REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : Numéro/M2/2019

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Image et Vie Artificielle

Simulation du comportement de mouvement en groupe avec prédateurs

---

Par :

Djouadi Sara

Soutenu le date septembre 2020, devant le jury composé de :

Nom et prénom	Grade	Président
Amina Bouguetitiche	MAA	Rapporteur
Nom et prénom	Grade	Examinateur

Remerciements

La première et dernière chose est pour Allah qui me fournit la capacité suffisante pour terminer ce travail.

Je n'oublierai pas de remercier les membres du jury qui ont accepté de lire et d'examiner ce travail.

J'aimerais remercier aussi mon encadreur, Mme Bouguetitiche Amina pour les conseils, les encouragements prodigués tout au long de mon parcours en tant qu'étudiante.

J'ai eu beaucoup de chance d'avoir un encadreur qui se souciait tellement de mon travail et qui a répondu à mes questions et requêtes si rapidement.

Je suis infiniment reconnaissante envers mes enseignants, qui m'ont fait profiter de leur savoir, qui m'ont encouragée et aidée durant mon cursus universitaire.

Je tiens à remercier toute ma famille sans exception, en particulier ma mère, mon mari, mes frères et ma belle sœur, Amina pour leur soutien continu afin de mener à bien ce travail.

Un merci spécial à tous ceux qui m'ont soutenu pour terminer ce travail.

Tables des matières

Tables des figures

Tables des tableaux

Résumé

Parmi les études menées par les chercheurs sur les systèmes vivants, le comportement collectif des animaux qui se déplacent en groupe, c'est un domaine fascinant qui analyse comment de simples actions d'un individu affectent la dynamique globale complexe d'un groupe. Des exemples typiques de comportement de groupe sont les troupeaux d'oiseaux, les troupeaux de poissons et les troupeaux d'insectes. Comme beaucoup de simulations de vie artificielle, il existe un programme informatique pour la vie artificielle (appelé Boids qui provient de Bird-oid) qui se présente sous la forme d'un oiseau, qui a été développé par Craig Reynolds en 1986, pour simuler le comportement d'une volée d'oiseaux en vol. Il a une forme géométrique qui lui est attachée pour l'affichage et son mouvement, est contrôlé par le comportement individuel. En fait, la complexité comportementale résulte ici de l'interaction de facteurs individuels tout en respectant un nombre limité de règles de Reynolds (séparation, cohérence, alignement). Notre objectif dans ce travail est de simuler le comportement collectif des poissons en présence de prédateurs en utilisant les règles que nous avons évoquées.

Abstract

Among the studies conducted by researchers on living systems, the collective behavior of animals that move collectively, this study is a fascinating area that analyzes how simple actions of an individual affect the complex overall dynamics of a group. Typical examples of group behavior are flocks of birds, flocks of fish and flocks of insects. Like many artificial life simulations, there is a computer program for artificial life (called Boids that comes from Bird-oid) that takes the form of a bird, which was developed by Craig Reynolds in 1986, to simulate the behavior of a flock of birds in flight. It has a geometric shape attached to it for display and its movement is controlled by individual behavior. In fact, the behavioral complexity here results from the interaction of individual factors while respecting a limited number of Reynolds rules (separation, coherence, alignment). Our objective in this work is to simulate the collective behavior of fish in the presence of predators using the rules that we have mentioned.

ملخص

من بين الدراسات التي أجراها الباحثون على الأنظمة الحية ، السلوك الجماعي للحيوانات التي تتحرك بشكل جماعي ،حيث تعد هذه الدراسة مجالًا رائعًا يحلل كيف تؤثر الإجراءات البسيطة للفرد على الديناميكيات الشاملة المعقدة للمجموعة. الأمثلة النموذجية لسلوك المجموعة هي قطعان الطيور وأسراب الأسماك وأسراب الحشرات. مثل العديد من محاكاة الحياة الاصطناعية، هناك برنامج كمبيوتر للحياة الاصطناعية (يسمى Boids) يأتي من Bird-oid الذي يأخذ شكل طائر، والذي طوره Craig.Reynolds في عام 1986 ، لمحاكاة سلوك سرب من الطيور في الرحلة. لها شكل هندسي متصل بها للعرض ويتم التحكم في حركتها من خلال السلوك الفردي. في الواقع ، ينتج التعقيد السلوكي هنا عن تفاعل العوامل الفردية مع احترام عدد محدود من قواعد رينولدز (الفصل ، التماسك ، المحاذاة). هدفنا في هذا العمل هو محاكاة السلوك الجماعي للأسماك في وجود الحيوانات المفترسة باستخدام القواعد التي ذكرناها.

Introduction Générale

L'étude des systèmes artificiels illustre les comportements caractéristiques des systèmes de vie naturels. Son objectif est de créer des systèmes artificiels inspirés des systèmes vivants (animaux, plantes, insectes), que ce soit sous forme de programmes informatiques ou sous forme de robots. Parmi les études menées par les chercheurs sur les systèmes vivants, il y a l'étude du comportement des troupeaux de poissons et d'oiseaux. L'étude du comportement de groupe dans les troupeaux est l'un des principaux domaines analysant comment des actions individuelles simples affectent la dynamique. Un ensemble universel complexe. Comme beaucoup de simulations de vie artificielle, Reynolds a simulé ce phénomène grâce à le programme Boids, c'est un programme informatique de vie artificielle, développé par Craig W. Reynolds en 1986, simulant le comportement d'une nuée d'oiseaux en vol. Le mot boid est par ailleurs une contraction de bird-oid (qui a la forme d'un oiseau), ce programme Boids permet de modéliser les comportements émergents. En fait, la complexité comportementale résulte ici de l'interaction de facteurs individuels (appelés boids) tout en respectant un nombre limité de règles simples où les fondements d'un groupe de comportements sont divisés en deux types, le comportement individuel et le comportement de groupe.

Sur la base de ce travail réalisé par Reynolds, nous cherchons à simuler le comportement émergent d'un groupe de poissons à l'aide de trois règles très simples où nous ajouterons une nouvelle entité, qui est le Prédateur, et pour cela nous aurons simulé deux types de personnages: Proie et Prédateur.

Pour atteindre l'objectif que nous sommes assignés, nous avons organisé notre mémoire selon la structure suivante :

- Le premier chapitre est un chapitre introductif sur la simulation des comportements, la modélisation comportementale et la modélisation de l'environnement virtuel.

- Le deuxième chapitre est tiré de la simulation comportementale, on y parle des modèles que Reynolds nous a donnés dans lesquels il illustre la simulation du comportement individuel et collectif à travers le programme Bird-Oid.

- La conception de notre application fera l'objet du troisième chapitre. Ce dernier explique la conception globale et détaillée de notre système, ainsi que chaque composante constituant le système.

- Dans le dernier chapitre, nous présentons l'implémentation et la réalisation de notre travail.

Enfin, nous terminons ce mémoire par une conclusion générale de notre projet et par différentes perspectives de recherche qui nous semblent intéressantes pour continuer ce travail.

1 Chapitre I
Simulation comportementale

1.1 Introduction

Les deux concepts, comportement et environnement, sont étroitement liés. En effet, le comportement découle de l'interaction de l'organisme avec l'environnement dans lequel il évolue, son environnement pouvant être constitué de types d'organismes. Une entité autonome, par exemple, est constamment dans une boucle d'interaction avec son environnement. Il peut utiliser ses capteurs (yeux, nez, oreilles, etc.) pour obtenir des données, c'est-à-dire ses effets (mains, pieds, etc.) Une personne doit prendre en compte de nombreux paramètres lorsqu'elle interagit avec son environnement. Prenant son déplacement comme exemple, avant de pouvoir le faire, il doit imaginer cet environnement. Cette représentation mentale sera en fait utilisée pour contempler cet environnement, que ce soit pour évaluer un chemin pour atteindre un lieu précis, ou pour organiser les tâches qu'il doit effectuer (comme les interactions avec des objets).

1.2 Animation Comportementale

L'animation comportementale cherche à offrir un comportement cohérent proche de l'organisme vivant simulé. Une entité perçoit l'environnement, décide de la prochaine action à exécuter et agit sur le monde. C'est important de remarquer que le processus décisionnel réalisé par l'entité peut être réactif ou cognitif. Avec un processus décisionnel réactif, l'entité choisit une action de manière inconsciente, c'est- adire, sans l'intervention d'aucune connaissance ni expérience. A l'opposé, dans un processus cognitif toutes les connaissances et expériences antérieures de l'entité influent sur sa décision ; ce type de processus permet de sélectionner une suite d'actions cohérentes visant à atteindre un but donnée et l'entité est alors capable de raisonner sur les conséquences de ses actions [Don04].

1.3 Simulation Comportementale

La Simulation Comportementale a pour finalité de permettre à une ou plusieurs personnes de réaliser des activités diverses dans des environnements ou mondes virtuels créés à partir de certains aspects du monde réel ou à partir de mondes imaginaires. [ABW06] les entités virtuelles sont interactives, adaptatives et autonomes. Interactives dans le sens où elles doivent pouvoir interagir avec d'autres entités, les objets de l'environnement et l'utilisateur de la façon la plus naturelle possible ,adaptatives dans le sens d'être capable d'agir conformément à des modifications introduites soit par l'utilisateur soit par des interactions antérieures ; et autonomes dans le sens ou elles doivent pouvoir décider parmi toutes les actions possibles à exécuter celle qui est la plus adéquate selon les conditions actuelles de l'environnement et les actions de l'utilisateur.

1.4 Le Comportement

Le mot comportement peut avoir plusieurs significations. Cela peut signifier l'action complexe d'un humain ou d'un animal basée sur la volonté ou l'instinct [Rey99]. Là où le fait demeure que la principale caractéristique qui nous rend humains est notre capacité à penser. Cette capacité de penser réside dans l'analyse et la réflexion. En fait, chaque être humain analyse le monde extérieur dans lequel il se développe à travers ces cinq sens. Après avoir analysé cet environnement, vous entrez dans la phase de réflexion pour guider nos décisions. Les sens parlent à notre cerveau de tout ce qui nous entoure. Par conséquent, penser à analyser nos sens guide nos décisions. Il est possible que nous agissions sur notre environnement grâce à nos corps et surtout nos extrémités (influences) [Avr08].

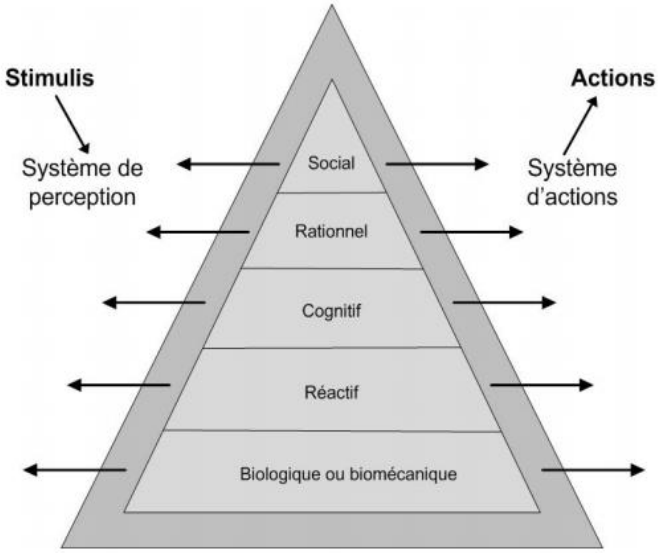


Figure 1 Pyramide décisionnelle.

Principe de décision

En fonction des informations perçues dans son environnement, l'être humain connait les actions qu'il peut y effectuer. Désormais il doit en choisir une à réaliser et pour cela le raisonnement entre en jeu. Newel propose dans ses travaux sur l'analyse de l'architecture cognitive, un modèle hiérarchique à cinq niveaux [New90]. Dans cette représentation chaque niveau de raisonnement est caractérisé selon deux critères couplés : le temps de réalisation des différentes actions et la complexité. Ils possèdent chacun des relations avec les niveaux supérieurs et inférieurs. La décomposition de Newel peut se représenter sous forme pyramidale et est définie comme suit (voir Figure 1) :

a. **Le niveau biomécanique :** Il correspond aux fonctions de base, comme le mouvement des muscles. Il a des actions allant des 100 µs à 10 ms.
b. **Le niveau réactif :** Il concerne les comportements réflexes et ceux ne nécessitant pas de connaissance.

c. **Le niveau cognitif :** Il représente les comportements utilisant la connaissance pour raisonner. Il peut être décomposé en plusieurs sous-niveaux :
    - Action délibérée permet de trancher entre une opération plutôt qu'une autre via les informations issues l'environnement.
    - Opération permet d'effectuer des opérations déjà existantes.
    - Tâche unitaire permet la composition d'opérations non élémentaires.
d. **Le niveau rationnel :** Il définit les tâches à effectuer et les priorités entre celles-ci.

Les actions peuvent aller d'une minute à plusieurs heures.

e. **Le niveau social :** Il est le sommet de la pyramide, c'est donc le niveau dépendant de tous les autres niveaux, par conséquent le plus complexe. Il situe les êtres humains entre eux, d'un point de vue social et relationnel. Ses actions peuvent aller d'un jour à plusieurs semaines voire plusieurs mois. Dans notre sujet, nous nous intéressons par exemple aux règles sociales régissant le comportement des piétons [Tho99] telles les règles piétonnières jouant, par exemple, un rôle important dans les étapes de navigation en groupe.

1.5 Modélisation comportementale

La modélisation comportementale est la modélisation du comportement, c'est-à-dire la détection et la mesure des éléments du comportement afin d'en réaliser un modèle mathématique. Elle peut concerner tant des systèmes physiques que des comportements humains individuels et collectifs et est utilisée notamment comme outil prévisionnel dans divers domaines d'activité. Autant cette modélisation peut être fiable pour prévoir et contrôler le comportement d'un système physique, autant les incertitudes affectant les comportements humains (distorsion des courbes probabilités, inversion de préférences, seuils d'émergence...) rend ces modèles impossibles.

1.6 L'autonomie

L'autonomie permet à un agent de produire des actions à partir de perceptions, et éventuellement d'états internes ou de mémoire. Aucune interférence extérieure n'est requise pour contrôler l'agent. Cette propriété est commune à toutes les entités d'animation comportementale. Une entité autonome se développe dans son environnement, elle continue à avoir une interaction consciente ou inconsciente avec elle. Ces relations entre l'entité autonome et le monde extérieur sont présentées sous la forme d'un «boucle de perception et de décision» (voir Figure 2) [Mar06].

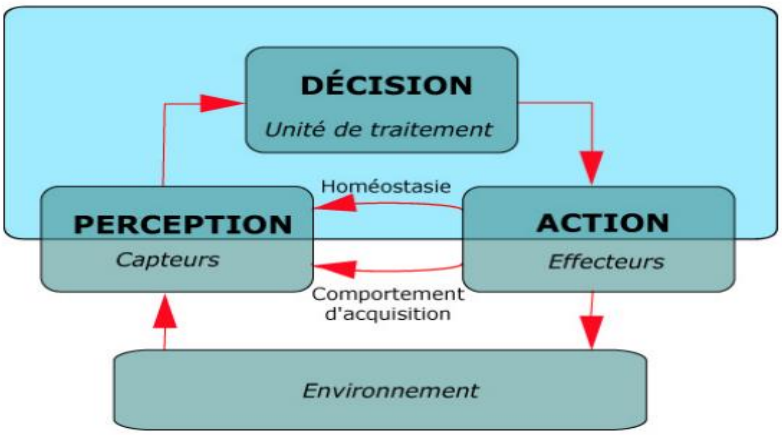


Figure 2 La boucle perception-décision-action.

L'entité autonome dispose d'une interface avec le monde extérieur comprenant notamment des capteurs ; il s'agit des cinq sens traditionnels et des capteurs internes du corps. Les informations reçus sont analysées, combinées et stockées par le centre décisionnel : l'unité de traitement. Celle-ci fonctionne de manière autonome, elle est isolée du reste du monde. La décision prise est ensuite exécutée grâce aux effecteurs. On regroupe sous ce terme les organes capables d'interagir avec l'environnement. Von Uexküll se base sur cette interaction pour définir l'environnement de l'entité autonome comme la partie du monde extérieur avec laquelle il peut naturellement interagir. L'environnement n'est donc que la partie locale à l'entité autonome du monde extérieur, celle à portée de ses capteurs et de ses effecteurs.

La figure 2 présente, en outre, trois boucles de retour :

- **Les interactions avec l'environnement :** il s'agit de la boucle de retour concernant les actions et la perception de l'environnement.
- **L'homéostasie :** elle représente la régulation interne du corps, c'est-à-dire les maintiens des constantes biologiques.
- **Les comportements d'acquisitions :** il s'agit des comportements permettant d'affiner la perception d'une partie de l'environnement.

1.7 Les techniques pour modéliser un comportement

1.7.1 Systèmes réactifs

Cette section présente différents systèmes mis en œuvre pour les modèles décisionnels. Il existe trois grandes catégories de modèles réactif : Cette section présente différents systèmes mis en œuvre pour les modèles décisionnels. Il existe trois grandes catégories de modèles réactif : Les systèmes stimuli-réponses, ceux à bases de règles et ceux à bases d'automates. Ces derniers sont apparus dans le but de créer des comportements de plus en plus proches du comportement humain. Ils sont

rapides et permettent de réagir rapidement aux changements environnementaux [Avr08].

a. Systèmes stimuli-réponses

Ce type de système est généralement représenté sous la forme d'un réseau de neurones. Leur principe se base sur l'interconnexion d'un grand nombre de neurones possédant des entrées et des sorties. Ces réseaux de neurones peuvent être représentés en couches avec ou non des boucles entre des couches. L'entrée correspond à la perception et la sortie à l'action. En fonction des neurones « stimulés » par la perception de l'environnement, une réponse en sortie détermine l'action à entreprendre. Ce système est fortement inspiré de l'étude biologique de l'homme [Avr08].

b. Systèmes à bases de règles

Dans ce type d'approche, les différents traits du comportement sont modélisés via un ensemble de règles. Par définition ces règles sont associées à une ou plusieurs pré-conditions qui, si elles sont respectées associent un comportement à adopter. Afin d'éviter l'exécution de plusieurs règles au même moment il existe des méthodes de choix de règles, permettant ainsi d'éviter les conflits [Avr08].

c. Systèmes à bases d'automates

Vouloir mener à bien une tâche précise consiste la plupart du temps à enchainer plusieurs tâches unitaires afin d'y parvenir. Prenons par exemple « Je dois finir ma bibliographie ». Afin de parvenir à ce but je dois effectuer plusieurs sous- tâches unitaires telles que « je vais à mon bureau, je prends mon crayon, j'allume mon ordinateur, je lis mes articles, j'écris ». Un modèle permettant l'enchaînement de ces tâches est l'automate. Les états de ce dernier représentent les actions (tâches unitaires) effectuables et les arcs reliant ces états représentent les conditions avec lesquelles il est possible de changer de tâches unitaires telles que « L'ordinateur est déjà allumé ». Plusieurs méthodes utilisant ces automates ont déjà été mise en œuvre [Lam03].

1.7.2 Systèmes cognitifs et orientés but

Ils nécessitent une représentation des connaissances de l'entité afin de calculer un enchainement d'actions permettant d'atteindre un but donne. Les différents modèles qui vous seront présentés diffèrent par leur coût de calcule, Leur façon dont les connaissances sont représentées, leur réactivé aux changements dans l'environnement ainsi que leur capacité à exploiter des opportunités [LD09].

a. **Le calcul situationnel:** Proposé par McCarty et P.J Hayes en 1969, il permet de décrire des mondes complexes ainsi que les comportements (actions) des entités qui les peuplent. Pour cela, il se base sur les éléments suivants:
   1. **Situations**: états complets du monde virtuel à un instant donné.
   2. **Les fluents**: fonctions qui permettent de décrire une propriété dynamique du monde.
   3. **Les causalités**: relations cause-effets utilisées pour déduire l'état d'un fluent (propriété) en fonction de l'état d'autres fluents.
   4. **Les actions**: sous la forme pré conditions => effets elles modifient une situation.
   5. **Les stratégies**: enchainements d'actions en fonction de certains raisonnements.
   6. **Les connaissances**: elles modélisent si une entité connait ou non l'état de certains fluents du monde.

A partir de ces éléments, une phase de planification à lieu donnant comme résultat la suite d'actions permettant de générer une situation du monde qui satisfait un but donne. Pendant cette phase tous les mondes possibles sont explores, cela suppose un temps de recherche considérable dans les cas ou l'ensemble des actions possibles est tr`es large.

b. **STRIPS – STandford Research Institute Planning System:** Du a la contrainte de temps de recherche présente dans le calcul situationnel, R.E. Fikes et al. Ont proposé en 1971une version simplifiée de ce calcul. Dans cette nouvelle version la notion des fluents disparaît et l'état du monde est exprimé en fonction des faits présents ou absents dans une situation particulière. Cette modification rend les algorithmes utilisés dans la recherche du monde attendu plus efficaces.

c. **Les réseaux de tâches hiérarchiques – HTN :**

Dans cette approche la planification du comportement d'une entité est faite hiérarchiquement en prenant en compte trois concepts de base :

1. **tâche:** elles équivalent aux buts a atteindre. Une tâche correspond a la satisfaction d'un ensemble de propriétés dans le monde, a la réalisation d'un ensemble d'actions ou `a une combinaison des deux [LD09].
2. **Méthode:** c'est la manière avec laquelle il est possible d'accomplir une tâche. Une méthode peut être une suite de sous- tâche ou d'actions. Comme dans les approches déjà décrites, avant d'exécuter une méthode, un certain état du monde (conditions) doit être vérifie.
3. **Actions:** sous la forme pré conditions => effets elles seront immédiatement réalisées si leur pré conditions est vérifié.

La planification consiste à décomposer la tâche cible en sous- tâche en appliquant les méthodes proposées. Cette décomposition donne comme résultat un type particulier d'arbre nommé arbre et/ou, ou chaque nœud peut correspondre soit a une succession

de (tâche et/ou) (nœud et) soit au choix entre les méthodes qui peuvent être utilisées pour la réalisation d'une même tâche (nœud ou). Le comportement final de l'entité consiste en une suite de (nœuds et/ou) de telle façon que les actions décrites dans chaque nœud permettent d'accomplir la tâche cible.

d. Les mécanismes de sélection d'actions: Le fonctionnement de ces systèmes se base sur une rapide réaction face aux changements de l'environnement ainsi que sur la possibilité d'utiliser au mieux ces changements. Le comportement final de l'entité est construit de fa¸con incrémentale, c'est-à-dire, `a chaque instant l'action qui promet la meilleure convergence vers le but final est choisie en prenant en compte les changements de l'environnement et l'acquisition de nouvelles connaissances [LD09].

1. **Les mécanismes de sélection d'action** : ce type de mécanisme possède deux objectifs principaux : Chercher une suite d'actions possibles pour atteindre le but fixé et saisir les opportunités qu'offre l'environnement.
2. **La planification d'action** : STRIPS et le calcul situationnel sont tout les deux des formalismes permettant l'analyse du monde et offrant ainsi certain raisonnement sur les différents changements d'états de ce monde lors de l'exécution d'une action quelconque. Dans ce type de formalisme, l'environnement est décrit à l'aide de propriétés présentes ou non au sein de celui-ci. Pour cela, on utilise un langage de description des actions, celles-ci possèdent un pré condition et plusieurs effets. Les effets découlant des pré-conditions se décomposent en deux listes : les propriétés à ajouter dans le monde où évolue l'entité autonome et les propriétés à supprimer de ce monde. Le calcul situationnel permet d'obtenir plusieurs effets différents d'une même action lorsque les conditions d'exécution varient. Ce qui entraîne pour ce formalisme un algorithme très complexe. A l'inverse les actions du STRIPS produisent toujours les mêmes effets, l'algorithmique qui en découle est donc plus simple [Avr08].

1.8 Environnement Virtuel

Un environnement virtuel peut être considéré comme un modèle de trois dimensions d'Environnements réels ou imaginaires que l'on peut visualiser et avec lesquelles on peut interagir en temps-réel (Figure3). Des informations sensorielles complémentaires (sonores, tactiles, ...) peuvent venir enrichir ce modèle. Pour Slater et al cet environnement virtuel devient immersif lorsque la représentation virtuelle du corps de l'utilisateur (tout ou partie) est affichée dans l'environnement virtuel, et qu'il peut réaliser un certain nombre d'actions sur cet environnement (à l'aide de capteurs de position et d'orientation, de gants de données).

En 1992, Zeltzer propose un nouveau tripler composé cette fois-ci de l'autonomie, de l'interaction et de la présence. L'autonomie désigne le comportement des entités virtuelles présentes dans l'environnement virtuel. L'interaction pointe la capacité d'interagir avec ces entités et l'environnement. La présence est ici associée à la stimulation sensorielle. Autonomie, interaction et présence sont vues comme les dimensions définissant l'espace des environnements virtuels [OOM92].

Figure 3 Environnements Virtuels.

1.9 Modélisation d'Environnement Virtuel

L'environnement virtuel peut être modélisé selon différents points de vue :

- un espace cartésien, permettant de construire un modèle d'environnement géométrique similaire au nôtre par la représentation des surfaces ou des volumes des obstacles.
- un espace des configurations où chaque dimension est liée à un degré de liberté du robot.

Les différents types de modèles de l'environnement virtuel sont possibles selon les informations que l'on souhaite exploiter. On peut distinguer les niveaux d'information présents dans la littérature en trois grands types :

- le niveau géométrique.
- le niveau topologique.
- et le niveau sémantique.

1.9.1 Modélisation géométrique et topologique

Pour synthétiser la représentation de l'espace libre, deux grandes familles de techniques existent : la décomposition de l'espace libre en cellules géométriques. Les différents modèles introduits dans cette section sont illustrés sur l'environnement 2D carré (encombré de 2 obstacles) introduit dans la (Figure 4).

Une description topologique de l'environnement permet de représenter et d'organiser l'environnement de simulation. Cette description peut être plus ou moins fine, exacte ou approximative, et peut éventuellement contenir des informations sémantiques. Les algorithmes utilisés en animation comportementale pour représenter l'environnement sont étroitement liés à ceux employés en robotique, domaine où cette représentation tient un rôle prépondérant.

a. Décomposition en cellules

Cette technique consiste à diviser l'environnement en cellules géométriques élémentaires. Les portions d'espace ainsi définies sont qualifiées comme libres ou occupées (même partiellement) par des obstacles. La partie libre est alors constituée de l'ensemble des cellules libres. On ne considère alors plus une partie continue, mais un ensemble de portions d'espace continu dont on connait la connectivité. On trouve ici une grande variété de décompositions:

Figure 4 Environnement 2D**.**

b. Représentations exactes de l'environnement

Les représentations exactes de l'environnement vont chercher à organiser les données spatiales tout en conservant intégralement les informations qu'elles contiennent à l'origine. La méthode utilisée consiste généralement à découper l'espace navigable en cellules convexes de différentes formes. Il existe plusieurs modèles pour effectuer cette subdivision :

- **Triangulation de Delaunay** : Créer un ensemble de triangles en réunissant des points fournis en entrée (Figure 5).

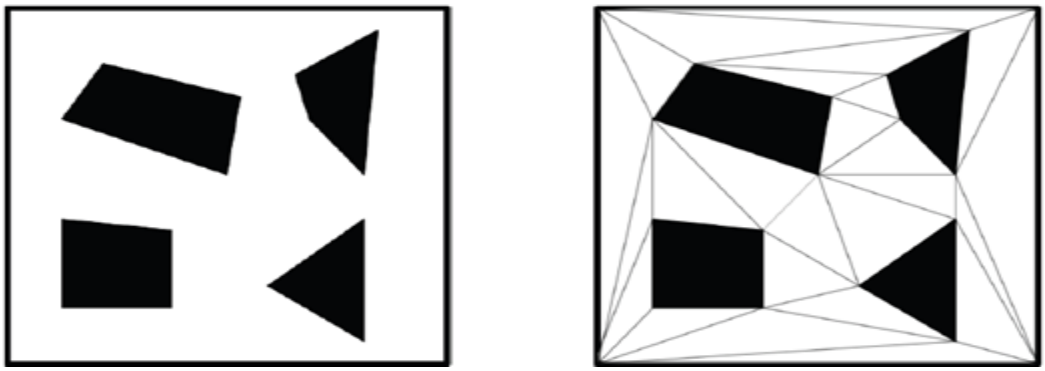


Figure 5 Triangulation de Delaunay.

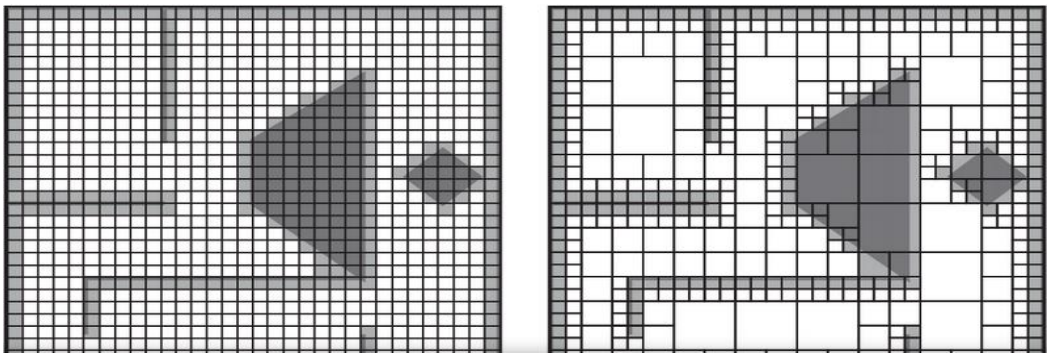
Ces représentations reproduisent exactement l'environnement d'origine tout en l'organisant de manière plus accessible. Et pour des environnements à géométrie complexe, notamment contenant des courbes, le nombre de triangles obtenus augmente très rapidement.

c. Représentations approximatives de l'environnement

Les représentations approximatives de l'environnement sont les plus utilisées en animation comportementale, du fait de leur facilité de mise en œuvre. Ces représentations vont décrire l'espace libre – de navigation – avec des formes géométriques simples telles que des carrés ou des segments. Deux modèles entrent dans cette catégorie : les modèles à base de grilles, et les cartes de cheminement.

- **Les modèles à base de grilles**

Le premier modèle de représentation approximative utilise des grilles régulières, formées de cellules carrées en deux dimensions et cubiques en trois dimensions (Figure 6). L'environnement est donc pavé par ces cellules, qui peuvent avoir trois états : libre, partiellement obstruée, et obstacle.

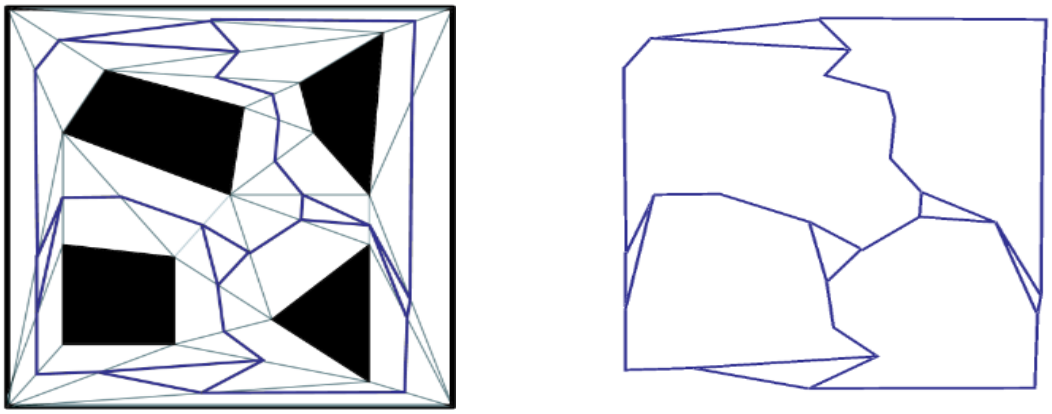


(a) Grille régulière composée de cellules carrées. (b) Grille régulière

Figure 6 couplée à un quad-tree afin de diminuer le nombre d'éléments pour représenter l'environnement.

- **Les cartes de cheminement**

Les cartes de cheminement discrétisent l'espace navigable sous la forme d'un réseau de chemins. Ce réseau est obtenu en reliant des points clefs répartis à l'intérieur de l'environnement (Figure 7 Exemple de carte de cheminement. A gauche, triangulation de Delaunay (gris) de l'environnement d'origine, et carte de cheminement déduite (bleu). A droite, un exemple de carte de cheminement obtenue.). Plusieurs méthodes existent pour créer des cartes de cheminement, différentes dans la manière de créer et de relier ces points clefs.

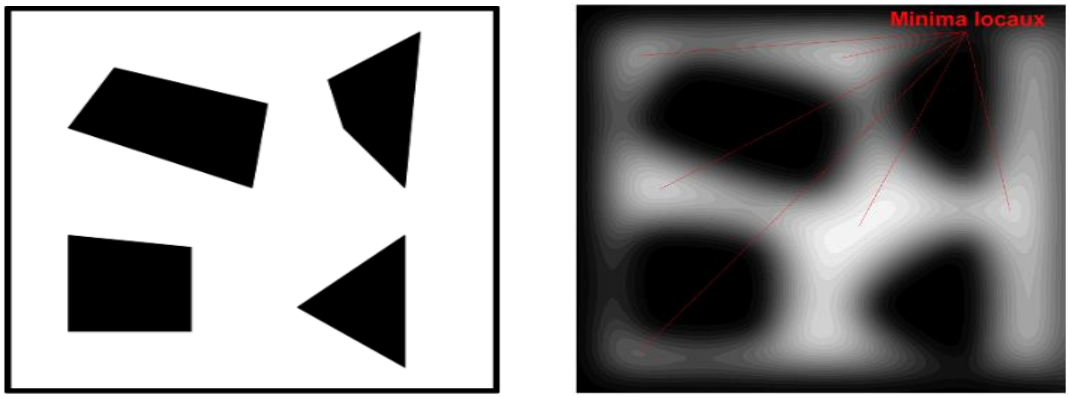


(a) Calculs sur l'environnement d'origine. (b) Exemple de carte de cheminement.

Figure 7 Exemple de carte de cheminement. A gauche, triangulation de Delaunay (gris) de l'environnement d'origine, et carte de cheminement déduite (bleu). A droite, un exemple de carte de cheminement obtenue.

d. Le modèle de champs de potentiels

Le modèle à base de champs de potentiels consiste en une définition de l'environnement permettant directement de résoudre les déplacements de personnes. Les champs de potentiels caractérisent les obstacles de l'environnement par des forces de répulsion, et les buts par des forces d'attraction. Un gradient de forces, ou champ de potentiel, est alors déduit en chaque point de l'environnement comme étant une somme pondérée, le plus souvent par la distance, des potentiels de répulsion et du potentiel lié au but (Figure 8 Carte de champs de potentiels : en noir les obstacles (répulsion), en blanc les zones de navigation (attraction). Le gradient de gris exprime la valeur de potentiel dans l'environnement. Cet exemple contient 6 minima locaux : zones isolées à potentiel d'attraction maximal.). La navigation d'une entité est alors simulée par une descente de gradient depuis sa position dans l'environnement.



(a)Environnement d'origine. (b)Exemple de champs de potentiels.

Figure 8 Carte de champs de potentiels : en noir les obstacles (répulsion), en blanc les zones de navigation (attraction). Le gradient de gris exprime la valeur de potentiel dans l'environnement. Cet exemple contient 6 minima locaux : zones isolées à potentiel d'attraction maximal.

1.9.2 Modélisation sémantique

On parle d'ensemble des propriétés fonctionnelles d'un objet liées à son usage. Elle est utilisée pour permettre à l'entité de raisonner sur des concepts symboliques. Elle consiste donc à associer aux lieux des informations permettant à l'entité intelligente:

- d'interagir avec un humain de manière quasi-naturelle : notification des opérations en cours et planifiées ("Je vais à la salle de bain puis, J'irai au bureau"), interprétation de requêtes ("Amène-moi au salon").
- de classer les lieux en fonction des éléments qui les composent,
- de traiter les ambiguïtés.
- de détecter les erreurs de localisation.

Cette modélisation est systématiquement associée à un modèle topologique et à un ancrage géométrique permettant de localiser les informations sémantiques. On peut aussi voir, dans certains travaux, une démarche incrémentale entre la modélisation topologique et la modélisation sémantique. Le modèle sémantique vient alors comme une sur couche du modèle topologique. Les lieux définis au niveau sémantique agrègent dans ce cas plusieurs lieux définis au niveau topologique [PMCJ10], [PJ11]. Parmi les lieux définis au niveau topologique, ceux permettant de changer de lieu sémantique sont alors identifiés comme des "lieux de passage". Plus généralement, les lieux identifiés dans les modèles sémantiques ne correspondent plus à des positions dans l'espace, mais à des parties de l'espace. Par exemple, dans un environnement intérieur, l'information sémantique distingue les différentes pièces d'un bâtiment pour permettre au robot de s'y localiser par exemple [UN00]. Dans cette application, les lieux identifiés sont interconnectés dans un graphe (topologique) en fonction de leur connectivité. Cette cartographie sans métrique est faite à la main au préalable de la navigation (le modèle de l'environnement est alors connu a priori).L'amélioration des

techniques de vision au cours des dernières années a permis de construire ces modèles en ligne en associant aux différents lieux une information sémantique déduite des éléments qui le composent [VSLM11]. Les lieux identifiés se rapprochent généralement de ceux définis par : on utilise une décomposition naturelle (couloir, hall, pièce avec leur fonction,...). Ces lieux sont identifiés en ligne [TMFR03], [VS04] par la reconnaissance des objets qu'ils contiennent (une pièce avec une douche est une salle d'eau, avec un ordinateur, un bureau,...). Leurs liens avec des modèles topologiques et géométriques les enrichissent et facilitent la navigation et la planification de trajectoires [GSC+05].

1.10 Conclusion

La simulation comportementale est une partie de l'animation qui se rapproche des systèmes réels de par son principe de fonctionnement en assignant aux acteurs ou systèmes animés des comportements indépendants. Ces derniers ne seront alors plus régis par un système global gérant le mouvement de tous les acteurs mais par un mécanisme de décision local placé dans chaque individu. Chaque personnage de la simulation prendra donc les décisions comportementales concernant son mouvement au pas de temps suivant, selon son état et celui de l'environnement l'entourant à cet instant de la simulation. La simulation comportementale est donc un moyen de faire interagir de manière naturelle des acteurs en simulant leurs capacités dans un environnement.

2 Chapitre II
Modèle comportementale de Reynolds

2.1 Introduction

Les chercheurs ont simulé le comportement de volées d'oiseaux, de poissons et d'autres types d'animaux qui se déplacent en groupe parce qu'il s'agit d'un phénomène naturel très important et complexe et qu'il est important de l'étudier et de le simuler. Parmi les simulations du comportement d'un troupeau d'animaux se trouve le modèle de Reynolds. Il s'agit d'un modèle à base des règles présenté comme un programme qui s'appelle « Boids » qui simule le comportement d'un troupeau d'oiseaux, voyageant en groupe coordonné. Parmi les comportements simulés dans ce programme, il y a le comportement de l'individu dans le groupe et le comportement de l'ensemble du groupe.

2.2 « Flocking » comportement autonome et de groupe

Le mouvement des troupeaux est un phénomène naturel hautement coordonné. Les troupeaux et les comportements de groupe synchronisés associés tels que les bancs de poissons ou les troupeaux d'animaux terrestres sont à la fois beaux à regarder et intrigants à contempler. Un troupeau présente de nombreux contrastes. Il est composé d'oiseaux discrets mais le mouvement global semble fluide; Il est simple dans son concept et pourtant si complexe visuellement, il semble rangé de manière aléatoire et pourtant magnifiquement synchronisé. Le plus troublant est peut-être la forte impression de contrôle intentionnel et centralisé. Pourtant, tout indique que le mouvement d'un troupeau doit être simplement le résultat global des actions d'animaux individuels, chacun agissant uniquement sur la base de sa propre perception locale du monde. L'un des domaines d'intérêt de l'animation par ordinateur est la description et le contrôle de tous les types de mouvement. Pour simuler un troupeau, nous simulons le comportement d'un oiseau individuel (ou au moins la partie du comportement de l'oiseau lui permettant de participer à un troupeau). Pour soutenir cette "structure de contrôle comportementale", elle doit également simuler des parties des mécanismes de perception de l'oiseau et des aspects de la physique du vol aérodynamique. Si ce modèle d'oiseau simulé a le comportement correct de membre de troupeau, tout ce qui devrait être nécessaire pour créer un troupeau simulé est de créer des instances du modèle d'oiseau simulé et de leur permettre d'interagir.

Définition d’un flock

Un flock c’est un groupe d’oiseau ou de mammifères qui se déplacent ensemble de façon aligné et cohérente, groupé et sans collision. Pour simuler un flock, il faut simuler le comportement individuel de chaque objet le composant :

- La perception.
- Le contrôle du déplacement.

Déplacement collectif

Le mouvement collectif peut être défini comme un groupe d’animaux qui décident de partir / se déplacer de manière assez synchrone, se déplacent ensemble dans la même direction (Figure 9 Déplacement collectif.) (ce qui implique que les animaux ont le choix entre différentes alternatives) et maintiennent la cohésion jusqu’à ce que le groupe cesse de bouger ou commence à nouvelle activité entraînant un changement de lieu. Cela suppose des décisions individuelles non indépendantes de se déplacer et des relations sur le transfert d'informations entre les membres du groupe médiatisés par des signaux ou signaux comportementaux, et des réponses sociales dont la dynamique peut être modulée par le mouvement collectif entrant lui-même. Le gabarit physique environnemental (photopériode, vent, sentier ou ruisseau) influe sur la trajectoire des groupes, mais ne peut pas être le principal facteur responsable de ces mouvements et de ces propriétés collectives [Tou16].

Figure 9 Déplacement collectif.

2.3 Modèles comportementaux

Dans le premier chapitre, nous avons parlé sur les modèles de comportement. Dans cette section nous passerons aux modèles informatiques du comportement de déplacement en groupe.

Qui et Hu [QH10] proposent une architecture pour modéliser les structures de groupes de piétons en considérant à la fois les relations intergroupes et intragroupes. Chaque groupe possède un leader qui est le seul à pouvoir être influencé par des individus d'autres groupes. Les relations entre agents sont stockées à l'intérieur de deux types de matrices : celles gérant les relations à l'intérieur d'un groupe et celles gérant les relations entre les différents groupes.

Goldenstein et al. [GKM+01] présentent une méthode de modélisation d'agent basée sur l'intégration de systèmes dynamiques non-linéaires et de structures de données cinétiques. Cette méthode est structurée en 3 niveaux. Le premier est le niveau local, il modélise le comportement de chaque agent. Le second est le niveau d'environnement global, il permet de gérer les obstacles et les collisions entre agents. Le dernier niveau, celui de planification globale gère quant à lui les objectifs des agents.

Kapadia et al. [KSHF09] proposent une architecture basée sur les affordances d'autonome, c'est à dire ses possibilités d'actions dans son environnement. Chaque autonome perçoit l'environnement à travers des champs de perception égocentriques (qui quantifient les propriétés de l'environnement local d'entité autonome) et d'affordance (qui donnent une valeur d'affordance basée sur l'objectif du cette entité et qui sont calculés comme fonction des champs de perception).l'entité autonome choisit l'affordance dont la valeur est la plus élevée.

2.4 Modèle de Reynolds

Parmi les modèles comportementaux à base des règles, il existe le modèle comportemental de Craig Reynolds qui fonctionne sur le même principe que les modèles précédents. Nous allons nous concentrer sur ce modèle dans notre travail car c'est l'essentiel pour nous, il est basé sur trois règles de base sur lesquelles nous allons travailler et y ajouter un modèle de prédateur qui n'existe pas dans les modèles de Reynolds.

Craig Reynolds : c'est l'initiateur de l'animation comportementale. Il a construit un modèle comportemental permettant d'animer un ensemble d'individus à l'aide de la description du comportement individuel de chaque animal appartenant au groupe.

Le modèle de Reynolds permet de modéliser un comportement émergent. En effet, la complexité comportementale résulte ici de l'interaction d'agents individuels (appelés boids) défini par (une masse, position, vitesse, accélération) en respectant un nombre limité de règles simples, telles que (**Error! Reference source not found.**

Les règles comportementales

- **La cohésion** : Le comportement de cohésion donne aux acteurs la capacité de générer une situation de cohérence dans le groupe qu'ils forment (se rapprocher et former un groupe) avec d'autres acteurs voisins .La direction de

la cohésion peut être calculée en déterminant tous les acteurs dans le voisinage local et en calculant "la position moyenne" (ou "centre de gravité") des acteurs voisins.

- **La séparation** : Le comportement de séparation donne à l'acteur la capacité de maintenir une certaine distance de séparation des autres caractères qui lui sont proches pour que les forces répulsives pour chaque acteur voisin sont additionnées ensemble pour produire la force de direction, c'est-à-dire 2 boids ne peuvent pas se trouver au même endroit au même moment
- **L'alignement** : Le comportement d'alignement donne à l'acteur la capacité de s'aligner sur d'autres acteurs voisins par conséquence la direction d'alignement peut être calculée en déterminant tous les acteurs dans le voisinage local, en faisant la moyenne des ensembles des vitesses des acteurs voisins.

Dans les faits, cela se traduit par la réaction du boid à son voisinage. Celui-ci se divise en trois zones, de la plus proche à la plus éloignée: une zone de répulsion, une zone d'orientation et une zone d'attraction. Lorsqu'un voisin entre dans la zone de répulsion, le boid s'en éloigne, s'il est dans la zone d'orientation, le boid le suit, et s'il est dans la zone d'attraction, il s'en rapproche. Généralement, on ajoute également à la simulation un angle mort, dans lequel le boid ne peut pas percevoir ses voisins.

Ces trois règles de bases, permettent l'attraction et la répulsion de chacun des individus et permet la stabilité de l'ensemble.

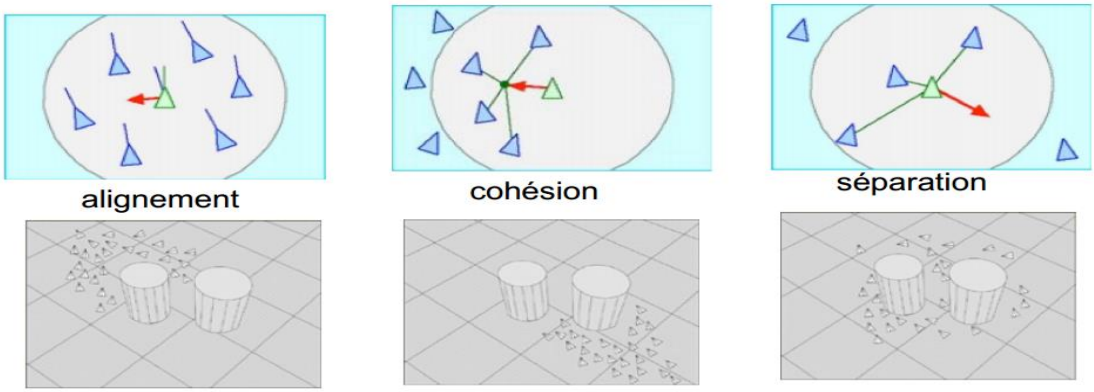


Figure 10 Les règles comportementales.

Les règles de Reynolds se divisent sur deux types de comportements, comportements d'individus et de groupes :

2.5 Comportements d'individus

a. Comportement d'évitement d'obstacles

Le comportement d'évitement d'obstacles donne à un acteur la capacité de manœuvrer dans un environnement encombré en esquivant autour des obstacles (Figure 11) Il y a une distinction importante entre l'évitement d'obstacles et le comportement de fuite.

Tandis que :
Flèche rouge : la force de A sur l'acteur.
B, C : des obstacles.

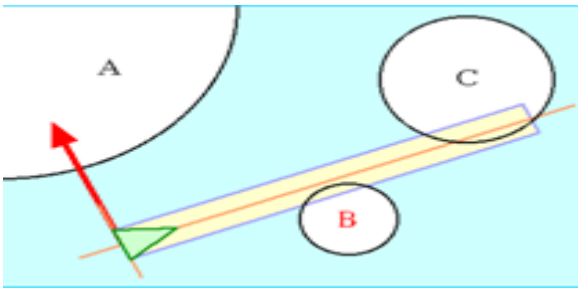


Figure 11 Comportement d'évitement d'obstacles.

b. Comportement de suivi de chemin

Ce type de comportement permet à un acteur de s'orienter le long d'une voie d'accès prédéterminée, telle qu'une chaussée, un couloir ou un tunnel (Figure 12 Comportement de suivi de chemin.**).**

Tandis que :
Ligne courbe : le chemin prédéfini.

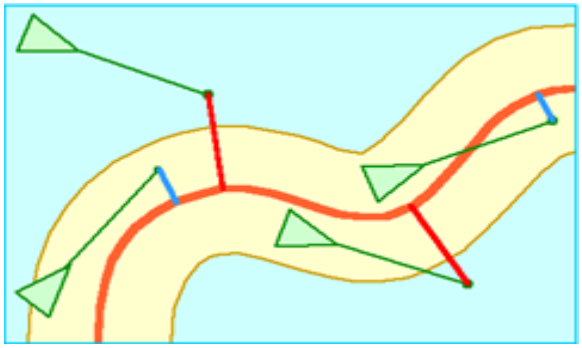


Figure 12 Comportement de suivi de chemin**.**

c. Comportement Evitement de collision non-alignée

Le comportement d'évitement de collision non alignée se doit de prévenir les collisions entre des acteurs se déplaçant dans une direction arbitraire (Figure 13 Comportement Evitement de collision non-alignée.)

Tandis que :

Les deux acteurs sont des obstacles dynamiques, ils vont entrer en collision, donc l'un d'eux se détourne, ou bien l'un ou l'autre change de cap, ou l'un ou l'autre diminue sa vitesse.

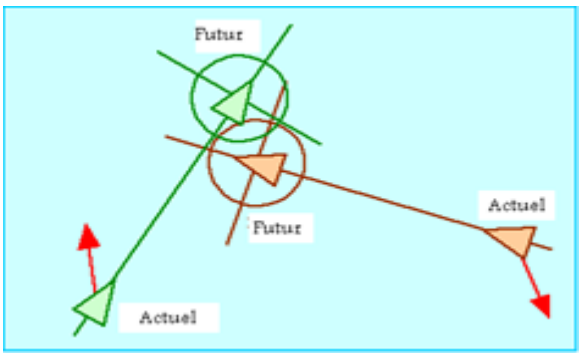


Figure 13 Comportement Evitement de collision non-alignée.

d. Comportement de recherche

La recherche est un comportement très simples qui déplace un véhicule vers une position de cible avec une vitesse constante (Figure 14 Comportement de recherche.**).**

Tandis que :
Triangle vert : proie.
Le cercle : la cible.
Seek : aligner le vecteur de vitesse vers une cible stationnaire.

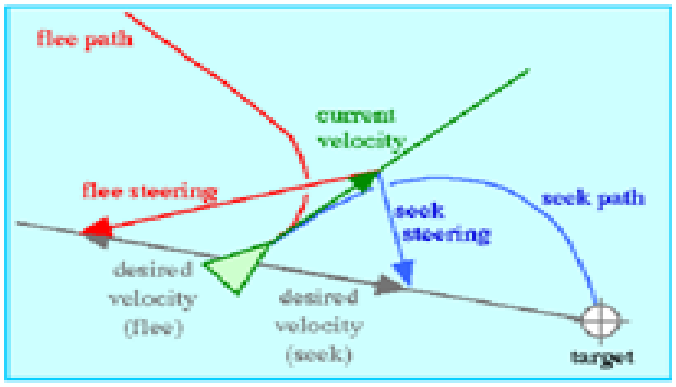


Figure 14 Comportement de recherche**.**

e. Comportement de poursuite

La poursuite est semblable à la recherche sauf que la carrière (cible) est un autre caractère mobile (Figure 15).

Tandis que :

Poursuite : Aligne radialement le vecteur de vitesse sur la position prédite d'une cible en mouvement dans un certain avenir.

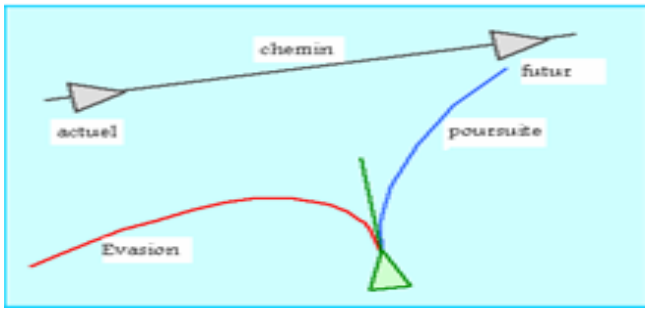


Figure 15 Comportement de poursuite**.**

f. Comportement d'arrivée

Le comportement d'arrivée est une extension du comportement de recherche (Figure 16). La différence importante se résume dans la manière selon laquelle le véhicule atteint la destination. Le comportement de recherche fait que notre véhicule atteint la cible à pleine vitesse.

Tandis que :

La flèche : la vélocité du boid.

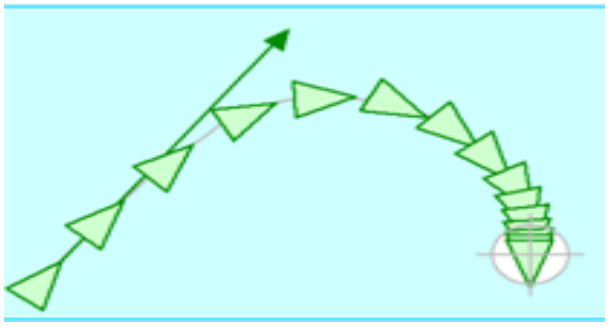


Figure 16 Comportement d'arrivée**.**

2.6 Comportements de groupes

Les trois comportements de base de direction (séparation, cohésion et l'alignement) touchent des groupes d'acteurs. Dans chaque cas, le comportement de direction détermine comment un acteur réagit à d'autres dans son voisinage local. Les acteurs à l'extérieur du voisinage local sont ignorés. Le voisinage est spécifié par une distance

qui est définit quand deux acteurs sont "voisin" et un angle qui définit la perception de l'acteur "champ de voisin (Figure 17)

Tandis que :

Le cercle gris : le champ de vision.

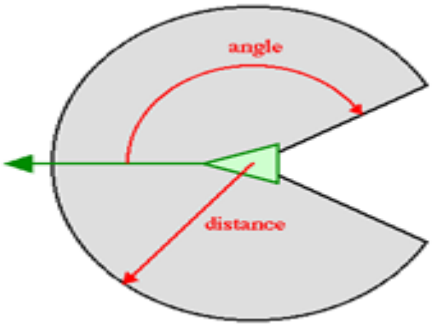


Figure 17 Notion de voisinage.

d. Le suivi de leader

Le comportement de suivi de leader, est provoqué par un ou plusieurs acteurs à l'effet de suivre un autre acteur en mouvement et considéré comme le leader
En outre, ces acteurs tentent des rester prés du leader (Figure 18 le suivi de leader.).

Tandis que :
Le triangle vert : les boids
Le triangle blanc : c'est le leader.
La flèche rouge : le suivie de leader.
Le rectangle : c'est la direction.

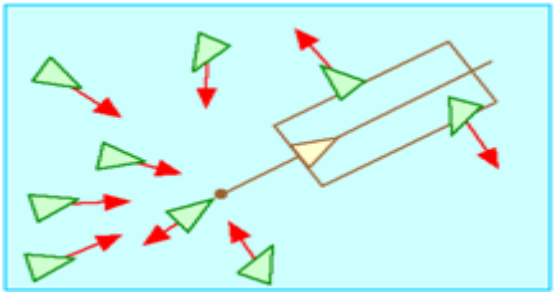


Figure 18 le suivi de leader.

Conclusion

Dans ce chapitre, nous avons présenté les modèles de simulation de comportement de déplacement en groupe et le modèle de Reynolds qui lui est associé. Il existe plusieurs types de règles de comportement, règles de comportement pour les individus et règles de comportement de groupe. Enfin, concernant la présentation de l'environnement, il faut faire attention à toutes les informations présentes dans la scène afin d'avoir un résultat proche de la réalité.

3 Chapitre III
Conception du modèle proposé

3.1 Introduction

En raison de l'importance du mouvement des animaux en groupe, nous nous sommes toujours intéressés à ce phénomène important, notre objectif aujourd'hui est donc de simuler leur comportement en présence de prédateur, de comprendre comment ils se comportent. Ce chapitre, qui traite de la conception de notre système, vise à décrire comment il fonctionne et qu'elle est sa structure. Nous développerons donc la conception générale et détaillée du système, dans le but d'expliquer le modèle que nous avons proposé pour atteindre nos objectifs.

3.2 Problématique

Dans le monde naturel, les organismes présentent certains comportements lorsqu'ils voyagent en groupe. Ce phénomène, mouvement en groupe, se produit à la fois aux échelles microscopiques (bactéries) et aux échelles macroscopiques (poissons). Dans le domaine de l'animation comportementale, ces modèles peuvent être simulés en créant des règles simples et en les combinant. Ceci est connu comme un comportement émergent, et a été utilisé dans des jeux pour simuler le mouvement en groupe. Le modèle de Craig W. Reynolds est le modèle le plus utilisé, il permet de modéliser le comportement d'individus au sein de groupes (poissons, oiseaux, troupeaux, ect.) au moyen de trois règles d'orientation très simples. Nous visons par ce sujet à étendre ce modèle pour inclure le comportement de prédateur. Notre objectif donc, est de simuler le comportement émergent d'un groupe de poissons en présence de prédateurs au moyen de trois règles d'orientation très simples de Reynolds (Séparation, Cohésion, Alignement).

3.3 Architecture générale du modèle proposé

3.3.1 Conception globale

À travers le schéma ci-dessous, nous expliquons la solution que nous avons proposée pour résoudre la problématique posée. Ce schéma décrit la conception globale de notre système et comporte trois parties principaux, la première partie c'est l'interface de l'utilisateur consiste à contrôler notre système de flockage, la deuxième partie c'est le fonctionnement du notre système, la troisième partie consiste à visualiser les comportements du mouvement des poissons en groupe en présence de prédateurs.

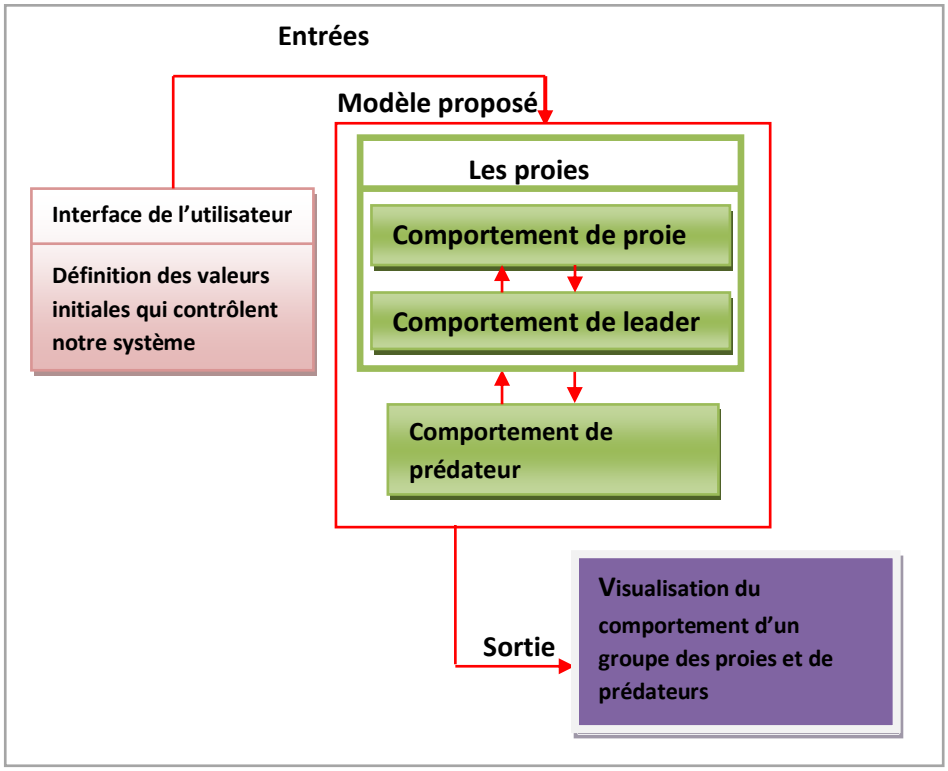


Figure 19 Conception globale.

3.3.2 Conception détaillée

L'étape de conception détaillée a pour but d'expliquer tout les modules qui composent notre système.

Les Entrées

L'utilisateur va donner des valeurs initiales pour définir les variables globales qui contrôlent notre système de flock.

Les comportements des différents types de poissons

Ils existent plusieurs types de poissons :

- Proies.
- Prédateurs.
- Leaders.

Les prédateurs et les proies peuvent influencer leur évolution mutuelle. Les traits qui améliorent la capacité d'un prédateur à trouver et à capturer des proies seront sélectionnés chez le prédateur, tandis que les traits qui améliorent la capacité de la proie à éviter d'être mangés seront sélectionnés chez la proie.

Chaque type a des motivations différentes qui produisent des modèles de comportement différents :

- Les proies et les leaders expriment un comportement sociale, c'est-à-dire qu'ils ont un comportement qui le rend intéressés par tout compagnon possible.
- Les prédateurs expriment un comportement individuel, c'est-à-dire qu'ils ont un comportement qui le rend intéressés par tout proies possible.

Module1 : modèles de comportement des proies

La nécessité de vivre en groupe peut s'expliquer par plusieurs raisons : Protection face aux prédateurs, devenir membre d'un groupe en l'occurrence d'un banc, et le rester sont une nécessité impérieuse pour la plupart des proies : en se "grouper" dans la masse, ils maximisent la probabilité d'échapper aux prédateurs.

Les règles qui régissent les mouvements d'une proie sont alors exprimées comme suit :

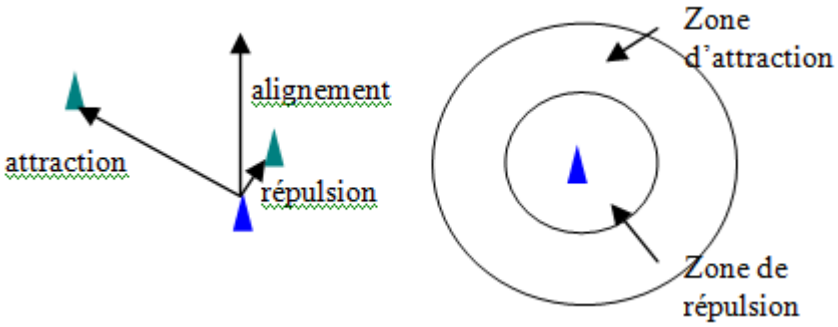


Figure 20 les trois influences du voisinage sur le vecteur de déplacement de l'individu.

- Lorsqu'un voisin entre dans la zone de répulsion, la proie tend à modifier sa direction pour s'éloigner de ses voisins.
- s'il est dans la zone d'attraction, il tend à modifier sa direction pour se rapprocher des voisins.
- et s'il est dans la zone d'orientation il tend à aligner sa direction avec les voisins les plus proches.

Les trois influences du voisinage nous présentent ces comportements suivants :

1. **Déplacement cohérent**

   Quand plusieurs proies sont proches ils ont besoin de se comportement de cohésion. Ce dernier consiste à calculer la direction moyenne de tous les poissons présents dans le champ de vision. Chaque poisson va ensuite adapter son orientation en fonction de la valeur.

2. **Evitement des collisions avec les voisins proches**

Ce comportement consiste en la récupération des deux directions (celle de poisson et de son plus proche voisin). Si ces deux directions mènent à une collision, les deux poissons tournent pour s'éviter.

3. **Le suivi d'un leader**

C'est la combinaison des comportements de séparation et arrivée, la cible à atteindre pour l'arrivé se trouve placée derrière. Un pour suivant s'éclatera s'il se trouve sur le chemin du leader.

4. **Eviter les prédateurs**

Lorsque la proie sent que le danger s'approche d'une certaine distance, elle change de trajectoire à mesure que son accélération et sa vitesse augmentent. Elle maintient son attachement sous forme de groupe pour éviter le danger des prédateurs.

Module2 : modèles de comportement des prédateurs

La figure 21 illustre le comportement d'un prédateur traquant sa proie.

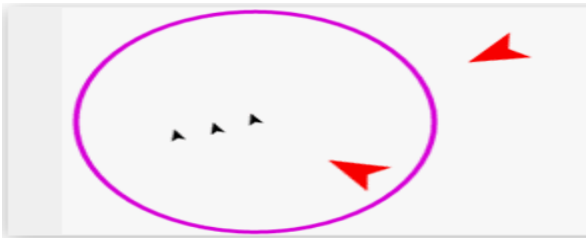


Figure 21 Comportement du prédateur (Le triangle rouge représente le prédateur, et les triangles noirs sont les proies).

1. **Attaquer les proies**

L'attaque du prédateur vise à répandre la tension dans le mouvement des proies pour disperser son comportement collectif, qui est considéré comme un élément de sa force, et l'attaque venant de différentes directions.

2. **Eviter les collisions avec les autres prédateurs**

Pendant l'attaque des prédateurs, il peut être entre à des collisions avec d'autres prédateurs, nous avons donc le créer une fonction pour éviter ces derniers.

Module3 : modèles de comportement des leaders

Le comportement de suivi des leaders amène un ou plusieurs acteurs à en suivre un autre acteur en mouvement le considère alors que le leader du groupe

a. Guider le groupe des proies

Souvent, dans les groupes d'animaux, il y a un élément de mouvement qui est considéré comme le leader de son groupe, en suivant son chemin et changer leur vitesse pour devenir égale à la vitesse du leader.

b. Eviter les prédateurs

Le leader est attaqué en étant à l'avant du groupe et en étant une proie cible pour les prédateurs.

3.4 Conclusion

Ce chapitre nous a permis d'expliquer notre objectif et de proposer une conception complète et détaillée de notre système. Pour ce faire, nous nous sommes appuyés sur l'utilisation des règles de Reynolds pour appliquer le comportement individuel des proies, entre proie et proie, et entre proie et leader. La combinaison de ces comportements individuels nous a fourni un modèle idéal pour le mouvement des proies en groupe avec une grande cohérence. Nous avons également pu citer le comportement d'un prédateur à travers ces derniers. Les règles pour que nous ayons étudié le comportement de cette espèce, qui consiste à attaquer ses proies à titre individuel ou collectif.

4 Chapitre IIII
Implémentation et résultat

4.1 Introduction

Après avoir établi l'étude conceptuelle de notre système où nous avons présenté toutes les tâches que nous allons réaliser, nous passons maintenant à l'implémentation de l'application, en présentant l'environnement et les outils utilisés pour arriver à notre objectif, comme l'environnement et le langage de programmation ainsi que les procédures exploitées, par la suite nous allons présenter les résultats que nous avons obtenus.

4.2 Outils de programmation

Nous avons implémenté notre travail en utilisant le langage de programmation C++ qui est la variante orientée objets du langage C et l'un des langages les plus populaires que nous pouvons utiliser sur divers systèmes d'exploitation.

Nous avons utilisé la bibliothèque graphique OpenGL (Open Graphics Library). Cette interface de programmation est disponible sur de nombreuses plateformes, elle est utilisée pour des applications de modélisation ainsi que des applications de jeux vidéo. OpenGl permet de visualiser la géométrie des objets sous forme de points, vecteurs et polygones et qui peuvent être utilisées pour afficher des scènes simples à deux dimensions ou des scènes tridimensionnelles complexes à partir de primitives géométriques.

4.3 Matériel et logiciel de configuration

Nom de l'ordinateur	MICRO-PC TOSHIBA
Processeur	Intel(R) Core(TM) i3-3110M CPU @ 2.4GHz 2.40GHz
Mémoire installée(RAM)	4.00 Go
Type de système	Système d'exploitation 64 bits
Windows	Windows 7 professionnel
Carte graphique	NVIDEA GeForce 710M + Intel(R) HD Graphics 4000

Tableau 1 Tableau de configuration**.**

4.4 Implémentation

Architecture de l'application

À travers le schéma ci-dessous, nous expliquons l'architecture du notre application qui comporte trois parties, la première partie c'est l'interface de l'utilisateur consiste à définir les variables globales de notre application, la deuxième partie comporte deux modules principaux module Flock et module Boid, l'utilité du module Flock c'est la création du Flock et Boids et l'ajout des boids, l'utilité du module Boid c'est l'utilisation des fonctions de comportement de boids et prédateurs, la troisième partie consiste à visualiser les comportements du mouvement des poissons en groupe en présence de prédateurs.

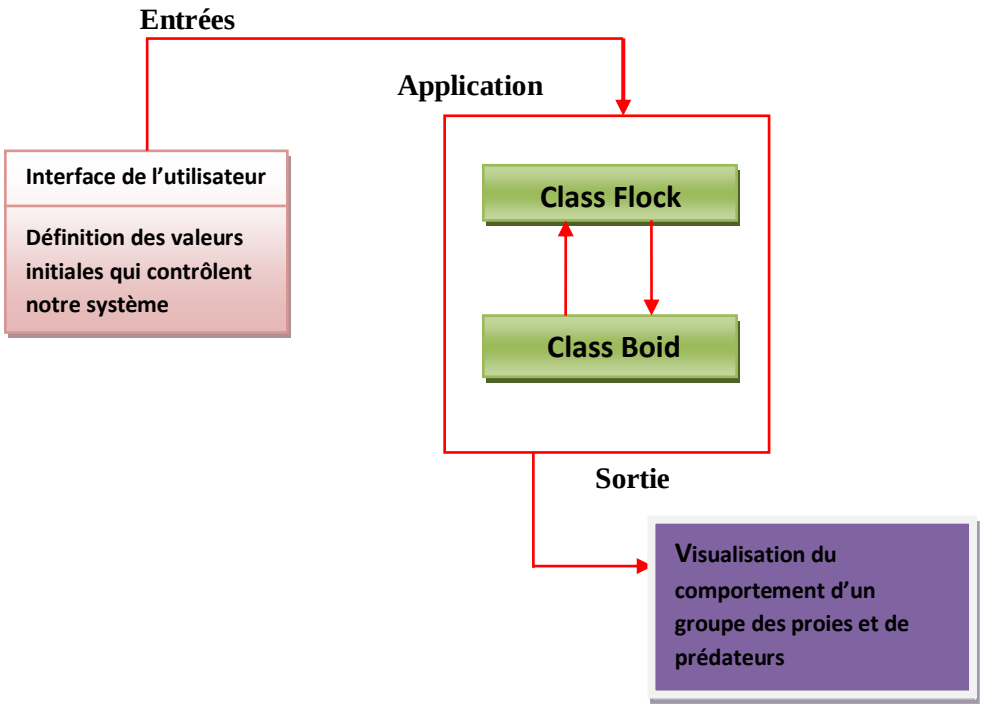


Figure 22 Architecture de l'application.

Les Entrées

Le contrôle se fera via la définition des variables globales suivant :

Variables	Signification
MAX_PREY	Le nombre des proies+ nombre des leaders).
MAX_LEADERS	Le nombre des leaders dans la scène.
MAX_PREDATORS	Le nombre des prédateurs dans la scène.
PREY_DETECTION_RANGE	Distance de détection des proies (proie détecté<valeur>non détecté).
PREDATOR_FLOWING_A_LEADER_SPEED_MODIFIER	Contrôlez la vitesse du prédateur alors qu'il poursuit le chef.
PREDATOR_SPEED_MODIFIER	Contrôlez la vitesse globale du prédateur.
BOUNDING_VOL_FOR_ALL	limitation en forme de carré pour tout les boids.
BOUNDING_VOL_FOR_ALLIES	La distance entre proie et proie (séparation).
BOUNDING_VOL_FOR_ENEMIES	La distance entre prédateur et prédateur (séparation).
LEADERS_DETECTION_RANGE	Distance de détection de leaders (leader détecté<valeur>non détecté).
GRAVITY_CONST	F= G*(m1*m2)/r², F c'est la force entre les masses, G c'est la gravité, m1 c'est la 1^{ere} masse, m2 c'est la 2eme masse, r c'est la distance entre le centre des masses, Dans notre système m1=m2=1, La masse est négligée donc F= G/r².
PREY_FLOWING_A_LEADER_SPEED_MODIFIER	Contrôlez la vitesse d'une proie en suivant un leader.
PREY_AVOIDING_A_PREY_SPEED_MODIFIER	Contrôle la vitesse de séparation des proies les unes des autres.
LEADERS_AVOIDING_A_PREY_SPEED_MODIFIER	Contrôlez La vitesse de séparation des leaders des proies.

LEADERS_SPEED_MODIFIER	Contrôlez la vitesse globale du leader.
PREDATOR_AVOIDANCE_SPEED_MODIFIER	Contrôle de la vitesse de séparation du prédateur.

Tableau 2 Signification des variables globales.

Après avoir défini les valeurs dans lesquelles notre utilisateur sera engagé, nous allons maintenant créer un flock et les Boids, qui sont la base du travail. Après cela, nous ajouterons ces éléments au flock.

1. Module Flock

On a utilisé dans notre classe Flock les variables : maxPrey, maxLeaders, maxPredators pour créer notre boids.

a. Création du Flock

```
void FlockSystem::addFlock_random(){
	unsigned int counter = 0;
	unsigned int max_prey_local = 0;
	max_prey_local = maxPrey -  maxLeaders;
	int maxPrey_sqrt = (int) (glm::sqrt(max_prey_local));
	int i, j, ID = 0;
	for(i = 0; i < maxPrey_sqrt; i++){
		for(j = 0; j < maxPrey_sqrt ; j++){
			flock.push_back(Boid(ID, (float)i, (float)j, false, false));
			counter++;
			ID++;}}
i++;
	if (counter < max_prey_local) {
		for (j = 0; j < max_prey_local - counter; j++) {
			flock.push_back( Boid(ID, (float)i, (float)j, false, false));
			ID++;}}
	addLeaders_random();
	addPredators_random();}
```

Figure 23 Création du Flock.

Explication :

=>On a utilisé la fonction addFlock_random () pour créer et ajouter les boids, où la création du boids est aléatoire (random) dans une matrice carré, *le nombre de proies local= max proies-max leaders* étant donné que les leaders sont des proies.

```
*for (i = 0; i < maxPrey_sqrt; i++){
*for (j = 0; j < maxPrey_sqrt ; j++){
```

=>Ces deux boucles imbriqués vont prendre la racine de valeur initiale de maxPrey et remplissent la matrice (i,j).

```
*flock.push_back (Boid(ID, (float)i, (float)j, false, false));
```

=>Le push_back va ajouter chaque fois un élément dans la liste des boids, ID la position, false =‘non prédateur’, false=’non leader’.

```
*if (counter < max_prey_local) {
*for (j = 0; j < max_prey_local - counter; j++) {
*flock.push_back( Boid(ID, (float)i, (float)j, false, false));
ID++;}}
```

=>On a utilisé cette boucle pour les nombre qui n'acceptent pas l'opération racine, on termine le remplissage de la matrice dans la dernière ligne ‘j’.

```
*addLeaders_random();
*addPredators_random();}
```

=>Après la création on fait l’appel des deux fonctions de l’ajout des prédateurs et des leaders au flock.

b. l’ajout des leaders

```
void FlockSystem::addLeaders_random() {

    for (int i = -maxLeaders; i < 0; i++) {
            flock.push_back( Boid(i, (float)i, (float)i/2, false, true));

    }


}
```

Figure 24 Ajout des leaders.

Explication :

On va ajouter les leaders dans le coté(-).

c. L’ajout des prédateurs

```
void FlockSystem::addPredators_random(){

    for (int i = -maxPredators; i < 0; i++) {
        flock.push_back(Boid(i, (float)i * 2, (float)i * 2, true, false));

    }

}
```

Figure 25 Ajout des prédateurs.

Explication :

=>On va ajouter les leaders dans le coté négatif.

2. Module Boids

Dans notre classe Boid, chaque boid est caractérisé par : ID, accélération, vélocité, position, masse.

a. Fonction de cohésion et séparation

```
void Boid::avoidPrey(std::vector<Boid> flock, unsigned int max_prey){

    glm::vec3 forces = glm::vec3(0.0f);
    float dsquared = 0.0f;float strength = 0.0f;float distance = 0.0f;
    for (int i = 0; i < max_prey; i++) {
            glm::vec3 dirVec = glm::vec3(0);
            dirVec = flock[i].position - position;
            distance = glm::length(dirVec);
            dsquared = distance * distance;
            if(distance > 0.0f && distance < bounding_volume_for_allies){
                strength = gravity_const / dsquared;
                dirVec *= strength;
                if (am_i_a_leader)
                    dirVec /= avoiding_speed_modifier_LEADER;
                else
                    dirVec /= avoiding_speed_modifier_PREY;
                forces +=  dirVec;
            }
    }
    if(forces !=  glm::vec3(0) ){
        prey_acceleration = prey_acceleration - forces;
    }
}
```

Figure 26 Fonction de séparation et cohésion.

Explication :

```
*if(distance > 0.0f && distance < bounding_volume_for_allies){
strength = gravity_const / dsquared;
=>dans cette condition on va utiliser une force d'attraction pour atteindre le
comportement de cohésion.


*if(forces !=  glm::vec3(0) ){
prey_acceleration = prey_acceleration - forces;
=>dans cette condition on va utiliser une force de répulsion pour atteindre le
comportement de séparation.
```

b. Fonction de l'alignement et suivi le leader

```
void Boid::followOneLeader(std::vector<Boid> flock, unsigned int maxPrey, unsigned int maxLeaders) {
    int i = maxPrey - maxLeaders;
    glm::vec3 force = glm::vec3(0.0f);
    std::vector <glm::vec3> directions;
    std::vector <float> distances_to_leaders;
    std::vector <int> leaders_IDs;
    float dsquared = 0.0f;
    int minID = 0;
    float strength = 0.0f;
    float forceModifier = 0.0f;

    if (maxLeaders != 0 ) {
        for (i = i; i < maxPrey; i++) {
            glm::vec3 temp  = flock[i].position - position;
            directions.push_back(temp);
            distances_to_leaders.push_back(glm::length(temp));
            leaders_IDs.push_back(i);
        }
        minID = std::min_element(distances_to_leaders.begin(), distances_to_leaders.end()) - distances_to_leaders.begin();
        dsquared = distances_to_leaders[minID] * distances_to_leaders[minID];
        if (dsquared > 0.0f ) {
            if(distances_to_leaders.at(minID) < leaderDetection_range){

                strength = following_speed_modifier_PREY;
                directions.at(minID) /= strength;

                force = directions.at(minID);
            }
        }
        prey_acceleration = prey_acceleration + force;
    }
```

Figure 27 Fonction de l'alignement et suivi de chef.

Explication:

*if (maxLeaders != 0) { ; => si maxLeaders=0 nous sortons de la boucle,si oui:
*for (i = i; i < maxPrey; i++) {
*glm::vec3 temp = flock[i].position – position;=> le temps va prendre la valeur du vecteur du distance entre les proies et les leaders.
*distances_to_leaders.push_back(glm::length(temp));=>la valeur absolu du vecteur = distance.
*leaders_IDs.push_back(i);=>position du
*leader.minID=std::min_element(distances_to_leaders.begin(),distances_to_leaders.end()) - distances_to_leaders.begin();=>fonction prédifini qui prend le min de distance.
*dsquared = distances_to_leaders[minID] * distances_to_leaders[minID];=>le carré de la distance.
        }
 *if(distances_to_leaders.at(minID) < leaderDetection_range){=> leader détectée.
*prey_acceleration = prey_acceleration + force;=>on va utiliser la force de l'attraction pour suivre la proie.

c. Fonction d'évitement de prédateur

```
void Boid::avoidPredators(std::vector<Boid> flock, unsigned int maxPrey) {
    glm::vec3 forces = glm::vec3(0.0f);
    float dsquared = 0.0f;
    float distance = 0.0f;
    float strength = 0.0f;
    for (int i = maxPrey; i < flock.size(); i++) {
        glm::vec3 dirVec = flock[i].position - position;
        distance = glm::length(dirVec);
        dsquared = distance * distance;
        if (distance != 0.0 && distance < bounding_volume_for_enemies)
        {
            strength = gravity_const / dsquared;
            dirVec *= strength;
            dirVec /= avoiding_speed_modifier_PREDATORS;
            forces += dirVec ;
        }
    }
    prey_acceleration = prey_acceleration - forces;
}
```

Figure 28 Fonction d'évitement de prédateur.

Explication:

*for (int i = maxPrey; i < flock.size(); i++) {
glm::vec3 dirVec = flock[i].position - position;=> cette boucle for va calculer le vecteur de distance entre les proies et prédateurs.
*if (distance != 0.0 && distance < bounding_volume_for_enemies)=>prédateur détecté.
*prey_acceleration = prey_acceleration - forces;=>si on détecte un prédateur on va utiliser une force de répulsion pour s'échapper.

d. Fonction d'attaque du prédateur

```
void Boid::followOnePrey(std::vector<Boid> flock, unsigned int maxPrey) {
    int i = maxPrey;
    glm::vec3 force = glm::vec3(0.0f);
    std::vector <glm::vec3> directions;
    std::vector <float> distances_to_prey;

    float dsquared = 0.0f;
    int minID = 0;
    float strength = 0.0f;
    float forceModifier = 0.0f;

    if (maxPrey != 0) {
        for (i = 0; i < maxPrey; i++) {
            glm::vec3 temp = flock[i].position - position;
            directions.push_back(temp);
            distances_to_prey.push_back(glm::length(temp));

        }
        minID = std::min_element(distances_to_prey.begin(), distances_to_prey.end()) - distances_to_prey.begin();
        dsquared = distances_to_prey[minID] * distances_to_prey[minID];
        if (dsquared > 0.0f) {
            if (distances_to_prey.at(minID) < preyDetection_range_toHunt) {

                strength = following_speed_modifier_PREDATOR;
                directions.at(minID) /= strength;

                force = directions.at(minID);
            }
        }
        predator_acceleration = predator_acceleration + force;
    }
```

Figure 29 Fonction d'attaque des proies.

Explication :

```
*if (maxPrey != 0) {
*for (i = 0; i < maxPrey; i++) {
glm::vec3 temp = flock[i].position - position;=> On va calculer la distance minimum entre les proies et les prédateurs .
*if (distances_to_prey.at(minID) < preyDetection_range_toHunt) {=> la proie est détectée.

predator_acceleration = predator_acceleration + force;=> on va utiliser la force de l'attraction pour suivre la proie.
```

4.1 Test et résultat

Processus de simulation

A travers le schéma ci-dessous nous expliquons le déroulement de notre simulation, les variables utilisées, les résultats obtenus, les configurations utilisées ce qui nous a permis d'accéder aux résultats présentés ci-dessous:

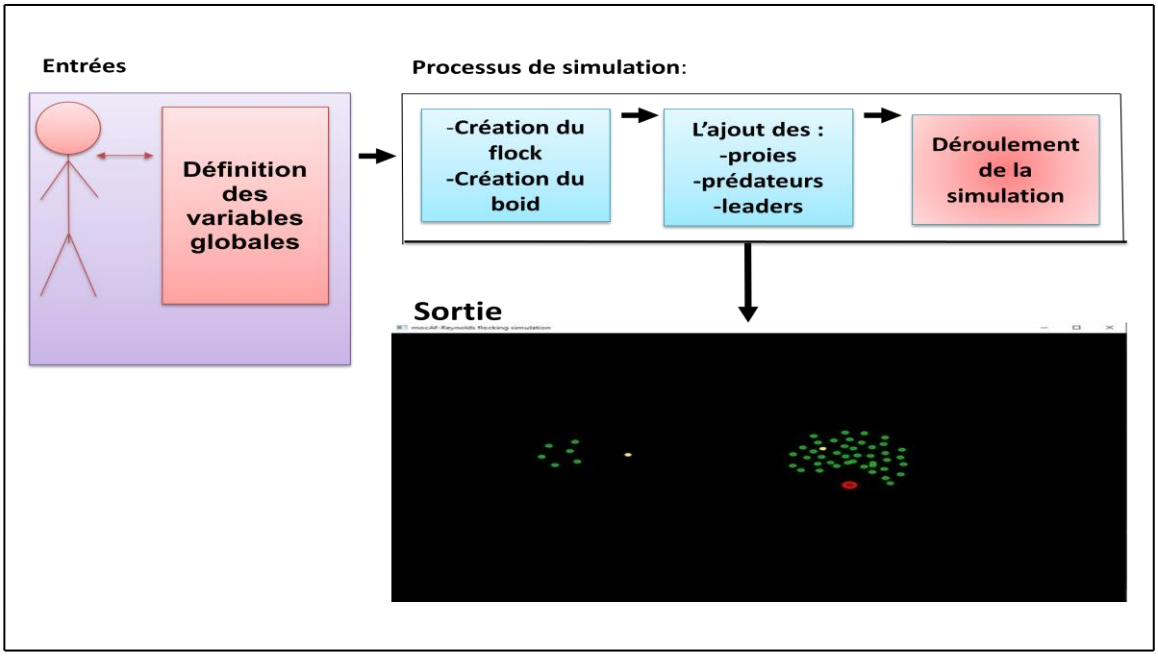


Figure 30 Processus de simulation.

Test1 : Le comportement de séparation et Cohésion

Afin d'obtenir un groupe de proies se déplaçant de manière cohérente séparée les uns des autres, nous avons utilisé la fonction de cohésion et de séparation et nous avons donné des valeurs initiales aux variables suivants :

- MAX_PREY 50;
-PREY_AVOIDING_A_PREY_SPEED_MODIFIER 100.0;
-BOUNDING_VOL_FOR_ALLIES 1.5f;
-GRAVITY_CONST 6.6f;

Résultat obtenu : Ici, nous obtenons un groupe de proies qui voyagent de manière cohérente séparément les unes des autres.

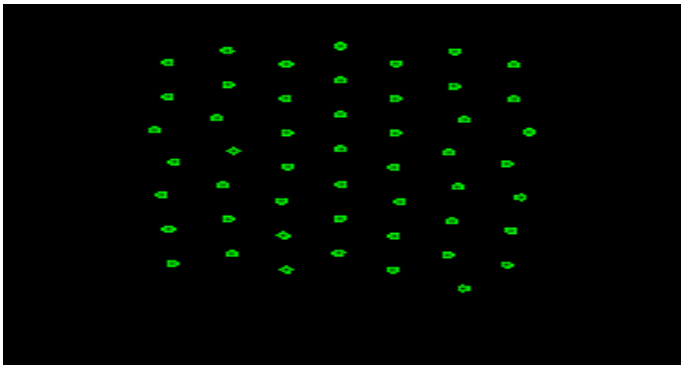


Figure 31 La séparation et cohésion.

Test2 : Le comportement de l'alignement et Suivi de chef

Nous voulons maintenant que ces proies suivent le même chemin en utilisant la fonction de l'alignement et en essayant de rester proches d'une autre proie en tant que leader. Nous avons utilisé les variables suivants :

MAX_LEADERS 2;
BOUNDING_VOL_FOR_ALLIES 1.5f;
LEADERS_DETECTION_RANGE 500.0f;
LEADERS_AVOIDING_A_PREY_SPEED_MODIFIER 50.0;
PREY_FLOWING_A_LEADER_SPEED_MODIFIER 10.0;
LEADERS_SPEED_MODIFIER 20.0;

Résultat obtenu : Ici, nous avons obtenus un groupe de proies voyageant sur le même chemin séparés les uns des autres et suivis son leader.

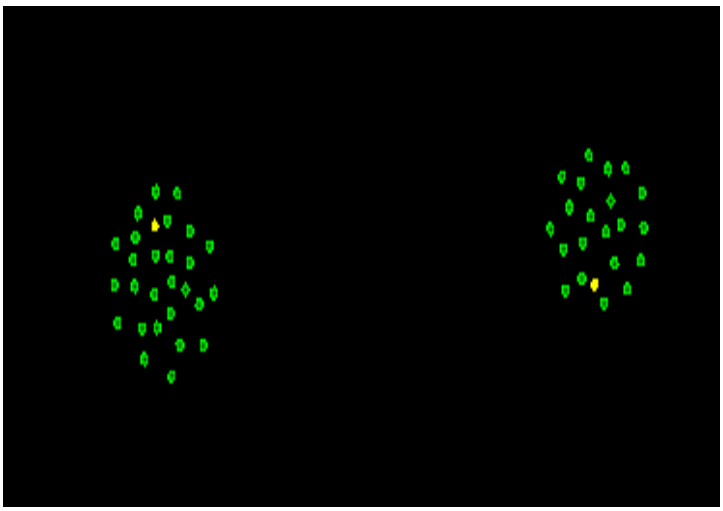


Figure 32 L'alignement et suivi de leader(le boid jaune c'est le leader).

Test3 : Le comportement d'évitement de prédateurs

Lorsque les proies sont attaquées en réponse, ils évitèrent cette attaque en utilisant la fonction d'évitement de prédateur, la configuration utilisée c'est :

PREDATOR_AVOIDANCE_SPEED_MODIFIER 5.0;

Résultat obtenu : Notre résultat montre un groupe de proies essayant d'échapper à l'attaque du prédateur en préservant sa cohésion, suivant le même chemin et son chef car sa force réside dans sa cohésion.

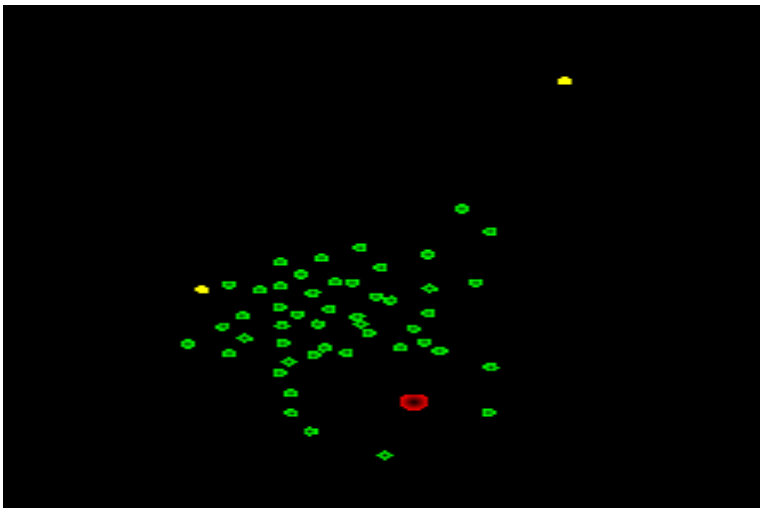


Figure 33 Evitement de prédateurs et suivi de leaders(le boid en rouge c'est le prédateur).

Test4 : Le comportement de prédateur

Les prédateurs dépendent de leur nourriture sur la prédation et attaquer les animaux pour les manger, la configuration utilisée c'est :

```
MAX_PREDATORS 2;
PREY_DETECTION_RANGE 1000.0f;
PREDATOR_FLOWING_A_LEADER_SPEED_MODIFIER 2.0;
PREDATOR_SPEED_MODIFIER 10.0;
BOUNDING_VOL_FOR_ENEMIES 2.0f;
```

Résultat obtenu : Le résultat que nous obtenons montre l'attaque des prédateurs sur les proies et chaque prédateur est indépendant de l'autre, donc il n'y a pas de collision entre eux.

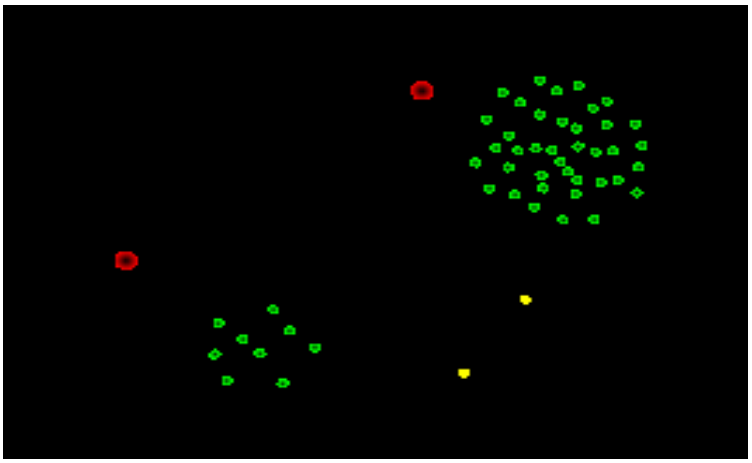


Figure 34 Comportement de suivi des proies.

Discussions des résultats

Lors de la soumission du formulaire original, Reynolds a noté qu'il n'était pas possible d'utiliser les scripts pour effectuer des animations réalistes de nuages d'oiseaux artificiels. Son idée était donc la suivante: les chaudières doivent être influencées par les autres afin de se déplacer de manière cohérente et crédible. Une citation de cette approche résume: "Le comportement de Boid dépend non seulement de l'état interne, mais aussi de l'état externe." Ainsi, chaque agent suit trois règles de comportement: R.1 Évitement de collision, R.2 Centrage, R.3 Adaptation de la vitesse. Notre travail était basé sur l'utilisation des règles de Reynolds pour modéliser le comportement d'un groupe de poissons afin d'obtenir un comportement de groupe cohérent, nous avons ajouté à cette modélisation le comportement de prédation, la réaction des proies et l'attaque des prédateurs. Où nous sommes arrivés à une conclusion similaire aux résultats étudiés par Reynolds.

4.2 Conclusion

Au terme de ce dernier chapitre, nous avons repris la démarche entreprise pour la réalisation de notre projet.

Tout d'abord, nous avons présenté notre environnement de travail et les outils de programmation utilisés avant de passer à la phase d'implémentation, en commençant par la modélisation boids.

Ensuite, nous sommes passés à la création d'un flock, et l'ajout des boids à notre système, par l'utilisation des règles du Reynolds on a pu simuler les comportements d'un groupe de boids dans un cas de prédation.

Finalement, nous avons terminé par la présentation de notre réalisation de notre travail, et c'en discutant les résultats que nous avons pu obtenir dans notre système.

Conclusion générale

Ce mémoire de fin d'études a eu pour objectif de simuler le comportement de mouvement en groupe avec prédateurs, Son but est de faciliter la compréhension du comportement d'une proie vérifiée se déplace en groupe et du comportement du prédateur lors l'attaque.
.
Pour arriver à ce résultat, nous avons d'abord dû étudier les règles de Reynolds qui se spécialisent dans le comportement des mouvements des oiseaux et comment les utiliser pour modéliser le comportement des poissons à travers ces règles.

Nous avons commencé le premier chapitre par une introduction et une étude sur les simulations comportementales et la modélisation des environnements comportementaux et virtuels.

Dans le deuxième chapitre, tiré de la vie artificielle, nous parlons des modèles que Reynolds nous a proposé dans l'illustre la simulation du comportement individuel et de groupe à travers Bird-Oid.

Au chapitre trois, nous avons, propose de concevoir notre systme, explorant les modèles les plus importants que nous simulerons.

Dans notre travail, nous nous sommes appuyés sur les règles de Reynolds pour visualiser le comportement des poissons et comment le comportement individuel affecté le groupe, dont nous avons ajouté une nouvelle perception du comportement, c'est le comportement des prédateurs durs et de la façade chacun interagit.

Le dernier chapitre est consacré à la réalisation de nos travaux. Nous avons décrit les différentes étapes qui ont été réalisées pour développer notre projet. Ce chapitre comprend également les résultats de l'enquête.

En analysant ces résultats, nous avons confirmé l'effet du comportement individuel sur le groupe.

Enfin, on peut dire que le comportement animal est un sujet très large, et leur rôle nous a incités à poursuivre et à comprendre ce domaine pour en apprendre de plus en plus sur ce phénomène important.

Perspectives

Simuler le comportement des poissons et des oiseaux est un très gros sujet qui a toujours besoin d'être amélioré pour approfondir dans ce domaine, car une meilleure compréhension du monde animal nous permet de mieux contrôler leur comportement, ce qui augmente l'opportunité de savoir comment interagir face aux différents défis qui entravent le développement du comportement de ce phénomène important.
Ce qui reste à faire pour les travaux futurs pour mieux développer ce travail :

- Simulation des comportements des poissons en trois dimensions;

Bibliographie

[ABW06] G. Antonini, M. Bierlaire, and M. Weber. Discrete choice models of pedestrian walking behavior. Transportation Research Part B: Methodological, volume 40 pages 667 – 687, 2006. 13

[Avr08] Quentin Avril. Articles Google Peuplement automatisé de bases géométrique urbaines. Université de Rennes1, [IFSIC], [IRISA].06-JUIN-2008.

[Don04] Stéphane Donikian. Modélisation, contrôle et animation d'agents virtuels autonomes évoluant dans les environnements informés et structurés. Habilitation à diriger des recherches, L'Université de Rennes1 Institut de Formation Supérieure en Informatique et en Communication 2004.

[GKM+01] S. Goldenstein, M. Karavelas, D. Metaxas, L. Guibas, E. Aaron, and A. Goswami. Scalable nonlinear dynamical systems for agent steering and crowd simulation. Computers & Graphics, volume 25 pages 983 – 998, 2001. 13

[GSC+05] Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernandez-Madrigal & Javier González. Multihier archical semantic maps for mobile robotics.In International Conference on Intelligent Robots and Systems, pages 2278–2283. IEEE/RSJ, 2005.

[KSHF09] M. Kapadia, S. Singh, W. Hewlett, and P. Faloutsos. Egocentric affordance fields in pedestrian steering. In I3D'09: Proceedings of the 2009 symposium on Interactive 3D graphics and games, pages 215–223, New York, NY, USA, 2009. ACM. 13

[Lam03] Fabrice Lamarche. Humanoïdes virtuels, réaction et cognition : une architecture pour leur autonomie. PhD thesis, 2003.

[LD09] Fabrice Lamarche and Stéphane Donikian. Crowd of virtual humans : a new approach for real time navigation in complex and structured environments. Comput. Graph. Forum, volume 23(3) pages 509 518, 2009.

[Mar06] Clodéric Mars. Navigation piétonnière en environnement informer pour des humanoïdes virtuels, Université de Rennes, IRISA, IFSIC, 07-02-2006.

[New90] A. Newell. Unified theories of cognition. Harward University Press, 1990.

[OOM92] Nassima Ouramdane, Samir Otmane, Malik Mallem. Interaction 3D en Réalité Virtuelle - Etat de l'art. hal.archives-ouvertes.fr. 18 Avril 2009.

[PJ11] Andrzej Pronobis & Patric Jensfelt. Hierarchical multi-modal place categorization. European Conference on Mobile Robots, vol. 11, pages 159-164, 2011.

[PMCJ10] Andrzej Pronobis, O Martinez Mozos, Barbara Caputo & Patric Jensfelt. Multi-modal semantic place classification. The International Journal of Robotics Research, vol. 29, no. 2-3, pages 298–320, 2010.

[QH10] F. Qiu and X. Hu. Modeling group structures in pedestrian crowd simulation. Simulation Modelling Practice and Theory, volume 18, pages 190 – 205, 2010.12.

[Rey99] C. W. Reynolds. Steering behaviors for autonomous characters. In Game Developers Conference. http://www.red3d.com/cwr/steer/gdc99, 1999.8, 12, 20, 77, 96, 19, 9.

[Tho99] Gwenola Thomas. Environnements virtuels urbains : modélisation des informations nécessaires à la simulation de piétons. PhD thesis. Décembre 16 1999.

[TMFR03] Antonio Torralba, Kevin P. Murphy, William T. Freeman & Mark A. Rubin. Context-based vision system for place and object recognition. In International Conference on Computer Vision, pages 273–280. IEEE, 2003.

[Tou16] Sylvain Toulet. Déplacements collectifs auto-organisés : décision individuelle et transfert d'information. Biodiversité. Thèse de doctorat Université Paul Sabatier - Toulouse III, 2015. France.

[UN00] Iwan Ulrich & Illah Nourbakhsh. Appearance-based place recognition for topological localization. In International Conference on Robotics and Automation, volume 2, pages 1023–1029. IEEE, 2000.

[VSLM11] Pooja Viswanathan, Tristram Southey, James Little & Alan Mackworth.Place classification using visual object categorization and global information. In Canadian Conference on Computer and Robot Vision, pages 1–7. IEEE, 2011.

[VS04] Julia Vogel & Bernt Schiele. A semantic typicality measure for

natural scene categorization. In: Pattern Recognition Symposium DAGM, pages 195–203. Springer, 2004.