

UNIVERSITÉ DE LILLE
FACULTÉ DES SCIENCES ET TECHNOLOGIES

Étude d'un Phénomène Émergent au sein d'un Système Multi-Agents

Le Flocking des Boïds

Antoine Nollet

Master Informatique

Master mention Informatique



DÉPARTEMENT D'INFORMATIQUE
Faculté des Sciences et Technologies

juin, 2023

Ce mémoire satisfait partiellement les pré-requis du module de Mémoire de Master, pour la 2^e année du Master mention Informatique.

Disponible depuis ce lien :
https://github.com/The-GTN/Master_Dissertation/tree/main



Candidat: Antoine Nollet, N° 41816061,
antoine.nollet.etu@univ-lille.fr

Encadrant(e): Jean-Christophe Routier,
jean-christophe.routier@univ-lille.fr



DÉPARTEMENT D'INFORMATIQUE
Faculté des Sciences et Technologies
Campus Cité Scientifique, Bât. M3 extension, 59655 Villeneuve-d'Ascq

juillet, 2023

Je dédie ce mémoire à toutes les personnes qui ont vécu, qui vivent ou qui seront amenées à vivre une dépression. N'oubliez pas de lever les yeux et d'y observer la beauté de la vie. On oublie bien trop souvent l'utilité des choses simples que nous offre la vie, comment ces doux spectacles de nuées d'oiseaux peuvent être porteurs tant bien d'algorithmes que d'espoir...

Remerciements

Ce travail de mémoire a été particulièrement éprouvant tout le long des deux années passées dessus... Ainsi, je remercie toutes les personnes qui m'ont aidé de près ou de loin dans ces travaux.

Merci à vous, Craig Reynolds, pour nous avoir apporté le modèle des Boids. Merci également à la communauté scientifique d'avoir approfondi les travaux autour de ce même modèle et ce depuis 1987. Vous avez permis au grand public de profiter de bien beaux spectacles. Et merci aux Systèmes Multi-Agents, pour nous avoir débarrassé de Xana !

Merci à vous, Romain Rouvoy, pour avoir offert l'opportunité aux étudiants de pouvoir effectuer un tel travail sur deux ans. Bien que très éprouvant et malgré les difficultés pratiques, il s'agissait là d'un exercice très formateur et qui a le mérite de marquer le vécu des étudiants.

Merci à vous, Thomas Pietrzak, pour avoir toujours été source de conseils avisés quant aux travaux de mémoire et également pour avoir été un responsable de parcours à l'écoute et bienveillant.

Merci à vous, ma famille, les amis, pour tous vos soutiens et bonnes humeurs !

Merci à vous, Jean-Christophe Routier, pour avoir accepté d'être mon encadrant. Vous m'avez soutenu et encouragé même lorsque je n'arrivais à rien. J'ai toujours eu peur de vous décevoir mais vous avez toujours su trouver les mots pour me redonner confiance. Je m'excuse pour toutes les mésaventures qu'aura occasionné cet encadrement et ne vous en remercie que davantage pour l'honneur que vous m'avez fait et pour la sagesse dont vous avez toujours su faire preuve.

Merci à toi, mon amie, pour avoir pris soins de moi lorsque je paniquais et que je m'évanouissais. Tu as été la source de ma force et de mon courage face à l'adversité de la vie, je n'ai que trop peu de mots pour quantifier tout le bien que tu m'apportes...

Merci à vous tous... Bonne lecture à vous.. !

Résumé

Dans ce travail de mémoire, nous étudions de manière formelle le modèle des boids de C.Reynolds [1]. L'étude de sa structure, de ses paramètres, de ses métriques et de ses manipulations possibles est mise en pratique au sein d'un simulateur [2] implémenté en C# en usant du logiciel Unity. L'objectif est ici de comprendre et de valider *l'émergence* du phénomène de *flocking* au sein d'un *SMA* composé de boids.

Une fois le modèle validé, nous identifions et étudions différents paramètres qui influent sur le comportement de la nuée des boids. Pour mesurer l'influence de ces paramètres, nous étudions deux métriques principales : l'*aire* et la *polarité*.

Nous parcourons ensuite des extensions du modèle, le *leading* et la *prédatation*, et leurs influences sur le phénomène du *flocking*. Ces modèles permettraient une meilleure réalisation du *flocking*.

Enfin, nous parcourons, sans exhaustivité, quelques contextes d'application du modèle. Nous illustrons ainsi différentes contributions scientifiques du modèle des boids de C.Reynolds [1]. Du schéma du vol des drônes en passant par l'algorithmique, ces contributions sont nombreuses et couvrent différentes disciplines.

Mots-clés : Émergence, Systèmes multi-agents, Simulation, Flocking

Abstract

In this master dissertation, we formally examine the boids model by C. Reynolds [1]. We undertake a comprehensive investigation of its structure, parameters, metrics, and potential manipulations, all of which are practically implemented within a C# simulator [2] utilizing the Unity software. The primary objective is to validate and to improve our understanding of the *emergence of flocking* phenomena within a *agent-based system* composed of boids.

Once the model has been validated, we proceed to identify and examine various parameters that influence the behavior of the boids swarm. In order to measure the influence of these parameters, we analyze two key metrics: *area* and *polarity*.

Subsequently, we get into extensions of the model, namely *leading* and *predation*, and their impacts on the phenomenon of *flocking*. These models have the potential to enhance the realization of *flocking behavior*.

Finally, we explore not exhaustively some application contexts of the model. In doing so, we illustrate different scientific contributions of C. Reynolds' boids model [1]. From drone flight patterns to algorithmics, these contributions are numerous and encompass various disciplines.

Keywords: Emergence, Multi-agent system, Simulation, Flocking

Indice

Table des figures	vii
Introduction	1
1 Modèle de Reynolds	5
1.1 Description	6
1.2 Reproduction du modèle et validation	12
1.3 Tests sur les paramètres	17
2 Autour du modèle	25
2.1 Le leading	25
2.2 Le leading émergent	27
2.3 La prédatation	27
3 Applications du modèle	29
3.1 Multimédia	29
3.2 Drônes	30
3.3 Algorithmique	32
3.4 Autres applications	33
Conclusion	35
Références	36

Table des figures

1	Le flocking d'oiseaux	2
2	Popularité des SMA	3
1.1	Agent Boïd	6
1.2	Champ de vision du boid	7
1.3	Règles du <i>flocking</i> : <i>Séparation</i>	8
1.4	Règles du <i>flocking</i> : <i>Alignement</i>	9
1.5	Règles du <i>flocking</i> : <i>Cohésion</i>	9
1.6	Répulsion de l'obstacle	11
1.7	Extrait de la vidéo de présentation de <i>Flocking Simulator</i>	12
1.8	Diagramme UML de <i>Flocking Simulator</i>	13
1.9	Environnement de <i>Flocking Simulator</i>	14
1.10	Les différentes types de voisinage	19
1.11	Champ de vision du boid	20
1.12	Champ de vision limité du boid	21
2.1	Graphe de connexion	26
2.2	Exemple de prédation	28
3.1	Fouloscopie	30
3.2	Sureté des drônes	31
3.3	Spectacle de drônes	32
3.4	Particle System Optimization	32
3.5	Mesures sur des pigeons	33
3.6	Recherche par la foule	34
3.7	Blues joué avec les boids	34

Introduction

Ce document correspond au mémoire d'Antoine Nollet rédigé dans le cadre de l'unité d'enseignement (UE) dénommée *Le mémoire de Master* en vue de la validation du Master Sciences et Technologies mention Informatique (master INFO¹) parcours Réalité Virtuelle et Augmentée (RVA²). Ce diplôme est dispensé par le département informatique (FIL) de la Faculté des Sciences et Technologies (FST) de l'Université de Lille. Ce mémoire a été rédigé sur deux années entre 2021 et 2023 et a pour objectif d'effectuer une synthèse de l'état de l'art scientifique d'un domaine ciblé. J'ai choisi le sujet *l'émergence du flocking des boids*.

Ce mémoire traite donc de l'étude du phénomène de l'émergence d'un comportement de *flocking*, d'un vol en nuée, au sein d'un groupe d'agents dans un Système Multi-Agents (SMA). La position adoptée dans ce mémoire est de se poser la question des caractéristiques et utilisations d'un tel phénomène. Ainsi, avant de commencer le contenu principal de ce mémoire, il nous faut en identifier et expliciter les termes principaux : *flocking*, *SMA* et *émergence*.

Le *flocking* correspond au phénomène naturel du vol groupé des oiseaux³ qu'on peut aisément constater lorsqu'on regarde le ciel et qu'on a la chance de croiser un regroupement d'oiseaux, comme illustré dans la figure 1⁴. Jusqu'à la publication de Craig W. Reynolds [1], il était difficile de simuler un tel phénomène sur ordinateur. Pour la reproduction sur ordinateur de phénomènes naturels complexes comme le vol d'oiseaux, la nature du mouvement qui se veut imprédictible est difficilement reproduisible par des modèles basés sur des équations prédictives et solvables. Le problème d'un vol d'oiseaux scripté est qu'il ne correspond justement pas à la nature dite chaotique, aléatoire, d'un vol d'oiseaux qu'on pourrait observer dans la nature. Il ne pourrait par exemple pas s'adapter à tous les environnements ou situations dans lequel on voudrait le placer, car il correspondrait à un scénario précis et ne serait pas capable de s'adapter.

-
1. <https://www.fil.univ-lille.fr/portail/index.php?dipl=MInfo&label=Pr%C3%A9sentation>
 2. <https://sciences-technologies.univ-lille.fr/informatique/formation/master-informatique/parcours-realite-virtuelle-et-augmentee>
 3. <https://www.youtube.com/watch?v=bb9ZTbYGRdc>
 4. source : <https://www.nature.com/articles/d43978-022-00067-4>



FIGURE 1 – Le flocking d’oiseaux

Pour résoudre ce problème, adopter une approche basée sur un *SMA* semble une solution davantage pertinente. Les *SMA* sont des systèmes composés d’agents, d’un environnement et d’un système faisant tourner la simulation. Les agents sont dotés de caractéristiques qui leurs sont propres, ayant un accès particulier à un ensemble d’informations et agissant selon leurs propres comportements. Ils existent dans l’environnement et peuvent être analysés ou influencés par le système. Ils sont amenés à avoir des interactions non pas seulement avec leur environnement mais également entre eux.

Ce principe est très utilisé dans les simulations et les résolutions de problèmes de par le positionnement local face à une situation. Il est possible d’associer cette idée avec la réalité : les agents sont des oiseaux, l’environnement est un espace en trois dimensions et le système est notre monde. Cette association avec la réalité ne donne que plus de pertinence à l’utilisation de *SMA* pour simuler le *flocking* des oiseaux.

En effet, l’approche centrée-individus propose une manière de décrire simplement ce phénomène complexe comme étant un comportement de groupe *émergent* d’une succession de règles simples suivies par les agents-oiseaux. Les agents n’ont pas conscience du mouvement du groupe, mais prennent des décisions individuelles basées sur leurs connaissances, c’est-à-dire ici leur voisinage. Le comportement collectif de la nuée résulte de ces décisions, et donc des comportements de chacun des agents. On parle alors *d’émergence* du comportement. Le mouvement de groupe n’est pas scripté d’un point de vue global, ce mouvement naît, *émerge*, des perceptions individuelles des agents et des actions qui en ont découlé. La nature *émergente* de cette approche permet ainsi de simuler le *flocking* des oiseaux de manière à s’adapter aux situations et à proposer des mouvements qui se rapprochent de la réalité.

L'approche centrée-individus a d'ailleurs eu des applications populaires très diverses dans le cinéma, le jeu vidéo, la télévision, l'architecture et l'ingénierie. Cela se révèle très pratique dès lors qu'on souhaite simuler des mouvements de foules. Les logiciels Massive⁵ et Golaem⁶ ont notamment pu nous offrir de célèbres scènes de batailles respectivement dans Le seigneur des anneaux, illustrée en figure 2a⁷, et Game of Thrones, illustrée en figure 2b⁸.



(a) Massive et Le Seigneur des Anneaux



(b) Goleam et Game of Thrones

FIGURE 2 – Popularité des SMA

5. <https://www.massivesoftware.com/about.html>

6. <https://golaem.com/>

7. source : <https://www.youtube.com/watch?v=sJVAoWnJ7Fw>

8. source : <https://www.inria.fr/en/golaem-puts-its-crowd-simulation-tools-service-television-industry>

Ce document est donc un mémoire portant sur l'étude d'un phénomène *émergent* au sein d'un *SMA* : le *flocking* des boids. Il s'agit donc en principale partie d'une synthèse de l'état de l'art scientifique qui véhicule sa réflexion autour du modèle de flocking des boids de C.W. Reynolds.

Ce travail a pour objectif d'éclaircir notre compréhension du phénomène, d'analyser les différents paramètres possibles de la simulation, de décrire ses déclinaisons et de constater ses utilisations.

Cependant, il ne s'agit pas seulement d'une synthèse de l'état de l'art. Ce mémoire a également donné lieu à des travaux en lien avec l'UE Projet Individuel (PJI) du master INFO. Un simulateur a été implémenté afin de reproduire de manière empirique le phénomène émergent du flocking des boids mais aussi de valider ou infirmer les extensions du modèle qui ont été étudiées.

La structure de ce document se compose donc en trois parties :

- chapitre 1 - Modèle de Reynolds : Description poussée du modèle initial avec des tests sur les paramètres
- chapitre 2 - Autour du modèle : Présentations de variations du modèle initial
- chapitre 3 - Application : Présentations d'utilisations du modèle

Chapitre 1

Modèle de Reynolds

Le modèle que propose Reynolds dans son article de 1987 [1] se pose dans un contexte de simulation informatique de mouvements en nuée, notamment avec le vol des oiseaux. Des vols d'oiseaux étaient déjà animés via ordinateur, cependant ces vols étaient scriptés, très difficiles à produire et ne pouvaient pas s'adapter à tous les environnements. En 1983, WT Reeves décrit dans [3] le principe de particules, des objets à la taille de points, ayant leur comportement propre afin de rendre des mouvements comme des nuages, de la fumé, de l'eau ou du feu. En 1985, «Eurythmy» in SIGGRAPH Video Review¹ propose dans son court métrage un modèle où chaque oiseau et objet applique un champ de force, influençant des vecteurs déplacements des oiseaux afin d'éviter les collisions et de se déplacer ensemble. S'inspirant de ces deux références, Reynolds a utilisé l'approche centrée-individus et a décrit le vol groupé d'oiseaux comme étant un comportement *émergent* des interactions locales entre les différents oiseaux. Les oiseaux ont, comme les particules, leur comportement propre et individuel et sont influencés par des forces extérieures résultantes de règles d'interactions.

1. <https://www.youtube.com/watch?v=HLiD2yCBY8o>

1.1 Description

Le modèle de Reynolds utilise ainsi l'approche centrée-individus des SMA. Les agents qui composent le modèle, représentant chacun des oiseaux, sont donc nommés des *boïds*, illustrés en figure 1.1². Il s'agit d'une contraction de *birds* et *oids*, signifiant *oiseaux* et *qui ressemble, à propos*, appuyant le fait qu'ils sont identifiables à des oiseaux. On parlera des *boïds de Reynolds* afin d'évoquer le modèle de Reynolds.

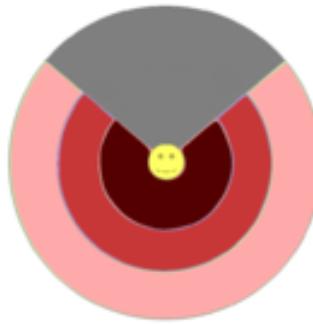


FIGURE 1.1 – Agent Boïd

Les boids

Ces agents boids sont caractérisés par une position, un vecteur déplacement et un champ de vision. Le vecteur déplacement décrit l'orientation que prend le boïd à chaque pas de simulation, de par son vecteur unitaire, ainsi que la vitesse de ce même boid, de par la norme du vecteur. La position du boïd dans la simulation sera trivialement impactée par son vecteur déplacement. Le champ de vision décrit un espace dans lequel le boid pourra obtenir des informations de son environnement et des autres boids qui se situent dans cet espace. Ce champ de vision est, selon que la simulation est en deux ou trois dimensions, soit un arc de cercle soit une portion de sphère. Comme illustré en figure 1.2³, la position du boid détermine le centre du champ de vision et son vecteur déplacement détermine la flèche. En effet, le boid regarde devant lui. Le champ de vision a pour paramètres un rayon et un angle, qui correspondent à la distance jusqu'à laquelle le boid peut voir et à l'étendue en largeur de ce qu'il peut voir.

2. source : <https://fr.wikipedia.org/wiki/Boids>

3. source : <https://www.red3d.com/cwr/boids/>

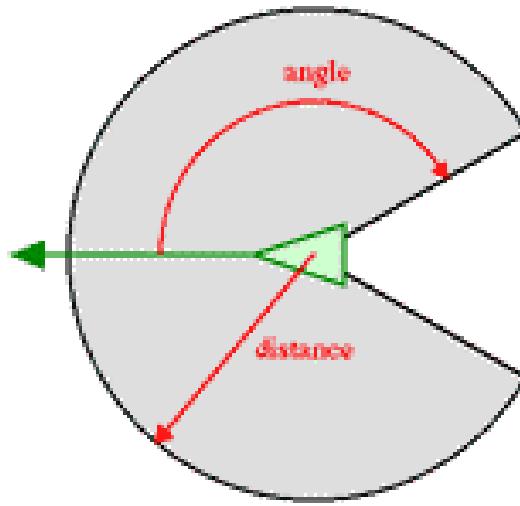


FIGURE 1.2 – Champ de vision du boid

Le voisinage du boid

Le modèle de Reynold est un modèle utilisant une approche centrée-individus. Les boids vont prendre en compte les informations qui sont à leur disposition, c'est-à-dire qui se situent dans leur champ de vision. Le champ de vision est paramétrable mais reste fixe lors de la simulation, ce qui n'est pas le cas du vecteur déplacement qui varie selon les boids se trouvant dans le champ de vision. Chaque boid peut accéder aux positions, orientations et vitesses de son voisinage, c'est-à-dire des boids se trouvant dans son champ de vision.

Avec l'ensemble B des boids, pour $b \in B$ on a les variables p_b de position, v_b de vecteur de déplacement, r_b de rayon de vision et α_b d'angle de vision, on décrit le voisinage N_b d'un boid b de la manière suivante :

$$N_b = \{i \in B, \|p_i - p_b\| \leq r_b \wedge \theta(b, i) \leq \alpha_b\} \quad (1.1)$$

avec $\theta(b, i)$ étant l'angle entre le vecteur déplacement du boid b et le vecteur défini par les positions p_i et p_b . Il s'exprime de la manière suivante :

$$\theta(b, i) = \arccos\left(\frac{v_b \cdot (p_i - p_b)}{\|v_b\| \cdot \|(p_i - p_b)\|}\right) \quad (1.2)$$

Le voisinage d'un boid a un impact sur la variation de son vecteur déplacement car le boid est soumis, de par son comportement individuel, à diverses règles qui s'appuient sur ce voisinage. On parle de comportement individuel, car même si les

boids suivent fondamentalement les mêmes règles, leurs décisions sont individuelles. Il en résulte donc qu'ils ont des vecteurs déplacements et positions différentes, donc également des voisinages différents, et donc ne vont pas réagir de la même manière. C'est en suivant des règles simples d'un point de vue individuel, que le phénomène du flocking, du vol en nuée, émergera d'un point de vue d'ensemble.

On peut d'ailleurs noter qu'en prenant en compte le nombre d'appels aux informations d'un autre boid comme étant une opération élémentaire, le modèle de Reynolds s'exécute avec une complexité algorithmique en $\mathcal{O}(n^2)$. Dans le pire des cas, chaque boid a tous les autres boids dans son voisinage.

Les règles du modèle

Les agents boïds suivent ainsi 3 règles : la *séparation*, l'*alignement* et la *cohésion*. Chacune de ces règles s'applique à chaque boid à chaque pas de simulation et permet de calculer le vecteur déplacement que le boid devra prendre en compte.

Séparation Cette règle, illustrée en figure 1.3⁴, se focalise sur les positions des boids. Elle prend en compte la distance entre le boid et chacun des voisins perçus, en fait la somme et renvoie l'opposé. Le but est de prévenir les collisions avec les voisins. Ainsi, avec N_b étant l'ensemble des voisins perçus par le boid b et p_i la position courante du boid i , nous avons l'expression de la règle $R_{sep}(b)$ suivante :

$$R_{sep}(b) = - \sum_{i \in N_b} p_i - p_b \quad (1.3)$$

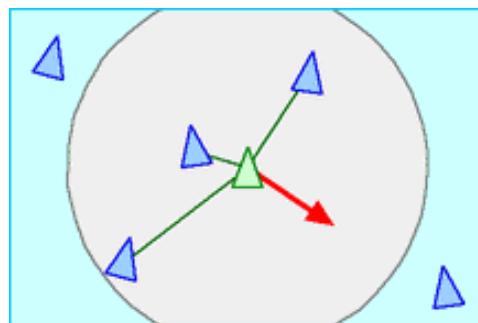


FIGURE 1.3 – Règles du *flocking* : *Séparation*

4. source : <https://www.red3d.com/cwr/boids/>

Alignement Cette règle, illustrée en figure 1.4⁵, se focalise sur les vitesses des boids. Elle prend en compte la direction de chacun de ses voisins et en fait la moyenne. Le but est de suivre le mouvement de la foule, de suivre le groupe. Ainsi, avec N_b étant l'ensemble des voisins perçus par le boid b et v_i le vecteur déplacement courant du boid i , nous avons l'expression de la règle $R_{ali}(b)$ suivante :

$$R_{ali}(b) = \frac{1}{|N_b|} \sum_{i \in N_b} v_i \quad (1.4)$$

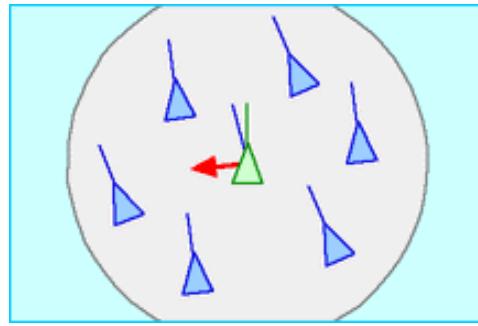


FIGURE 1.4 – Règles du *flocking* : *Alignement*

Cohésion Cette règle, illustrée en figure 1.5⁵, se focalise sur les positions des boids. Elle prend en compte la position de chacun des voisins perçus et en fait la moyenne pour calculer le barycentre des boids voisins. Cette règle renvoie la distance entre le boid et ce barycentre. Le but est de ne pas trop s'éloigner du centre de la nuée. Ainsi, avec N_b étant l'ensemble des voisins perçus par le boid b et p_i la position courante du boid i , nous avons l'expression de la règle $R_{coh}(b)$ suivante :

$$R_{coh}(b) = \frac{1}{|N_b|} \left(\sum_{i \in N_b} p_i \right) - p_b \quad (1.5)$$

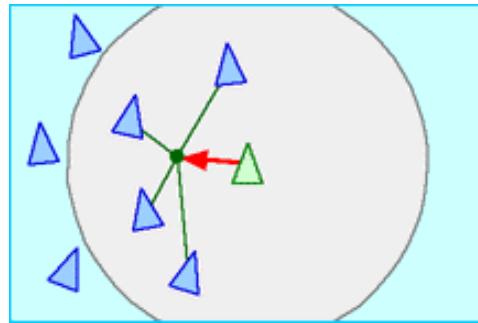


FIGURE 1.5 – Règles du *flocking* : *Cohésion*

5. source : <https://www.red3d.com/cwr/boids/>

Ces règles peuvent ne pas avoir les mêmes poids d'importance selon les situations. Si la distance entre le boid et un de ses voisins est jugée dangereusement faible, alors la règle R_{sep} doit s'appliquer davantage que les autres. Si le boid est éloigné des autres, alors la règle R_{coh} doit jouer un plus grand rôle. Il n'est pas explicité de méthode de décision propre, Reynolds décrit un module de pilotage. Mais cela peut se traduire par des seuils d'activations de règles et/ou des poids d'importances des règles, qui peuvent être constants comme variables selon l'usage qui en est souhaité.

Nous pouvons donc définir la règle $R_{Rey}(b)$ correspondant au modèle de C.Reynolds appliquée à un boid b , qui avec les règles $R_{sep}(b)$ (équation 1.3), $R_{ali}(b)$ (équation 1.4) et $R_{coh}(b)$ (équation 1.5) accompagnées de leurs poids d'importances w_{sep} , w_{ali} et w_{coh} , peut s'exprimer ainsi :

$$R_{Rey}(b) = \frac{w_{sep} * R_{sep}(b) + w_{ali} * R_{ali}(b) + w_{coh} * R_{coh}(b)}{w_{sep} + w_{ali} + w_{coh}} \quad (1.6)$$

Mise en œuvre du modèle

Afin que son modèle soit applicable à diverses situation, Reynolds a décrit son modèle de telle manière que chaque boid devrait fonctionner sous trois modules :

- module de navigation : analyse le champ de vision et envoie les règles à appliquer selon la situation ;
- module de pilotage : récupère les règles du module de navigation et interprète l'ordre de priorité en donnant des poids d'importance selon la situation ;
- module de vol : effectue le déplacement demandé par le module de pilotage.

Les boids sont amenés à se déplacer dans un espace où se trouvent non seulement d'autres boids mais également des obstacles qu'ils vont devoir esquiver.

Obstacles

Dans son champ de vision, le boid peut être amené à croiser d'autres boids ainsi que des éléments de décor de l'environnement : des obstacles. Les boids doivent éviter de rentrer en collision avec ces obstacles. Pour cela, il est nécessaire de disposer d'une règle prenant en compte ce cas de figure. Ainsi, une nouvelle règle de comportement des boids consiste à retourner un vecteur qui les éloignent des obstacles et dont l'intensité croît lorsque la distance entre le boid et l'obstacle perçu diminue. Ainsi, avec $rep(v_b)$ le vecteur répulsion en réponse au vecteur déplacement v_b d'un boid b et avec O_b l'ensemble des obstacles perçus par ce même boid b , de la même manière que les boids sont perçus pour le voisinage, nous avons la règle R_{rep} telle que :

$$R_{rep}(b) = \frac{1}{|O_b|} \sum_{o \in O_b} \frac{1}{||p_o - p_b||^2} rep(v_b) \quad (1.7)$$

Dans l'expression précédente, il y a un choix à effectuer : le vecteur répulsion $rep(v_b)$ peut être simplement la direction opposée à v_b . Dans ce cas, si un boid fonçait en ligne droite sur un obstacle, un mouvement non naturel en résulterait. Comme illustré en figure 1.6, la réponse $rep(v_b)$ peut également être la tangente entre v_b et le point d'impact, ou bien même le vecteur moyen entre cette tangente et la direction opposée, nous approfondissons ce point dans la section suivante. Cependant, réagir à un obstacle que le boid n'est pas sur le point de percuter est-il vraiment pertinent ? Reynolds propose que le boid ne soit repoussé par un obstacle que s'il est orienté vers ce même obstacle.

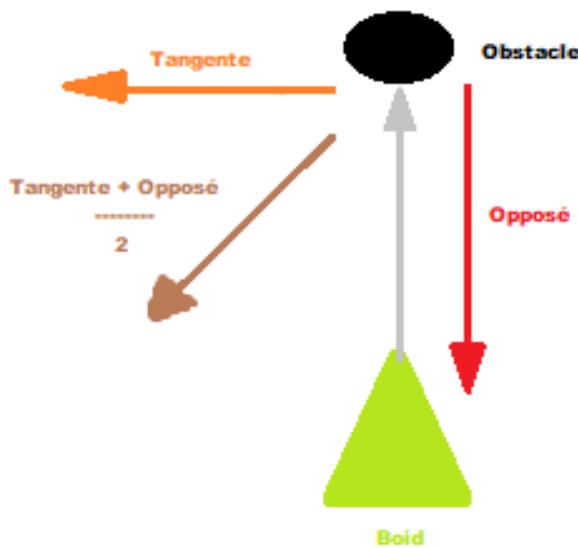


FIGURE 1.6 – Répulsion de l'obstacle

Ouvertures

Dans son article [1], Reynolds a énoncé diverses ouvertures :

- simplification algorithmique par décentralisation : attribuer un processeur par boid réduirait la complexité algorithmique du modèle de Reynolds de $\mathcal{O}(n^2)$ en $\mathcal{O}(n)$. Chaque boid individuellement pourrait, dans le pire des cas, voir tout les boids de son entourage ;
- implémentation de leaders : ce point est plus approfondi dans le prochain chapitre. Il se peut que le fait que des boids suivent certains d'entre eux en priorité favorise l'émergence de *flocking* ;

- implémentation de prédateurs : ce point est également approfondi dans le prochain chapitre. On peut penser à un nouveau type d'obstacle qu'il faudrait ici esquiver. Cela pourrait correspondre à une meilleure fidélité au phénomène qu'on peut constater dans la nature ;
- validation du modèle : en effet, Reynolds indique qu'il n'a pas testé de manière formelle son modèle afin de prouver qu'il passe le test de Turing⁶. Ici le test de Turing, pour le flocking des oiseaux, est de regarder si ce phénomène est crédible d'un œil humain. Afin de s'assurer du bon fonctionnement de ce dit modèle et pour pouvoir ensuite effectuer des tests dessus, un modèle [2] de Reynolds a été implémenté dans le cadre de ce mémoire.

1.2 Reproduction du modèle et validation

Comme précédemment cité, ce mémoire a donné lieu à des travaux en lien avec l'UE Projet Individuel (PJI) du master INFO. Ces travaux correspondent à un simulateur de flocking de boids nommé *Flocking Simulator* [2] qui a permis de reproduire de manière empirique le modèle de Reynolds, de faire des tests sur ses différents paramètres et de tester, valider ou infirmer des extensions étudiées du modèle tout au long du travail du mémoire.

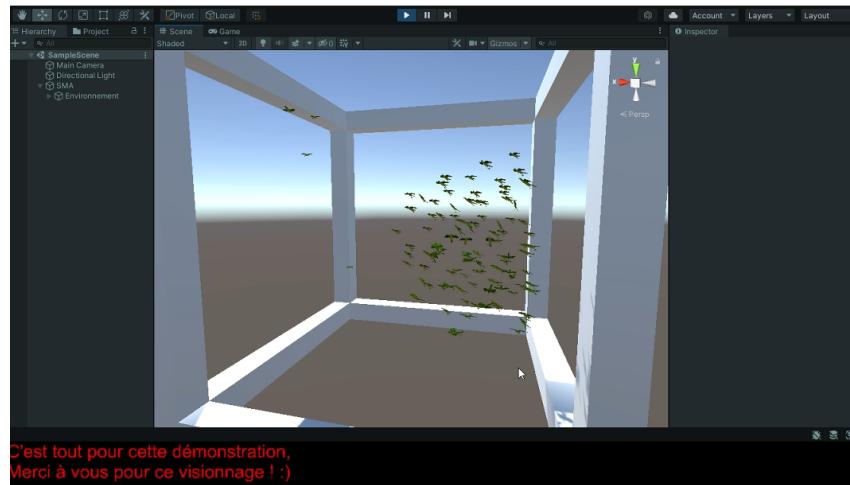


FIGURE 1.7 – Extrait de la vidéo de présentation de *Flocking Simulator*

6. https://fr.wikipedia.org/wiki/Test_de_Turing

Implémentation

Flocking Simulator a été en implémenté sous Unity version LTS 2020.3.31f1⁷ avec le langage C#. Il s'agit d'un simulateur de volées d'oiseaux dans lequel on peut incorporer diverses comportements et obstacles et qui respecte une certaine structure comme illustrée en figure 1.8.

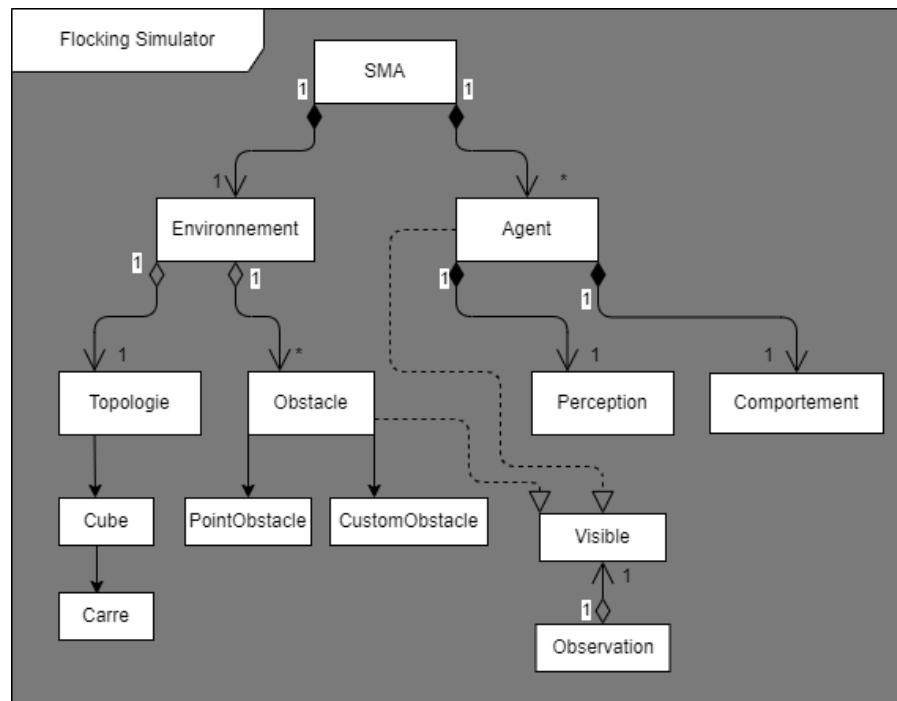


FIGURE 1.8 – Diagramme UML de *Flocking Simulator*

Les agents boids évoluent donc dans un SMA qui leur offre un environnement dans lequel agir. Un agent boid est défini par sa perception de son environnement et son comportement qui définit ses interactions avec l'environnement et avec les autres agents. Bien qu'on utilise les mêmes scripts pour chacun des agents, leurs positions et donc leurs perceptions sont différentes et ils agissent donc chacun différemment des autres. C'est par cette différence qu'il est intéressant de visualiser les phénomènes émergents qui peuvent se créer à partir des interactions des agents, comme par exemple, un *flocking* des boids. C'est en créant des scripts qui héritent de Comportement qu'il est possible, via *Flocking Simulator*, de reproduire un comportement donnée dans la littérature. Ce niveau d'abstraction a justement été pensé pour reproduire différents modèles comme par exemple celui de C.Reynolds [1]

7. <https://unity.com/releases/editor/whats-new/2020.3.3>

Néanmoins la mise en œuvre d'un tel simulateur a poussé à la réflexion de critères à prendre en compte que C.Reynolds n'explicitait pas lorsqu'il a proposé son modèle. Bien que voulu comme assez abstrait pour pouvoir être utilisé dans bon nombre de contextes, il a fallu poser le contexte d'environnement dans lequel le modèle allait exister.

Environnement Borné

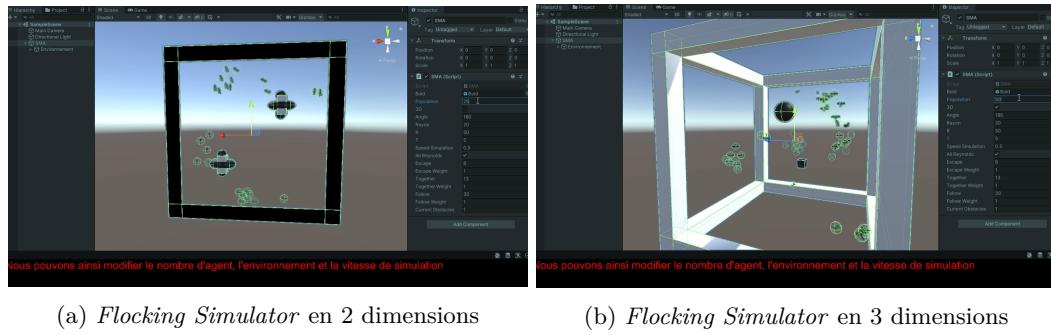


FIGURE 1.9 – Environnement de *Flocking Simulator*

En effet, comme illustré ci-dessus, nous avons fait en sorte de pouvoir reproduire des modèles tant en 2 dimensions qu'en 3 dimensions. Si ce choix a été simple à prendre, car il est cohérent de vouloir pouvoir mettre en place des *flocking* des boids en 2 ou 3 dimensions, il a été moins simple de décider de mettre en place un environnement clos. Toujours comme illustré précédemment, *Flocking Simulator* propose un environnement fermé dans lequel les agents agissent, pour la principale raison qu'on souhaite pouvoir observer les phénomènes *émergents* des différentes interactions. Unity proposant un espace large dans lequel les entités peuvent se déplacer, les agents pouvaient donc se disperser dans des directions opposées sans interagir entre eux, ce qui est contraire aux objectifs de reproductions expérimentales et d'analyses de *Flocking Simulator*.

En appliquant le comportement du modèle de C.Reynolds [1] dans un tel environnement clos, on s'approche davantage d'une nage-schooling de poissons en aquarium⁸ que d'un vol-*flocking* d'oiseaux. Mais ceci n'entre pas en contradiction avec le modèle car C.Reynolds indiquait justement que son modèle pouvait servir également à reproduire ce phénomène. Malgré cela, il reste à déterminer comment les agents agissent selon ce cadre borné qu'ils peuvent considérer comme un obstacle.

8. <https://www.youtube.com/watch?v=YSfaeHlyeN8>

Répulsion des obstacles

Comme cité dans la section précédente et dans la continuité des conditions que C.Reynolds n'a pas explicité dans son modèle, il n'y a pas de manière claire d'éviter les obstacles que les agents peuvent suivre. Seul le fait que l'agent ne soit pas repoussé par un obstacle ne se trouvant pas en face de lui et le fait qu'il soit davantage répulsé par l'obstacle se trouvant proche de lui sont décrits en même temps que son modèle. Nous reprenons d'ailleurs l'équation 1.7 pour y intégrer un paramètre r qui permettra de régler l'intensité de la force de répulsion.

$$R_{rep}(b) = \frac{1}{|O_b|} \sum_{o \in O_b} \frac{r}{\|p_o - p_b\|^2} rep(v_b) \quad (1.8)$$

Cependant, on souhaite que la définition de $rep(v_b)$ soit en effet une moyenne entre la direction opposée et la direction tangente. L'agent arrivant face à un obstacle, il y a le choix de quelle tangente il prend en considération : on itère simplement sur les quatres tangentes haut-bas et droite-gauche et on conserve la tangente dont la différence d'angle est la moins importante par rapport à la direction courante de l'agent. Ainsi, en prenant en compte la direction $tan(b, o)$ qui décrit la direction tangente qu'un agent b prend en fonction d'un obstacle o , on redéfinit R_{rep} de la manière suivante :

$$R_{rep}(b) = \frac{1}{|O_b|} \sum_{o \in O_b} \frac{r * (-v_b) * tan(b, o)}{2\|p_o - p_b\|^2} \quad (1.9)$$

Expérimentalement, on s'est rendu compte que la répulsion opposée à l'obstacle devait être davantage important que la répulsion par direction tangente lorsque l'agent est très proche de l'obstacle car naturellement l'objectif principal de l'agent est d'esquiver l'obstacle. Parallèlement, on s'est également rendu compte que la répulsion par la tangente pouvait être davantage importante que la répulsion par l'opposée lorsque l'agent n'est pas proche de l'obstacle, car cela représente un mouvement plus simple à faire : l'agent n'a qu'à légèrement se décaler pour éviter l'obstacle. De ces observations, nous traduisons donc une nouvelle définition de R_{rep} avec également un paramètre t de force de répulsion par tangente accompagnant le paramètre r de force de répulsion par l'opposé.

$$R_{rep}(b) = \frac{1}{2|O_b|} \left(\sum_{o \in O_b} \frac{r * (-v_b)}{\|p_o - p_b\|^2} + \sum_{o \in O_b} \frac{t * tan(b, o)}{\|p_o - p_b\|} \right) \quad (1.10)$$

Comportement Composé

Le comportement de prévention des collisions avec les obstacles étant défini, il a fallu définir comment l'intégrer à d'autres modèles, d'autres comportements, comme par exemple celui de C.Reynolds. Dans son article [1], C.Reynolds décrit les modules de navigation, de pilotage et de vol. Les modules de navigation et de vol sont assurés respectivement par la structure du projet *Flocking Simulator* comme illustré en figure 1.8 et par le logiciel Unity qui permet le déplacement des entités entre les pas de simulations qui se font entre chaque rendu visuel.

À noter que cette particularité, couplée avec l'ouverture de la complexité algorithmique en $\mathcal{O}(n^2)$, occasionne du délai d'exécution lorsque le nombre n d'agents boids est trop important. Avec un ordinateur portable doté d'un processeur Intel(R) Core(TM) i5-5300U, d'un CPU d'une fréquence de 2.30GHz, de 8Go de RAM et d'une carte graphique Intel HD Graphics 5500⁹, *Flocking Simulator* est capable de lancer une simulation contenant jusqu'à une centaine d'agents boids.

Le module de pilotage revient ici à regrouper les règles que suivent les agents et à leur donner un ordre de priorité à suivre. Nous avons donc fait le choix de moyennner avec pondération les différents comportements que suit un agent. Cela se traduit par le fait qu'un agent suit la règle de répulsion des obstacles et en reçoit un vecteur, puis suit les règles du modèle de Reynolds et en reçoit également un vecteur réponse, puis nous moyennons ces vecteurs pour obtenir le nouveau vecteur déplacement de l'agent. Nous rajoutons également, à poids moindre, un vecteur aléatoire *alea* dans la moyenne afin de brouter le mouvement des agents boids les rendant davantage réalistes.

Ainsi, avec un pas t de simulation, un boid b , un vecteur $R_{Rey}(b, t)$ en réponse aux règles du modèles de Reynolds, w_{Rey} son poids d'importance, un vecteur $R_{rep}(b, t)$ en réponse à la prévention des collisions avec les obstacles, w_{rep} son poids d'importance, un vecteur aléatoire R_{alea} et w_{alea} son poids d'importance, on définit le prochain vecteur déplacement $v_b(t + 1)$ d'un agent boid b comme la somme pondérée des trois vecteurs :

$$v_b(t + 1) = \frac{w_{rep} * R_{rep}(b, t) + w_{Rey} * R_{Rey}(b, t) + w_{alea} * R_{alea}}{w_{rep} + w_{Rey} + w_{alea}} \quad (1.11)$$

9. <https://fr.gadgetversus.com/carte-graphique/intel-hd-graphics-5500-fiche-technique/>

Notons que nous prenons maintenant en compte la variable de pas de simulation t non sans raison. À l'état initial $t = 0$, tous les boids sont placés aléatoirement et ont des vecteurs déplacements aléatoires. Ensuite, en $t = 1$, les agents boids se réfèrent aux positions et vecteurs déplacements qu'ils perçoivent en $t = 0$ afin de décider de leur nouveau vecteur déplacement pour enfin se déplacer en conséquence. Ainsi, un pas t de simulation se conclut par le fait que chaque agent a déterminé son vecteur déplacement pour ensuite l'effectuer. Quand chaque agent a effectué son déplacement, on passe au prochain pas de simulation. Nous n'avions donc jusqu'ici pas cité la variable t car les règles $R_x(b, t)$ s'appliquent selon les positions $p_b(t)$ et vecteurs déplacements $v_b(t)$, nous avions donc simplifié les équations.

Le simulateur *Flocking Simulator* implémenté et le modèle de C.Reynolds reproduit, il nous reste à utiliser ce même simulateur afin d'analyser l'influences des divers paramètres possibles du modèle dans le phénomène *émergent* du *Flocking des boids*.

1.3 Tests sur les paramètres

Identification des paramètres

Pour reprendre la section précédente, on peut d'ores et déjà identifier le nombre d'agents, les forces de répulsions ainsi que les poids des différentes règles comme étant des paramètres sur lesquels on peut agir. De plus, et cela a également été prévu dans les travaux effectués, nous pouvons considérer des seuils d'activation. Lors des explications des règles du modèle de Reynolds, on indiquait déjà une nuance de distance : les boids trop proches doivent s'éloigner, les boids trop éloignés doivent se rapprocher.

Ainsi, il faut reprendre les équations 1.3, 1.4 et 1.5 en définissant les nouveaux voisinages $N_{sep}(b)$, $N_{ali}(b)$ et $N_{coh}(b)$. Ainsi, avec les seuils s_{sep} , s_{ali} et s_{coh} , on obtient :

$$N_{sep}(b) = \{i \in N_b, \|p_i - p_b\| \leq s_{sep}\} \quad (1.12)$$

$$N_{ali}(b) = \{i \in N_b, \|p_i - p_b\| \leq s_{ali}\} \quad (1.13)$$

$$N_{coh}(b) = \{i \in N_b, \|p_i - p_b\| \leq s_{coh}\} \quad (1.14)$$

On peut donc redéfinir les trois règles du modèle de C.Reynolds en :

$$R_{sep}(b) = - \sum_{i \in N_{sep}(b)} p_i - p_b \quad (1.15)$$

$$R_{ali}(b) = \frac{1}{|N_{ali}(b)|} \sum_{i \in N_{ali}(b)} v_i \quad (1.16)$$

$$R_{coh}(b) = \frac{1}{|N_{coh}(n)|} \left(\sum_{i \in N_{coh}(n)} p_i \right) - p_b \quad (1.17)$$

On peut également opter pour une variante des définitions des différents voisinages précédemment cités. En effet, dans son article [1], C.Reynolds ne spécifie pas la manière dont les distances ont un impact, on peut donc avoir ces voisinages :

$$N'_{sep}(b) = \{i \in N_b, \|p_i - p_b\| \leq s_{sep}\} \quad (1.18)$$

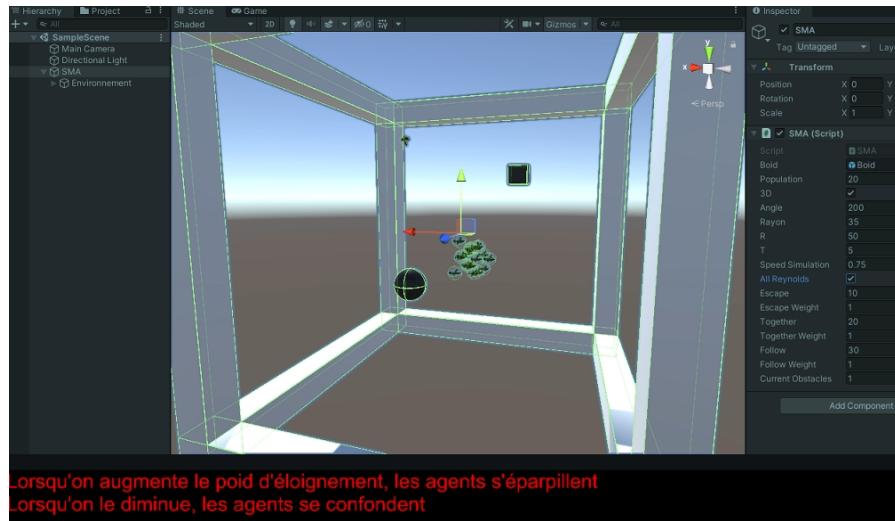
$$N'_{ali}(b) = \{i \in N_b, \|p_i - p_b\| \leq s_{ali} \wedge \|p_i - p_b\| \geq s_{sep}\} \quad (1.19)$$

$$N'_{coh}(b) = \{i \in N_b, \|p_i - p_b\| \leq s_{coh} \wedge \|p_i - p_b\| \geq s_{ali}\} \quad (1.20)$$

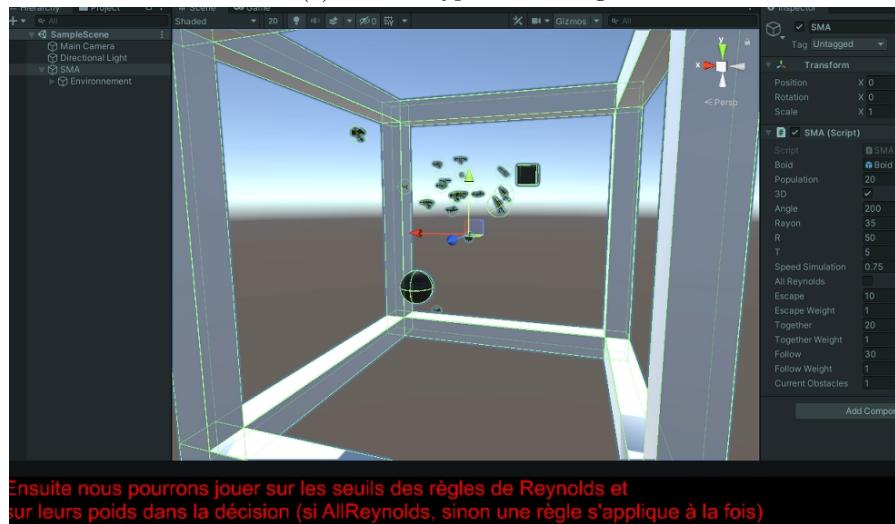
La principale différence ici réside dans le fait que, lorsqu'on prend en compte les précédents voisinages, on considère que les agents doivent se rapprocher et s'éloigner des autres agents trop proches. Ceci car, pour un modèle pertinent, il faut respecter la relation suivante :

$$s_{sep} \leq s_{ali} \wedge s_{ali} \leq s_{coh} \quad (1.21)$$

Donc il est cohérent de considérer la deuxième version des voisinages : on ne voudrait pas se rapprocher d'un agent qu'on est censé éviter. Cependant, les deux types de voisinages, illustrés en figure 1.10, permettent d'obtenir des *flocking* des boids stables et visuellement réalistes. La seule différence que cela induit, et qui est assez prévisible, est que le premier type des voisinages produit des nuées plus compactes, plus denses, là où le deuxième type de voisnage produit des nuées plus élargies, moins denses. On va cependant préférer le premier type de voisnage car il produit moins de bruits, de mouvements parasites, en faisant davantage de moyennage que le second.



(a) Premier type de voisinage



(b) Deuxième type de voisinage

FIGURE 1.10 – Les différentes types de voisinage

Nous avons donc comme paramètres sur lesquels nous pouvons influer : les seuils s_{sep} , s_{ali} et s_{coh} des règles du modèle de C.Reynolds, les poids w_{sep} , w_{ali} et w_{coh} de ces mêmes règles, le nombre n de boids lors de la simulation, le choix de voisinage précédemment cité, les coefficients r et k d'intensité de force de répulsion, la distance parcourue entre deux pas de simulation ainsi que la vision qui permet à un boid de définir son voisinage.

La vision

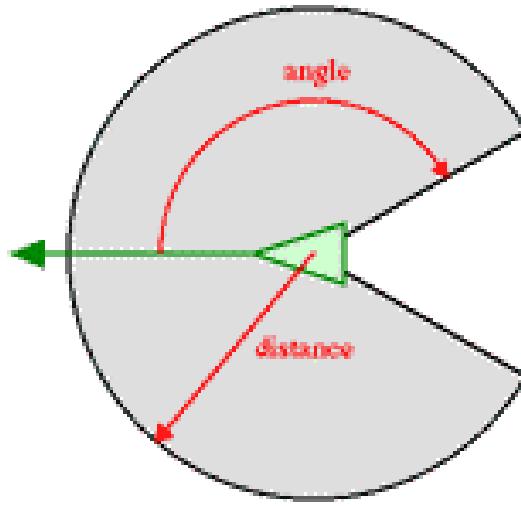


FIGURE 1.11 – Champ de vision du boid

En reprenant cette figure 1.11¹⁰, on constate deux paramètres clés dans la description de la vision d'un boid : son rayon de vision et son angle de vision.

Dans *Flocking Simulator* [2], ces deux paramètres définissent la perception des boids. La perception qui permet à l'environnement du *SMA* de retourner les voisinages lorsque les boids le demandent.

[4] et [5], tous deux basés sur le modèle *Flock Synthesis Rules* (FSR) qui sera présenté au chapitre 2, proposent d'utiliser un angle total de vue défini entre 180 et 270 degrés afin de favoriser l'*émergence* du *flocking* des boids.

Cependant, dans [6], dans un contexte de vol de drônes et pour de l'optimisation de contrôle de la nuée, un champ de vision non-centrée vers l'avant du boïd est proposé. Dans ce contexte d'application pratique du modèle de C.Reynolds [1], avec les contraintes matérielles de placement des caméras, la question de l'angle et de l'orientation de la vue est posée, inspirée par la constitution biologique des oiseaux. Comme illustré en figure 1.12¹¹, en plus de l'angle de vision on peut définir un angle de restriction de la vision qui se trouve en face du boid. En effet, dans la figure, on retrouve un agent i avec un vecteur déplacement \dot{x}_i doté d'un angle α de vision de 120° . L'angle θ , ici de 90° , correspond à l'angle azimut entre le vecteur déplacement d'un boid et le vecteur traversant le centre d'un de ses demi-champs de vision.

10. source : <https://www.red3d.com/cwr/boids/>

11. source : figure 1 of [6]

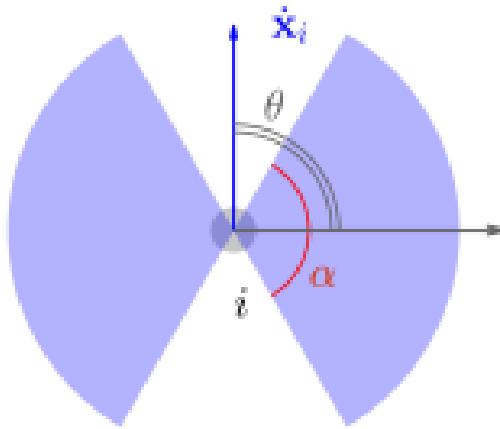


FIGURE 1.12 – Champ de vision limité du boid

On détermine l'angle de restriction θ_{restr} de la vision ainsi :

$$\theta_{restr} = \theta - \frac{\alpha}{2} \quad (1.22)$$

Avec les valeurs précédemment citées, l'angle θ_{restr} est de 30°.

[6] nous dit qu'un grand angle de vision favorisera l'émergence de la nuée. Bien que cela soit trivial, la question de l'angle à choisir dépendra donc de la volonté de créer un mouvement de nuée au détriment de la volonté de le faire avec une moindre complexité algorithmique. En effet, [1] établit en $\mathcal{O}(n^2)$ la complexité de la gestion du voisinage, où n est le nombre de boids de la simulation. Restreindre le champ de vision permet de prendre en compte un moindre nombre de boids dans les voisinages et donc occasionne un moindre nombre d'accès aux informations des boids.

L'orientation non-centrée du champ de vision est jugée plus pratique en application réelle, car maximisant la sûreté avec un angle θ_{restr} allant entre 20 et 30 degrés, comme c'est le cas pour la plupart des oiseaux. La figure 1.12 représente donc une vision pertinente d'un boid.

Métriques de description

Les paramètres du modèle sont identifiés, il reste à savoir comment mesurer leur influence dans le phénomène *émergent* du *flocking*. Nous déterminons les métriques du phénomène comme décrit dans [7] : l'*aire* et la *polarité*.

L'*aire* d'une nuée correspond à l'espace occupé par l'ensemble des boids formants cette nuée. Il s'agit très souvent d'un espace convexe calculé selon les positions des boids. Plusieurs formules de calculs sont ainsi possibles afin de calculer la valeur de cette métrique, mais l'idée à retenir est que la valeur de cette métrique est basse lorsque la nuée est dense et compacte, et qu'elle est haute lorsque les boids sont espacés les uns des autres.

La *polarité* d'une nuée correspond au degré d'alignement des boids d'une nuée. Cette métrique est calculée selon les vecteurs déplacements des boids qui constituent la nuée. Plus les boids se dirigent dans la même direction, plus cette métrique a une grande valeur. Plus les boids se dispersent, plus cette métrique a une basse valeur.

Ainsi, on peut décrire, comme le fait [4], l'apparition d'un *flocking* des boids comme étant le fait que les boids aient une grande polarité entre eux et qu'ils occupent une petite aire. Cela correspond au fait que les boids forment en effet une nuée ensemble, et qu'ils se dirigent dans la même direction.

Dans le cadre de mise en pratique tel que dans [6], on peut proposer d'autres métriques comme : l'*ordre* de la nuée, la *connectivité*, l'*union* de la nuée ainsi que sa *sûreté*. L'*ordre* correspond au fait que les boids aillent dans la même direction et la *connectivité* à la mesure du fait que les boids se perçoivent. Ces deux métriques peuvent se déduire de la métrique de la *polarité*. L'*union* correspond au fait que des boids forment un groupe restreint et la *sûreté* indique le fait que la proximité des boids ne présente pas de dangers. Ces deux métriques peuvent se déduire de la métrique de la *l'aire*.

Influences des paramètres

Nous allons donc reprendre chacun des paramètres en incluant les seuils et poids des règles, les paramètres de vision ainsi que la vitesse des boids.

En effet, la vitesse des boids, abordée que brièvement jusqu'ici, correspond à la distance qu'un boid effectue entre deux pas de simulation. On peut la décrire également comme étant la norme du vecteur déplacement du boid. La vitesse $\|v_b\|$ des boids b peut être constante, comme dans [2], ou variante, comme dans [7]. Lorsque la vitesse est variante, il est important de déterminer une vitesse maximale v_{max} que les boids ne peuvent dépasser. En vitesse constante, on considère v_{max} comme étant la vitesse que les boids ont constamment.

Le simulateur [2] ainsi que [7] nous permettent de faire les descriptions suivantes :

- Plus la vitesse v_{max} est grande, plus l'aire de la nuée devient grande et moins la polarité devient importante. À trop grande vitesse, les boids n'ont pas le temps de prendre en compte leur voisinage afin d'interagir avec. Une trop grande distance parcourue entre deux pas de simulation ne fait que perturber les interactions entre les boids et, par conséquence, perturbe également l'*émergence du flocking*. Cependant, une valeur trop basse de v_{max} ne fait que perturber le bon fonctionnement de la simulation qui serait en suspend et ferait du "sur place".
- Plus le nombre n de boids dans la simulation est important, plus l'aire ainsi que la polarité de la nuée vont avoir des valeurs importantes. Plus il y a de boids, plus il y a de chance qu'un boid prenne en compte un autre boid dans son voisinage et de conduire à un comportement groupé de *flocking*. Cependant, un nombre n trop élevé de boids risque de ralentir le bon fonctionnement de la simulation, de par la complexité algorithmique en $\mathcal{O}(n^2)$ de la simulation.
- Les coefficients r et k d'intensité de force de répulsion ne semblent pas avoir d'impact sur le comportement de *flocking* mais seulement sur son mouvement. Plus ils sont importants, plus la foule sera repoussée par les bords de la simulation.
- Les paramètres de vision, c'est-à-dire l'angle de vue α et le rayon de vue r_{vision} , impactent la perception du voisinage de chacun des boids. Si leurs valeurs sont grandes, davantage de boids sont perçus et l'*émergence du flocking* a plus de chance d'arriver. Cependant, comme pour le paramètre n , si les paramètres de visions sont trop élevés, cela revient à considérer tout les boids de la simulation et à ralentir le fonctionnement de la simulation.
- Le poids w_{sep} de la règle de *séparation* ainsi que son seuil d'activation permettent d'augmenter l'aire et de diminuer la polarité lorsqu'ils sont élevés. La règle consistant à faire se séparer les boids, leur faisant prendre de directions opposées, il est cohérent de constater une telle dispersion. Le poids intensifie cette règle face aux autres règles et le seuil permet d'activer plus souvent cette même règle.

- Le poids w_{ali} de la règle d' *alignement* ainsi que son seuil d'activation permettent d'augmenter la polarité lorsqu'ils sont élevés. L'aire n'est pas impactée par cette règle. Celle-ci consistant à faire prendre une direction commune à tous les boids, cette stabilisation du mouvement est cohérente. Le poids intensifie cette règle face aux autres règles et le seuil permet d'activer plus souvent cette même règle.
- Le poids w_{coh} de la règle de *cohésion* ainsi que son seuil d'activation permettent d'augmenter l'aire lorsqu'ils sont élevés. La polarité n'est pas impactée par cette règle. Tant le fait que la règle rassemble les boids diminuent directement l'aire est intuitif, tant on pourrait penser que la polarité en est diminuée car les boids ne suivent pas la même direction. La raison de cette non-influence de la polarité est que, bien qu'ils n'aient pas les mêmes vecteurs déplacements, les boids suivent une même position de centroïd de la nuée. Le poids intensifie cette règle face aux autres règles et le seuil permet d'activer plus souvent cette même règle.

Manipulations

Nous pouvons donc influer l'aire et la polarité de la nuée selon les paramètres précédemment citées, modifier ainsi la forme de cette nuée. Vouloir contrôler la forme de la nuée a notamment des avantages pratiques qui sont davantage détaillés en section 3.2.

En usage pratique, comme c'est le cas dans [8] et [9], les boids sont des drônes qui effectuent des tâches et qui peuvent être ralenti selon les tâches. La forme souhaitée de la nuée peut varier selon la situation à un moment donnée.

[9] a décidé d'affecter ses poids de règles de Reynolds, selon les distances où se trouvent les boids et selon les objectifs communs courants. Par exemple la nuée peut vouloir migrer vers une direction, on favorise donc la polarité plutôt que l'aire de la nuée. [9] suit une adaptation itérative de ses paramètres, d'autres solutions sont possibles. On peut décider, comme le fait [8], d'utiliser des algorithmes d'apprentissage afin qu'émerge des paramètres optimaux à une situation. [8] propose un apprentissage de ces différents paramètres par utilisation d'algorithme génétique¹².

Après avoir fait le tour du modèle de reynolds, il est temps de creuser les ouvertures que C.Reynolds avait posées. Quelles variations et extensions peuvent être implémentées à partir de celui ci ?

12. https://en.wikipedia.org/wiki/Genetic_algorithm

Chapitre 2

Autour du modèle

À la conclusion de son article [1], C.Reynolds fait une ouverture sur la possibilité de désigner des meneurs de la nuée et de considérer des obstacles mobiles comme étant des prédateurs. Il est maintenant question de considérer ces pistes.

2.1 Le leading

Le modèle du *leading* correspond à donner un rôle spécifique à un ou plusieurs des boids de la simulation, que les autres boids vont suivre en priorité : on parle de mener la nuée. Donner le rôle de leader permet d'inciter au phénomène de *flocking* et même d'influencer la direction de la nuée et ainsi de favoriser une haute polarité. En effet, le leader influence les vecteurs directeurs des autres boids en les incitant à le suivre. On peut penser à plusieurs techniques d'implémentation de leading mais elles respectent a priori toutes le principe expliqué. On peut notamment établir des graphes orientés de connexions, comme illustré en figure 2.1¹, qui mesurent le fait qu'un boid est suivi par les autres. On constate que lorsqu'un boid perçoit un autre boid, une flèche partant du boid et allant vers le boid perçu est présente. Comme illustré dans cette même figure, on constate par exemple que le boid 2 est mené par les boids 3 et 5. Par effet récursif, un boid est mené par un autre qui lui-même peut être mené par un autre. Le cas de base de cette récursion représente le leader de la nuée. Dans cette figure 2.1, le boid 6 est le leader.

1. source : figure 7 de [5]

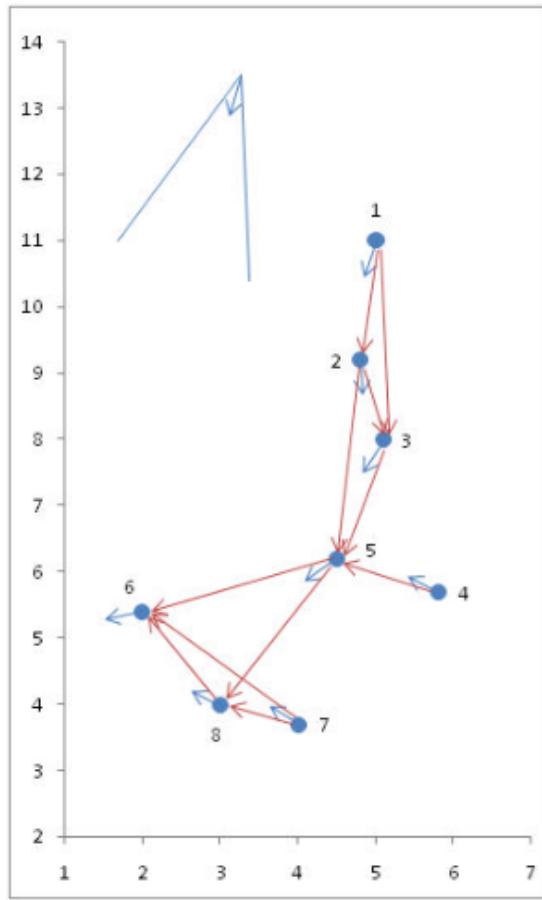


FIGURE 2.1 – Graphe de connexion

Un premier exemple de *leading* est illustré par le modèle de [10], où il est expliqué comment un ou plusieurs leaders informés peuvent mener leur nuée non-informée à une cible prédéterminée. Ici les leaders ont un objectif et possèdent des informations globales et arrivent, du fait qu'ils sont suivis en priorité, à mener la nuée vers une cible. Le leader préfère se diriger vers la cible, il se déplace plus rapidement quand il s'y dirige et a conscience qu'il doit être suivi de la nuée. Ce mécanisme permet d'obtenir un phénomène de *flocking* qui se dirige en effet vers la cible, cependant il faut adapter la vitesse maximale du leader avec précaution car une valeur trop grande de cette vitesse peut faire sortir le leader du champ de vision des suiveurs.

Un deuxième exemple, présenté dans [11], propose un modèle où chacun des boids a une probabilité de devenir leader. Il est ici considéré que les leaders ne sont pas prédéterminés. Basé sur des équations différentielles, chaque boid a en effet la même probabilité de devenir leader, mais les leaders peuvent très vite redevenir suiveur s'ils sont leaders depuis longtemps et que leurs plus proche voisin n'est pas proche. Il peut y avoir plusieurs leaders. Ce modèle est stable et permet de reproduire des *flockings* des boids.

2.2 Le leading émergent

Le modèle du *leading* peut ne pas être explicitement implémenté et nous pouvons le constater dans le phénomène du *flocking*. Il s'agit d'un phénomène *émergent* du phénomène *émergent* qu'est le *flocking*. Lorsqu'il y a nuée, les boids finissent par se suivre, et donc suivent indirectement un ou plusieurs meneurs. Les nuées peuvent se diviser en plusieurs nuées, dirigées par différents leaders.

Dans [4] et [5], l'*émergence* de leaders est mesuré grâce au graphe orienté des connexions, illustré précédemment en figure 2.1. Des questions pertinentes peuvent se poser pour mieux comprendre le phénomène du *leading*, comme par exemple la prédisposition d'un boid à devenir un leader selon sa position initiale dans la simulation. Même si leurs nuées sont stables, et même si lorsque la nuée est stable, son leader est stable, ils n'ont pu établir de tel lien entre emplacement spatial et probabilité de devenir leader. Le modèle du *flocking* est manipulable de par ses paramètres, mais reste suffisamment chaotique pour maintenir de telles questions en suspend.

[4] et [5] ont également pu mettre en oeuvre un nouveau modèle permettant l'*émergence* du *flocking* des boids : le modèle Flock Synthesis Rules (FSR). Ce modèle est basé sur les interactions dyadiques entre les boids et les notions de distances réelles et de distances optimales. Une interaction dyadique est une interaction qui concerne exactement deux entités. Il s'agit ici du fait que chaque boid, après avoir déterminé son voisinage comme vu dans le modèle de C.Reynolds [1], va déterminer une distance idéale à laquelle se trouver d'un boid de son voisinage. Il ne s'agira plus de prendre en compte les trois règles du modèle de C.Reynolds, mais de prendre en compte toutes les distances optimales de chacun des boids de chaque voisinage, et de retourner le vecteur déplacement qui permet de satisfaire autant que possible l'ensemble de ces distances à maintenir.

2.3 La prédation

La *prédation* consiste, comme cité précédemment en ouverture de [1] et dans l'introduction de ce chapitre, à ce que les boids considèrent un nouveau type d'obstacle : les prédateurs. Contrairement aux obstacles décrits dans [1] et [2], ici les prédateurs sont mobiles et se déplacent même avec la volonté de rentrer en collision avec les boids. Les boids doivent non seulement adopter une règle de répulsion exercée à la vue d'un prédateur, mais doivent adapter leurs comportements pour éviter en priorité d'être en collision avec les prédateurs.

Précédemment les boids se suffisaient à éviter un obstacle que lorsqu'ils allaient entrer en collision avec. Maintenant, pour leur survie, les boids doivent véritablement fuir le prédateur, comme illustré en figure 2.2². Plusieurs techniques sont envisageables, comme une simple fuite en allant dans la direction opposée ou bien une formation en essaim pour perturber le prédateur ou d'autres encore. Le but étant de minimiser le nombre de collisions effectuées par les prédateurs.

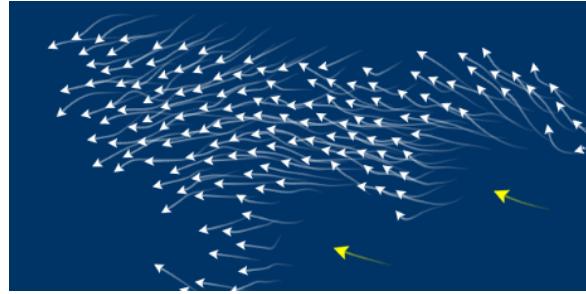


FIGURE 2.2 – Exemple de prédation

Bien que des stratégies différentes peuvent être pensées, la *prédation* fait apparaître une particularité étonnante : l'*émergence* du *flocking* des boids en sa seule présence. En effet, les modèles de [12] et [13] ont montré qu'il est possible de re créer un *flocking* des boids seulement en leur indiquant d'esquiver un prédateur. Les boids suivent un apprentissage par renforcement, comme par exemple le Q-Learning³, afin de survivre au mieux au prédateur, de l'esquiver le mieux possible. De cet apprentissage *émerge* le phénomène de *flocking*. Cela se veut crédible avec la réalité, chaque boid cherche à optimiser sa survie et le mouvement de nuée va perturber le prédateur. Dans ces modèles, l'environnement est en tore et les boids ne décident seulement de l'orientation qu'ils vont prendre au prochain pas de simulation, la vitesse est constante. Le modèle de [12] propose qu'un seul boid apprenne à la fois, et en démontre le fonctionnement en vidéo⁴. Le modèle de [13] propose que tous les boids apprennent en même temps. En définitif, les deux modèles nous démontrent étonnamment que le *flocking* peut *émerger* en réponse à la *prédation*.

Bien des modèles sont implantables à partir de celui de [1], ils ont leurs spécificités, leurs comportements... Mais quelles utilisations concrètes pouvons nous mettre en place ? C'est ce que nous allons aborder dans le prochain chapitre.

2. source : <https://swharden.com/csdv/simulations/boids/>

3. <https://fr.wikipedia.org/wiki/Q-learning>

4. <https://www.youtube.com/watch?v=SY59CYaqWpE>

Chapitre 3

Applications du modèle

Il est tout naturel de se demander quelles pourraient être les applications pratiques de la modélisation d'une nuée d'oiseaux. Dans ce chapitre nous nous focalisons sur les différents usages pratiques du phénomène du *flocking* des boids. Bien évidemment, ce chapitre ne se veut pas exhaustif, mais a pour objectif de donner un ordre d'idée des possibilités offertes par ce modèle.

3.1 Multimédia

Dans l'introduction, nous évoquions les logiciels de simulations Massive¹ et Golaem². Reprenant l'approche de vie artificielle³, ces logiciels ont pu reprendre le principe du modèle des boids de C.Reynolds [1], ajouter des perceptions, des actions possibles et des apparences revisitées aux agents de la simulation, afin de nous offrir de superbes spectacles comme nous pouvons en voir au cinéma. En effet, il était humainement compliqué d'engager des milliers de figurants afin de mettre en œuvre des scènes de batailles de grandes envergures, cette solution informatique a révolutionné les œuvres cinématographiques que nous apprécions aujourd'hui.

1. <https://www.massivesoftware.com/about.html>

2. <https://golaem.com/>

3. https://fr.wikipedia.org/wiki/Vie_artificielle

Il existe un bon nombre de simulateurs de foules différents, mais Massive et Goleam restent les plus connues de par leurs différentes contributions. On parle d'une centaine de films pour le logiciel Massive⁴ qui s'illustre notamment par sa participation au dernier film Avatar⁵. Ses contributions s'étendent aussi aux séries télévisées, et aux simulations de foules pour l'ingénierie et l'architecture. Le logiciel de Goleam lui aussi compte plus d'une centaine de productions⁶ pour le cinéma, les séries télévisées, les publicités, le jeu vidéo et tout autre usage professionnel de simulation de foule.

Le chercheur et foulologue Mehdi Moussaïd⁷ de l'institut Max Planck de Berlin est d'ailleurs un très bon vulgarisateur des sciences des foules et a notamment publié une vidéo⁸ sur la contribution du modèle de Reynolds [1] dans le cinéma. Publiant tant des livres que des articles, tenant une chaîne youtube nommée Fouloscopie, Mehdi Moussaïd contribue grandement à la compréhension des mécaniques des phénomènes de foules, comme illustré en figure 3.1.



FIGURE 3.1 – Fouloscopie

3.2 Drônes

Comme évoqué dans la section 1.3, le modèle a grandement contribué dans le domaine de pilotage de drônes⁹. Ce domaine d'application a notamment donné lieu à de nombreuses recherches sur le modèle de Reynolds, comme cela est le cas dans [7], [6], [8] et [9]. Dans la plupart des cas, il est question d'optimisation des paramètres du modèle de Reynolds [1] afin d'en obtenir la meilleure application possible en situation réelle.

-
- 4. <https://www.massivesoftware.com/film.html#>
 - 5. [https://fr.wikipedia.org/wiki/Avatar_\(La_Voie_de_l%27eau\)](https://fr.wikipedia.org/wiki/Avatar_(La_Voie_de_l'eau))
 - 6. <https://goleam.com/content/customers/gallery>
 - 7. <http://www.mehdimoussaïd.com/>
 - 8. <https://www.youtube.com/watch?v=w-Oy4TYDnoQ>
 - 9. <https://www.youtube.com/watch?v=87y3AsUOCaQ>

Le vol en nuée des drônes est notamment très utilisé dans le domaine de l'exploration et du sauvetage. Les drônes peuvent en effet se déplacer malgré des tremblements de terre ou des fuites de gaz. Ils peuvent faire de la reconnaissance pour des missions de sauvetage, minimisant ainsi le risque de pertes humaines et optimisant la recherche d'individus à secourir. Dans ce domaine d'application, il s'agit de pouvoir s'adapter à des environnements de diverses topologies. Il s'agit donc également de s'assurer du vol groupé des drônes et de leur sûreté, comme énoncé dans le chapitre 1.3 lorsqu'il était question de métrique de sûreté, illustré en figure 3.2¹⁰. S'assurer de la sûreté des drônes leur permet d'effectuer leurs tâches et s'assurer d'un bon usage des paramètres du modèle de Reynolds leur permet d'effectuer efficacement leurs tâches. Lorsque les drônes vont explorer une zone, on favorise une grande aire et une faible polarité. Lorsque les drônes veulent se déplacer dans une direction déterminée, on favorise une faible aire et une grande polarité et on utilise la mécanique du *leading*.

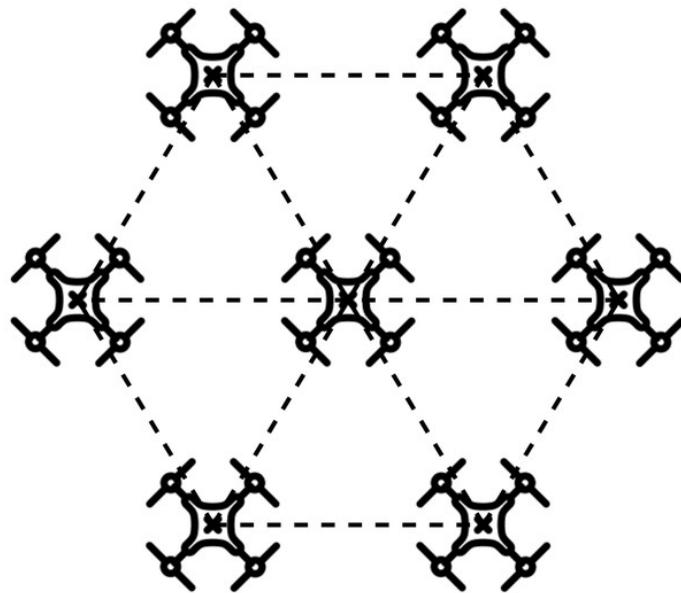


FIGURE 3.2 – Sureté des drônes

Le pilotage des drônes en usant du modèle des boids de Reynolds peut avoir bien d'autres usages. L'un d'entre eux est simplement la représentation artistique, comme illustrée en figure 3.3¹¹. Il ne faut pas oublier que le spectacle du *flocking* est bien satisfaisant à regarder !

10. source : figure 1 de [9]

11. source : figure 5 de [8], crédit : Zsolt Bézsenyi



FIGURE 3.3 – Spectacle de drônes

3.3 Algorithmique

Le modèle des boids de Reynolds a également aidé dans le domaine des problèmes d'optimisation. [14] a conçu un modèle d'optimisation de fonctions à N paramètres : l'optimisation de la nuée de particules, ou Particle System Optimization (PSO), illustrée en figure 3.4¹².

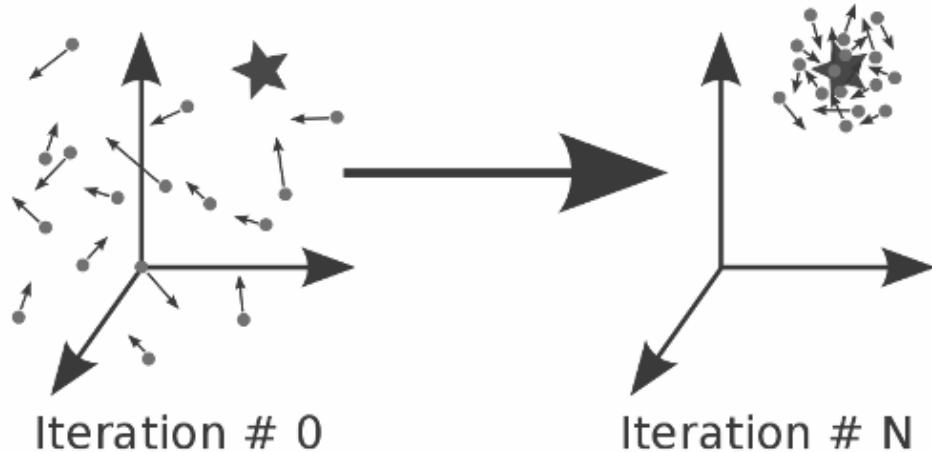


FIGURE 3.4 – Particle System Optimization

12. source : <https://esa.github.io/pagmo2/docs/cpp/algorithms/pso.html>

Le modèle du PSO a été pensé par le principe du "make it simple, make it better", qui dit que plus un algorithme est simple plus il est robuste. Initialement, le modèle s'inspire des boids de Reynolds [1] et y ajoute un mouvement hasardeux pour couvrir davantage de surface. Puis, il est ajouté un objectif spatial, un "épi de maïs", que les boids essaient d'atteindre. Enfin, on retire ce mouvement hasardeux, on retire des règles du modèle et on étend l'objectif non plus à un espace en deux ou trois dimensions mais à un espace à N dimensions. Le PSO est ainsi apparenté à un réseau de neurones, qui a notamment réussi à résoudre des problèmes complexes comme le XOR à 13 dimensions¹³.

3.4 Autres applications

Ce chapitre est non-exhaustif sur les applications pratiques possibles du modèle des boids de C.Reynolds [1]. Nous essayons d'en parcourir un certain panel. Nous concluons donc ce chapitre en ajoutant deux domaines divers auxquels a contribué le modèle : la biologie et la sociologie.

Dans le cadre de la biologie, et plus précisément dans le cadre de l'ornithologie, le modèle a contribué dans l'analyse comportementale du vol des pigeons. C'est dans [15] qu'est justement repris le phénomène de *leading*, abordé dans les sections 2.1 et 2.2. Dans ces travaux, il effectue des mesures GPS sur les pigeons, illustré en figure 3.5¹⁴, durant un vol groupé pour mesurer si un leader est pré-déterminé ou non, et si la présence ou non d'un leader est propice à l'émergence de *flocking*. Il est déterminer que le leader change et n'est donc pas pré-déterminé mais qu'en effet il aide au mouvement du groupe.

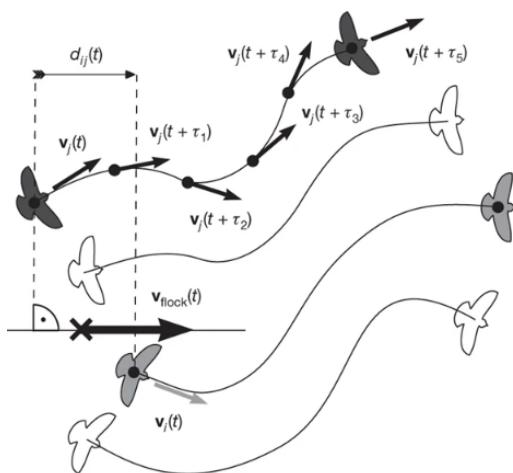


FIGURE 3.5 – Mesures sur des pigeons

13. <https://dev.to/jbahire/demystifying-the-xor-problem-1blk>

14. source : figure 1.a de [15]

Dans le cadre de la sociologie, l'émergence au sein de groupes d'individus est considéré avec intérêt de par ce que cela nous apprend de l'humain et de comment il peut s'organiser. Dans le cadre de la thèse [16], il nous est décrit qu'un groupe d'individus peut se retrouver à effectuer plus ou moins un modèle de Reynolds [1] afin d'organiser une recherche, comme la recherche d'une issue comme illustrée en figure 3.6¹⁵.

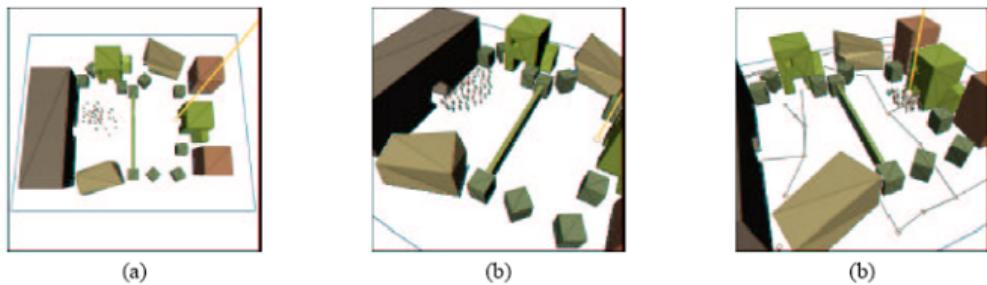


FIGURE 3.6 – Recherche par la foule

L'émergence de *flocking* peut également amener à de nouvelles idées, de nouvelles expressions artistiques et de nouvelles expériences de tout genre. Par exemple, l'improvisation musicale¹⁶ peut se résumer en un groupe d'invidus dont l'action est de jouer de la musique, et que chacun calibre son instrument selon les instruments des autres. [17] et [18] montrent notamment des performances musicales basées sur le modèle des boids, comme illustré en figure 3.7¹⁷.



FIGURE 3.7 – Blues joué avec les boids

15. source : figure 5 de [16]

16. https://fr.wikipedia.org/wiki/Improvisation_musicale

17. source : vidéo récupérée de <http://www.intlab.soka.ac.jp/unemi/1/DT1/>

Conclusion

Dans ce travail de mémoire nous avons pu décrire de manière formelle le modèle de C.Reynolds [1], en expliciter les différents paramètres qui peuvent influencer le phénomène et les tester dans le simulateur implémenté dans le cadre du Projet Individuel (PJI) [2]. Le prototype a permis de valider expérimentalement le modèle de C.Reynolds. Il nous a également permis d'expérimenter l'influence des paramètres sur la simulation. Les paramètres identifiés peuvent donc en effet modifier l'aire et la polarité de la nuée : bien que restant chaotique, il est possible de manipuler le *phénomène émergent* selon le cas d'usage souhaité.

Nous avons pu parcourir certaines ouvertures évoquées par C.Reynolds en étudiant des modèles de *leading* et de *prédition*. L'apparition de *leader* s'est avérée être un phénomène *émergent* du *flocking* des boids. Nous avons également constaté, de par les propositions de nouveaux modèles, que le *leading* ainsi que la *prédition* peuvent favoriser l'*émergence* du *flocking* des boids.

Enfin, nous avons parcouru un large panel d'applications concrètes du modèle des boids de C.Reynolds. Le cinéma, l'optimisation algorithmique, la vie artificielle, la robotique pour de l'exploration et du sauvetage, la biologie, la sociologie, l'expressivité artistique et bien d'autres domaines ont été grandement impactés par ce modèle. Ce mémoire n'a pas pour ambition d'être exhaustif quant aux applications possibles du modèle. Il cherche surtout à exprimer l'idée que ce modèle reste et restera une source d'inspirations tant scientifiques qu'artistiques. L'usage du *flocking* des boids pourrait servir à automatiser certains processus d'improvisation au sein de groupes d'individus.. ? On peut même se demander si d'autres phénomènes *émergents* au sein de groupes d'individus peuvent avoir autant de cas d'usages que celui du *flocking*. Les phénomènes *émergents*, comme celui du *flocking*, constituent donc toujours un grand sujet ouvert à la recherche. Le modèle des boids de C.Reynolds n'a pas fini de nous rendre "tête en l'air", à nous inviter à contempler le doux spectacle des oiseaux volants dans le ciel...

Références

- [1] C. W. Reynolds, “Flocks, herds, and schools : A distributed behavioral model.” Computer Graphics, Available at <https://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/>, July 1987. [Cité en pages i, iii, 1, 5, 11, 13, 14, 16, 18, 20, 21, 25, 27, 28, 29, 30, 33, 34 et 35]
- [2] A. Nollet, “Flocking-simulator.” Computer Science project for Master Degree, Availble at <https://gitlab.univ-lille.fr/antoine.nollet.etu/flocking-simulator>, June 2022. [Cité en pages i, iii, 12, 20, 22, 23, 27 et 35]
- [3] W. T. REEVES, “Particle systemsma technique for modeling a class of fuzzy objects.” ACM Transactions On Graphics, Availble at <https://dl.acm.org/doi/pdf/10.1145/357318.357320>, July 1983. [Cité en page 5]
- [4] F. S. B. Vicenç Quera, Salvador Herrando, L. Salas, and M. Minano, “An index for quantifiying flocking behavior.” Perceptual and Motor Skills, Availble at <http://deposit.ub.edu/dspace/bitstream/2445/34615/1/552693.pdf>, Oct. 2007. [Cité en pages 20, 22 et 27]
- [5] R. D. Vicenç Quera, Francesc S. Beltran, “Flocking behaviour : Agent- flocking behaviour : Agent-based simulation and hierarchical leadership.” Journal of Artificial Societies and Social Simulation, Available at <http://jasss.soc.surrey.ac.uk/13/2/8.html>, Mar. 2010. [Cité en pages 20, 25 et 27]
- [6] F. S. Enrica Soria and D. Floreano, “The influence of limited visual sensing on the reynolds flocking algorithm.” 2019 Third IEEE International Conference on Robotic Computing (IRC), Availble at https://infoscience.epfl.ch/record/264152/files/soria_influence_2018.pdf, Feb. 2019. [Cité en pages 20, 21, 22 et 30]
- [7] V. R. J.D. Eversham, “Experimental analysis of the reynolds flocking model.” IEEE 9th International Conference on Cybernetic Intelligent Systems, 2010, Reading, UK, Availble at <https://www.degruyter.com/document/doi/10.2478/s13230-012-0001-8/pdf>, Jan. 2012. [Cité en pages 21, 22, 23 et 30]
- [8] G. S. Gábor Vásárhelyi, Csaba Virágħ, T. Nepusz, A. E. Eiben, and T. Vicsek, “Optimized flocking of autonomous drones in confined environments.” Science Robotics, Availble at <https://hal.elte.hu/~vasarhelyi/doc/vasarhelyi2018optimized.pdf>, July 2018. [Cité en pages 24, 30 et 31]

- [9] T.-S. T. Duc N. M. Hoang, Duc M. Tran and H.-A. Pham, “An adaptive weighting mechanism for reynolds rules-based flocking control scheme.” PeerJ Computer Science, Available at <https://peerj.com/articles/cs-388/>, Feb. 2021. [Cité en pages 24, 30 et 31]
- [10] K. J. P. Sara Bernardi, Raluca Eftimie, “Leadership through influence : What mechanisms allow leaders to steer a swarm ?” Bulletin of Mathematical Biology, Available at <https://doi.org/10.1007/s11538-021-00901-8>, May 2021. [Cité en page 26]
- [11] M. P. Emiliano Cristiani, Marta Menci and L. Brafman, “An all-leader agent-based model for turning and flocking birds.” Journal of Mathematical Biology (2021) 83 :45, Available at <https://doi.org/10.1007/s00285-021-01675-2>, Oct. 2021. [Cité en page 26]
- [12] C. Hahn, T. Phan, T. Gabor, L. Belzner, and C. Linnhoff-Popien, “Emergent escape-based flocking behavior using multi-agent reinforcement learning.” ALIFE 2019, Available at <https://arxiv.org/pdf/1905.04077.pdf>, May 2019. [Cité en page 28]
- [13] K. Morihiro, H. Nishimura, T. Isokawa, and N. Matsui, “Learning grouping and anti-predator behaviors for multi-agent systems.” 12th International Conference, KES 2008, Zagreb, Croatia, Proceedings, Part II 12, Available at https://www.researchgate.net/profile/Haruhiko-Nishimura/publication/226881555_Learning_Grouping_and_Anti-predator_Behaviors_for_Multi-agent_Systems/links/576e26ea08ae842225a40dee/Learning-Grouping-and-Anti-predator-Behaviors-for-Multi-agent-Systems.pdf, Sept. 2008. [Cité en page 28]
- [14] J. Kennedy’ and R. Eberhart, “Particle swarm optimization.” IEEE international conference on neural networks, 1995, Available at https://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6%201995%20particle%20swarming.pdf, Nov. 1995. [Cité en page 32]
- [15] D. B. Máté Nagy, Zsuzsa Ákos and T. Vicsek, “Hierarchical group dynamics in pigeon flocks.” Nature, Available at <https://www.nature.com/articles/nature08891>, Apr. 2010. [Cité en page 33]
- [16] D. Rosen, “Flock theory : Cooperation and decentralization in communication networks, p23-53.” A Dissertation Presented to the Faculty of the Graduate School of Cornell University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, Available at https://ecommons.cornell.edu/bitstream/handle/1813/7656/Devan%20Rosen_Dissertation_4thSubmit.pdf, Aug. 2007. [Cité en page 34]

- [17] C. Hongchan and W. Ge, “Lush : An organic eco+ music system..” NIME 2010, Available at <https://www.gewang.com/publish/files/2010-nime-lush.pdf>, 2010. [Cité en page 34]
- [18] U. Tatsuo and B. Daniel, “Playing music by conducting boid agents-a style of interaction in the life with a-life.” Proceedings of A-Life IX, Available at <http://www.intlab.soka.ac.jp/~unemi/1/DT1/ALife9Unemi.pdf>, 2004. [Cité en page 34]