 5.3** Condensé des réponses au questionnaire de l'atelier.

Les deux premières questions sont les questions concernant le passif des enfants. On peut voir que si la plupart sont intéressées par l'informatique (ce qui était attendu puisque les ateliers demandaient une inscription préalable) mais la plupart ne connaissaient pas (ou peu) le *Machine Learning*. Cela peut s'expliquer par l'âge puisque le ML est une branche de l'IA assez complexe et même si les enfants entrent en contact avec, il est nécessaire d'avoir de bonnes connaissances pour se poser des questions. On peut voir en observant les réponses aux questions de manière indépendante que l'atelier a plutôt été un succès: la majorité des enfants ont trouvé l'atelier amusant et surtout instructif. Cependant, même si la majorité ont répondu que l'atelier les avait aidé à comprendre le ML, elles admettent ne pas comprendre totalement le MJ à la fin. Vivarium à pour but de se poser comme une ouverture pour comprendre les mécanismes du ML, donc c'est encore une fois une réponse que l'ont attendait.

Une question intéressante est de savoir si l'atelier était trop compliqué: en effet le ML est un sujet compliqué et Vivarium n'était pas à la base conçu spécialement pour les enfants, un objectif des activités de l'atelier était donc de simplifier le propos et d'accompagner la réflexion. C'est un objectif qui semble réussi puisque les réponses sont presque unanimes sur la complexité de l'atelier, toutefois on relativiser les réponses à cette question puisqu'il est facile de passer à côté du propos de la simulation (et de s'arrêter à l'observation des résultats) et donc de ne pas comprendre en quoi l'atelier était compliqué. Les réponses à la dernière question sont encourageantes puisque l'atelier a plutôt donné envie d'approfondir la question. C'est un résultat important puisqu'il est impossible d'expliquer complètement le fonctionnement du ML en 30 minutes, par conséquent le but de la vulgarisation est non seuleument de données les clés pour comprendre des explications plus complexes, mais aussi d'intéresser suffisement pour donner envie d'approfondir la question plus tard.

On peut aussi mettre ces différentes réponses en relation, en dessinant une surface où le point $(x, y)$ est le nombre d'enfants ayant répondu $x$ à la première question et $y$ à la deuxième, comme on s'intéresse à l'influence de la première réponse sur la deuxième, on normalise les résultat par rapport au premier choix, c'est-à-dire qu'on divise le résultat par le nombre d'enfants ayant répondu $x$. Cela nous permet par exemple de distinguer les réponses aux question par rapport à l'âge. Voici ce qu'on obtient en observant l'nfluence de l'âge sur la réponse à la question "L'atelier était trop compliqué":

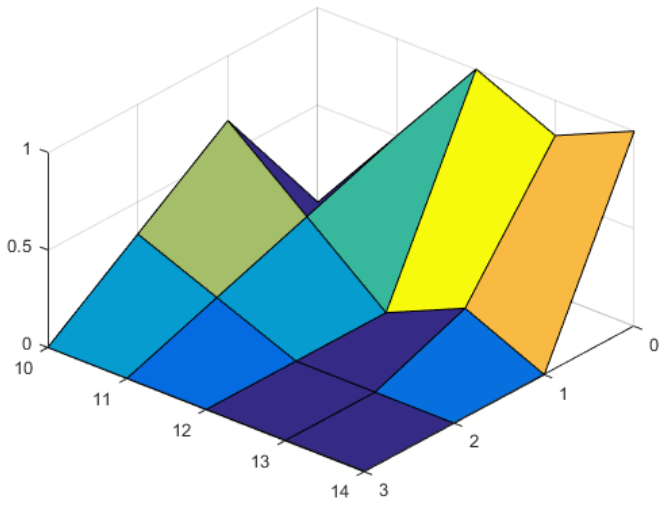



**FIGURE 5.4** Influence de l'âge sur la réponse à la question "L'atelier était trop compliqué".

Il n'y avait que 6 réponses "un peu" et 2 "moyennement" à cette question, ce qui semble peu mais on remarque que l'âge a une grande influence à la réponse à cette question puisque les 10-11 ans ont été les seules à répondre "moyennement" et la classe des 10 ans contient presque toutes les réponses "un peu". Même si le nombre d'échantillons dans ces classes est assez faible (3 enfants de 10 ans et 6 de 11 ans), on peut avancer l'hypothèse que 11-12 ans est la limite d'âge pour présenter cet atelier et que les enfants plus jeunes trouvent les thèmes abordés trop compliqués, ce qui est compréhensible. Cette tendance se confirme lorsqu'on étudie l'influence de l'âge sur la réponse à la question "L'atelier m'a aidé à comprendre le Machine Learning":

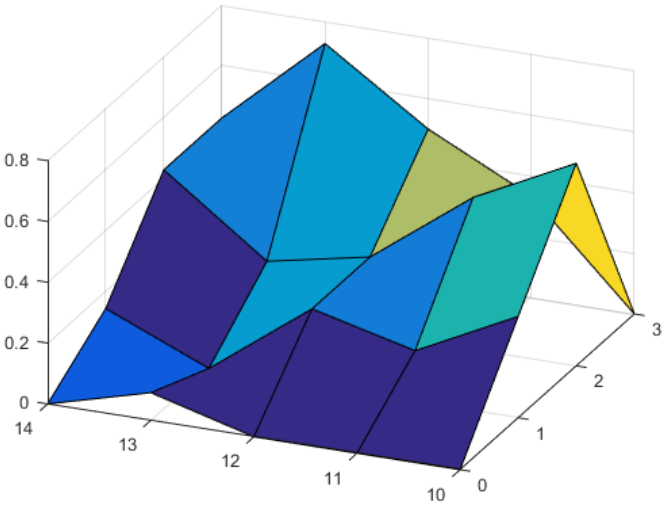



**FIGURE 5.5** Influence de l'âge sur la réponse à "L'atelier m'a aidé à comprendre le Machine Learning".

On remarque que l'atelier a bien fonctionné sur la tranche 13/14 ans (d'après le pic dans le fond) et moins sur les 10/11 ans (d'après le pic sur la droite). En continuant sur cette lancée et en mettant

en relation d'autres réponses, on s'apperçoit cependant de ce qui peut être un biais important. En effet, souvent les réponses aux différentes questions sont souvent très liées: globalement le retour est négatif ou positif et on trouve peu de nuances dans les réponses aux différentes questions, ce qui se traduit par une diagonnale sur la surface. Cela indique qu'un questionnaire immédiat après l'expérience n'est pas forcement la meilleure méthode pour analyser précismeent les sentiments des sujets.

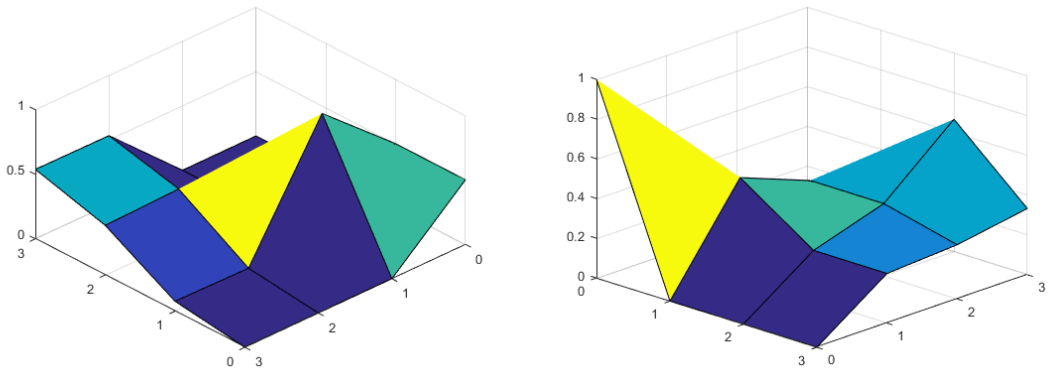



**FIGURE 5.6** Relations entre les réponses des questions 4, 5 et 6.

Le questionnaire comportait un espace libre dédié aux remarques diverses sur l'atelier. Ce cadre était remplit sur 15 des 31 questionnaires, la plupart des remarques paraphrasait les réponses du questionnaire mais portent aussi sur ce qu'elles ont apprécié le plus dans l'atelier. Sans surprises les parties de Puissance 4 contre une IA simulée et la partie des Vivarium sont les parties qui ont rencontré le plus de succès, ce qui conforte l'idée que le jeu peut être un bon outil de vulgarisation. De plus, ces parties étaient les plus interactives et nous avons eu beaucoup de questions pendant ces deux parties, plutôt que pendant les parties explicatives, ce qui peut s'expliquer par le fait qu'avec un support visuel sous les yeux, il est plus facile d'exprimer et de formuler ses questions.

CHAPTER 6

# Conclusion

L'objectif de ce projet était de concevoir un outil de démonstration du Machine Learning, dans le but d'en vulgariser les mécanismes de manière ludique. Si la littérature en matière de vulgarisation du ML est assez pauvre, il existe de nombreuses applications qui possèdent naturellement un potentiel ludique qui ont servi d'inspiration. Le projet aura toutefois prit du temps pour mûrir, de nombreuses idées ont été couchées sur papier et la plupart n'ont pas été exploitées. Trouver un équilibre entre simplicité des mécanismes et potentiel ludique n'a pas été une chose aisée. Le développement n'aura pas posé beaucoup de problèmes et m'aura permit de perfectionner mes compétences en matière de Machine Learning, en effet même si les algorithmes appliqués dans Vivarium sont assez simplistes, il aura fallu compulser beaucoup d'études pour proposer un système simple mais qui reflète correctement les algorithmes plus complexes utilisés aujourd'hui.

Les plus gros problèmes rencontrés ont été de fournir un outil agréable à suivre, dans lequel l'utilisateur se sente impliqué malgré l'interactivité assez faible. Plusieurs pistes ont été testées pour donner à l'utilisateur un rôle plus actif comme le carnet ou les jumelles mais se sont révelées insuffisantes. Si le concept de Vivarium a peu évolué durant le développement, le concept d'atelier s'est greffé au projet par l'opportunité d'exposer Vivarium et corrige finalement plusieurs défauts du jeu d'origine. Au final, il s'agit d'un ajout très satisfaisant et qui a donné lieu a une expérience enrichissante et couronée de succès au vu des retours collectés.

Au final, Vivarium est plutôt un succès: les retours collectés sont bons et s'il est difficile d'en tirer une conclusion sur son caractère éducatif, il conforte la piste selon laquelle le jeu est un excellent outil d'exemple pour illustrer et expliquer le *Machine Learning*. Il m'aura en effet permi de retenir pendant 35 minutes l'attention d'un groupe d'enfants pendant des explications parfois techniques sur un sujet très abstrait, ce qui constitue déjà une réussite. La vulgarisation scientifique est un pari important à réussir pour la communauté scientifique et je pense que le jeu est une piste à exploiter pour le réussir. Néanmoins, il s'agit d'un investissmeent conséquent: lors de ce projet le temps nécessaire pour discuter avec les utilisateurs et revoir le concept du jeu a fortement été sous estimé. Le développement initial n'est au final qu'une part minime du temps à investir comparé aux allers-retours incessants entre test utilisateurs et analyse des retours qui permet d'avoir une expérience beaucoup plus adaptée au public, de fournir une expérience plus agréable et donc de

fournir un cadre idéal à la vulgarisation.

## 6.1 Pistes pour de futurs travaux

Comme inidqué plus haut, Vivarium n'est très interactif et l'orienter beaucoup plus vers un jeu a été envisagé, la principale objection qui a été retenue est que le principe est de laisser l'algorithme travailler seul pour faire émerger un comportement, une intervention humaine briserait le côté automatique de l'apprentissage. De plus, le ML travaille par itérations et l'apprentissage nécessite une répétition des situations ce qui n'est pas très adapté pour un jeu. Néanmoins, plusieurs idées pourraient être mise en application dans le futur:

- Enrichir le systèmes de génétique: perfectionner le réseau de neurones pour prendre en compte plus de paramètres ou ajouter des actions comme sprinter ou nager
- Ajouter des mutations physiques qui modifieraient les caractéristiques de l'individu comme la vitesse de course, l'endurance ou la couleur
- Donner plus d'ampleur au mode "jumelles", on pourrait ajuster les variables en cours de simulation ou voir les réseaux de neurones en temps réel
- Permettre à l'utilisateur d'interagir avec l'environnement, l'utilisation d'appeau vrtuel pourrait être une bonne idée, ou la possibilité de récompenser soit même une créature, augmentant ainsi sa *fitness*
- Pouvoir changer la manière dont la transition entre génération se fait afin de pouvoir choisir soi-même les couples ou créer soit même les mutations

Évidement toutes ces modifications altéreraient drastiquement l'apprentissage automatique et s'inscrirait plus dans une démarche de produire un véritable jeu et moins dans une démarche pédagogique mais ce n'est pas une piste à négliger dans la vulgarisation, un *s*erious game qui retient ses joueurs pendant de nombreuses heures à un grand potentiel pour transmettre un message.

Concernant les résultats obtenus par Vivarium, le but était de tester la possibilité de toucher un public non-spécialiste à l'aide d'un outil de vulgarisation ludique et les tests effectués s'inscrivent dans cette démarche mais une étude plus approfondie serait nécessaire pour vérifier les connaissances acquises après utilisation de Vivarium pour mesurer son impact mais ces tests s'éloignent du domaine des sciences informatiques et de mes compétences et pourraient être un sujet de travail futur. De plus, bien qu'anonymes, les réponses à un questionnaire immédiatement après expérience et dont les questions portent explicitement sur l'atelier peut être sujet à de nombreux biais (d'autant plus pour des questionnaires remplis par des enfants) et bien que ça n'ait pas été le sujet de ce travail, une étude portant sur le caractère éducatif plus précise pourrait être enrichissante.

Tout ces points font de Vivarium un bon point de départ pour des projets plus ambitieux mais il ouvre aussi la voie à de nouveaux projets de vulgarisation, en effet aujourd'hui la science prend une importance considérable et si beaucoup de sujets (comme l'intelligence artificielle) créent la

polémique, on trouve finalement peu d'études de vulgarisation de la part des spécialistes autre que des cours ou des articles journalistiques, ce qui est dommageable pour l'image de ces secteurs aux yeux du public. Toutefois la vulagrisation est un sujet qui prend de plus en plus d'ampleur et l'on peut donc s'attendre à voir une nette amélioration à l'avenir.

# Bibliography


[1] I. Asimov, *I, Robot*. Gnome Press, 1950.

[2] O. Tezuka, *Tetsuwan Atomu*. Kobunsha, Kodansha, 1952-1968.

[3] S. Kubrick, *2001: A Space Odyssey*. Metro-Goldwyn-Mayer, Polaris, 1968.

[4] L. McCauley, "Countering the frankenstein complex." in *AAAI spring symposium: Multidisciplinary collaboration for socially assistive robotics*, 2007, pp. 42–44.

[5] A. Turing, "Computing machinery and intelligence," *MIND*, vol. 59 (236), pp. 433–460, 1950.

[6] W. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5 (4), pp. 115–133, 1943.

[7] M. Minsky and D. Edmonds, "Crevier1993," p. 17, 2003.

[8] F. Rosenblatt, "The perceptron: a probabilistic model for information sotrage and organization in the brain," *Psychological Review*, vol. 65 (6), pp. 386–408, 1958.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521(7553), pp. 436–444.

[10] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern the recognitron unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[11] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, p. 2554–2558, 1982.

[12] C. Watksin, "Learning from delayed rewards," 1989.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20 (3), pp. 273–297, 1995.

[14] Y. LeCun, Y. Bengio, P. Haffner, and L. Bottou, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86 (11), pp. 2278–2324.

[15] D. Steinkraus, I. Buck, and P. Simard, “Using gpus for machine learning algorithms,” *12th International Conference on Document Analysis and Recognition*, p. 1115–1119, 2005.

[16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[17] A. Mordvintsev, C. Olah, and M. Tyka, “Deepdream-a code example for visualizing neural networks,” *Google Res*, 2015.

[18] F. Heider and M. Simmel, “An experimental study of apparent behavior,” *The American Journal of Psychology*, vol. 57, no. 2, pp. 243–259, 1944.

[19] *https://www.youtube.com/watch?v=Gl3EjiVlz_4*, Accessible le 20/01/2018.

[20] *https://www.youtube.com/watch?v=uwz8JzrEwWY*, Accessible le 20/01/2018.