

N° d'ordre :

N° de Série :



Faculté des Mathématiques et de l'Informatique

Département d'Informatique

THÈSE

Présentée pour obtenir le diplôme de

DOCTORAT EN SCIENCES

Filière : Informatique

Spécialité : Informatique

Par

Mohamed CHATRA

THÈME

**Fouille de données des comportements réalistes
des foules de piétons**

Soutenue publiquement le :21/03/2024

Devant le jury composé de :

M. Samir AKHROUF	Professeur	Université de M'sila	Président
M. Mustapha BOURAHLA	Professeur	Université de M'sila	Rapporteur
M. Farid NOUIOUA	Professeur	Université de BBA	Examineur
M. Abdelouahab ATTIA	MCA	Université de BBA	Examineur

Année universitaire : 2023-2024

Dédicaces

« Louange à Dieu, le seul et unique »

A mes très chers parents...

Témoignage d'affection et de grande reconnaissance,

Que Dieu les garde pour moi

A toute ma famille,

Mes très chers frères et sœurs,

Mes chers Grands-parents,

Mes Oncles et Tantes

A tous mes amis,

A tous ceux que j'aime

A tous ceux qui ont participé de loin ou de près à la réalisation de ce travail.

Avec l'expression de tous mes sentiments de respect,

Je dédie ce modeste travail.

Mohamed Chatra

Remerciements

Je remercie tout d'abord le bon dieu pour m'avoir donné le courage et la santé pour accomplir ce travail.

Je tiens à remercier vivement mon directeur de ma thèse, le Professeur Mustapha BOURAHLA pour m'avoir proposé ce sujet et dirigé mon travail ainsi que pour la marque de confiance qu'ils m'ont manifestée.

Je remercie le président de jury monsieur Samir AKHROUF, professeur à l'université de M'Sila qui m'a fait l'honneur de présider le jury.

Je remercie également monsieur Farid NOUJOUA, professeur à l'université de Bordj Bou Arreridj pour m'avoir fait l'honneur d'évaluer mon travail.

Je remercie aussi monsieur Abdelouahab ASSIA, professeur à l'université de Bordj Bou Arreridj pour l'intérêt qu'il a bien voulu porter à ce travail en acceptant d'être examinateur.

Enfin, que tous ceux qui mon ont aidé et encouragé de près ou de loin dans la concrétisation de ce travail, trouvent ici ma gratitude et mes sincères remerciements.

ملخص

في هذه الأطروحة، قمنا بتطوير وصف لنموذج قائم على الوكيل لمحاكاة إخلاء الحشود من الأماكن المادية المعقدة للهروب من المواقف الخطرة. يصف هذا النموذج مساحة مادية تحتوي على مجموعة من الأسوار والعوائق ذات الأشكال المختلفة، بالإضافة إلى باب الخروج. يوصف المشاة الذين يشكلون الحشد ويتحركون عبر هذا الفضاء من أجل إخراجهم بأنهم عملاء أذكىاء مجهزون بالتعلم الآلي الخاضع للإشراف باستخدام المعرفة المستخرجة باستخدام استخراج البيانات والتي تعتمد على إدراك بيئة معينة بشكل مختلف. تم تطوير وصف هذا النموذج باستخدام لغة بايثون حيث يمثل تنفيذه محاكاة له. قبل المحاكاة، يمكن التحقق من صحة النموذج باستخدام رسم متحرك مكتوب بلغة بايثون لحل المشاكل المحتملة في وصف النموذج. تم تقديم تقييم لأداء النموذج باستخدام تحليل نتائج المحاكاة ويبين هذا التقييم أن هذه النتائج مشجعة للغاية.

الكلمات الدالة: سلوك إخلاء الحشود، محاكاة النماذج المعتمدة على الوكيل، الذكاء الاصطناعي، الرسوم المتحركة للسلوك، تقييم الفعالية والكفاءة

Summary

In this thesis, we developed a description of an agent-based model to simulate the evacuation of crowds from complex physical spaces to escape dangerous situations. This model describes a physical space containing a set of fences and obstacles of different shapes, as well as an exit door. Pedestrians making up the crowd and moving through this space in order to be evacuated are described as intelligent agents equipped with supervised machine learning using knowledge extracted using data mining on synthetic data and which is based on perception to perceive a particular environment differently. The description of this model is developed with the Python language where its execution represents its simulation. Before simulation, the model can be validated using an animation written in the Python language to resolve possible model description problems. An evaluation of the model performance is presented using an analysis of the simulation results and this evaluation shows that these results are very encouraging.

Keywords: *Crowd Evacuation Behavior, Agent-based Model Simulation, Artificial Intelligence, Behavior Animation, Effectiveness and Efficiency Evaluation*

Résumé

Dans cette thèse, nous avons développé une description d'un modèle basé sur des agents pour simuler l'évacuation de foules d'espaces physiques complexes pour échapper à des situations dangereuses. Ce modèle décrit un espace physique contenant un ensemble de clôtures et d'obstacles de formes différentes, ainsi qu'une porte de sortie. Les piétons composant la foule et se déplaçant dans cet espace afin d'être évacués sont décrits comme des agents intelligents dotés d'un apprentissage automatique supervisé utilisant des connaissances extraites à l'aide de fouille de données sur des données synthétiques et qui sont basées sur la perception pour percevoir différemment un environnement particulier. La description de ce modèle est développée avec le langage Python où son exécution représente sa simulation. Avant la simulation, le modèle peut être validé à l'aide d'une animation écrite avec le langage Python et ce pour résoudre d'éventuels problèmes de description du modèle. Une évaluation des performances du modèle est présentée à l'aide d'une analyse des résultats de simulation et cette évaluation montre que ces résultats sont très encourageants.

Mots clés: *Comportement d'évacuation des foules, simulation de modèles basés sur des agents, intelligence artificielle, animation comportementale, évaluation de l'efficacité et de l'efficience.*

Liste des abréviations

AC	Automate Cellulaire
D	La distance
EPFL	École polytechnique fédérale de Lausanne
GSB	Goal-Seeking Behavior (comportement de recherche d'objectifs)
HCSM	Hiérarchie de Chomsky
HFSM	Hierarchical Finite State Machine
IA	Intelligence Artificielle
IRISA	Institut de Recherche en Informatique et Systèmes Aléatoires
IRIT	Institut de Recherche en Informatique de Toulouse
ISI	Ingénierie des Systèmes d'Information
KDD	Knowledge Discovery in Databases
MACEMACES	<i>Multi Agent Communication for Evacuation Simulation</i>
MAGS	Multi-Agent Geo-Simulations
P	Piéton (Pedestrian)
PMFserv	Performance Moderator Functions server
s	espace physique
s_p	position du secteur visuel
ViCrowd	Virtual Crowd
HiDAC	High-Density Autonomous Crowds

Table des matières

Introduction générale.....	14
1. Contexte général.....	14
2. Objectifs et contributions	14
3. Les travaux similaires.....	15
4. Organisation du manuscrit.....	16
État de l’art	19
Chapitre 1 : Animation comportementale	19
5. Contexte et définition	19
6. Domaines d’applications	26
2.1 Les jeux vidéo	26
2.2 Les films et effets spéciaux	26
2.3 Validité ergonomique des sites.....	26
2.4 Mise en situation	26
7. Les premiers travaux de l’animation comportementale	26
8. La boucle de l’animation comportementale	26
9. Conclusion.....	26
Chapitre 2 : Foule de piétons.....	27
1. Introduction	27
2. Piéton virtuel	34
2.1 Propriétés des piétons virtuels	34
2.2 Classification des humains virtuels	34
2.3. Animation d’humain virtuel	34
3. La foule.....	34
3.1. Comportement de foule	34
3.2. Animation de foule	34
4. Conclusion.....	34
Chapitre 3 : La modélisation de foules.....	35
1. Introduction	35
2. Les aspects sociologiques et d’interaction de foule.....	46
3. Les deux approches de foule	46
3.1 Les modèles microscopiques.....	46
3.2 Les modèles macroscopiques	46
4. Les systèmes récents de simulation piétonniers	55
5. Classifications des méthodes de foule	55
6. Synthèse sur les modèles de foule	57
7. Conclusion.....	57
Contribution	59
Chapitre 4 : Modèle de simulation proposé.....	59
1. Introduction	59
2. Fouille de données (Data mining)	60
3. Description du modèle d’évacuation des foules.....	62
3.1 Description de l'espace physique.....	62
3.2 Description des piétons.....	63
3.3 Les comportements des piétons.....	69
4. Conclusion.....	75
Chapitre 5 : Analyse de résultats de simulation	76
1. Introduction	76
2. Vitesse de mouvement d’individu.....	76
3. Vitesse de mouvement et densité	76
4. Notions de comportement.....	78
5. Validation du modèle et simulation.....	78
6. Expériences et évaluation des performances	80
7. Etude comparative	86

8. Conclusion.....	87
Conclusions et perspectives.....	89
Références	91

Table des figures

Figure 1.1 : Le jeu vidéo « Les Sims 2 » de Electronic Arts [35].	26
Figure 1.2 : L'environnement virtuel du jeu en ligne « Second Life » [35].	26
Figure 1.3 : Le film d'animation « Mulan » de Disney [35].	26
Figure 1.4 : La simulation d'entraînement Firefighter Training (IRIT) [35].	26
Figure 1.5 : Visite virtuelle de la ville de Grenoble (Artesia) [35].	26
Figure 1.6 : Visite virtuelle de la cité des sciences et de l'industrie (IRISA) [35].	26
Figure 1.7 : la boucle d'animation comportementale [36].	26
Figure 1.8 : Hiérarchisation des comportements illustrée par A. Newell dans sa pyramide comportementale [36].	26
Figure 2.1 : Structure du modèle comportemental [34].	34
Figure 2.2 Acteurs autonomes (Design : Mireille Clavien) [40].	34
Figure 2.3 interaction entre deux acteurs [40].	34
Figure 2.4 : Système à base de particules [33]. S. Goldenstein et al [50].	34
Figure 2.5 : Règles comportementales du modèle Flocks of Boids.	34
Figure 2.6 : Les boids de Reynolds et leur utilisation au cinéma.	34
Figure 2.7 : Projet de MAGS [57].	34
Figure 3.1 : La simulation de modèle de Helbing avec 10.000 individus [33].	46
Figure 3.2 : Simulation de foule par AC 2D structure étant à la base de grille [71].	46
Figure 3.3 : Carreaux de flux et Champs de Vitesse.	46
Figure 3.4 : interface du logiciel permettant de choisir des carreaux prédéfinis [66].	46
Figure 3.5 : Deux applications du système Flow Tiles [66].	46
Figure 3.6 : Capture d'écran d'un brouillard tourbillonnant généré à l'aide de champs de vitesses dynamiques.	46
Figure 3.7 : Les 3 règles du système de Reynolds (le point rouge représente le centre du système de Boids).	46
Figure 3.8 Installation	46
Figure 3.9 : Robots unijambistes à rapatrier dans l'enclos.	46
Figure 3.10 cyclistes virtuels.	46
Figure 3.11 : Installation.	46
Figure 3.12 : Forces dynamiques d'un cycliste virtuel.	46
Figure 3.13 : Simulation de l'évacuation d'un immeuble.	46
Figure 3.14 : Comportements des agents.	46
Figure 3.15 : Gestion d'une foule dans un musée.	46
Figure 3.16 : Gestion de la foule de la Figure 1 après introduction d'un élément de panique.	46
Figure 3.17 : Différents états émotionnels des individus.	46
Figure 3.18 : Différentes façon de marcher selon les émotions des agents.	46
Figure 3.19 : Simulation dans une gare.	46
Figure 3.20 : Distributions de densité pendant une évacuation dans l'exode [85].	55
Figure 3.21 : Simulation d'une évacuation avec l'exode de bâtiment et de sa visualisation en différé dans 3D avec le vrExodus [86].	55
Figure 3.22 : Densité et simulation de foule dans Simulex.	55
Figure 3.23 : Simulation de légion d'une station de train (légion 2003).	55
Figure 3.24 : simulation 2D par STEPS.	55
Figure 3.25 : L'interface d'Open street [94].	55
Figure 3.26 : Simulation d'évacuation avec Massive WS pour la conception architecturale et l'évaluation.	55
Figure 3.27 : Navigation réactive [95].	55
Figure 3.28 : Les poissons artificiels.	55
Figure 3.29 : Le système ACUMEN est appliqué aux sphères et les écoliers [96].	55

Figure 3.30 : Les piétons autonomes [100].	55
Figure 3.31 : Graphique de visibilité de syntaxe de l'espace [104].	55
Figure 4.1: Le processus de KDD (Knowledge Discovery in Databases)	60
Figure 4.2 : l'environnements de programmation	62
Figure 4.3 : les éléments de l'espace physique modélisé	63
Figure 4.4 : Vue générale du modèle de décision.	63
Figure 4.5 : Moteur d'inférence pour chaque piéton.	64
.....	64
Figure 4.6 : les champs visuels de piéton	64
Figure 4.7 : Représentation géométrique de valeur de fonction $tg(\alpha)$.	65
Figure 4.8 : Principe de la mesure de distance par la fonction tangente	65
Figure 4.9 Basse de connaissances	66
Figure 4.10 : les caractéristiques de piétons	67
Figure 4.11 Architecture de navigation pour chaque piéton (vue schématique)	68
Figure 4.12 : les champs visuels de piéton	69
Figure 4.13 : Les comportements de piéton	70
Figure 4.15 : comportement d'évitement des obstacles.	74
Figure 5.1 : Relation débit-densité pour la marche de piéton	77
Figure 5.2 : Relations vitesse-densité	77
Figure 5.3 : Animation avec 8 piétons	80
Figure 5.4 : Simulation du modèle d'évacuation des foules	80
Figure 5.5 : Simulation de foule avec différents situations nombre des obstacles et nombre de piétons	83
Figure 5.6 : Relations de densité entre la vitesse moyenne (à gauche), le nombre total de mouvements (au centre) et la distance totale parcourue (à droite)	84
.....	86
Figure 5.7 : Évaluation des performances	86
Figure 5.8 : comparaison entre les vitesses d'évacuation (notre simulation nommée par SimLing)	86

Liste des tableaux

Tableau 3.1 : les automates cellulaires	46
Tableau 3.2 : Les différentes approches dans le domaine [67].	55
Tableau 3.3 : Comparaison de différents systèmes pour l'animation de grands groupes.	57
Tableau 5.1 : Vitesses moyennes de marche dans différents Continents [130]	76
Tableau 5.2 : Résultats expérimentaux	84
Tableau 5.3 : comparaison entre notre simulation et quelques simulateurs de foule (facteur de temps /s)	86

Introduction générale

Introduction générale

1. Contexte général

L'analyse des comportements réalistes des foules de piétons est un domaine de recherche important dans la fouille de données et la modélisation des systèmes complexes. Cela peut avoir des applications dans la planification urbaine, la sécurité des foules, la conception de bâtiments, les simulations de désastres, et bien d'autres domaines.

Il est devenu courant que les gens se rassemblent dans des espaces publics comme les centres commerciaux, les parcs à thème et les stations de métro. Cet assemblage présente des difficultés pour les planificateurs et les administrateurs ainsi qu'une variété de risques pour la vie des gens (tels que le piétinement et la surpopulation). Lorsqu'une urgence survient, la conduite irrationnelle des foules provoquée par la peur ou même la panique pendant le processus d'évacuation peut créer des pertes incommensurables. Pour cette raison, les inquiétudes quant à la compréhension de la dynamique des foules dans les situations d'évacuation de routine et d'urgence se sont développées rapidement [1, 2].

Les simulations à base-modèle sont souvent les principales méthodologies de recherche utilisées pour analyser l'évacuation de la foule au lieu des expériences du monde réel, qui sont très difficiles à examiner le comportement de la foule lors d'une évacuation d'urgence [3, 4, 5]. Dans ce cas, les modèles de simulation sont pilotés par des informations environnementales, qui devraient être qualitatives plutôt que quantitatives.

Les principaux éléments des modèles sont les humains, qui obtiennent leurs informations environnementales par perception plutôt que par mesure. Parce que chaque piéton perçoit différemment un environnement particulier, il est difficile d'estimer dans quelle mesure les stimuli environnementaux réels sont présents. À la lumière de cela, il est logique de modéliser et d'étudier les actions et les caractéristiques humaines en utilisant des données linguistiques (mots) plutôt que des quantités numériques [6, 7, 8].

2. Objectifs et contributions

Dans cette thèse, nous avons développé un modèle d'évacuation de foule, où un piéton est considéré comme un individu distinct (autonome). Ce modèle intégrera des interactions complexes entre les personnes et leurs environnements. Les comportements d'un piéton sont influencés à la fois par la conscience (perception) de l'individu et par l'environnement. Compte tenu des liens complexes avec l'environnement, il est difficile de développer un modèle qui décrit et prédit efficacement les activités d'un piéton.

Nous avons utilisé la fouille de données (KDD : Knowledge Discovery in Databases) pour classer les comportements de piétons dans des foules dans les cas de paniques où des données synthétiques générées par des simulateurs sont utilisés dans ce processus de la fouille de données.

Dans ce modèle, les données basées sur la perception sont pleinement utilisées, ainsi que l'expérience et l'expertise humaines pour agir sainement en cas d'évacuation de foule tout en tenant compte de l'impact de l'environnement. Les valeurs précises des interactions complexes avec l'environnement, telles que les vitesses, les directions et les distances, ont des effets sur les comportements d'un piéton.

Nous classerons les informations obtenues à partir de paramètres considérés comme base de la perception en utilisant les techniques de fouille de données et l'intelligence artificielle (IA) [9]. Ainsi, nous devons créer un modèle utilisable qui peut utiliser pleinement les données basées

sur la perception et capturer le lien entre la conception environnementale et la perception des piétons.

Nous développons un modèle basé sur l'IA pour atteindre ces objectifs. Les perceptions d'un piéton de son environnement immédiat sont généralement exprimées à l'aide d'un langage naturel, qui est par nature ambigu et imprécis. Cette approche basée sur l'IA est capable de gérer l'imprécision et l'imprévisibilité inhérentes aux informations de perception.

Les processus de raisonnement et de prise de décision des piétons peuvent être articulés à l'aide d'un ensemble de règles d'inférence simples qui présentent les avantages de données d'entrée facilement disponibles et de sorties intelligibles [10]. L'aspect novateur de cette étude est la suggestion d'un modèle d'évacuation basé sur une technique d'IA qui peut efficacement prendre en compte les connaissances et l'expérience humaines ainsi que les perceptions des piétons de l'environnement.

Lors du processus de modélisation, les impacts des interactions complexes avec l'environnement sur la dynamique des piétons sont pris en compte qualitativement. Le modèle utilise quelques règles d'inférence simples pour représenter divers facteurs pouvant avoir une influence sur certains mouvements de piétons. En tant que système d'inférence d'IA avec des variables d'entrée et de sortie prédéterminées, le piéton a deux comportements combinés : éviter les obstacles et rechercher le but.

Ces actions sont entreprises pour inciter les piétons à se déplacer dans la direction de leurs objectifs, à choisir le chemin avec le moins d'énergie négative et à éviter les obstacles en avant. Les deux comportements sont combinés à l'aide d'une approche de priorité, où les décisions de direction et de vitesse de déplacement sont prises à chaque étape de la simulation du modèle.

Notre contribution est validée par une publication d'un article scientifique original dont le titre est « Agent-Based Simulation of Crowd Evacuation through Complex Spaces » qui est publié dans une revue de renommée mondiale dont le titre est « Ingénierie des Systèmes d'Information (ISI) ».

3. Les travaux similaires

Divers modèles de simulation ont été développés au cours des dernières décennies pour analyser la dynamique de l'évacuation des foules dans des situations régulières et d'urgence. Nous organiserons la discussion en modélisant des méthodologies : force sociale, automate cellulaire, logique floue et intelligence artificielle.

Helbing [11] a suggéré un modèle de force sociale qui considère les piétons comme des particules entraînées par la force et relève le défi d'évacuer les piétons effrayés. Un modèle d'automate cellulaire a été utilisé par Varas et al. [12] pour imiter l'évacuation des piétons d'une pièce avec des obstacles fixes. Afin d'imiter une expérience d'évacuation menée dans une salle de classe avec des barrières, Liu et al. [13] modification du modèle d'automate cellulaire.

Les modèles de gaz en treillis [14, 15], les modèles de théorie des jeux [16, 17], les contrôles de rétroaction basés sur l'optimisation [18] et d'autres sont fréquemment utilisés pour analyser les problèmes d'évacuation dans les espaces publics. De plus, plusieurs méthodologies ont été utilisées pour étudier les comportements d'évacuation dans diverses circonstances, y compris les tunnels routiers enfumés [19], les hôtels en feu [20], les immeubles de grande hauteur [21] et le bioterrorisme dans des contextes micro-spatiaux [22].

Ces résultats montrent que les types et les amplitudes d'influence du comportement des foules varient considérablement en fonction des situations et de l'environnement associé. Entre-temps,

de nombreux conseils utiles pour assurer une évacuation en toute sécurité sont également fournis par les réalisations de l'étude qui ont été publiées dans la littérature.

Nasir et al. [23] ont présenté un système génétique flou. La perception floue et la peur sont ancrées dans la pensée humaine, et Mauro et al. [24] ont proposé un modèle comportemental d'évacuation des foules basé sur une logique floue et tenant compte de ces aspects. De plus, plusieurs systèmes d'inférence floue sont conçus pour fournir un échappement, un délai de sortie et une direction de mouvement [7, 8].

Notamment, des études sur les comportements dynamiques des piétons dans des situations calmes et frénétiques ont été réalisées grâce à la modélisation et à la simulation basées sur l'intelligence artificielle [25, 26], grâce auxquelles des techniques d'intelligence artificielle peuvent être utilisées pour prédire l'esprit et la perception humains.

Pour imiter et modéliser le comportement de guidage des piétons dans des environnements construits, les auteurs de [27] ont présenté une étude sur la dynamique des mouvements des piétons lors d'une évacuation d'urgence à l'aide de l'apprentissage automatique. Dans les travaux présentés dans [28], une méthode d'apprentissage par renforcement est utilisée pour produire un modèle basé sur les données pour l'évacuation des foules. Li et al. [29] ont combiné les techniques d'apprentissage en profondeur et sociales pour développer un modèle de simulation d'évacuation de foule.

Toutes ces méthodologies présentent des avantages et des inconvénients. On trouve des techniques, qui traitent de l'aspect quantitatif de l'information qui auront des limites de perceptions sur l'environnement et d'autres, qui ont des limites sur la description de l'espace physique.

4. Organisation du manuscrit

Ce document est organisé en deux parties :

La première partie de cette thèse est la présentation de l'état de l'art sur tous les aspects qui entrent dans le domaine de l'animation comportementale et la simulation de foule de piétons. Cette partie est organisée en trois chapitres :

- Le premier chapitre introduit l'animation comportementale avec sa boucle, son domaine d'application, ainsi que les premiers travaux dans ce domaine.
- Le deuxième chapitre présente les propriétés des piétons, caractérisés par le comportement, l'autonomie, l'adaptation, la perception, la mémorisation, l'émotion, et la conscience. Le comportement de piéton et la classification des piétons seront traités par la suite.
- Le troisième chapitre est dédié à la présentation des outils développés dans ce domaine. Nous présentons aussi les différentes techniques de modélisation de foules, puis nous présentons les approches de simulation de foule macroscopique et microscopique avec leurs méthodes. Nous présentons à la fin de ce chapitre une synthèse sur les différents systèmes de modélisation.

La deuxième partie de cette thèse illustrée notre contribution elle est constituée de deux chapitres :

- Dans Le quatrième chapitre, la description du modèle d'évacuation des foules basé sur l'intelligence artificielle. Une section décrit comment valider le modèle proposé à l'aide de l'animation du modèle et comment produire des résultats de simulation.
- Le dernier chapitre présente notre modèle de simulation avec les programmes utilisés pour déterminer les différents comportements, nous avons également présenté quelques figures résultats pour notre animation. Puis, nous terminons ce chapitre par deux points :

- des rapports de simulation, une évaluation de la performance du modèle proposé avec analyse des résultats de simulation.
- Une étude comparative entre nos résultats de simulation et d'autres simulateurs.

Enfin, nous concluons cette thèse par des critiques sur ce travail et nous donnons certaines perspectives.

Première partie

État de l'art

État de l'art

Cette partie propose une analyse des travaux existants portant sur les travaux et les notions de l'animation comportementale des foules de piétons.

Chapitre 1 animation comportementale.

Chapitre 2 s'intéresse spécifiquement aux foules de piétons.

Enfin, le chapitre 3 se concentre sur les différentes techniques de modélisation de foules.

Chapitre 1 : Animation comportementale

1. Contexte et définition

Le but de l'animation est de créer une séquence d'images représentant une discrétisation temporelle de l'état d'un système évoluant au cours du temps. Cette évolution se matérialise par le mouvement des objets ou de la caméra dans la scène, la déformation géométrique des objets et la transformation de leurs caractéristiques non géométriques. Une suite définissant l'état du système aux instants et correspondant à un échantillonnage régulier du temps représente le système. L'animation implique à la fois l'expression de la phase de spécification et l'évaluation de la phase de calcul. Le cœur d'un système d'animation est constitué des modèles de contrôle du mouvement des objets, qui peuvent être divisés en trois grandes familles :

- 1) Les modèles descriptifs qui cherchent à reproduire les effets sans comprendre les causes profondes.
- 2) Les modèles générateurs qui permettent de décrire les causes qui produisent les effets,
- 3) Les modèles comportementaux visent à reproduire le comportement externe ou interne d'entités dotées d'une certaine autonomie. Par conséquent, les modèles comportementaux sont classés en deux catégories :

3.1 Les modèles de transformation interne :

- Provoquant des changements externes de l'objet (par exemple croissance de plantes ou modèle de muscle).
- Les relations entre l'entité et son environnement sont quasi inexistantes, c'est l'élément important pour fournir une perception la plus réaliste possible du comportement de la plante (croissance) ou de l'individu (animation faciale, musculaire).

3.2 Les modèles d'animation externe :

- Définissant le comportement extérieur d'un être, ses actions et ses réactions, de manière individuelle (animal, humain) ou collective (nuée d'oiseaux, banc de poissons, troupeau de mammifères, foules d'humanoïdes).
- Sont tous fondés sur un système relationnel entre des acteurs et leur environnement ce qui implique deux types de relations : la perception et l'action.

L'animation comportementale utilise une boucle d'avertissement fermée à niveau élevé pour simuler des entités autonomes telles que des organisations et des êtres vivants. De tels acteurs peuvent percevoir leur environnement, prendre des décisions, interagir avec d'autres entités et faire quelques choses, comme marcher dans la rue ou saisir un objet. Si le rapport entre l'acteur et son environnement peut être simulé, des comportements réalistes des acteurs autonomes

évoluant dans des environnements complexes et structurés peuvent être obtenus. L'information qui doit être extraite ou interprétée de l'environnement dépend du niveau d'abstraction du raisonnement des acteurs autonomes.

2. Domaines d'applications

Les domaines d'application de l'animation comportementale sont très variés et s'étendent des domaines ludiques comme les jeux vidéo aux domaines médicaux et de soins aux phobies.

2.1 les jeux

L'amélioration de la puissance de calcul des ordinateurs et de la puissance des cartes graphiques permet désormais d'obtenir une bonne qualité de rendu graphique en utilisant relativement peu de temps processeur. Cette évolution permet aux créateurs de jeux vidéo de se concentrer sur le comportement des personnages avec lesquels les joueurs peuvent interagir. Les jeux de créatures de *Cyberlife* et *Lionhead Black and White* sont deux exemples notables qui présentent des mondes peuplés de créatures autonomes. Ces créatures peuvent prendre des décisions seules, l'intervention du joueur, qui peut, dans certaines circonstances, influencer sur leur comportement, est de tenter de leur apprendre des choses. Une nouvelle tendance : la fiction interactive a vu le jour avec le développement de l'animation comportementale. L'idée est de fournir les grandes lignes d'un scénario tout en laissant le joueur interagir avec des entités semi-autonomes à sa guise. Cependant, ces entités peuvent prendre des décisions pour interagir avec le joueur de manière « intelligemment ». Cela offre une grande sensation de liberté, mais cela peut également entraîner des situations imprévues en fonction des actions de l'utilisateur. En plus de la qualité des animations et des graphiques, la crédibilité et la cohérence du comportement sont cruciales pour le réalisme et l'implication du joueur dans le domaine du jeu [34].



Figure 1.1 : Le jeu vidéo « Les Sims 2 » de Electronic Arts [35].



Figure 1.2 : L'environnement virtuel du jeu en ligne « Second Life » [35].

2.2 Les films

Depuis un certain temps, l'animation comportementale a fait son apparition dans le domaine de la cinématographie et des effets spéciaux. La tendance actuelle est de l'augmenter en développant des outils de modélisation du comportement pour les logiciels de production d'animation en 3D. Cette tendance est due à un besoin accru de productivité à travers des circonstances de plus en plus complexes mais allant de pair avec des exigences de qualité croissantes. Le deuxième volet de l'adaptation cinématographique du seigneur des anneaux est l'un des meilleurs exemples actuels. Un outil unique, MASSIVE, a été utilisé lors de la grande bataille du gouffre de Helm pour reproduire les comportements de foule et simuler la partie décisionnelle des entités. Il existe actuellement de nombreux outils disponibles dans le même domaine, tels que l'outil AI-Implant de BIOGRAPHICS, qui peut être utilisé comme module pour 3D studio Max et Maya, ou encore les modules RTK-Crowd et RTK-Behavior de SOFTIMAGE. Ces outils offrent de grands avantages en termes de conception :

- La gestion automatisée des comportements de groupe simplifie le travail du concepteur.
- Les animations adaptées au contexte peuvent être générées automatiquement grâce aux modèles décisionnels.
- Les scènes sont plus variées, ce qui les rend plus réelles.

Ces techniques permettent de décrire, de manière implicite, des scènes d'une complexité inégalable à l'aide d'outils dédiés grâce à l'automatisation. [34].



Figure 1.3 : Le film d'animation « Mulan » de Disney [35].

2.3 Validité ergonomique des sites

La création d'espaces publics présente des problèmes d'ordre ergonomique, c'est-à-dire la qualité de leur utilisation. Les problèmes de navigation à l'intérieur des lieux, la lisibilité des différents panneaux de direction ou les situations de panique peuvent survenir. En utilisant des modèles basés sur l'analyse du comportement humain, l'animation comportementale peut être utilisée pour effectuer des validations sur des maquettes virtuelles. Avant la construction des différents aménagements, les problèmes potentiels peuvent alors être identifiés et résolus. Dans ce contexte, son utilisation présente des avantages à la fois sécuritaires et ergonomiques et économiques [34].



Figure 1.4 : La simulation d'entraînement Firefighter Training (IRIT) [35].

2.4 Mise en situation

La réalité virtuelle permet de mettre un individu en situation dans un scénario en interagissant avec des agents autonomes. Ces compétences peuvent être utilisées dans un cadre pédagogique où l'interaction avec des humanoïdes intelligents augmentera la rapidité d'apprentissage grâce à des moyens d'interaction plus proches de la réalité. L'armée est un autre domaine d'application où la définition du comportement des humanoïdes permet de tester des scénarios catastrophes ou de mettre les soldats en situation. L'animation comportementale et la réalité virtuelle peuvent être utilisées dans les soins aux personnes phobiques dans les hôpitaux.

L'utilisation de la réalité virtuelle vise à plonger la personne dans un environnement qui provoque sa phobie. Les simulations de foules s'avèrent utiles dans le traitement de l'agoraphobie, par exemple, pour peupler les environnements virtuels et donner au patient un environnement réaliste [34].



Figure 1.5 : Visite virtuelle de la ville de Grenoble (Artesia) [35].



Figure 1.6 : Visite virtuelle de la cité des sciences et de l'industrie (IRISA) [35].

3. Les premiers travaux de l'animation comportementale

Les débuts de l'étude peuvent être remontés à la fin des années 80, en particulier l'article de C. Reynolds [38] qui traite de l'animation de nuées d'oiseaux. Jusqu'au milieu des années 90, un premier ensemble d'approches a été étudié en parallèle dans la littérature pour définir le modèle décisionnel :

Stimulus / réponses : Cette méthode transforme l'information passée pour définir le comportement des objets à partir d'un ensemble de capteurs et d'effecteurs connectés par un réseau de nœuds intermédiaires. Les modèles de type réseau de neurones sont inclus dans ce type d'approche. La façon dont un objet se comporte dépend de la façon dont il perçoit son environnement et de la façon dont cette perception est transmise aux effecteurs qui produisent le mouvement. L'avantage de cette méthode est qu'elle peut produire une grande variété de mouvements différents grâce au choix du jeu de paramètres, mais elle reste très abstraite. De plus, ces modèles sont classés comme des boîtes noires, ce qui signifie qu'il est impossible de modifier n'importe quel paramètre sans recommencer la configuration complète. De plus, avec cette méthode, il est impossible de vouloir contrôler un mouvement complexe car le pourcentage de bons contrôleurs diminue de manière exponentielle avec la croissance du nombre de paramètres à prendre en compte [36].

Les Règles de comportement : L'approche par règles de comportement, comme l'approche précédente, reçoit des informations restituant une certaine perception de l'environnement et produit un certain contrôle sur la motricité des objets en sortie. Ici, un ensemble de règles régissent le comportement des objets. Un arbre de décision, avec chaque branche représentant un comportement différent, peut être utilisé pour représenter les comportements potentiels d'un objet. L'application d'un algorithme de parcours d'arbre choisira une action satisfaisant les conditions de l'environnement courant. Les comportements autorisés par cette méthode sont plus abstraits que dans l'approche précédente. Le cœur du problème avec cette méthode est la pondération des différents comportements. La solution la plus simple consiste à faire un choix implicite sur l'ordre des règles (suite de si... alors... sinon), mais cette solution ne permet pas de spécifier des comportements complexes. Avec une pondération des branches, le parcours de l'arbre des possibilités permet de ne pas privilégier toujours la même règle, tandis que la prise en compte d'une hiérarchie d'experts permet de confronter plusieurs comportements concurrents, le choix final restant à l'expert de plus haut niveau. La multitude d'algorithmes de parcours d'arbre de décision qui intègrent des fonctions d'évaluation de diverses branches (essais successifs, séparation) peut être utilisée de cette manière [36].

Automates d'états finis : L'automate établit les différents enchaînements potentiels entre les comportements. Dès qu'il s'agit de modéliser un comportement un tant soit peu complexe, cette méthode trouve rapidement ses limites. La modélisation du comportement d'un conducteur a démontré la nécessité de passer à l'utilisation de plusieurs automates, dont l'exécution coordonnée doit être définie.

Le constat que l'on peut faire de ces différents travaux est que se sont des modèles ad hoc conçus pour être appliqués à chaque fois sur des cas particuliers, dans lesquels les objets et leur environnement sont relativement simples et les champs de perception et d'action sont limités. De plus, il existe une grande variété de comportements autorisés par chacune des approches :

- Si le niveau d'abstraction est très faible, seuls des comportements de type réflexes pourront être identifiés,
- Si le niveau d'abstraction est plus élevé, l'environnement est réduit, voire complètement défini, et donc la perception et l'action de l'entité sont très faibles.

Ces modèles sont relativement simples, ont des champs de perception et d'action limités et ne prennent pas en compte l'aspect temporel, qui est essentiel (mémorisation, prédiction, durée d'une action, séquençement de tâches). Les deux premières approches ont été rapidement abandonnées au profit de l'approche à base d'automates d'états finis dans leurs versions parallèles et hiérarchiques, car il est nécessaire de traiter conjointement les aspects continus et discrets, de coordonner les comportements concurrents et de gérer leur structure organisationnelle afin de rendre compte de la complexité décisionnelle : piles d'automates (EPFL, Suisse), ensembles d'automates communicants (PaT-Nets) (Université de Pennsylvanie, Etats-Unis), hiérarchie d'automates parallèles (HCSM) (Université d'Iowa, Etats-Unis) et systèmes de transitions parallèles hiérarchisés (HPTS) (IRISA, France). Le modèle HPTS1 est le seul qui peut traiter les aspects temporels et non déterministes et intègre un langage de spécification ainsi qu'un interpréteur et un générateur de code. Du côté industriel, Motion Factory (orientation jeux) a commercialisé un produit (Motivate) pour la spécification du comportement à base de machines à états finis hiérarchiques (HFSM) et est maintenant intégré au logiciel d'animation Softimage sous le nom de RTK Behavior [36].

4. La boucle de l'animation comportementale

L'environnement humain est dynamique. Il dispose de trois capacités essentielles qui lui permettent de réagir aux modifications de cet environnement et d'agir : la perception, la décision et l'action. La boucle perception-décision-action [37], qui est le modèle le plus courant pour représenter un organisme vivant, autonome et évoluant dans un environnement dynamique, est constituée de ces trois blocs. Ce modèle, qui est à la base d'un grand nombre de théories relatives au comportement, dans lesquelles chaque brique a un rôle bien défini, est illustré à la figure 1.7 :

La perception

Utilisez les informations sur l'environnement pour faciliter la prise de décision. Ce sont des informations locales ; un organisme, quel qu'il soit, ne peut percevoir que les informations à proximité. Il va sans dire que cette notion de proximité, en comparaison avec la taille du monde, peut être considérée comme ponctuelle.

L'action

Représente la capacité de l'organisme à modifier l'état du monde qui l'entoure. Parler, manipuler un objet, se déplacer sont quelques exemples d'actions....

La décision

Elle joue un rôle important en corrélant la perception et l'action. La perception donne aux capteurs (ou sens) des informations « abstraites » qui peuvent être mémorisées et exploitées pour aider à prendre des décisions. La partie décisionnelle utilise ces informations pour sélectionner une ou plusieurs actions qui correspondent au contexte et aux motivations de l'individu.

Il existe un certain nombre de retours/interdépendances entre la perception et l'action. Ceux-ci sont de natures différentes :

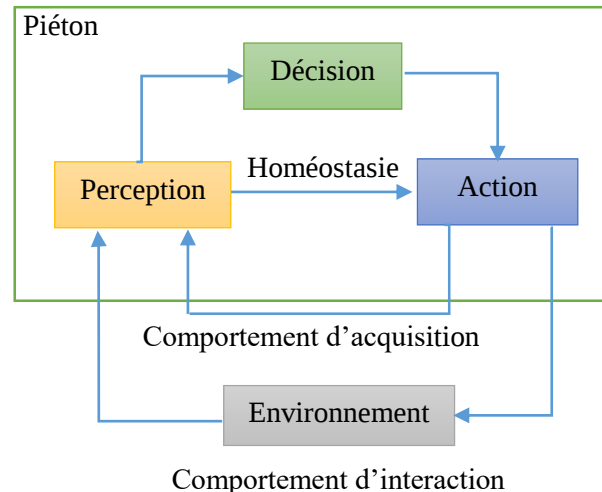


Figure 1.7 : la boucle d'animation comportementale [36].

- **L'homéostasie** est la boucle de régulation interne de l'entreprise. Elle reflète la préférence de l'être humain à maintenir ses paramètres biologiques en réponse aux changements dans son environnement. Cette boucle ne sera pas prise en compte dans la suite de cette étude car elle n'agit pas directement au niveau du comportement.
- **Le comportement d'acquisition** est la composante du comportement qui vise à améliorer la perception. Il peut s'agir, par exemple, de fixer une chose particulière pour obtenir les informations nécessaires au processus de prise de décision.
- **Les comportements d'interaction** constituent la boucle principale. Les actions, qui sont supervisées par le processus décisionnel et la perception, modifient l'état de l'environnement, traduisant l'activité de l'organisme dans son environnement.

Ces données illustrent le lien subtil qui existe entre l'état humain et l'environnement dans lequel il se développe. L'environnement semble déterminer le comportement, mais l'être humain l'influence inversement. Cette relation d'influence reflète l'autonomie des êtres vivants qui peuvent décider de leurs propres actions en fonction de ce qu'ils perçoivent et de leur état interne. [58].

H. Mallot a classé les boucles d'animation comportementale ou les boucles rétroaction [111]. De plus, il classe les comportements humains en quatre niveaux de complexité :

- 1) Le premier niveau décrit les comportements réflexes de type attraction / répulsion peuvent être définis par une interconnexion entre les capteurs et les effecteurs.
- 2) Le deuxième niveau décrit des comportements de type manœuvre qui nécessitent une intégration spatio-temporelle.
- 3) Le troisième niveau s'occupe de l'apprentissage, et est intimement lié aux données sensorielles.
- 4) Enfin, le quatrième niveau, dénommé cognitif, tient compte du but courant de l'individu, et s'inscrit donc dans un processus à plus long terme.

La théorie du contrôle est une méthode similaire à cette classification. Il a été créé grâce à des recherches en biologie qui ont impliqué des expériences sur des animaux qui ont été asservis. Selon son principe de base, lorsque l'état d'un organisme est différent de celui désiré, il va modifier son comportement en effectuant une série d'actions pour réduire cet écart. Par conséquent, le principe de rétroaction est repris dans cette théorie du contrôle par R. Lord et P. Levy [41], mais ils l'étendent à la régulation des comportements sociaux. Pour eux, ces boucles de rétroaction sont organisées hiérarchiquement et ont une idée de complexité croissante, ce qui

leur permet de prendre des décisions avec une grande généralité. Par conséquent, chaque niveau traitera d'un ensemble de tâches spécifiques tout en utilisant le principe de minimisation des erreurs. A. Newell [42] proposant une classification de ces niveaux de comportement hiérarchiques. Il précise que chaque niveau a ses propres fonctions, mais qu'il est capable de communiquer avec et d'influencer ses voisins immédiats sans pour autant avoir besoin de connaître comment ils fonctionnent. En informatique, ce principe est particulièrement fascinant car il peut être directement appliqué à un modèle de boîtes noires qui doivent communiquer des informations. La transmission d'informations doit être suffisamment synthétique tout en conservant une grande expressivité.

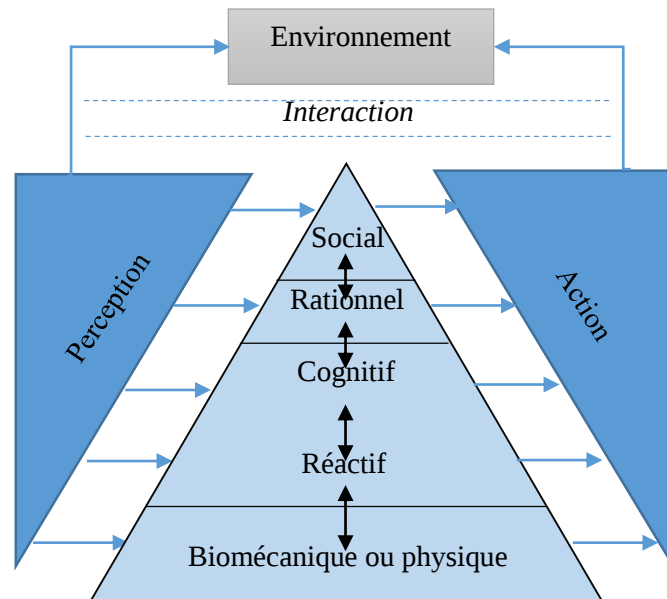


Figure 1.8 : Hiérarchisation des comportements illustrée par A. Newell dans sa pyramide comportementale [36].

5. Conclusion

Dans de nombreux domaines de la technologie et du divertissement, l'animation comportementale est un domaine en constante évolution. En créant des comportements réalistes et crédibles, elle vise à rendre les interactions entre les humains et les personnages virtuels plus naturelles et immersives. En assignant des comportements indépendants aux acteurs ou systèmes animés, l'animation comportementale est une partie de l'animation qui se rapproche des systèmes réels. Il n'y aura plus de système global contrôlant le mouvement de tous les acteurs, mais plutôt un mécanisme de prise de décision local intégré à chaque individu. Ainsi, l'animation comportementale est une méthode pour inciter les acteurs à interagir naturellement en reproduisant leurs compétences dans un environnement.

Chapitre 2 : Foule de piétons

1. Introduction

La simulation de foule est un sujet très ouvert, même si elle a été abordée depuis un certain temps. De nombreuses études continuent de s'intéresser à ce vaste sujet, que ce soit dans le but de créer une animation ayant un aspect réaliste ou encore de réaliser des simulations pour caractériser et étudier ce phénomène. Nous avons étudié l'entité qui compose la foule d'humains virtuels au début de ce chapitre. Et en deuxième lieu, nous définissons le concept de foule.

2. Piéton virtuel

2.1 Propriétés des piétons virtuels

L'objectif majeur de la modélisation des comportements des piétons est de construire des agents intelligents autonomes virtuels avec adaptation, perception et mémoire, capable d'agir librement et avec émotion, d'être conscient et imprévisible (figure 2.1).

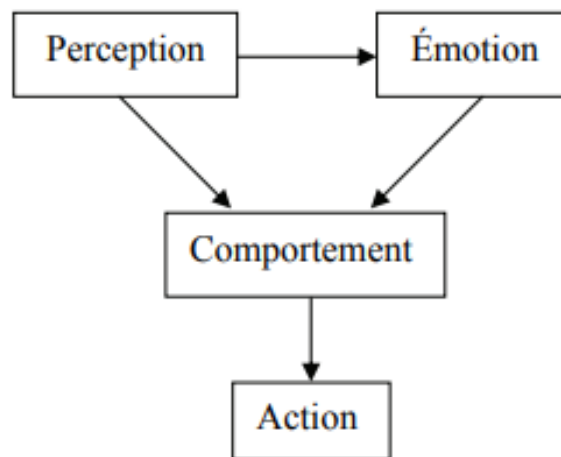


Figure 2.1 : Structure du modèle comportemental [34].

2.1.1 La perception

La conscience des éléments dans l'environnement à travers des sensations physiques est connue sous le nom de perception. Le processus consiste à doter les agents de détecteurs auditifs, tactiles et visuels afin de reproduire les actions quotidiennes des humains telles que l'aspect visuel, le mouvement et la réaction. Le système visuel est le sous-système perceptuel le plus important. Pour modéliser une animation comportementale, une approche basée sur la vision est idéale. Dans le contexte de la recherche de chemin, elle offre une approche universelle pour le transfert d'informations de l'environnement à l'acteur.

Au niveau supérieur, la perception peut être divisée comme indiqué par Un acteur peut voir les objets et les autres acteurs du quartier. Cependant, cela limite le nombre de comportements possibles car seules la présence et les caractéristiques d'un objet ou d'un acteur sont impliquées dans la sélection d'un comportement. Les actions des autres parties prenantes ne sont pas prises en compte. Trois types de perception sont créés par le module de perception : la perception de la présence des acteurs et des objets, la perception des actions des acteurs et la perception des acteurs effectuant des actions sur des objets [36].

2.1.2. L'émotion

L'émotion peut être définie comme l'aspect affectif de la conscience : un état de sentiment, une réaction psychique et physique (comme la colère ou la peur), subjectivement ressenti comme un sentiment fort et physiologique qui augmente les changements préparant le corps à une action vigoureuse immédiate. Les acteurs doivent être capables de répondre, physiquement et émotionnellement, à leur situation. Les designers ont un moyen direct d'influencer un état émotionnel de l'utilisateur grâce aux émotions visibles. Ainsi, les acteurs sont équipés d'un modèle informatique simple de comportement émotionnel, qui est lié au comportement comme les expressions de visage et qui peut influencer leurs actions. La réaction d'une personne à une perception est connue sous le nom d'émotion. Elle est amenée à répondre à l'aide d'une expression du visage, d'un geste ou de la sélection d'un comportement particulier. Une émotion survient entre une perception et une réaction. La façon dont deux personnes réagissent à la même perception peut varier. La réaction aux objets, les actions des agents et les événements provoquent des émotions. Trois types d'émotions peuvent être utilisés pour classer les émotions causées par des événements :

- Les émotions causées par des événements potentiels.
- Les événements affectant le destin d'autres.
- Les événements affectant le bien-être de l'acteur.

Chaque classe est caractérisée par des conditions d'apparition pour chacune de ses émotions et variables qui affectent son intensité. La perception d'un acteur détermine les sentiments qu'il ressent [36].

2.1.3. Le comportement

La façon dont les animaux et les humains agissent est souvent appelée comportement. Il est généralement décrit en termes de langage naturel, qui a une signification sociale, psychologique ou physiologique, mais qui n'est pas nécessairement facilement réductible au mouvement d'un ou plusieurs muscles.

Le comportement est également la réponse d'une personne, d'un groupe ou d'une espèce à son environnement. Le comportement comprend non seulement la réaction à l'environnement, mais aussi le flux d'information par lequel l'environnement affecte la créature vivante et la façon dont la créature code et utilise ces informations.

Le comportement peut être expliqué de manière hiérarchique. Le modèle comportemental divise un comportement en comportements plus petits qui peuvent être étudiés plus en détail.

Chaque étape de cette division hiérarchique comprend un ou plusieurs comportements qui sont exécutés séquentiellement ou simultanément. Le comportement est un niveau de hiérarchie qui comprend plusieurs comportements qui doivent être exécutés séquentiellement.

Chaque comportement qui appartient à un ordre de comportement est connu sous le nom de cellule comportementale. Lorsque les règles d'inhibition sont spécifiées, la cellule comportementale contient les comportements qui sont exécutés simultanément ou exclusivement. Les comportements d'une cellule comportementale sont composés ou fondamentaux.

Un comportement permet une décomposition récursive de la hiérarchie. Au bas de la décomposition hiérarchique, un comportement élémentaire encapsule un comportement spécialisé qui contrôle directement une ou plusieurs actions. Les comportements fondamentaux des entités peuvent être exécutés à chaque niveau de la structure de la hiérarchie, ce qui signifie qu'un comportement est exécuté récursivement.

On utilise une pile de comportements pour contrôler le comportement global d'un acteur. L'utilisateur pousse un ordre de comportements dans la pile de l'acteur au début de l'animation. Lorsque le comportement actuel est terminé, le système d'animation passe au comportement

suivant et l'exécute. Ce processus est répété jusqu'à ce que la pile de comportement de l'acteur se vide. Avec En utilisant ce contrôle de comportement avec une pile, un acteur devient plus indépendant et peut créer ses propres objectifs en exécutant le scénario original [36].

2.1.4. L'action

Le mécanisme comportemental d'un acteur détermine les actions à exécuter en fonction des informations perceptuelles. Les actions peuvent varier en niveau de complexité. Un acteur peut évoluer dans son environnement, interagir avec son environnement ou communiquer avec d'autres acteurs.

Les actions sont réalisées à l'aide d'une architecture de mouvement commune. En animant un modèle d'homme générique basé sur une hiérarchie de nœud, le module d'action gère l'exécution des actions utilisées par un comportement. En gérant des transitions fluides entre les actions finales et les actions amorçantes, il permet l'exécution simultanée ou séquentielle d'actions. L'animation est dirigée par une boucle comportementale, qui gère l'état du monde virtuel.

Le temps est incrémenté à chaque itération et le monde virtuel est mis à jour, en particulier avec une mise à jour de l'état de chaque objet et acteur. Dans le cas d'un acteur, la perception est d'abord exécutée, puis ses émotions sont produites, puis son comportement et ses actions [36].

2.1.5. La mémoire

Le pouvoir ou le processus de reproduction ou de rappel de ce qui a été appris et conservé, principalement par les mécanismes associatifs, est généralement défini comme mémoire. La mémoire est également le stockage d'informations apprises et à conserver provenant de l'activité d'un organisme ou d'une expérience. La mise en œuvre de la mémoire pour un acteur n'est pas très difficile car la mémoire est déjà un concept important en informatique. Les auteurs proposent, par exemple, une troisième mémoire visuelle globale qui permet à un acteur de retenir l'environnement, de le percevoir et de s'adapter à ses changements [36].

2.2 Classification des humains virtuels

Pour mieux comprendre les différentes catégories d'humains virtuels et leur développement, nous utiliserons la division de thalman [39] des humains virtuels :

- Avatar
- Acteur guidé
- Acteur autonome
- Acteur perceptif interactif

2.2.1 Les Avatars

Dans notre vocabulaire, les avatars sont des représentations fidèles de l'utilisateur. La méthode implique d'enregistrer directement les mouvements de l'utilisateur et de les transférer sur un humain virtuel qui aura également son apparence. Avoir des séries de capteurs comme les Flock of Birds est nécessaire pour implanter de tels avatars.

2.2.2 Les acteurs guidés

Les acteurs guidés sont ceux qui sont conduits par l'utilisateur mais dont les mouvements ne sont pas directement synchronisés avec ceux de l'utilisateur. Ils reposent sur l'idée de la métaphore directe en temps réel [40]. Pour mieux comprendre l'idée, voyons un exemple de métaphore classique : prendre en main une marionnette. Une marionnette est une poupée dont les membres peuvent être déplacés par des fils ou des ficelles. Le mouvement de la marionnette est contrôlé par les doigts humains. Le meilleur exemple de guidage d'acteurs en Réalité Virtuelle est la navigation effectuée dans les systèmes d'Environnements Distribués.

2.2.3 Les acteurs autonomes

Ils sont chargés de représenter un service ou un programme, tel que la navigation. Comme ces acteurs virtuels ne sont pas guidés par les utilisateurs, ils doivent avoir des comportements leur permettant d'accomplir leurs tâches de manière autonome. Il faut créer des comportements pour le mouvement et des mécanismes pour l'interaction. La recherche de pointe se produit dans ce domaine. Examinons l'exemple de l'évitement des obstacles, un problème que chacun d'entre nous résout fréquemment chaque jour. Les roboticiens ont introduit des techniques mathématiques qui permettent d'évaluer le ou les chemins possibles pour aller d'un point à un autre en évitant les obstacles à partir d'une configuration donnée. Cette approche est très utile pour un bras robotique qui se déplace mécaniquement dans un environnement connu et statique, mais elle n'est pas applicable au comportement d'un humain dans un environnement souvent inconnu et parfois dynamique. Lorsqu'on marche sur un trottoir, il est important de ne pas faire attention aux parcomètres et d'éviter les piétons qui passent en face de vous, sans oublier qu'ils essaieront également de vous éviter. Il est impossible de reproduire ce comportement dynamique en animation sans utiliser une approche sensorielle. En effet, c'est le but de nos études. Une vision, une audition et un toucher de synthèse sont désormais accessibles à nos humains virtuels. Ils peuvent se diriger vers un bruit déterminé ou éviter la table qu'ils voient, ou ils peuvent enfin réagir même dans l'obscurité en reconnaissant qu'ils ont cogné un mur. Ils peuvent même jouer au tennis en imitant leur adversaire.



Figure.2.2 Acteurs autonomes (Design : Mireille Clavien) [40].

Il s'agit d'une boucle appelée perception-action. Chaque instant, une perception de l'espace virtuel est créée pour chaque être virtuel, c'est-à-dire que l'ordinateur simule ce que chaque être virtuel peut voir, entendre et toucher à l'aide de ses sens virtuels. En conséquence, l'être virtuel disposera d'un champ de vision réel, ce qui lui permettra d'entendre les sons générés par d'autres personnes ou de se blottir sur une table. Ces êtres sont capables de prendre des décisions simples en fonction de ce qu'ils ressentent. Par exemple, un être virtuel, attiré par un son en dehors d'une pièce, pourra chercher la porte et, lorsqu'il l'aura aperçue, pourra trouver le chemin jusqu'à elle et sortir, sans qu'un vrai humain intervienne.

2.2.4 Les acteurs perceptifs et interactifs

Nous définissons un acteur perceptif et interactif comme un acteur conscient d'autres acteurs et des gens réels. Un tel acteur est aussi supposé être autonome bien sûr. De plus, il est capable de communiquer interactivement avec les autres acteurs quel que soit leur type et avec les gens réels (voir la figure 2.3).



Figure 2.3 interaction entre deux acteurs [40].

2.3. Animation d'humain virtuel

Il est un organisme vivant qui retient l'attention de l'être humain dans le cadre de l'animation comportementale. C'est un être vivant avec un comportement très complexe. Il est à la fois capable de raisonner et de réagir à l'environnement. Cette complexité est illustrée par l'immense potentiel d'interaction entre l'être humain et son environnement, ainsi que par sa capacité, souvent exploitée, de réaliser plusieurs tâches à la fois. Elle soulève de nombreux problèmes liés à l'animation comportementale et met l'accent sur le besoin d'outils et de modèles de description de comportement plus sophistiqués.

Il existe plusieurs niveaux d'animation pour un humain virtuel de synthèse : le niveau géométrique, le niveau cinématique, le niveau de contrôle du mouvement, le niveau réactif et enfin le niveau cognitif.

La représentation de l'humain virtuel dans l'environnement virtuel, qui se caractérise par un grand nombre de degrés de liberté, est fournie par le niveau géométrique. La couche cinématique, chargée de la gestion de l'animation, calcule les valeurs associées à ces degrés de liberté. La couche de contrôle du mouvement fournit des interfaces de haut niveau pour déterminer les gestes à effectuer par l'humain virtuel. Ces trois premiers niveaux, plus précisément la précision de représentation du modèle géométrique, ainsi que la qualité visuelle des mouvements qui peuvent être effectués, déterminent le réalisme de l'animation.

Les enchaînements d'actions atomiques définis par le niveau réactif permettent de traduire le comportement de l'humain virtuel sous la forme de ses interactions avec l'environnement et effectuent ainsi le lien avec l'animation. Enfin, le niveau cognitif représente les capacités de raisonnement dont peut être doté l'humain virtuel. Il utilise une image abstraite de l'environnement, ce qui lui permet de raisonner sur ses actions et leurs répercussions. Au moyen de la cohérence globale du comportement exhibé par l'humain virtuel, ces deux niveaux permettent d'obtenir un certain réalisme. Cette cohérence dépend principalement de la pertinence de l'enchaînement d'actions par rapport à l'état perceptible de l'environnement et aux motivations de l'humain virtuel. finalement, la brique d'évolution représente le même phénomène biologique que l'animation comportementale ne traite pas réellement et qui est plutôt du ressort de la vie artificielle [58].

3. La foule

3.1. Comportement de foule

Il est bien connu que dans les mouvements de foules, les flux de piétons marchant dans des directions opposées ont tendance à se séparer, ce qui conduit à la formation dynamique de files de piétons marchant dans la même direction. Il est démontré que les lois sur l'évolution des fluides peuvent être utilisées pour calculer une approximation de la tendance générale lorsque la densité des piétons dans un espace augmente. Dans les situations de panique, les piétons souhaitent se déplacer beaucoup plus rapidement que d'habitude, ce qui les pousse à interagir physiquement avec les autres. De plus, ils ont tendance à imiter les gens dans le flux. Turner

[43] utilise la théorie des normes pour dire que l'homogénéité dans une foule se produit non pas parce que des personnes avec les mêmes prédispositions se sont rassemblées spatialement, mais parce que les membres de la foule ont tendance à se conformer à une norme émergente. Selon Yamori [44], la notion de norme est inévitablement associée à la notion de régulation, qui est un comportement acquis comme élément normatif au sein d'une macrostructure ou d'une collectivité spécifique. Cependant, le rôle du cœur peut être joué par de petits groupes qui peuvent influencer de plus grandes unités. Boles [45] a vu des bandes dans les foules de piétons qui se déplaçaient sur les trottoirs et a démontré que les piétons avaient tendance à utiliser ces structures pour maximiser l'efficacité du trafic piétonnier. K. Yamori a cherché à examiner ces structures de bande à l'aide d'un microscope [44]. Il suppose qu'une densité critique de personnes est nécessaire pour que ce type de structure existe, et il cherche à déterminer les facteurs qui déterminent l'émergence de ces structures. Selon Milgram et Toch [46], une structure en anneau se formait lorsque les membres d'une foule avaient un point d'intérêt commun. Il est possible d'observer cette formation circulaire dans les foules qui se regroupent autour d'artistes de rue. Les relations entre les comportements micro et macroscopiques au sein d'une foule sont l'un des problèmes majeurs soulevés par Yamori [44]. L'enjeu est d'expliquer comment une personne peut affecter une collectivité et comment la collectivité va agir en retour sur le comportement individuel au fil du temps. Yamori se concentre sur l'étude de la formation des structures de bandes macroscopiques sur des passages piétons dans des situations où le trafic piétonnier est très dense, comme c'est le cas au Japon. Selon l'étude, une structure en forme de bande doit atteindre une masse critique.

3.2. Animation de foule

En raison de son faible coût de calcul, la simulation macroscopique a été historiquement utilisée pour simuler les déplacements de personnes. En effet, dans cette méthode, le piéton n'est pas considéré comme un individu, mais plutôt comme un composant d'une matière plus large. Henderson [47] suggère l'utilisation d'un modèle gazeux pour les densités faibles en comparant les mouvements des personnes à ceux des molécules de gaz en comparant leurs densités respectives. Le modèle hydraulique pour les foules à forte densité, qui compare le mouvement des personnes dans un espace à celui d'un liquide dans un tuyau, a été proposé.

Des modèles statistiques supplémentaires basés sur un large éventail d'observations ont été proposés. En traitant séparément les zones de circulation horizontale et verticale, ils donnent des équations sur le temps d'évacuation d'un immeuble. L'étude bibliographique de début de thèse de Sébastien Paris [48] sur la caractérisation des niveaux de services et la modélisation des circulations de personnes dans les lieux d'échange offre une description détaillée de ces modèles pour les lecteurs curieux. Les systèmes physiques à base de particules peuvent décrire les forces d'attraction et de répulsion associées à des obstacles ou à des entités simulées. Les forces d'une entité sont combinées pour déterminer sa nouvelle direction. Ce modèle fait l'analogie entre l'écoulement des particules dans des compartiments et le déplacement d'individus en forte densité, n'ayant quasiment pas de degrés de liberté dans leurs déplacements. Selon I. Peschl [49], cette modélisation est utile dans les situations où une population importante est confinée et en état de panique. Lorsque le flot se dirige vers une seule issue, un phénomène d'agglutination se produit et le débit de l'issue devient presque nul. En ajoutant la dimension sociale de la connaissance d'autrui au sein du groupe, Braun et al [112] proposent une extension du modèle d'Helbing : Les entités peuvent s'entraider pour sortir d'une situation critique lorsqu'elles se connaissent. Ils modélisent ainsi les forces d'attraction entre agents, pondérées par des facteurs d'altruisme et de dépendance. Dans le cas d'une densité de population significative, les modèles à base de particules montrent un comportement macroscopique réaliste. Cependant, ces modèles ignorent les règles sociales et la perception visuelle.

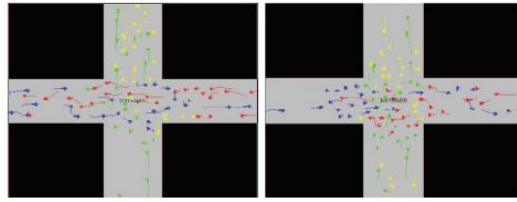


Figure 2.4 : Système à base de particules [33]. S. Goldenstein et al [50].

Ils proposent une méthode dynamique en trois couches pour simuler le comportement d'un grand nombre de personnages qui évoluent dans des environnements virtuels. La première couche, la plus locale, est un système dynamique non linéaire décrivant le mouvement de l'agent modélisé comme une particule par deux paramètres : son angle d'orientation et son module de vitesse. En conséquence, un ensemble de répulseurs et d'attracteurs est défini par les obstacles et les cibles. Une distance-seuil, qui indique à quelle distance un obstacle est pris en compte, est utilisée pour prendre en compte des facteurs comportementaux tels que l'agressivité. La deuxième couche, appelée couche environnementale, est constituée d'une structure de données cinématiques qui modifie dynamiquement les relations avec les objets en mouvement ainsi que les obstacles grâce à une triangulation de Delaunay. La troisième couche permet d'avoir une vision globale nécessaire pour planifier des tâches et de déterminer l'orientation désirée pour s'approcher d'une cible.

Le modèle de nuée introduit par Reynolds [38] est un modèle basé sur des règles de comportement qui permettent de définir comment un oiseau se comporte en fonction de ses entités les plus proches. En conséquence, il propose d'utiliser seulement trois règles pour définir le comportement d'un oiseau au sein de la nuée, tout en attribuant un rôle spécifique au meneur chargé de suivre une trajectoire :

- La séparation afin d'éviter d'éventuelles collisions avec les voisins ;
- L'alignement afin de réguler sa vitesse sur celle de ces voisins ;
- La cohésion afin de rester proche de ses voisins

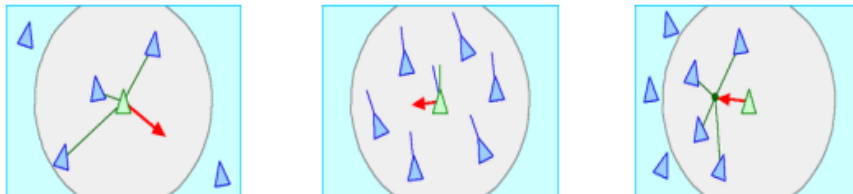
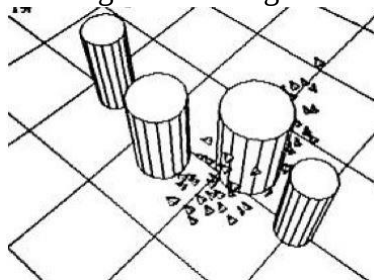


Figure 2.5 : Règles comportementales du modèle Flocks of Boids.



(a) Les boids de Reynolds sont très convaincants en groupe.



(b) Dans « Batman Returns », des centaines de boids envahissent les rues de Gotham City

Figure 2.6 : Les boids de Reynolds et leur utilisation au cinéma

En combinant des approches à base de règles et des automates d'états finis, Ulicny et al [51] utilisent une approche en plusieurs niveaux pour simuler le comportement d'une personne au sein d'une foule. Toutes ces méthodes sont confrontées au problème de déterminer les voisins

les plus proches, ce qui constitue un obstacle à la détermination du nombre de piétons simulables en temps réel. En utilisant des structures de données comme des treillis (bin-lattice) [53] ou des arbres K-d, plusieurs méthodes ont été proposées pour optimiser ce calcul [54]. D'autres méthodes basées sur des connaissances sur l'environnement informé ont été proposées, permettant ainsi de modéliser des comportements qui utilisent la nature de l'espace de circulation et sa localité. Des études se sont concentrées sur les niveaux d'autonomie et de contrôle pour la simulation de foules afin de proposer des comportements de foule réalistes dans des environnements virtuels à un coût calculatoire inférieur. Par conséquent, Soraia Raupp Musse [Mus00c] a présenté dans sa thèse trois niveaux de contrôle des foules : la programmation, l'autonomie et le guidage.

Dans le cadre du projet MAGS, les chercheurs se concentrent sur la simulation de plusieurs milliers de piétons dans une représentation discrétisée de la ville de Québec, comme illustré à la figure 2.6. Limiter la vision à un cône de 90 degrés de secteur angulaire et 20 cellules de rayon (pour une carte 300x300) résout le problème du goulet d'étranglement du calcul de la perception. Des recherches supplémentaires ont été menées sur l'optimisation de la visualisation de foules en temps réel en utilisant des imposteurs hiérarchiques et des modèles d'ombrage d'imposteurs en temps réel. Ashida et al [55] ont effectué des analyses statistiques sur les déplacements des piétons le long d'un trottoir pour rendre les animations plus réalistes. Afin de contrôler leur activation, ils exhibent des actions subconscientes qu'ils intègrent dans leur moteur d'animation à l'aide d'un processus stochastique. Tsutsuguchi et al. [56] proposent un modèle de gestion de l'animation de foule qui dépend de la distance à l'observateur : si la distance à l'observateur est élevée, la position des piétons ne sera pas mise à jour sur chaque image, mais plutôt à une fréquence plus faible.



Figure 2.7 : Projet de MAGS [57].

4. Conclusion

Si le piéton était l'élément central de la foule, elle ne serait pas significative. Le piéton, l'élément principal de la foule, a été présenté dans ce chapitre. Pour donner plus de réalisme à l'animation de foule basée sur l'autonomie de chaque piéton, nous avons présenté ses propriétés.

Chapitre 3 : La modélisation de foules

1. Introduction

La littérature sur la simulation de foule, les études connexes sur les facteurs sociologiques, les différentes modélisations de foule et les aspects d'interaction sont présentés dans ce chapitre. On propose également une classification des techniques de simulation de foule pour présenter les travaux dans le domaine de la simulation de foule. Enfin, il est mentionné que de nombreuses applications ont été créées pour reproduire les comportements des foules. Il serait impossible de faire une liste complète de toutes les choses. Nous présentons également des travaux qui utilisent divers modèles de comportement.

2. Les aspects sociologiques et d'interaction de foule

Beaucoup d'aspects des conventions sociales, la proxémie d'individu et de groupe (étude de l'espace personnel), l'importance des relations entre groupes et les effets des interactions sur un groupe de personnes ont été étudiés par les sociologues et les psychologues. L'espace personnel d'une personne est son environnement direct. Selon les intentions et les perceptions individuelles de l'espace, cet espace peut être utilisé pour permettre l'interaction avec les autres ou pour être isolé dans les autres (concept d'intimité) [65]. L'étude de l'utilisation sociale de l'espace et des foyers personnels pour atteindre une communication confortable et réaliser les niveaux préférés de participation interpersonnelle est appelée proxémie. [59] [52].

Les normes sociales régissent le comportement dans le monde physique. Les expériences passées, l'observation des autres et l'identification des repères qui indiquent si un comportement est socialement acceptable influencent le comportement des gens [61]. Les conventions sociales sont les normes qui dictent son comportement. Les conventions sociales peuvent être influencées par l'espace personnel, l'intimité, l'espace de groupe et les facteurs environnementaux. Les normes sociales dictent comment les gens se comportent dans la société. Parfois, ils ne sont pas dits et peuvent être acquis par l'expérience. Elles peuvent avoir un impact significatif sur le comportement de l'utilisateur et ses attentes quant au comportement des autres. Les normes du groupe peuvent "se former à partir d'instructions données par les superviseurs, d'événements importants dans l'histoire du groupe, ou comportement à partir des situations passées » [63].

Une foule acceptable est un groupe de personnes dans le même environnement physique, avec un objectif commun ou les deux. Les foules peuvent être en activité ou inactives. Dans le dernier cas, les groupes ou les individus partagent des objectifs et des tendances spécifiques. Ensuite, les gens dans une foule peuvent agir différemment de ceux dans un petit groupe ou seuls, ce qui signifie que les gens peuvent abandonner leurs comportements et intentions individuels et adopter des comportements de foule ou de groupe [62]. Les rapports sociaux entre les individus qui forment temporairement une masse ou une foule peuvent entraîner le développement de certains comportements sociaux.

Quelques chercheurs ont examiné les normes nécessaires pour faciliter la communication dans les mondes virtuels [65]. Par exemple, Becker et Mark (1998) ont défini quelques conventions d'interaction entre groupes pour déterminer l'adhésion et l'intimité. Cependant, le thème de produire l'entité, dont le genre est un composant, peut être développé par les conventions sociales [64]. Rodden (1996), Greenhalgh et Benford (1997) ont examiné les effets des

environnements virtuels de collaboration à grande échelle utilisés par les utilisateurs d'Internet [60].

3. Les deux approches de foule

Il existe de nombreuses méthodes appropriées pour la simulation de foule. Nous présentons ici divers systèmes destinés à la dynamique d'animation ou d'évacuation. Elles partagent la condition pour les modèles de mouvement piétonniers, mais elles diffèrent dans la façon de créer des chemins de mouvement. Nous examinerons d'abord les techniques microscopiques, puis les techniques macroscopiques.

3.1 Les modèles microscopiques

Les modèles microscopiques examinent le comportement, les décisions et l'interaction des différents piétons dans la foule. Le comportement d'espace-temps de différents piétons est illustré par des modèles microscopiques. Les modèles de force sociale et les modèles des automates cellulaires (AC) font partie des deux catégories. La discrétisation de l'espace et du temps est ce qui les distingue. Les domaines sociaux, également appelés forces sociales virtuelles, qui sont influencés par le comportement social des individus, sont expliqués par les modèles de force sociale [33]. L'approche AC représente l'espace à l'étude par une grille uniforme de cellules avec des états locaux conformément à un ensemble de règles qui décrivent le comportement des piétons [66]. Ces règles calculent l'état d'une cellule en fonction de son état précédent et de celui des cellules adjacentes. Les modèles microscopiques sont plus fascinants pour animer des foules virtuelles d'agents avec des comportements autonomes réalistes. Par conséquent, nous expliquerons plus en détail ces méthodes.

3.1.1 Modèles de force sociale

Une méthode microscopique pour simuler le mouvement de piétonnier est le modèle de force sociale. De vraies forces telles que l'interaction répulsive, les forces de frottement, la dissipation et les fluctuations sont similaires aux forces sociales « virtuelles ». Il résout toutes les équations du mouvement de Newton. Il est possible d'utiliser ce modèle pour simuler des scénarios de mouvement dans le monde réel. Les modèles de force sociale, à l'inverse d'autres modèles, décrivent le comportement piétonnier plus précisément. Cependant, ils sont conçus pour rester aussi simple que possible. Un cercle avec son propre diamètre représente chaque agent dans le plan de locomotion, et le modèle décrit les coordonnées, les vitesses et les interactions continues avec d'autres objets. Chaque paramètre de force ayant une interprétation normale est unique pour chaque piéton et est souvent choisi aléatoirement ou découvert à l'aide d'une expérience. Avec une combinaison de facteurs socio-psychologiques et physiques, les forces sociales modélisent le comportement de la foule humaine. Le modèle de force sociale dérivé le plus important de forces sociales est le modèle de Helbing comme illustré dans la figure 3.1 [67].

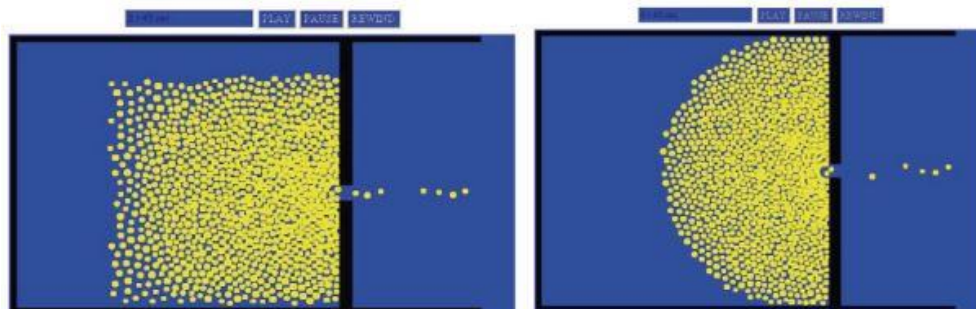


Figure 3.1 : La simulation de modèle de Helbing avec 10.000 individus [33].

3.1.2 Les modèles d'automates cellulaires

AC est une approche pure de l'intelligence artificielle qui simule les modèles comme des spécifications mathématiques de systèmes physiques dans lesquels l'espace et le temps sont discrets et quantifiés physiquement et un ensemble fini de valeurs discrètes est pris en compte. Un automate cellulaire se compose d'une grille uniforme régulière (2D rangée) avec un ou plusieurs variables discrètes à chaque emplacement (cellules). Les cellules blanches sont des zones où les piétons peuvent circuler, tandis que les murs et autres obstacles fixes sont noirs [68] [69].

Les valeurs des variables, comme chaque cellule, déterminent complètement l'état d'un automate cellulaire. Un automate cellulaire évolue en étapes discrètes, avec la variable d'une seule cellule qui affecte les cellules voisines. Les variables de chaque cellule sont mises à jour en même temps selon un ensemble de règles locales et en fonction des valeurs des variables voisines de l'étape de temps précédente [70]. Ces règles définissent le comportement de prise de décision (intelligent) des automates, permettant ainsi la création et l'émulation de comportements réels. Chaque automate évalue ses chances cas par cas. Les interactions des règles locales, telles que chaque piéton examine les cellules disponibles dans son voisinage, contribuent à l'émergence du comportement global de groupe. En général, les AC offrent un cadre pour les modèles discrets d'interactions homogènes localement. Le tableau ci-dessous montre les propriétés fondamentales (L, S, N, f) qui les caractérisent. Puisque l'ensemble d'états, S, contient des informations sûres, comme la capacité d'une cellule à être accessible ou non (par exemple, des portes ou des murs entre les cellules), l'acceptation d'un treillis régulier et d'un voisinage uniforme est compatible avec les géométries présentées dans le tableau [67].

Tableau 3.1 : les automates cellulaires	
L	Se compose d'une grille discrète régulière des cellules
$t \rightarrow t+1$	L'évolution prenne place dans des étapes de temps discret
S	Ensemble d'états finis
$F : S^2 \rightarrow S$	Chaque cellule évolue selon la même règle (fonction de transition), qui dépend seulement de l'état de la cellule et d'un nombre fini de cellules voisines.
$N : \forall c \in N, \forall r \in L : r+c \in L$	La relation de voisinage est locale et uniforme

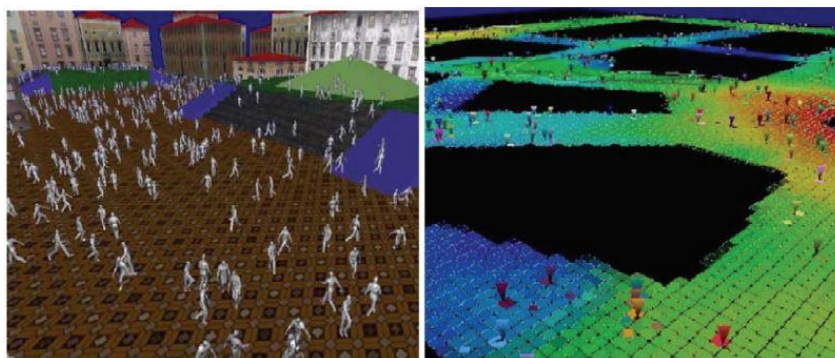


Figure 3.2 : Simulation de foule par AC 2D structure étant à la base de grille [71].

Parmi les tâches de AC Stephen Chenney est un chercheur à l'université du Wisconsin à Madison aux États-Unis. En 2004, il a publié un article au Siggraph sur l'un de ses projets appelé « Flow Tiles » [66], qui signifie littéralement « carreaux de flux ». Dans cet article, nous présentons une nouvelle méthode de création de champs de vitesses. Un champ de vitesse est une grille avec une vectrice de vitesse attribuée à chaque case. Par conséquent, lorsqu'un

élément de la foule passe au-dessus d'une case, le vecteur spécifique à celle-ci lui est attribué, ce qui détermine sa direction et sa vitesse. Ainsi, le comportement des agents de la foule est lié à leur position sur un terrain. Pour créer du vent qui dirige plusieurs objets dans la même direction, il est possible de combiner plusieurs flux de base pour créer une superposition. Cependant, cette méthode ne permet pas de gérer efficacement le comportement des agents lorsqu'ils s'approchent des bords du terrain ou rencontrent un obstacle. En effet, ce sont des forces globales et linéaires qui s'appliquent à l'ensemble de la population. L'avantage du système de Stephen Chenney est qu'il peut contrôler la vitesse d'une seule personne et non de l'ensemble. De plus, cela permet l'application de forces non linéaires. En effet, chaque carreau est constitué d'un champ de vitesse, qui est composé de plusieurs vectrices de vitesse applicables aux agents de la foule (voir la figure 3.3). Cela permet d'avoir des mouvements fluides à l'intérieur d'un carreau et de détourner les objets lorsqu'un obstacle se présente.

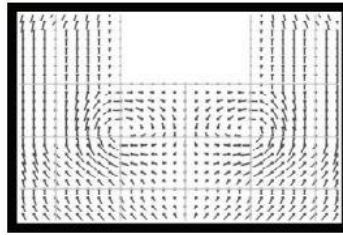


Figure 3.3 : Carreaux de flux et Champs de Vitesse.

Ainsi, Stephen Chenney a créé une application qui permet à l'utilisateur de créer facilement de vastes espaces d'influence en utilisant des carreaux de flux.

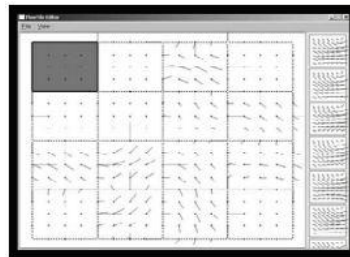


Figure 3.4 : interface du logiciel permettant de choisir des carreaux prédéfinis [66].

Le programme affiche les carreaux susceptibles d'être placés à ses côtés lorsque l'utilisateur choisit un carreau de flux. En effet, pour éviter les sautes d'animation lorsqu'un agent passe d'un carreau à l'autre, les vitesses sur les bords des carreaux doivent être alignées.

Ce programme peut également être utilisé pour simuler le comportement d'objets flottants sur une rivière ou pour gérer une foule dans les rues d'une ville. Parce que les calculs sont simples, cela est très utile pour générer des foules massives d'individus.

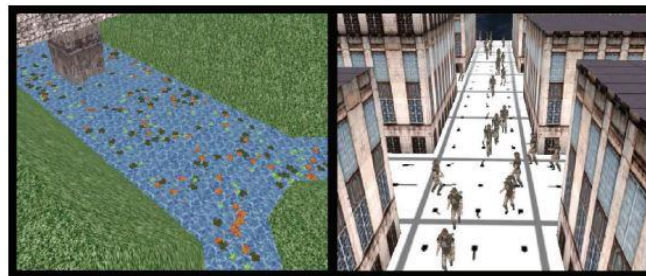


Figure 3.5 : Deux applications du système Flow Tiles [66].

De plus, des champs de vitesses dynamiques, c'est-à-dire avec des vectrices de vitesse qui changent au fil du temps, peuvent être utilisés (voir figure 3.6)

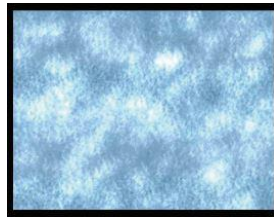


Figure 3.6 : Capture d'écran d'un brouillard tourbillonnant généré à l'aide de champs de vitesses dynamiques.

Ainsi, les groupes de personnes présentés ici sont créés sans objectif explicite. Ils ne sont pas acteurs du système, ils se contentent de suivre un chemin. En d'autres termes, le comportement d'une personne ne modifie pas le comportement général du groupe. Modèles AC Bien que rapide et facile à mettre en œuvre, [71][69][66][72] ignore les interactions entre les agents. Les personnes ne peuvent se déplacer qu'à une cellule libre adjacente car la surface du sol est discrète. Cette méthode de damier donne des résultats réalistes pour les foules à faible densité, mais elle donne des résultats peu réalistes lorsque des agents à haute densité sont nécessaires dans les cellules discrètes. En précalculant les chemins vers des buts et en les stockant dans la grille, il est possible d'obtenir des chemins non locaux plus longs et plus réalistes dans la grille. [73].

3.1.3 Les Modèles basés sur les règles

3.1.3.1 Les « Boids » de Craig Reynolds

Craig Reynolds a inventé un algorithme pour créer des agents autonomes appelés « Boids » en 1986 [74]. Ces derniers étaient capables d'improviser leurs actions et leurs mouvements. Il a cherché à reproduire des mouvements d'animaux tels qu'une nuée d'oiseaux ou un banc de poissons de manière complètement automatique. Craig Reynolds est rapidement devenu un expert en animation comportementale grâce à la simplicité de ce système.

Il est possible de simuler un comportement appelé émergent à l'aide du système des Boids. Il est effectivement impossible de prédire dans quelle direction une nuée d'oiseaux se dirigera à une instante t . L'interaction des Boids entre eux crée ici une complexité comportementale. Comme indiqué dans la figure 3.7, ces derniers doivent respecter un certain nombre de règles simples :

- La cohésion : pour former un groupe, chaque Boid doit se rapprocher des autres.
- La séparation : chaque Boid doit garder une distance minimale avec ses congénères. Cela permet d'éviter les collisions entre deux Boids.
- L'alignement : Les Boids doivent suivre le même chemin pour rester groupés.

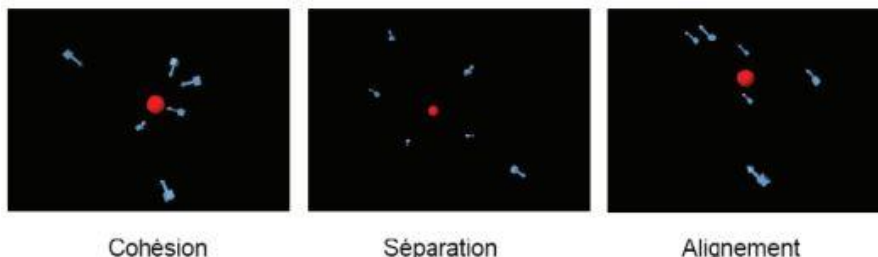


Figure 3.7 : Les 3 règles du système de Reynolds (le point rouge représente le centre du système de Boids).

En combinant ces trois règles, il est possible de déterminer une force spécifique à chaque Boid qui lui permet de progresser dans une direction spécifique. L'application de paramètres globaux (qui affectent tout le système) et locaux (tels que la vitesse maximale) permet d'augmenter les options d'animation. Ainsi, les trajectoires de plusieurs éléments peuvent être calculées à l'aide de ce système sans utiliser la méthode du « keyframing ». Il était nécessaire d'établir toutes les positions de chaque membre de la nuée à un moment donné. Ce travail, qui était à la fois difficile

et long, n'a pas permis d'obtenir un résultat similaire à celui des Boids en termes de fluidité d'animation. Chaque Boid est attribué deux fonctions par le système de Craig Reynolds : être acteur du système et être soumis aux lois qui le régissent. Cette méthode reste la plus fréquemment utilisée pour modéliser des groupes de personnes en temps réel ou pré-calculé, malgré son âge d'une vingtaine d'années.

3.1.3.2 Les systèmes de particules

Pour étudier les mouvements de foule où les êtres humains sont modélisés comme un ensemble interactif de particules, Bouvier (1997) a utilisé des systèmes de particules adaptés. Les forces newtoniennes, ainsi que les buts et les décisions humains, guident le mouvement des personnes. Ils ont introduit les concepts de "charge de décision" et de "champs de décision" en utilisant les concepts de charges électriques dans le sens qu'un champ électrique affectera une particule avec une charge électrique comme une personne avec une charge de décision [75].

3.1.3.3 Mouvements contrôlés par les lois de la dynamique

Trois chercheurs américains, David Brogan, Ronald Metoyer et Jessica Hodgins, travaillent à l'Institut de Technologie de Géorgie aux États-Unis. En 1998, ils ont créé deux établissements dans le cadre de leurs études sur les interactions humaines dans des environnements virtuels [76]. Ils proposent une méthode pour augmenter la taille des environnements virtuels en utilisant une simulation dynamique pour créer divers mouvements des personnages qui répondent aux actions de l'utilisateur en temps réel.

Ainsi, la machine est entièrement responsable de l'animation, mais elle reste liée au comportement de l'utilisateur. Ces trois chercheurs ont créé une première installation qui consiste en un jeu dans lequel un troupeau d'animaux unijambistes robotiques se déplace dans un environnement virtuel. En portant des lunettes avec l'application en temps réel, l'utilisateur navigue avec ce troupeau en conduisant une sorte de pédalo statique. Le but du jeu est de rassembler le troupeau de robots dans un enclos, comme un chien de berger.



Figure 3.8 Installation

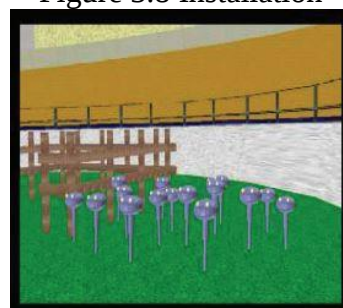


Figure 3.9 : Robots unijambistes à rapatrier dans l'enclos.

Chaque robot est une entité virtuelle capable de percevoir la position de l'utilisateur et ses congénères. Il utilise cette information pour éviter les conflits avec tous ces individus.

La deuxième application consiste en une réplique virtuelle du circuit de cyclisme des Jeux Olympiques d'Atlanta (1996). Le but était de créer un environnement virtuel dans lequel un

athlète pourrait s'immerger visuellement et physiquement en présence d'autres cyclistes virtuels. Un tel système peut être utile pour les professionnels qui ont du temps ou des conditions météorologiques difficiles. Ils pourraient s'entraîner avec ce type de programme.



Figure 3.10 cyclistes virtuels.

Ainsi, pour simuler les côtes et les pentes du circuit, l'utilisateur monte sur un vélo de course (représenté à la figure 3.11) qui tangue entre -12 et 12 degrés. Le vélo est équipé de capteurs qui peuvent mesurer à la fois la vitesse de la roue arrière et la direction de la roue avant. Ces capteurs peuvent être mesurés entre -20 et 20 degrés en utilisant la rotation du guidon.

Au moment où le cycliste se trouve dans une pente, la roue arrière est connectée à un générateur qui permet de la faire tourner. Cela augmente l'effet immersif de l'installation, qui est déjà obtenu par le casque que porte l'utilisateur qui diffuse les images de l'environnement virtuel.



Figure 3.11 : Installation.

Le principal avantage de cet environnement virtuel est la présence de cyclistes virtuels dynamiques. Rien n'est préétabli. Comme le montre la figure 3.12, leurs déplacements sont contrôlés par les forces appliquées sur leurs vélos, notamment l'interaction « cycliste-vélo » et l'interaction « sol-vélo ».



La force appliquée sur le guidon dirige le vélo

Les roues tournent sur le sol sans glisser

Les forces appliquées par le cycliste sur les pédales sont transmises à la roue arrière et entraînent le vélo

Figure 3.12 : Forces dynamiques d'un cycliste virtuel.

En ce qui concerne l'animation comportementale, les cyclistes virtuels peuvent utiliser des stratégies de course assez sophistiquées, telles que les blocages, les échappées solitaires, etc... Chaque personne a son propre comportement, ce qui rend cette modélisation de foule très complexe. Il peut agir en groupe ou seul.

3.1.3.4 Modélisation de foules avec facteur psychologique

Quatre chercheurs américains, Noria Pelechano, Kevin Brien, Barry Silverman et Norman Badler, travaillent à l'Université de Pennsylvanie à Philadelphie. Comme indiqué dans la figure 3.13, ils ont créé une application qui peut simuler l'évacuation d'un bâtiment dans une situation de panique. Ils ont intégré un modèle psychologique dans leur groupe virtuel [77].

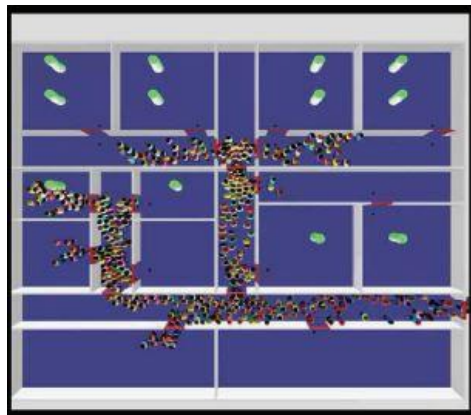


Figure 3.13 : Simulation de l'évacuation d'un immeuble.

Cette simulation est en fait la réunion de deux applications :

- **MACEMACES : Multi Agent Communication for Evacuation Simulation.** Ces algorithmes de détermination de chemins permettent aux agents de la foule d'explorer un environnement inconnu. De plus, cela leur permet de communiquer entre eux et de partager des informations sur l'environnement.
- **PMFserv: Performance Moderator Functions server.** L'aspect psychologique du projet, qui consiste à gérer les différentes émotions et donc leur comportement, a été développé par cette librairie. Cela augmente la réalisme du projet en contrôlant la capacité, le stress, les émotions, les décisions et la motivation des agents à différents degrés.

Les comportements ne sont jamais programmés ou prédéfinis. L'attribution de paramètres aux piétons est la seule action de l'utilisateur. Les piétons réagiront individuellement à cela et conduiront au caractère émergent du système. Le programme divise les piétons en différentes parties. Seuls les agents qui travaillent ensemble peuvent interagir et échanger des informations. Les agents occupent divers rôles basés sur leurs qualités de « chef » et « expérimenté » :

- Un chef compétent connaît la structure interne du bâtiment et aidera les autres pendant l'évacuation.
- Un chef sans expérience correspond aux personnes qui réussissent à gérer leur stress et à aider les autres tout en explorant l'édifice pour trouver de nouveaux chemins.
- Un piéton qui n'est ni chef ni expérimenté dépend des autres et panique dans de telles circonstances au point d'être incapable de prendre des décisions.

Ainsi, le système qui en découle est un système multi-agents, qui est un ensemble d'entités qui s'auto organisent et dont la force réside dans la simplicité de l'individu, le nombre d'agents et la communication. Il devient évident que tous les individus finissent par se diriger vers la bonne sortie en évitant les obstacles à leur passage, tels que des poteaux ou même des agents tombés au sol (voir la figure 3.14). Il s'agit également d'un système en développement où les

comportements individuels finissent par conduire à des comportements de groupe non programmés.

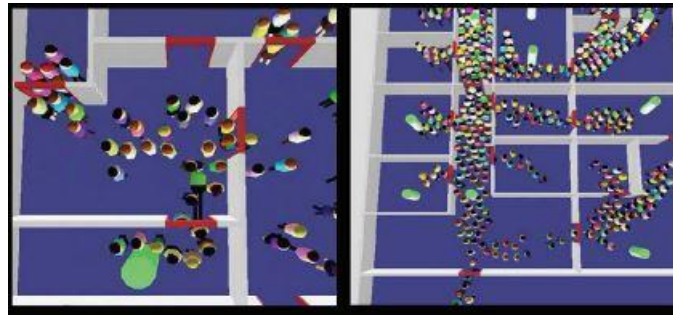


Figure 3.14 : Comportements des agents.

3.1.3.5 Méthode utilisé des Règles locales

Un système qui applique ses règles à chaque membre de la foule est un modèle comportemental utilisant des règles locales. L'individu a une perception des autres et son comportement est calculé en fonction de cette perception.

Le Laboratoire de Réalité Virtuelle (VRlab) de l'EPFL (Ecole Polytechnique Fédérale de Lausanne) est dirigé et professé par Daniel Thalmann. C'est un innovateur dans le domaine de la recherche sur les humains virtuels. Les Humains Virtuels en temps réel, les Environnements Virtuels Immersifs, la simulation de foules et l'interaction multimodale sont ses domaines de recherche actuels. Daniel Thalmann et Soraia Raupp Musse ont créé "ViCrowd", un modèle comportemental d'êtres humains en temps réel [80]. (Virtual Crowd, la foule virtuelle). Soraia Raupp Musse a travaillé sur ce projet lors de sa thèse. Ce travail est basé sur plusieurs concepts de sociologie et permet par exemple de simuler l'évacuation d'un lieu en raison d'une situation de panique, comme illustré dans les figures 3.15 et 3.16, et de pouvoir gérer par la suite de tels événements dans la vie réelle en utilisant l'analyse de la simulation virtuelle.

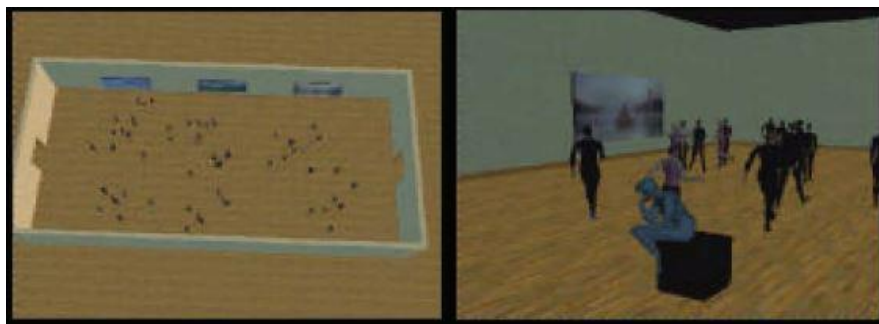


Figure 3.15 : Gestion d'une foule dans un musée.



Figure 3.16 : Gestion de la foule de la Figure 1 après introduction d'un élément de panique. La hiérarchie de "ViCrowd" est composée de groupes, d'individus et de foules virtuelles. Comparé à d'autres projets similaires, il met l'accent sur l'animation de groupe plutôt que sur la

singularité des individus, qui dans d'autres projets similaires sont autonomes et « intelligents ». Cela permet une réduction significative des temps de calcul. Comme le montrent les figures 3.17 et 3.18, ViCrowd utilise des règles de comportement simples pour gérer les événements et les réactions des êtres virtuels, bien qu'il soit basé sur le principe de la « gestion de foules ». Un modèle sociologique a été utilisé pour illustrer les affinités et les répulsions entre les individus dans ce projet qui visait à représenter les mouvements et les réactions des humains.

Ces derniers ont également une variété d'émotions attribuées par l'utilisateur, telles que « triste », « calme », « état normal », « heureux », « très heureux », etc. Les mouvements de chaque personne sont influencés par ses émotions (posture, vitesse de marche, etc.). (Voir les images 3.17 et 3.18).



Figure 3.17 : Différents états émotionnels des individus.



Figure 3.18 : Différentes façon de marcher selon les émotions des agents.

Les émotions et le comportement de la foule sont liés au type d'endroit où les agents se trouvent. Par exemple, ils marcheront plus vite dans une gare que dans un parc à cause du stress. De plus, certains comportements sont préétablis pour les individus. Par exemple, ils peuvent toujours acheter un billet de train, passer un coup de fil, attendre quelqu'un, etc.



Figure 3.19 : Simulation dans une gare.

3.2 Les modèles macroscopiques

Historiquement, les modèles de simulation macroscopique ont été les premiers modèles informatiques à permettre la simulation de foules. En effet, ils simulent la totalité des individus par un ensemble de fonctions mathématiques (une vue globale des comportements de foule comme des flux), ne nécessitant qu'une puissance de calcul assez faible. Par conséquent, ces modèles préfèrent simuler rapidement un grand nombre d'entités plutôt que tout comportement individuel évolué. Nous verrons que l'objectif de ces modèles est de décrire un phénomène spécifique qui est généralement reproductible dans des conditions environnementales modifiées.

3.2.1 Les modèles de régression

Les modèles de régression utilisent les relations statistiquement établies entre les variables de flux pour prévoir les opérations d'écoulement des flux piétonniers dans des conditions particulières. Les caractéristiques de ce flux dépendent de l'infrastructure (escaliers, couloirs, etc.) [78][67].

3.2.2 Les modèles de choix de route

Basés sur l'idée d'utilité, les modèles de choix de route décrivent comment trouver un piéton. Pour maximiser l'utilité de leur voyage (confort, temps de déplacement, etc.), les piétons choisissent leurs destinations [79] [67].

3.2.3 Les modèles de queue (file d'attente)

Les modèles de chaîne de Markov [83] sont utilisés pour décrire comment les piétons se déplacent d'un nœud à l'autre dans un réseau. En raison du fait que les nœuds sont généralement des chambres, les liens sont généralement des portails ou des portes. Un ensemble d'états avec des probabilités de transition est utilisé pour définir des modèles de chaîne de Markov. À chaque étape d'extrapolation, un état successeur est choisi à partir de la distribution de transition ou de l'identifiant de successeur le plus probable. Les probabilités de transition d'état sont calculées en prenant le prototype de comportement le plus proche chaque instant dans les données d'apprentissage et en calculant la fréquence relative des transitions entre les prototypes de comportement observés. Seulement en prenant en compte les transitions qui déclenchent le changement d'état [67].

3.2.4 Gaskinetics

Dès 1971, LF Henderson suggère que les mouvements des personnes dans des conditions de faible densité doivent être comparés à ceux des molécules de gaz. En assimilant les différences d'états des piétons (arrêt, marche, course) aux différences d'énergie des gaz, il fait le parallèle entre la densité des piétons et celle des particules d'un gaz en utilisant la théorie cinétique des gaz de Maxwell-Boltzmann. Ce modèle repose sur deux hypothèses concernant les piétons, en négligeant toutes les règles sociales (telles que les liens interpersonnels) :

- Leurs positions et vitesses sont indépendantes les unes des autres ;
- Leur vitesse est indépendante de leur position.

Toute simulation d'humanoïdes virtuels repose sur ces premiers constats, même s'ils peuvent paraître évidents aujourd'hui.

Pour expliquer comment la densité et la vitesse de foule changent avec le temps, Gaskinetics utilise une analogie avec la dynamique de fluide ou de gaz. Ils utilisent des équations différentielles partielles [81].

3.2.5 Modèle hydraulique

- J. Archea [82] a suggéré un modèle hydraulique pour reproduire des foules denses. Il compare le mouvement des personnes dans les couloirs, les escaliers et les portes à celui de l'eau dans les tuyaux, les vannes, etc. Le processus de simulation de l'évacuation d'un bâtiment se déroule en deux étapes. :
- Initialiser le mouvement de tous les individus recevant simultanément un message d'alerte ;
- Simulation de l'écoulement (écoulement de l'eau causé par la gravité). Les hypothèses suivantes sont la base de ce modèle :
 - Prise de conscience et décision de l'évacuation immédiate et simultanée de tous les individus.
 - Répartition uniforme des occupants.
 - Débits uniformes.

- Flux unidirectionnels.
- Trajets prédéfinis.

Même si ce modèle a été largement utilisé, en raison de sa simplicité de mise en œuvre, il souffre de la simplicité des hypothèses sur lesquelles il repose : les personnes ne partent pas toutes en même temps et leurs directions ne sont pas aussi faciles à prévoir.

3.2.6 Modèle de Togawa

K. Togawa [84] propose un modèle qui est basé sur les observations effectuées dans des établissements recevant du public au cours d'une décennie. En conséquence, il fournit une formule pour déterminer la vitesse de marche d'une foule par rapport à sa densité :

$$V = V_0 \cdot \rho^{-0.8} \quad (01)$$

- Avec $V_0 = 1.3 \text{ m.s}^{-1}$, $\rho = \text{densité en personnes} / \text{m}^2$

Le débit est alors déduit par :

$$N = \rho \cdot V = V_0 \cdot \rho^{0.2} \quad (02)$$

- Avec $N = \text{débit en personnes/s/m}$, $1.3 < N < 1.7$

K. Togawa suggère quelques modifications suivant le degré d'inclinaison du plan de circulation :

- Si le sens est ascendant, V est réduite de 2% par degré d'angle d'inclinaison ;
- Si le sens est descendant, la vitesse augmente pour des pentes faibles, mais diminue pour des pentes fortes.

Ce modèle a été appliqué dans le cadre de la simulation de l'évacuation d'un immeuble en cas d'incendie. Le produit de sortie de cette simulation est un temps total d'évacuation, défini comme la somme des temps :

- De détection du feu ;
- D'alarme ;
- De préparation ;
- Et d'évacuation net, qui est lui-même la somme des temps d'évacuation des escaliers et des circulations horizontales.

Le temps d'évacuation des circulations horizontales est calculé ainsi (formule 03) :

$$T = \frac{P}{C_c \cdot \sum L_e} + \frac{L_h}{v} \quad (03) \text{ avec}$$

- T = temps d'évacuation en secondes
- P = nombre de personnes dans le bâtiment
- L_e = largeur des zones de circulation et des issues en mètres
- C_c = coefficient de circulation (évalué à 1.3 pers. / m / s)
- L_h = longueur totale maximale des chemins d'évacuation
- v = vitesse de circulation (généralement 0.6 m / s) Le temps de parcours d'un escalier est quant à lui calculé par, quel que soit le sens de parcours :

$$t = 4H, \text{ avec}$$

- t = temps de parcours en secondes, et H = hauteur d'un étage en mètres

K. Togawa [84] remarque que, pour l'évacuation des établissements à plusieurs étages, la largeur seule des escaliers n'est pas suffisante pour prédire leur débit. Il pense que les paramètres les plus significatifs sont la hauteur des marches et l'allure des personnes. Il constate que le temps de parcours d'un escalier est inversement proportionnel à son angle et sa hauteur, et que la vitesse moyenne en descente est de 0.5 m/s. Pour conclure, le modèle de K. Togawa propose un ensemble de règles permettant une évaluation approximative du débit maximal admissible en sortie d'un bâtiment. Il semble difficile d'intégrer telles quelles ces règles dans

un modèle plus complexe, mais elles peuvent tout de même servir comme valeur de comparaison lors d'une première étape de validation.

3.2.7 Modèle de Melinek et Booth

S. J. Melinek et S. Booth [84] ont amélioré le modèle de K. Togawa en prenant en compte les éventuels phénomènes de saturation. Ils découpent un bâtiment en cellules (une par étage) et calculent le temps d'évacuation pour chacune d'elles. Les personnes sont supposées être à proximité des escaliers, et leur flot continu et ne rencontrant pas d'obstacle.

Il est aussi admis que les personnes quittant le rez-de-chaussée ne réduisent pas le débit des niveaux supérieurs. Il est défini :

- Q_r = occupation du niveau r en personnes
- br = largeur de l'escalier entre les niveaux r et $(r-1)$ en mètres
- $N0$ = débit unitaire par mètre de largeur des escaliers descendants (généralement 1.1 pers/m/s)
- t_s = temps mis par une personne pour descendre librement un étage (généralement 16 s)
- La population P_r des niveaux r et supérieur se calcule ainsi (formule 04) :

$$P_r = \sum_{i=r}^n Q_i \quad (04)$$

- Le temps minimum pour descendre l'escalier situé entre les niveaux r et $(r-1)$ est :

$$\frac{P_r}{N' \cdot b_r} \quad (05)$$

- Le temps pour que la queue de la foule atteigne le rez-de-chaussée est : $r \cdot t_s$
- Le temps minimum d'évacuation T_r (formule 06) des niveaux r et supérieurs est donc :

$$T_r = \frac{P_r}{N' \cdot b_r} + r \cdot t_s, \quad r \in \{1, \dots, n\} \quad (06)$$

Soit

$$T_r = \sum_{i=r}^n \frac{Q_i}{N' \cdot b_r} + r \cdot t_s, \quad r \in \{1, \dots, n\} \quad (07)$$

Dans les cas où plusieurs escaliers sont disponibles à un étage r , b_r est considéré comme étant la somme de leurs largeurs. En supposant que le nombre d'occupants et la largeur d'escaliers soient identiques à chaque niveau, on pose $Q = Q_r$ et $b = b_r$. On obtient donc le temps minimum T_r d'évacuation des étages r et supérieurs par :

$$T_r = \frac{(n-r+1) \cdot Q}{N' \cdot b} + r \cdot t_s \quad (08)$$

Deux cas sont alors possibles pour le temps net d'évacuation T_e :

$$\frac{Q_i}{N' \cdot b_r} > t_s \quad T_r \text{ est maximum pour } r = 1$$

$$\frac{Q_i}{N' \cdot b_r} < t_s \quad \left\{ \begin{array}{l} T_e = \frac{n \cdot Q}{N' \cdot b} + t_s \quad (\text{saturation à chaque niveau}) \\ T_r \text{ est maximum pour } r = n \\ T_e = \frac{Q}{N' \cdot b} + n \cdot t_s \quad (\text{aucun saturation}) \end{array} \right. \quad (09)$$

Malheureusement, les temps prédits par ce modèle sont généralement bien inférieurs à ceux observés en situation réelle (avec un écart d'environ deux minutes). S. J. Melinek et S. Booth proposent donc de rajouter systématiquement ces deux minutes aux temps calculés car ils estiment qu'elles correspondent au délai de réponse à l'alarme ainsi qu'au temps d'accès aux

escaliers. Comme pour le modèle de K. Togawa, le modèle présenté ici fournit des approximations quant aux débits maximum de personnes dans des situations particulières. Les corrections approximatives apportées par S. J. Melinek et S. Booth [84] peuvent amener à douter quant à la généralisation de leur modèle. En effet, il est difficilement concevable de pouvoir approximer la complexité du comportement humain par une approximation constante du temps de réponse à un stimulus externe. Ainsi, d'autres situations nécessiteront certainement des corrections différentes.

3.2.8 Modèle de Paule

J. L. Pauls [84] oriente son étude sur l'observation d'exercices d'évacuation d'immeubles de bureaux de grande hauteur au Canada. Celui-ci justifie l'utilisation de son modèle par le fait que de tels exercices, menés en situations réelles, peuvent être considérés comme potentiellement dangereux. Son étude porte surtout sur l'évacuation au niveau des escaliers, en établissant la loi suivante :

$$f = 0.206 \cdot p^{0.27}, \text{ Avec}$$

- f = débit unitaire en pers /s/m de largeur d'escalier
- p = nombre de personnes par mètre de largeur d'escalier

J. Pauls [84] affine son modèle quantitatif par des précisions qualitatives provenant de diverses observations (vidéos, photographies). Il développe ainsi le concept de largeur effective d'un escalier en constatant qu'un espace d'une quinzaine de centimètres est systématiquement laissé entre les évacuants et les murs latéraux. La largeur effective correspond donc à la largeur réellement utilisée d'un escalier, soit la largeur réelle diminuée de trente centimètres. Ainsi, il indique que la formule permettant de calculer le débit est :

$$Q = 0.206 \cdot (L - 0.3) \cdot \left(\frac{P}{L - 0.3} \right)^{0.27} \quad (10), \text{ avec}$$

- Q = débit en pers/s
- L = largeur effective en mètres
- P = nombre de personnes évacuant ($P = p \cdot L$)

Le temps d'évacuation est alors lié à la population initiale du bâtiment et à la largeur effective des escaliers par :

$$\frac{L}{P} = 8040 \cdot T^{-1.37} \quad (11), \text{ avec}$$

- L = largeur effective en mètres
- P = nombre de personnes évacuant
- T = temps d'évacuation en secondes (au maximum 600)

3.2.9 Le modèle de Predtechenskii et Milinskii

Un modèle fluide déterministe basé sur l'analyse statistique d'un grand nombre de mouvements de foules est proposé par V. M. Predtechenskii et A. I. Milinskii [84]. Ces travaux ont eu un impact important en Europe et sont toujours utilisés aux États-Unis, en particulier par la Society of Fire Protection Engineers. Ils introduisent les concepts de débit unitaire et d'intensité (produit de la densité et de la vitesse du mouvement) pour déterminer la capacité d'une issue. La densité locale de la foule, qui est la grandeur fondamentale de ce modèle, est le rapport entre la somme des projections horizontales des individus et la surface totale occupée par la foule. L'encombrement d'une personne est calculé ici en fonction de son apparence physique, de son âge et de sa tenue vestimentaire, et correspond environ à une ellipse :

$$s = \frac{\pi}{4} \cdot l \cdot e, \text{ avec}$$

- s = surface au sol occupée par l'individu en m²

- l = largeur de l'individu en mètres
- e = épaisseur de l'individu en mètres

3.2.10 Le modèle de Takahashi, Tanaka, et Kose

En 1988, au Japon, K. Takahashi, T. Tanaka et S. Kose [84] proposent un modèle. Ils décrivent un bâtiment avec huit éléments différents, dont deux représentants d'espaces virtuels :

- Une pièce est un espace pourvu d'issues, contenant initialement des personnes.
- Un couloir est un espace pourvu d'une entrée et d'une sortie, initialement vide. Il peut être divisé en unités pour les besoins du calcul.
- Un escalier est un espace reliant entre eux deux étages, fonctionnant comme un chemin.
- Un vestibule est un espace avec une entrée et une sortie vers un escalier, fonctionnant comme un chemin et permettant, par comptage, de juger de l'état d'évacuation d'un étage.
- Un hall est un espace à entrées et sorties multiples, connectées par des chemins multiples.
- Un refuge est une destination finale d'évacués.
- Un lien est un espace imaginaire sans capacité qui régule un flot d'évacués à l'intersection de deux éléments.
- Une zone d'entassement est un espace imaginaire qui reçoit les évacués ne pouvant passer immédiatement à l'élément suivant.

Le modèle suppose des mouvements d'évacuation homogènes (modèle fluide, vitesse constante) et une répartition initiale uniforme dans les pièces. Les occupants de chaque pièce vont vers la sortie au début de l'évacuation, en privilégiant la plus proche si elles sont multiples. Par conséquent, une pièce ayant plusieurs sorties sera divisée en unités. Si l'issue ne se trouve pas dans un coin, elle doit être divisée en deux unités.

4. Les systèmes récents de simulation piétonniers

Les classes microscopiques et macroscopiques comprennent les modèles commerciaux de la circulation des piétons. La classe microscopique étudie les caractéristiques de différents piétons, telles que la vitesse et l'interaction avec les autres piétons, tandis que la classe macroscopique étudie des groupes de piétons plutôt que des caractéristiques particulières, et leur analyse se concentre sur la densité et les systèmes à grande échelle. Parce que notre intérêt principal est dans les simulations où chaque agent est guidé par ses propres objectifs, sa propre personnalité et un processus de prise de décision. L'accent sera mis sur les systèmes microscopiques.

Cette section examine certains systèmes de simulation de foule qui ont été développés à la fois dans le milieu professionnel et universitaire.

Exodus : Le « Fire Safety Engineering Group » de l'université de Greenwich a créé cet outil [85][86]. (Les figures 3.20 et 3.21 montrent des projectiles d'écran du commutateur d'Exodus.) [67].

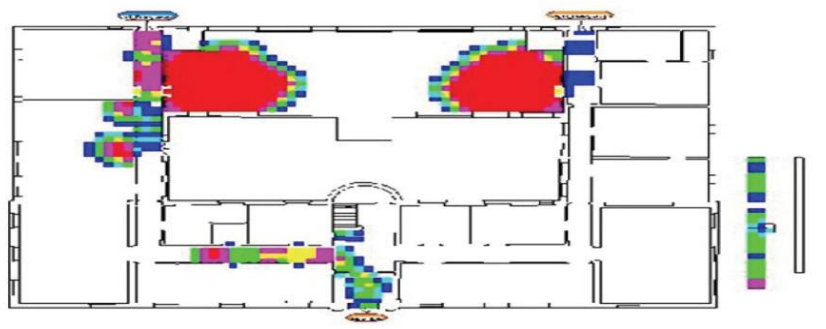


Figure 3.20 : Distributions de densité pendant une évacuation dans l'exode [85].

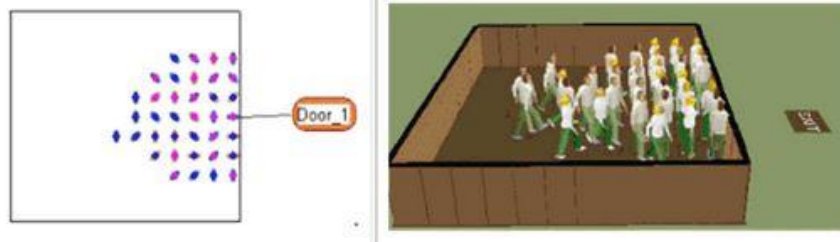


Figure 3.21 : Simulation d'une évacuation avec l'exode de bâtiment et de sa visualisation en différé dans 3D avec le vrExodus [86].

Pedroute : Il a été initialement créé par « London Underground Limited » et a été largement utilisé pour simuler les paramètres de foule dans les réseaux souterrains du monde entier [87][67].

CROSSES (Crowd Simulation System for Emergency Situations) est un système de simulation de foule d'urgence qui utilise un modèle pour créer et simuler une foule virtuelle afin de former les personnes à répondre efficacement aux urgences urbaines [88]. Avec l'environnement virtuel et les vrais participants humains, ce système multi-agents fournit des agents autonomes et des comportements préétablis [67].

Simulex : Il a été créé comme modèle d'évacuation qui peut représenter un grand nombre de personnes dans des bâtiments géométriquement complexes [89]. La représentation de la distance et la simulation d'une foule sont illustrées à la figure 3.22 [67].

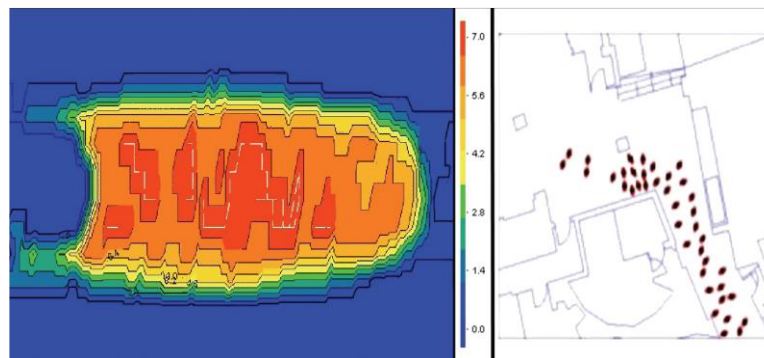


Figure 3.22 : Densité et simulation de foule dans Simulex.

Rampage : C'est un système de simulation de particules pour animer des explosions et d'autres primitives élémentaires ; il divise le comportement humain en réactions réflexes et la prise de décision basée sur la connaissance de la scène [90]. L'équation de gaz de Boltzmann sert de base à ses théories pour simuler le comportement humain [67].

Egress : Le développement de l'Egress a été initié par la technologie AEA en 1991 (technologie AEA 2002). C'est un logiciel de simulation de foule commercial. Le modèle utilise des techniques d'intelligence artificielle pour déterminer comment une personne réagirait dans une

variété de circonstances de feu et de fumée. Le rendement d'Egress comprend une analyse du temps d'évacuation [67].

Legion : Ce système n'a pas été conçu pour l'analyse comportementale de foule, mais plutôt pour l'étude de systèmes interactifs de grande échelle. Le modèle informatique rend plus facile de représenter le comportement des gens. Pour représenter la nature complexe des comportements de l'individu, le modèle utilise seulement quatre paramètres (point de but, vitesse, distance à d'autres et temps de réaction) et une règle de décision basée sur l'hypothèse du principe de moindre-effort. La station de train avec le système Legion est simulée à la figure 3.23 [67].

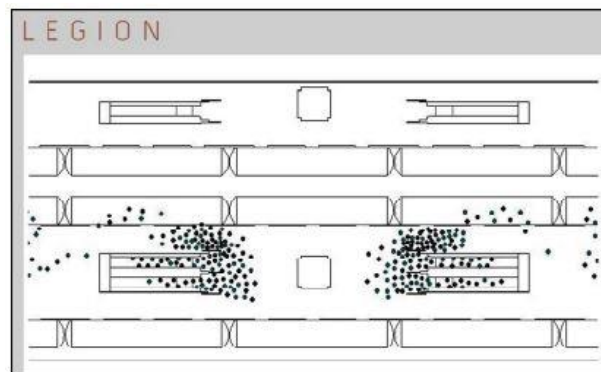


Figure 3.23 : Simulation de légion d'une station de train (légion 2003).

STEPS : C'est un modèle à base d'agent qui suit la géométrie des automates cellulaires (voir la figure 3.24) [92]. Si la prochaine cellule est vide, chaque individu occupe une cellule à un moment donné et se déplace vers une direction désirée. Chaque cellule occupée a ses propres caractéristiques, sa patience et son comportement typique [67].

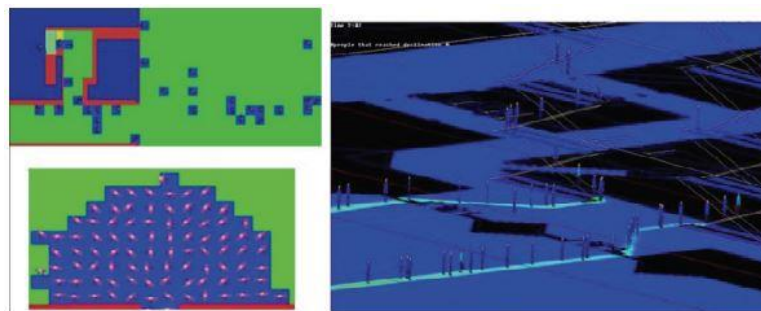


Figure 3.24 : simulation 2D par STEPS.

ViCrowd : représente un modèle pour générer automatiquement des foules de personnes en fonction des caractéristiques de groupe plutôt que des individus [93]. Pour satisfaire les conditions en temps réel, les membres d'un groupe suivraient les normes du groupe plutôt que leurs propres préférences. Il inclut une définition simple des règles comportementales utilisant des événements et des réactions conditionnelles, bien qu'il soit basé sur le système de flocking de Reynolds. Les effets de l'affinité et de la répulsion dans la simulation de foule peuvent être manipulés à l'aide d'un modèle sociologique, ce qui entraîne des comportements plus complexes.

OpenSteer : fournit une trousse à outils qui définit les comportements de direction en termes d'agent mobile abstrait [94][67].

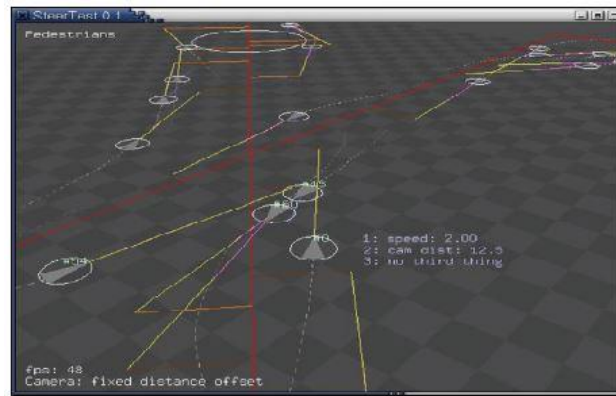


Figure 3.25 : L'interface d'Open street [94].

Massive SW. Massive Software, Inc. a développé ce système basé sur la technologie de la vie artificielle qui utilise une combinaison de règles simples et de logique floue (Massive Software, Inc., 2005). Pour obtenir des informations supplémentaires. La simulation d'évacuation d'une foule est illustrée à la figure 3.26 [67].

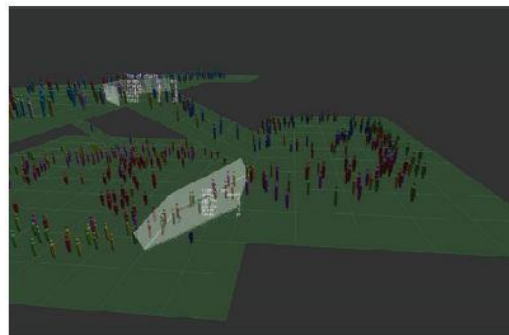


Figure 3.26 : Simulation d'évacuation avec Massive WS pour la conception architecturale et l'évaluation.

Reactive Navigation : Les agents de Lamarche et de Donikian (2000, 2004) naviguent dans des environnements virtuels complexes qui sont représentés par une structure topologique hiérarchique qui est extraite de la géométrie de l'environnement virtuel, comme illustré à la figure 3.27. Cette structure permet une recherche de chemin rapide et un algorithme de navigation réactif efficace. Ils utilisent un processus d'optimisation itératif pour éviter les collisions tout en atteignant leurs cibles. Les calculs de voisinage sont la base de la prévision de collision, qui crée des relations de voisinage dans les foules clairsemées et des relations de courte distance dans les foules denses [67].



Figure 3.27 : Navigation réactive [95].

Artificial fishes : Une approche de la vie artificielle a été créée par TU et Terzopoulos (1994) pour reproduire l'aspect, le mouvement et le comportement de différents poissons dans un monde marin virtuel, ainsi que les comportements complexes de groupe observés dans de vrais écosystèmes aquatiques. Le comportement des poissons est illustré par chaque poisson [67].



Figure 3.28 : Les poissons artificiels.

ACUMEN : Il utilise une interface de langage naturel dans un environnement virtuel pour synthétiser et identifier les mouvements globaux [96]. Il comprend une interface interactive de commande globale basée sur une collection de paramètres d'extension d'un modèle existant. La représentation paramétrisée d'action [97] a été utilisée par le système ACUMEN pour capturer la sémantique du mouvement global pour la génération et l'identification. Le mouvement est principalement basé sur un système de particules, tel que le modèle de la simulation de groupe. Les forces dynamiques agissant sur des corps rigides produisent le mouvement souhaité, comme indiqué dans la figure [67] [98] [99].

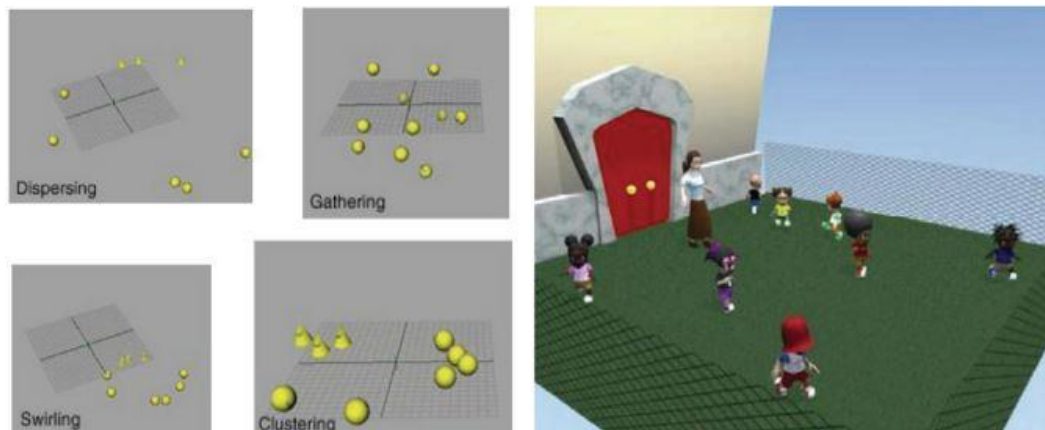


Figure 3.29 : Le système ACUMEN est appliqué aux sphères et les écoliers [96].

Autonomous Pedestrians : Terzopoulos l'a préconisé car il offre un moyen d'intégrer l'approche de la vie artificielle. Un modèle des piétons intègre des éléments cognitifs, perceptifs et comportementaux [100]. Les structures de données hiérarchiques représentent l'environnement et soutiennent efficacement les requêtes perceptuelles sur les piétons autonomes. Ces requêtes conduisent leurs réponses comportementales et maintiennent leur capacité à prévoir leurs actions à un niveau local et global [67].

Six routines réactives de base de comportement sont exécutées par les agents : éviter l'obstacle statique, éviter l'obstacle statique lors d'un tour complexe, maintenir la séparation dans une foule mobile, éviter les piétons approchants, éviter les piétons dangereusement étroits et vérifier les obstacles relatifs dans de nouvelles directions. Les agents exécutent l'action d'évitement de collision ; donc, si une intersection avec un autre piéton est sur le point de se produire, l'agent s'arrêtera, essaiera de tourner pour faire face à l'autre côté et attendra jusqu'à ce que l'espace soit disponible autour de sa position actuelle [101] [102].

Des arbres de décision sont construits pour différents comportements, et leurs probabilités d'obtenir les comportements souhaités sont données.



Figure 3.30 : Les piétons autonomes [100].

Space syntax : Les planificateurs urbains ont suggéré un graphique de visibilité pour le mouvement piétonnier [103]. Cette méthode examine les points visuellement accessibles dans un secteur pour déterminer la direction probable de la navigation d'un piéton. Sans tenir compte d'autres facteurs socio-économiques (atteindre une destination, rencontrer des personnes, prendre quelque chose, etc.) ou physiques granulaires (particules mobiles vers un but, poussée, formant des ruelles, etc.), cette méthode utilise le champ visuel pour guider le mouvement normal. Ils calculent "une architecture visuelle exo-somatique" où les connexions entre les endroits mutuellement évidents dans une configuration sont préenregistrées dans une table de consultation pour simuler une foule relativement grande [67].

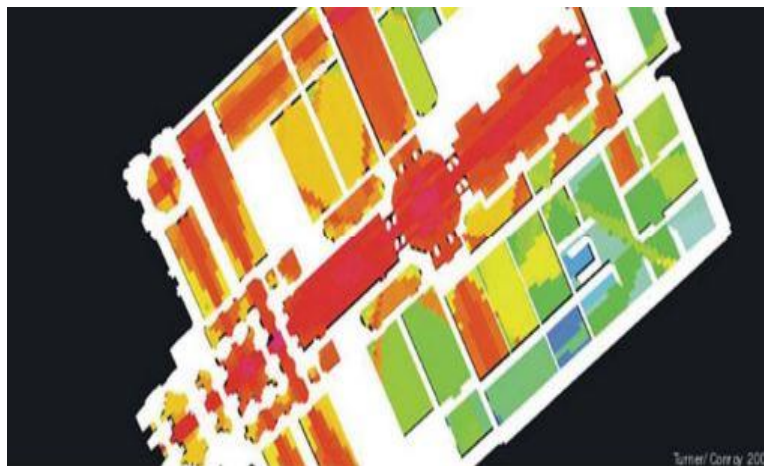


Figure 3.31 : Graphique de visibilité de syntaxe de l'espace [104].

5. Classifications des méthodes de foule

Nous proposons dans cette section deux classifications pour les travaux effectués dans le domaine des foules après avoir examiné les études dans ce domaine. La première est liée aux techniques utilisées pour représenter les comportements et les mouvements de foules. La seconde classe les foules en fonction de leurs utilisations.

Les méthodes qui peuvent être utilisées pour représenter les foules sont décrites dans la classification suivante :

- Méthodes de foules à base d'image : définissent les techniques pour transformer les images et les visions en données de vraies foules.
- Modèles de foule à base physique : utilisé dans les fondements de la physique pour expliquer le mouvement de foule.
- Modèles procéduraux : basé sur des équations paramétriques.

- **Modèles comportementaux** : fournissent un outil pour simuler des foules basées sur le comportement.

Une classification de diverses méthodes et systèmes de modelage de groupes ou de foules est présentée dans le tableau 3.2. Chaque méthode est caractérisée par divers paramètres, tels que la communication interpersonnelle, l'apprentissage et l'utilisation en temps réel.

	Communication ou signale	Individus ou des rôles	Temps réel	Apprentissage	Structure spatiale pour le mouvement
modèles de forces sociaux	Non	Certains	Oui	Non	Construction
Basée sur les règles	Non	Non	Oui	Non	Construction
AC	Non	Non	Oui	Non	Grille 2D
Simulex	Non	Certains	Non	Non	Construction
Egress	Non	Certains	Non	Non	Grille hexagonal
ViCrowd	Non	Oui	Oui	Non	Construction
OpenStreet	Non	Certains	Oui	Non	Construction
Legion	Certains	Certains	Non	Non	Construction
Exodus	Non	Certains	Non	Non	Grille 2D
Steps	Non	Certains	Non	Non	Grille 2D
Massive SW	Non	Oui	Non	Non	Construction
Reactive navigation	Non	Oui	Oui	Non	Construction
Artificial fishes	Non	Oui	Oui	Oui	Construction
ACUMEN	Oui	Oui	Oui	Non	Construction
Crosses	Non	Oui	Oui	Non	Construction
Autonomous Pedestrians	Non	Oui	Oui	Non	Construction
Space Syntax	Non	Non	Oui	Non	Construction
MACES+HiDAC	Oui	Oui	Oui	Oui	Construction

Tableau 3.2 : Les différentes approches dans le domaine [67].

En fonction de la nature des foules requises dans les applications, les méthodes présentées dans le tableau 3.2 peuvent être utilisées. En effet, les contraintes varient selon le but de l'application. Par exemple, si l'application vise à simuler un mouvement de foule lors d'une évacuation, la commande interactive et la simulation de comportement ne peuvent pas être considérées comme cruciales. Sur la base de divers objectifs, nous avons classé ces applications en quatre catégories:

- **Les applications** de divertissement comprennent la production de films et les jeux de film. Dans ce cas, les individus et les groupes doivent être facilement dirigés et commandés. Les films récents "AntZ" [105] et "désinsectise la vie" [106] en sont des exemples. La simulation de plusieurs personnes et la génération automatique d'individualités sont des défis. De plus, la commande simple des groupes est une condition de ce type d'application.
- **Le but de la simulation de mouvement de foule** est d'évaluer le mouvement de plusieurs personnes dans un espace limité. Pour évaluer l'environnement, elle nécessite généralement une visualisation simple (incorporation de foule) et un compromis solide avec les résultats numériques. Des applications de masse comprennent des logiciels commerciaux tels que SIMULEX [107] et LEGION [108]. Une méthode réaliste d'action pour éviter les collisions, un lien fort avec l'environnement et des résultats numériques et statistiques sont les principaux défis dans ce cas.
- Afin **d'augmenter la taille des environnements virtuels de collaboration**, comme dans un système CVE, il est nécessaire d'effectuer une simulation en temps réel ainsi que de nombreux aspects interactifs pour guider et diriger la foule pendant la simulation. Dans

les environnements virtuels de collaboration où les participants peuvent créer et interagir avec des groupes, la programmation du comportement de groupes est un défi important. Quelques expériences ont été effectuées dans le cadre du projet COVEN. [TS97].

- **La modélisation comportementale des foules** est le dernier type d'application de foule. Le but est de fournir des comportements autonomes ou semi-autonomes qui peuvent être utilisés pour la self-animation des agents formant la foule dans ce cas. Pour faire face aux exigences imposées par le grand nombre d'acteurs, qui peuvent être de centaines ou de milliers, cette forme d'application présente des difficultés telles que : description des aspects sociologiques et sociaux qui surgissent dans les foules en traitant les comportements et les connexions dirigées et émergentes avec l'environnement virtuel. Afin de modéliser et de simuler l'intelligence de foule et les capacités de décision, des optimisations et des simplifications sont également nécessaires.

6. Synthèse sur les modèles de foule

La simulation de foules ou de groupes a été largement étudiée, en particulier la dynamique d'évacuation. Mais il reste des choses à améliorer. Les applications commerciales se concentrent principalement sur la validation de leurs systèmes en termes de sortie. Ils utilisent à la fois des techniques microscopiques et macroscopiques. Les modèles microscopiques les plus couramment utilisés dans les applications industrielles sont les simulations de particules et les modèles d'automate cellulaire (AC). Ces méthodes se sont avérées manquer du réalisme lorsqu'elles sont utilisées pour des systèmes d'animation avec des humains virtuels 3D car elles l'un ou l'autre regarde plus près des particules que au vrai mouvement humain (les modèles de forces sociales et la simulation de particules en général) ou sont limitées aux configurations de damier.

Cependant, la recherche sur la création d'agents autonomes s'est concentrée sur leur capacité à diriger de grands environnements virtuels complexes tout en évitant des obstacles statiques et d'autres agents. Cependant, dans la plupart des cas, ils négligent les problèmes qui surviennent lorsque vous devez gérer les foules à haute densité. Ces systèmes appliquent généralement un certain nombre de règles pour éviter les collisions lorsque les agents modifient leur vitesse ou leur trajectoire. Par conséquent, ces modèles sont suffisants pour les foules de densité moyenne-basse. Cependant, lorsqu'il y a une foule très dense, ils rapportent des comportements émergents artificiels, comme cesser et attendre l'espace pour éclaircir. Il n'y a aucun concept de contact physique qui pousse des agents dans une foule ou traîne des individus dans la foule. D'autres effets, tels que la chute, les dommages, l'incapacité et d'autres qui marchent sur l'agent tombé, sont également ignorés. Le principal objectif du système HiDAC est de traiter toutes ces configurations qui émergent dans de vraies foules à haute densité [109].

En termes de navigation globale, les systèmes édités pour la simulation de foule supposent que les agents disposent d'informations complètes sur l'environnement. Un agent peut accéder à la structure interne complète des algorithmes d'environnement et d'utilisation comme A* (plusieurs techniques ont été utilisées pour réaliser le temps réel en effectuant la planification globale de chemin pour de grands groupes d'agents), ou l'environnement peut être discrétisé comme une grille qui stocke des potentiels ou des cartes de distance que les agents suivront pour atteindre localement le but. Un agent doit être autorisé à explorer des environnements en partie connus et à apprendre les nouveaux dispositifs. Les foules ont également la capacité de communiquer avec d'autres pour échanger des informations importantes de navigation, ce qui est ignoré ailleurs. En les permettant d'avoir des informations partielles sur l'environnement et de pouvoir prolonger leur mémoire (ou cartes mentales) pendant qu'ils explorent

l'environnement et communiquent avec d'autres agents dans la foule, notre travail se concentre également sur améliorer le réalisme du comportement des agents. Basés sur leurs différents rôles, les agents peuvent également adopter divers comportements [110]. Pour résumer, pour modifier l'exécution globale d'une personne basée sur son état mental, quelques facteurs psychologiques doivent être incorporés.

Une comparaison de certains des modèles les plus importants de l'animation de foule est présentée dans le tableau 3.3. Pour comparer les contributions de notre modèle (MACES + HiDAC) à d'autres, cela souligne les caractéristiques principales des simulations multi-agent et des évacuations piétonnières.

	Réponse de collision	Secousse de particules corrigée	Méthode de comportement
modèles de forces sociaux	Oui	Non	Les forces
Basée sur les règles	Non	Pas besoin	Les règles
AC	Non	Pas besoin	AC
Simulex	Oui	Non	Cartes de distance
Egress	Non	Pas besoin	AC
ViCrowd	Non	Pas besoin	Les règles
OpenStreet	Non	Pas besoin	Les règles
Legion	Non	Pas besoin	Moindre-effort
Exodus	Non	Pas besoin	AC
Steps	Non	Pas besoin	AC
Massive SW	Non	Pas besoin	les règles + la logique floue
Reactive navigation	Non	Pas besoin	Les règles
Artificial fishes	Non	Pas besoin	Les règles
ACUMEN	Oui	Non	Simulation des particules
Crosses	Non	Pas besoin	Les règles
Autonomous Pedestrians	Non	Pas besoin	Approche vie artificielle
Space Syntax	Non	Pas besoin	Graphiques de visibilité
MACES+HiDAC	Oui	Oui	Forces sociales prologées

Tableau 3.3 : Comparaison de différents systèmes pour l'animation de grands groupes.

7. Conclusion

Nous avons présenté dans ce chapitre quelques différents modèles et techniques existant pour la simulation de foule, ici nous avons remarqué que la simulation comportementale et la création d'un environnement virtuel habité et peuplé par un grand nombre de piétons d'une manière réaliste est introduite dans plusieurs domaines d'applications tels que : le graphisme, la robotique, les jeux vidéo et les films. Dans ce contexte deux approches ont été traitées qui sont les méthodes microscopiques et macroscopiques. La méthode macroscopique étudie la foule dans sa totalité et ne considère pas l'interaction entre les individus, et elle est caractérisée par son faible coût de calcul, mais ils ne prennent pas en compte ni la perception visuelle, ni les règles sociales.

Deuxième partie

Contribution

Contribution

Chapitre 4 : Modèle de simulation proposé

1. Introduction

Plusieurs modèles de simulation ont été développés au cours des dernières décennies pour analyser la dynamique de l'évacuation des foules dans les situations normales et les situations d'urgence. Helbing [11] a proposé un modèle de force sociale qui considère les piétons comme des particules animées par une force et s'attaque au problème de l'évacuation des piétons effrayés. Varas et al [12] ont utilisé un modèle d'automate cellulaire pour simuler l'évacuation de piétons d'une salle comportant des obstacles fixes.

Afin d'imiter une expérience d'évacuation menée dans une salle de classe avec des barrières, Liu et al [13] ont modifié le modèle d'automate cellulaire. Les modèles de gaz en treillis [14,15], les modèles de théorie des jeux [16,17], les contrôles de rétroaction basés sur l'optimisation [18] et d'autres sont fréquemment utilisés pour analyser les problèmes d'évacuation dans les espaces publics.

En outre, de nombreuses méthodologies ont été utilisées pour étudier les comportements d'évacuation dans diverses circonstances, notamment dans les tunnels routiers enfumés [19], les hôtels en feu [20], les immeubles de grande hauteur [21] et le bioterrorisme dans des contextes micro-spatiaux [22]. Ces résultats montrent que les types et l'ampleur de l'influence sur le comportement des foules varient considérablement en fonction des situations et des environnements associés. Dans le même temps, de nombreux conseils utiles pour garantir une évacuation en toute sécurité sont également fournis par les études publiées dans la littérature.

Nasir et al [23] ont présenté un système génétique flou. La perception floue et la peur font partie intégrante de la pensée humaine, et Mauro et al [24] ont proposé un modèle comportemental basé sur la logique floue pour l'évacuation des foules qui tient compte de ces facteurs. En outre, certains systèmes d'inférence floue sont conçus pour offrir la possibilité de s'échapper, de retarder l'évacuation et d'indiquer la direction du mouvement [7,8].

Il est important de noter que les études sur les comportements dynamiques des piétons dans des conditions normales et de panique ont été menées à l'aide de modèles et de simulations basés sur l'intelligence artificielle [25,26], où l'esprit et la perception humaine peuvent être prédits à l'aide de techniques d'intelligence artificielle.

Afin de modéliser et de simuler le comportement de direction des piétons dans les environnements bâtis, les auteurs de [27] ont présenté une étude sur la dynamique du mouvement des piétons lors d'une évacuation d'urgence en utilisant l'apprentissage automatique. Dans le travail présenté dans [28], une méthode d'apprentissage par renforcement est utilisée pour produire un modèle basé sur des données pour l'évacuation des foules. Li et al [29] ont combiné les techniques d'apprentissage profond et social pour développer un modèle de simulation pour l'évacuation des foules. Avant de discuter de la description de notre modèle, il est préférable de définir brièvement quelques notions sur les données et les algorithmes utilisés.

2. Fouille de données (Data mining)

Le data mining peut être appelée exploration de connaissances à partir de données, extraction de connaissances, analyse de données/modèles. Le data mining, également connue sous le nom de découverte de connaissances dans les bases de données KDD (Knowledge Discovery in Databases), fait référence à l'extraction non triviale d'informations implicites, auparavant inconnues et potentiellement utiles à partir de grands ensembles de données.

Le besoin de data mining est d'extraire des informations utiles à partir de grands ensembles de données et de les utiliser pour faire des prédictions ou une meilleure prise de décision. De nos jours, data mining est utilisée dans presque tous les domaines.

Le processus KDD est un processus itératif et il nécessite plusieurs itérations des étapes ci-dessus pour extraire des connaissances précises des données. Les étapes suivantes sont incluses dans le processus KDD (voir figure 4.1) :

- Le nettoyage des données (Data Cleaning)
- Intégration des Données (Data Integration)
- Sélection des Données (Data Selection)
- Fouille de données (Data Mining)
- Évaluation du Modèle (Pattern Evaluation)
- Représentation des Connaissances (Knowledge Representation)

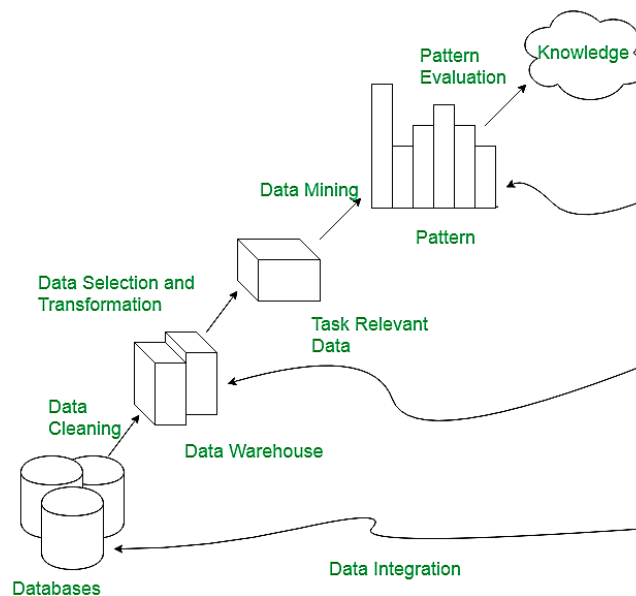


Figure 4.1: Le processus de KDD (Knowledge Discovery in Databases)

Les données utilisées dans cette recherche sont des données liées aux piétons, utilisées pour la découverte de connaissances KDD (Knowledge Discovery in Databases), ces connaissances déterminent le comportement de la foule. Pour cela, nous avons utilisé les techniques de certains modèles de l'intelligence artificielle comme ANN (artificial neural network) pour simuler les comportements des piétons, et compte tenu des comportements des piétons, nous pouvons collecter une image complète du mouvement de la foule et extraire des comportements classés en fonction du besoin de la simulation.

Cette tâche est ostensiblement une simulation, mais en fait c'est un processus d'exploration du comportement des piétons basés sur des données (fouille de données).

Nous avons utilisé la classification afin de connaître le comportement d'un piéton et nous avons implicitement ajouté une idée inspirée de la logique floue pour rapprocher la simulation des piétons aux comportements des êtres humain réel (les variables linguistiques).

Le KDD est un processus itératif où les mesures d'évaluation peuvent être améliorées, l'exploration peut être affinée, de nouvelles données peuvent être intégrées et transformées afin d'obtenir des résultats différents et plus appropriés. Le prétraitement des bases de données consiste en un nettoyage et une intégration des données (voir figure 4.1).

Nous avons utilisé le KDD pour étudier les comportements de piétons dans les foules selon leur perception de leur vitesse (stop, lent et rapide), leur distance par rapport au but (proche, loin, ...) et leur direction vis à vis le but (zéro, aigu et large).

Nous avons utilisé l'algorithme K-means et nous suivons les étapes suivantes :

1. Collecte des données : collectez des données sur les positions, directions et les vitesses de mouvements des piétons au sein de la foule. Nous avons obtenu en utilisant la technique des modèles de simulation (des données synthétiques).
2. Extraction des caractéristiques : convertissez les données collectées en caractéristiques appropriées pour le clustering K-means. Ces caractéristiques pourraient inclure les coordonnées spatiales, les vitesses, les directions ou toute autre variable pertinente décrivant le comportement des mouvements de piétons.
3. Déterminez la valeur de k : Choisissez le nombre approprié de clusters, k, en fonction de notre objectifs d'évitement des obstacles, d'évitement de collision et atteindre le but.
4. Appliquez K-means : Utilisez l'algorithme K-means pour regrouper les piétons en fonction de leurs caractéristiques de mouvement. Attribuez chaque individu au centroïde le plus proche représentant un cluster. Les centroïdes agiront comme des points représentatifs pour chaque cluster.
5. Analysez les clusters : Une fois le regroupement terminé, nous avons analysés les clusters résultants. Vous pouvez examiner les caractéristiques et les comportements des individus au sein de chaque cluster, tels que leurs itinéraires préférés, leurs schémas d'interaction ou leur comportement.

```
def normalize_Peds():
    data_array=[ped.loggedData for ped in Pedestrians]
    df = pd.DataFrame(data_array)
    return min_max_normalize(df).fillna(0)
def get_optimal_k_silhouette(min_clusters,max_clusters,normalized_df):
    # Silhouette Method
    silhouette_scores = []
    # Iterate over different values of k to find the optimal number of clusters
    for k in range(min_clusters, max_clusters):
        kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
        cluster_labels = kmeans.fit_predict(normalized_df)
        silhouette_avg = silhouette_score(normalized_df, cluster_labels)
        silhouette_scores.append(silhouette_avg)
    return silhouette_scores.index(max(silhouette_scores)) + min_clusters
def do_clustering(optimal_k,normalized_df):
    # Perform K-means clustering with the optimal number of clusters
    kmeans = KMeans(n_clusters=optimal_k, init='k-means++',
random_state=42)
    cluster_labels = kmeans.fit_predict(normalized_df)
    for i,c in zip(normalized_df.index ,cluster_labels):
        Pedestrians[i].cluster = c
```

L'application de k-means dans notre simulation de foule peut fournir des informations précieuses sur la dynamique des foules ainsi que leur comportement, l'algorithme K-means aide à optimiser les stratégies de gestion des foules et améliorer la sécurité et l'efficacité dans divers scénarios, tels que la planification d'évacuation, la gestion d'événements ou la conception urbaine. A la fin l'algorithme k-means ne sont qu'une approche parmi tant d'autres disponibles. Selon vos besoins spécifiques, d'autres algorithmes de clustering ou techniques de simulation peuvent également convenir.

3. Description du modèle d'évacuation des foules

La description du modèle d'évacuation de la foule est basée sur la description de l'espace physique, qui est composé d'obstacles, de frontières (des murs), d'une porte de sortie (le but), des piétons, qui sont représentés par des positions physiques (indiquées par des coordonnées de localisation) et des comportements.

Nous utilisons Python [30] (voir la figure 4.2 (a)), un langage de programmation polyvalent de haut niveau fonctionnant sur Jupyter Notebook (une plate-forme informatique interactive basée sur le Web) (voir la figure 4.2 (b)) pour développer un modèle de simulation pour l'analyse de l'évacuation des foules à l'aide de l'intelligence artificielle (le code Python complet est disponible à la suite), (voir la figure 4.2).



(a) Python



(b) Jupyter Notebook

Figure 4.2 : l'environnements de programmation

3.1 Description de l'espace physique

La simulation de foule est le processus de création de modèles informatiques du comportement et du mouvement des personnes dans des espaces surpeuplés, tels que des stades, des gares ou des centres commerciaux. Il s'agit d'utiliser des algorithmes mathématiques et de l'infographie pour simuler le flux et le comportement de grands groupes de personnes dans un espace donné.

Pour définir notre espace de simulation nous introduisons d'abord une méthode de représentation de l'espace physique "s", qui joue un rôle central dans la modélisation et la simulation. La simulation sera réalisée dans un hall carré (rectangle) en utilisant le modèle piétonnier proposé, dont l'échelle est *Largeur d'espace* × *Longueur d'espace* avec une sortie de *goal_width* au milieu du mur. L'espace physique "s" est composé de bordures (murs) et d'obstacles internes de formes différentes et d'une sortie. Ce qui suit est une partie du code Python pour générer un modèle de l'espace physique "s".

```
def generate_physical_space(pnumber = 200,
                           nobstacles_vertexes = [], goal_width = 500):
    Goal, goal_position = generate_goal(goal_width)
    positions = generate_positions(pnumber)
    obstacles = generate_obstacles(positions, nobstacles_vertexes)
    borders = generate_borders(goal_position, goal_width)
    return positions, obstacles, borders, Goal, goal_position
```

Tout d'abord, nous générons la position de sortie (but) par l'exécution de la fonction *Goal*, $goal_position = generate_goal(goal_width)$, où *Goal* est le polygone de sortie (c'est un rectangle). L'appel à la fonction $positions = generate_positions(pnumber)$ générera aléatoirement un ensemble de positions $pnumber$ (points) dans l'espace pour représenter les positions initiales des piétons en "s". L'exécution de la fonction $obstacles = generate_obstacles(positions, nobstacles_vertexes)$ générera aléatoirement un ensemble d'obstacles, qui sont des polygones remplis, chacun est spécifié avec un certain nombre de sommets. La liste *nobstacles_vertexes* contient le nombre de sommets de chaque obstacle. Les bordures (qui sont des obstacles externes, en forme de rectangles) sont générées par l'exécution de l'appel à la fonction $borders = generate_borders(goal_position, goal_width)$ (voir figure 4.3).

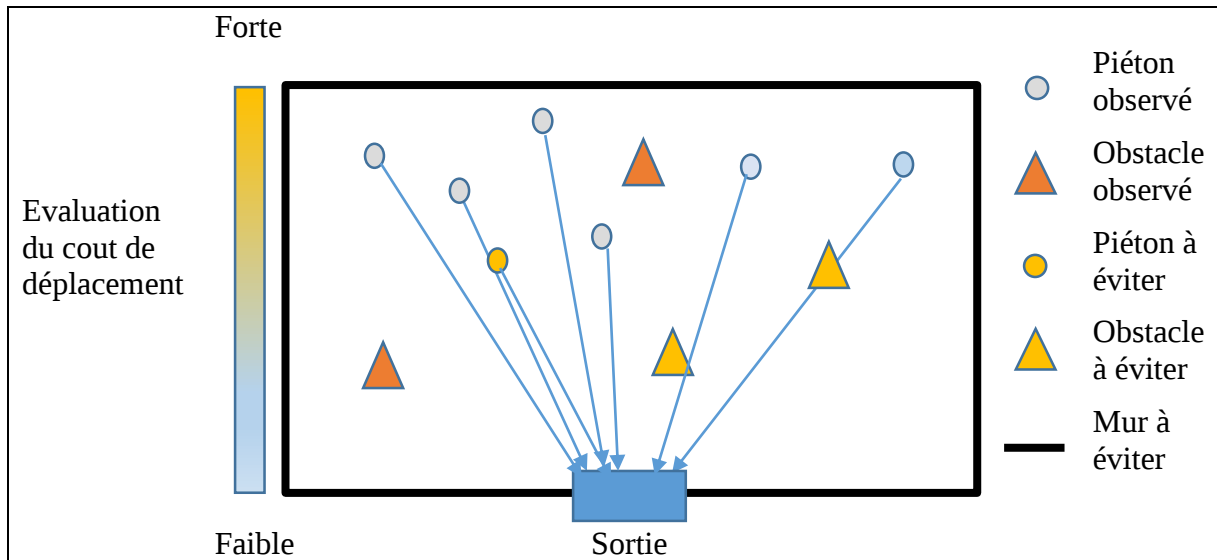


Figure 4.3 : les éléments de l'espace physique modélisé

3.2 Description des piétons

En plus de l'espace physique "s", le modèle est composé d'une foule de piétons à évacuer d'une situation dangereuse. L'évacuation est essentiellement basée sur le comportement des piétons en réponse à l'environnement. Ce comportement humain rend notre modèle intelligent, où les connaissances utilisées pour décrire le modèle intelligent reposent sur l'expérience humaine dans la catégorisation des valeurs pour les distances, les directions (angles) et les vitesses (voir figure 4.4).

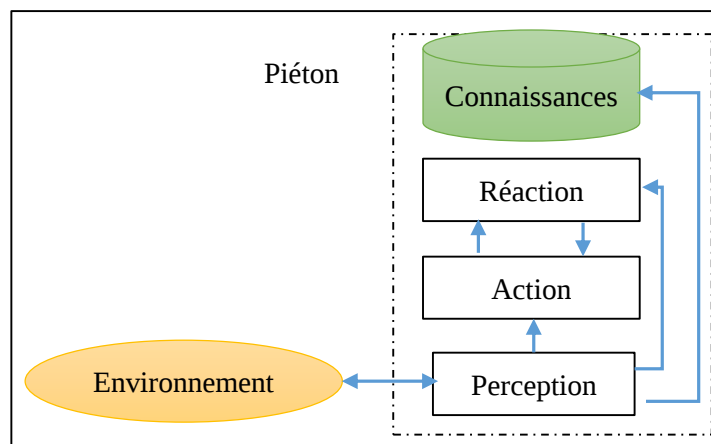


Figure 4.4 : Vue générale du modèle de décision.

Leurs classifications aboutissent à un certain nombre de classes linguistiques (catégories), qui sont utilisées pour couvrir le discours de l'univers. Pour ce modèle intelligent structuré, la précision et la complexité temporelle augmentent de façon exponentielle à mesure que le nombre de classes augmente. Ainsi, nous choisissons un certain nombre de classes linguistiques pour représenter les variables d'état afin de compenser l'efficacité et la précision du calcul (voir figure 4.5).

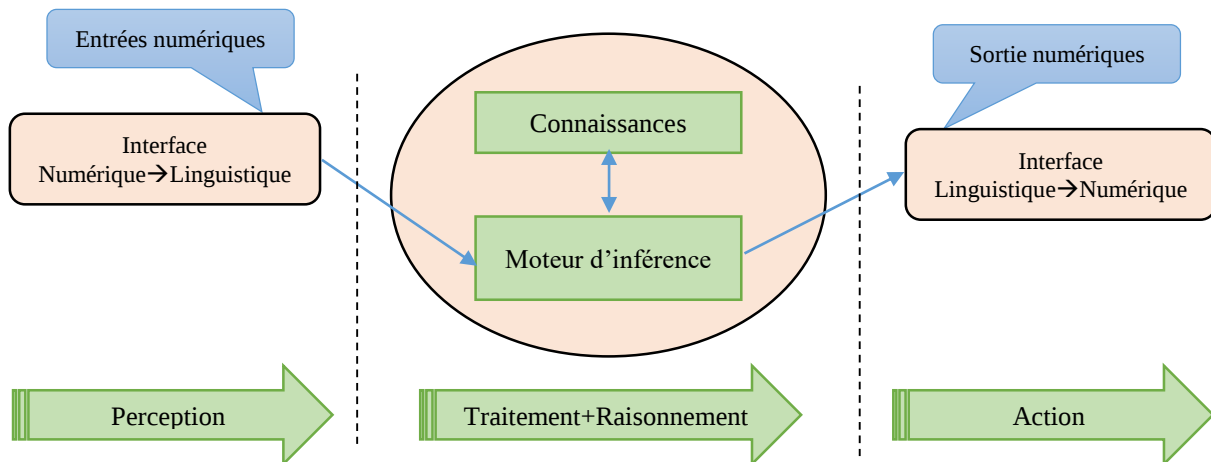


Figure 4.5 : Moteur d'inférence pour chaque piéton.

D'après l'expérience humaine, les distances sont représentées par deux catégories (ensembles de connaissances) : "*Proche*" et "*Lointain*". Un piéton percevra n'importe quelle position dans l'espace physique comme une distance proche si la distance entre sa position et cette position est comprise entre 0 et 80 unités, et c'est une distance lointaine si elle est dans l'intervalle [100, distance visuelle] (voir figure 4.6).

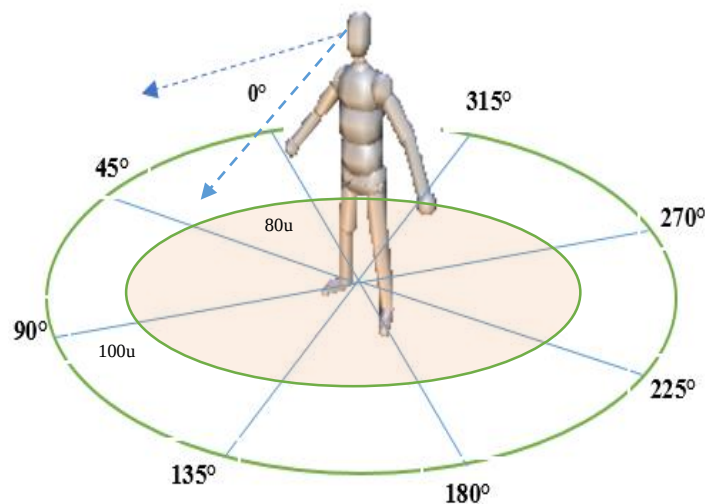


Figure 4.6 : les champs visuels de piéton

Le code Python suivant construira un classificateur Perceptron multicouche, nommé *DistanceModel*, qui est entraîné avec des données spécifiées pour que chaque piéton perçoive une valeur de distance “*d*” en appelant la méthode *perceiveDistance* (*self*, *d*), où *self* est la référence à l'objet piéton.

```

Near = np.arange(0, 80, 0.1)
Far = np.arange(100, VisualDistance, 0.1)
DistanceX = np.reshape(np.concatenate([Near, Far]), (-1, 1))
DistanceY = np.reshape(np.concatenate([np.full((len(Near), 1), [0]),
np.full((len(Far), 1), [1])]), len(Near)+len(Far), ))
DistanceModel =
MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(15, ),
              random_state=1).fit(DistanceX, DistanceY)

```

Pour calculer l'angle de direction nous utilisons la fonction tangente permet de déterminer la longueur d'un côté de l'angle droit connaissant un angle et la longueur d'un des autres côtés. Ceci est utilisé pour la mesure optique de longueurs. Par exemple, avec un télémètre à parallaxe, la distance *D* d'un objet observé est déterminée à partir de la distance *L* séparant entre deux lunettes d'observation et de l'angle θ d'observation, déterminé en faisant coïncider les images des deux lunettes en faisant pivoter un miroir : $D = L \cdot \tan \alpha$

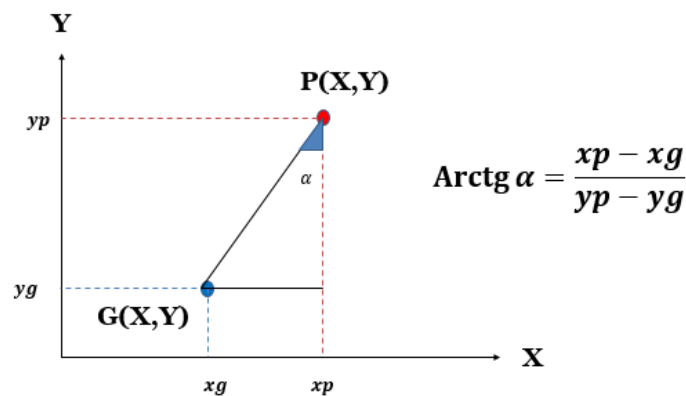


Figure 4.7 : Représentation géométrique de valeur de fonction $\tan(\alpha)$.

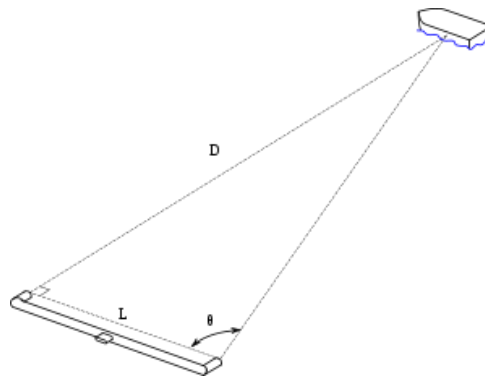


Figure 4.8 : Principe de la mesure de distance par la fonction tangente

L'angle de direction (l'angle entre la position du piéton et la position du but (sortie)) peut être perçu par le piéton (décideur) comme l'une des classes (ensembles de connaissances) : “Zéro”, “SmallPos”, “LargePos”, “SmallNeg”, ou “LargeNeg”, où, “Neg” et “Pos” font tourner les piétons dans les directions gauche et droite, respectivement. De la même manière, leurs intervalles sont définis dans le code Python suivant (voir figure 4.9).

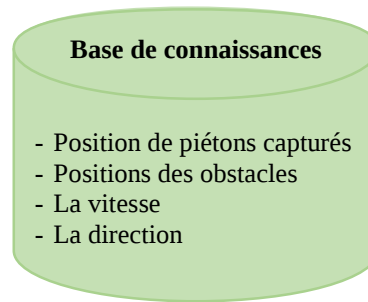


Figure 4.9 Base de connaissances

Un classificateur Perceptron multicouche, appelé *DirectionModel*, est construit et entraîné avec ces données, où *DirectionX* est une concaténation de tous les intervalles de valeurs de direction et *DirectionY* est la concaténation des codes de catégorie (classe).

```
Zero = np.arange(-10, 10, 1)
SmallPos = np.arange(15, 110, 1)
LargePos = np.arange(120, 180, 1)
SmallNeg = np.arange(-110, -15, 1)
LargeNeg = np.arange(-180, -120, 1)
DirectionX = np.reshape(np.concatenate([Zero, SmallPos, SmallNeg,
                                         LargePos, LargeNeg]), (-1, 1))
DirectionY = np.concatenate([np.full((len(Zero), 3), [0,0,0]),
                              np.full((len(SmallPos), 3), [0,0,1]),
                              np.full((len(SmallNeg), 3), [0,1,0]),
                              np.full((len(LargePos), 3), [0,1,1]),
                              np.full((len(LargeNeg), 3), [1,0,0])])
DirectionModel = MLPClassifier(solver='lbfgs',
                              alpha=1e-5, max_iter=1000,
                              hidden_layer_sizes=(15, ),
                              random_state=1).fit(DirectionX, DirectionY)
```

Les piétons peuvent percevoir la vitesse de déplacement d'un type venant dans la direction opposée comme l'une des catégories suivantes “*Stop*”, “*Slow*” et “*Fast*”. Un classificateur Perceptron multicouche, appelé *SpeedModel*, est construit et entraîné avec les données, où *SpeedX* est une concaténation de tous les intervalles de valeurs de vitesse et *SpeedY* est la concaténation des codes de catégorie (classe).

```
Stop = np.array([0])
Slow = np.arange(1, 20, 0.1)
Fast = np.arange(30, 50, 0.1)
SpeedX = np.reshape(np.concatenate([Stop, Slow, Fast]), (-1, 1))
SpeedY = np.concatenate([np.full((len(Stop), 2), [0,0]),
                              np.full((len(Slow), 2), [0,1]),
                              np.full((len(Fast), 2), [1,0])])
SpeedModel = MLPClassifier(solver='lbfgs', alpha=1e-5,
                           hidden_layer_sizes=(15, ), random_state=1).fit(SpeedX, SpeedY)
```

Pendant la simulation, en plus des fonctions de prédiction du modèle de direction *DirectionModel.predict()* et le modèle de vitesse *SpeedModel.predict()*, il est nécessaire de définir les fonctions *direction_crisp()* et *speed_crisp()* pour calculer la valeur *apparent (crisp)* de l'angle de direction et de la vitesse de classe correspondants, respectivement en utilisant la méthode de sélection aléatoire pour mettre à jour l'état du piéton.

```
def direction_crisp(d):
    if d == "Zero": return random.choice(Zero)
    if d == "SmallPos": return random.choice(SmallPos)
    if d == "SmallNeg": return random.choice(SmallNeg)
    if d == "LargePos": return random.choice(LargePos)
    return random.choice(LargeNeg)
def speed_crisp(s):
    if s == "Stop": return random.choice(Stop)
    if s == "Slow": return random.choice(Slow)
    return random.choice(Fast)
```

Un piéton "*p*" se déplaçant dans l'espace physique "*s*" est un objet de la classe Python "*Pedestrian*", qui se caractérise par des informations (voir figure 4.10) :

- Une localisation : qui sont un point avec des coordonnées (x_p , y_p), pour suivre ses mouvements.
- Des informations de direction : qui sont l'angle entre la position du piéton et la position (sortie) du but.
- La vitesse de déplacement : pour suivre sa vitesse de déplacement d'une position à l'autre.

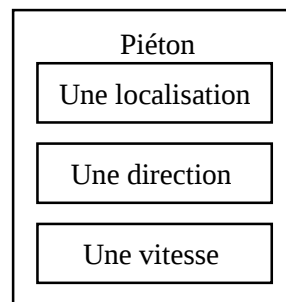


Figure 4.10 : les caractéristiques de piétons

```
class Pedestrian:
    def __init__(self, p):
        self.position = p
        self.direction =
        self.perceiveDirection(self.get_angle(p, goal_position))
        self.speed = "Stop"
        self.MovementsNbre = 0
        self.MovementsDistance = 0
        self.Blocked = False
        self.GoalReached = False
```

Les informations *MovementsNbre*, *MovementsDistance* et *MovementsSpeed* sont utilisées pour suivre le nombre de mouvements (un mouvement est un changement de direction, de vitesse ou les deux), la distance des mouvements des piétons et leurs vitesses totales, respectivement, pour atteindre la position d'arrivée (objectif). Ils sont utilisés pour calculer l'énergie piétonne pour sortir de l'espace physique. L'élément de données de l'information bloqué "*Blocked*" indiquera

si le piéton est dans un état de blocage en raison d'obstacles physiques l'empêchant de se déplacer et l'information objectif atteint "*ReachedGoal*" indiquera si le piéton a déjà atteint l'objectif.

Les objets de piétons (on les appelle aussi décideurs) peuvent mettre à jour leurs membres de données lors des étapes de simulation du modèle en appelant la méthode *update member (self, sp, speed)* pour changer leurs positions, directions et vitesses, où le paramètre *self* est le décideur, "*sp*" est la position du secteur visuel pour établir la direction du mouvement vers l'objectif et la vitesse de déplacement spécifique du piéton (voir figure 4.11).

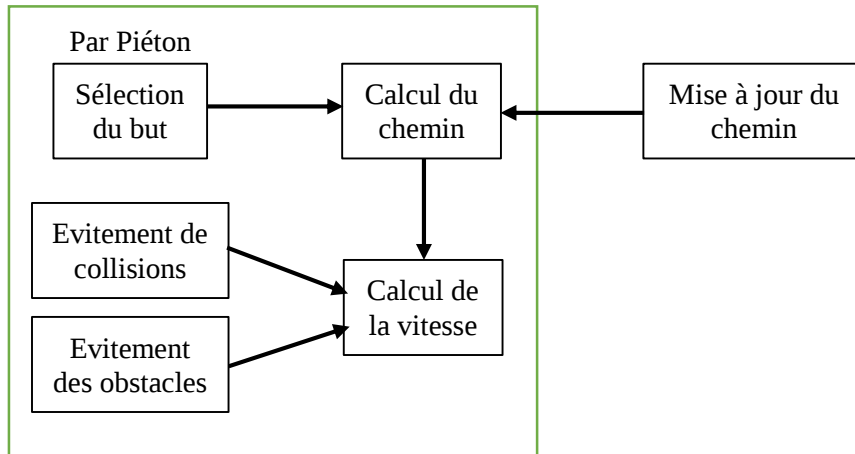


Figure 4.11 Architecture de navigation pour chaque piéton (vue schématique)

Les piétons peuvent avoir des perceptions des données de distance, de direction et de vitesse à l'aide de leurs méthodes membres qui perçoivent la distance (*perceiveDistance(self, d)*), percevoir la direction (*perceiveDirection(self, a)*) et la vitesse (*perceiveSpeed(self, s)*), qui appellent respectivement les fonctions de prédiction du modèle correspondantes.

Pour compléter la description du modèle, nous devons créer un ensemble de piétons sans chevauchement entre eux. Pour une simulation, on appelle la fonction *generate_pedestrians(n)*, pour générer *n* piétons, où le *i*-ème piéton occupe la position *positions[i]* dans l'espace '*s*'.

```

def generate_pedestrians(n):
    Pedestrians = []
    if n > len(positions): n = len(positions)
    for i in range(n):
        Pedestrians.append(Pedestrian(positions[i]))
    return Pedestrians

```

Le champ visuel d'un piéton est défini comme une zone sous la forme d'un cercle avec un point central, qui est la position du piéton, les unités de rayon *R* et l'angle central (*Central_Angle* = 360°). Cette région est divisée en *n* secteurs similaires mais non congruents occupant les angles centraux du *Sector_Angle_i*, *i* = 1, ..., *n* ($\sum_{i=1, \dots, n} \text{Sector_Angle}_i = \text{Central_Angle}$).

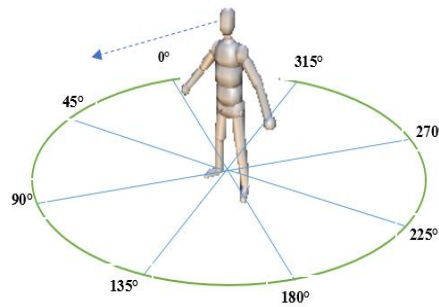


Figure 4.12 : les champs visuels de piéton

Le nombre de secteurs constitutifs du champ visuel est déterminé par un compromis entre la précision et la complexité du modèle. Avec une méthode de représentation spatiale prédéterminée, les états de mouvement des piétons peuvent être mis à jour en combinant les détails des objectifs et les informations environnementales déjà perçues dans tous les secteurs du champ visuel (voir figure 4.12).

3.3 Les comportements des piétons

La notion de comportement est fondamentale en évacuation, car elle définit les décisions des individus et influence par conséquent fortement les mouvements individuels et collectifs. Ceci est encore plus vrai dans le cas dangereux, où les effets dus à l'urgence et la panique qu'il engendre sont multipliés.

L'intention des piétons est d'évacuer les zones dangereuses et de survivre. Dans les sections 3.1 et 3.2, nous avons proposé un modèle structurel basé sur l'intelligence artificiel IA du comportement des piétons dans une foule. Les hypothèses suivantes sont formulées afin de simplifier et de clarifier le modèle proposé :

- 1) Tous les piétons ne connaissent pas les informations globales sur les environnements, mais ils connaissent les informations régionales dans leur champ visuel et les détails de leurs objectifs.
- 2) Le piéton peut décider de basculer entre deux états prédéterminés quelconques dans un temps de relaxation ignoré.

La structure générale du modèle basé sur l'intelligence artificielle consiste en un système de prédiction, qui est utilisé pour décrire une combinaison de comportement d'évitement d'obstacles et de comportement de recherche d'objectifs. Les informations d'entrée comprennent principalement des informations sur les obstacles, des informations sur les piétons et des informations sur les objectifs (voir figure 4.13).

Par exemple, la distance la plus proche entre le décideur (piéton) et les obstacles domine le comportement d'évitement d'obstacles, et la distance et la vitesse du piéton qui a marché dans la direction opposée au décideur influencent considérablement le comportement de recherche de chemin (objectif).

Les informations de sortie du système sont les angles de braquage (directions) et les vitesses de déplacement, qui sont utilisés pour déterminer les états du mouvement final (voir figure 4.13).

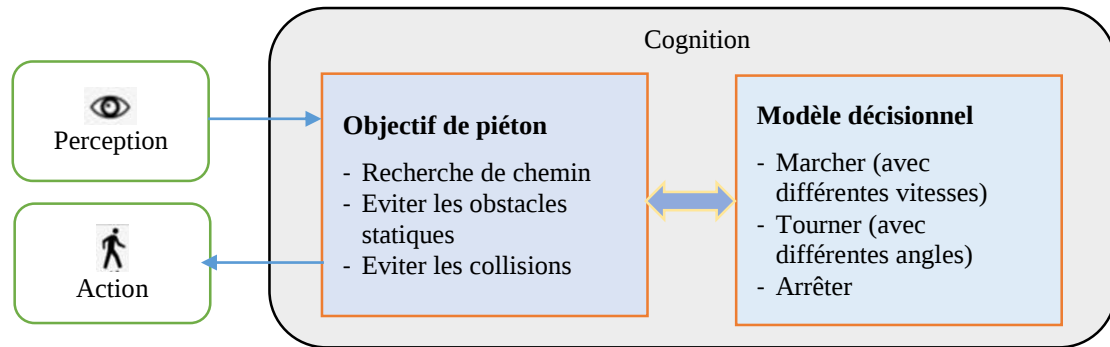


Figure 4.13 : Les comportements de piéton

Les comportements d'évacuation de la foule sont déterminés par une combinaison de comportements locaux d'évitement d'obstacles et de comportements de recherche d'objectifs. Les besoins des piétons évitent les collisions avec son espèce, les obstacles et les frontières qui apparaissent dans son champ visuel, tout en se déplaçant vers la direction du but avec la vitesse de déplacement appropriée.

Il est évident que les comportements des piétons sont principalement affectés par l'emplacement de l'objectif et la répartition de leur espèce et des obstacles. Tous les piétons sont censés connaître l'emplacement de l'objectif à l'avance. Nous spécifions le champ visuel, qui a la forme d'un cercle avec un rayon d'unités de rayon (par défaut, il est égal à 50 unités).

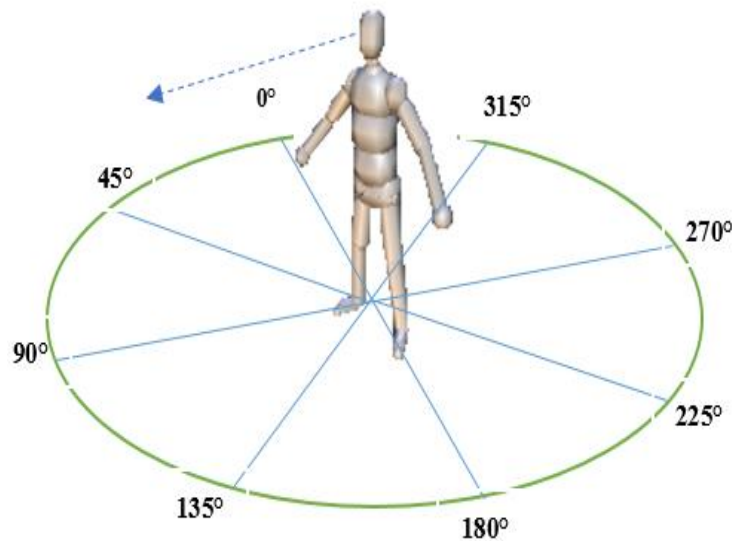


Figure 4.14 : les champs visuels

L'angle central du champ visuel du piéton "*Central_Angle*" est subdivisé en angles de secteur identiques avec la même valeur d'angle *Sector_Angle*, alors nous avons $n = \text{Central_Angle} / \text{Sector_Angle}$ qui composent le champ visuel. Chaque secteur définit une direction de la position piétonne à la position de secteur " s_p ".

En général, le processus décisionnel des piétons peut être décrit comme suit. Tout d'abord, le décideur (le piéton) balaie son champ visuel, secteur par secteur, et choisit le secteur à traverser pour atteindre la sortie (goal) en fonction de considérations personnelles (voir figure 4.14).

Ensuite, il marche dans la direction de la position de secteur sélectionnée " s_p " pour atteindre l'objectif et évite les collisions avec les piétons et les obstacles frontaux avec une vitesse et une direction de mouvement appropriées.

Les entrées de l'algorithme d'évacuation des foules sont les distances piéton-obstacle les plus proches dans chaque secteur et les sorties sont l'angle de braquage et la vitesse de déplacement. L'angle de braquage final et la vitesse de déplacement du décideur sont calculés en utilisant une combinaison de comportement d'évitement d'obstacles et de comportement de recherche de but (goal), avec le décideur évitera d'abord l'obstacle devant lui, au lieu de chercher le but (goal).

Ainsi, le comportement d'évitement d'obstacles est très important que le comportement de recherche d'objectifs. Par conséquent, les piétons peuvent atteindre l'objectif avec succès, tout en évitant les obstacles et les autres piétons apparaissant sur leur chemin.

Pour contrôler le mouvement des piétons, les informations suivantes doivent être connues. La position (x_p, y_p) du décideur (piéton) "p", un but "g", qui est situé à la position (x_g, y_g) représentant l'endroit où les piétons veulent atteindre en s et est situé à l'angle but y_g (l'angle entre la position piétonne "p" et la position de but "g", qui est appelé l'angle de direction du piéton) et qui se trouve à la distance but d_g de la position du piéton "p".

Lors de la simulation du modèle, la méthode *get_distance_perceptions(self)* (self est la référence du piéton), sera appelée par la méthode *behavior(self)* définie ci-dessous, pour obtenir la perception du piéton sur les distances minimales entre son emplacement et les obstacles environnants, les frontières et les autres piétons venant dans des directions opposées dans différents secteurs de son champ visuel.

Ces perceptions de distance sont enregistrées dans une liste appelée *perceptions_distance*. Sa longueur maximale est le nombre de secteurs, qui est égal *Central_Angle/Sector_Angle* (le secteur d'angle du *Sector_Angle* est égal, par défaut, à 45°).

Après l'exécution de *get_distance_perceptions(self)*, la liste *distance_perceptions* contient des informations sur les distances perçues par le piéton lui-même dans chaque secteur de son champ visuel.

Toutes les informations sur le secteur sont des triplets, le premier élément étant la position du secteur et le deuxième élément étant la distance minimale perçue entre la position du piéton et tous les obstacles, bordures et autres piétons observés à travers la fenêtre du champ visuel du secteur.

Le troisième élément d'information dans le triple, il est utilisé pour déterminer si l'obstacle est un piéton marchant dans la direction opposée ou un autre obstacle. Lorsque ces informations sont collectées pour le piéton actuel (décideur), il prendra la décision de changer de position avec la vitesse et l'angle de direction appropriés par rapport à sa stratégie.

La méthode *behavior (self)* est définie ci-dessous pour changer l'état du piéton pour le faire passer d'un endroit à un autre jusqu'à ce qu'il atteigne la sortie (goal). Le piéton actuel peut également entrer dans un état de blocage (arrêter de bouger) s'il se trouve devant des obstacles physiques dans tous les secteurs du champ visuel.

```

def get_distance_perceptions(self):
    sector_angle = 0
    obstacles_borders = obstacles + borders
    distance_perceptions = []
    while sector_angle < Central_Angle:
        sector_pos = Point(
            (
                self.position.x+Radius*math.sin(sector_angle),
                self.position.y+Radius*math.cos(sector_angle)
            ))
        if (sector_pos.x < -SpaceWidth+Radius or sector_pos.x >
            SpaceWidth-Radius or
            sector_pos.y < -SpaceLength+Radius or sector_pos.y >
            SpaceLength-Radius):
            sector_angle += Sector_Angle
            continue
        MinDistance = VisualDistance
        for ob in range(len(obstacles_borders)):
            d = Poly(obstacles_borders[ob]).boundary.distance
                (sector_pos)
            if d > Poly(obstacles_borders[ob]).boundary.distance
                (self.position): continue
            if d < MinDistance: MinDistance, kind = d, -1
        for k in range(len(Pedestrians)):
            if self != Pedestrians[k]: # and Pedestrians[k].GoalReached
                == False:
                d = sector_pos.distance(Pedestrians[k].position)
                if d > self.position.distance(Pedestrians[k].position):
                    continue
                if d < MinDistance: MinDistance, kind = d, k
        if MinDistance < VisualDistance:
            distance_perceptions.append((sector_pos,
                self.perceiveDistance(MinDistance), kind))
        else:
            distance_perceptions.append((sector_pos, "Far", -1))
        sector_angle += Sector_Angle
    return distance_perceptions

```

Pour déterminer le mouvement de chaque piéton, nous appelons la fonction *get_distance_perceptions (self)* définie ci-dessus pour obtenir des informations décisionnelles pour le piéton lui-même.

La méthode *behavior(self)* combine le comportement d'évitement d'obstacles et le comportement de recherche d'un but (goal) pour obtenir de nouvelles positions.

Ces nouveaux postes sont candidats au déplacement du piéton (décideur). La sélection de la meilleure nouvelle position est basée sur une formule donnant la petite pondération entre la distance et l'angle de direction du but en utilisant deux paramètres de pondération alpha et bêta pour définir la préférence entre eux.

Tout d'abord, le piéton lui-même (le décideur) vérifie s'il est proche d'une espèce, qui a déjà atteint l'objectif, puis son état change au fur et à mesure qu'il a atteint l'objectif (sortie).

Sinon, il vérifiera si tous les obstacles/frontières/piétons sont loin de lui pour décider de suivre le chemin de recherche du but en appelant la méthode *goal_seeking_behavior (self, distance_perceptions)*.

Une sorte de comportement global connu sous le nom de "comportement de recherche des buts" est la propension du décideur à se déplacer constamment dans la direction de son objectif, quel que soit l'environnement qui l'entoure. Il est défini par l'angle de but y_g (qui peut appartenir à l'une des classes *LargeNeg*, *SmallNeg*, *Zero*, *SmallPos* et *LargePos*) et la distance de but d_g (qui peut appartenir à l'une des classes *Near* ou *Far*).

```
def behavior(self):
    if not self.GoalReached and not self.Blocked:
        distance_perceptions = self.get_distance_perceptions()
        if [k for (x, (sp, od, k)) in enumerate(distance_perceptions)
            if od == "Near" and k != -1 and Pedestrians[k].GoalReached]
            != []: self.GoalReached = True
        else:
            new_positions =
self.goal_seeking_behavior(distance_perceptions)
            if new_positions == []:
                # The obstacle-avoiding behavior
                new_positions =
self.obstacle_avoiding_behavior(distance_perceptions)
            if new_positions != []:
                new_positions.sort(key=lambda x:x.get('index'))
                if new_positions[0].get('speed') != "Stop":
                    self.update(new_positions[0].get('sp'),
new_positions[0].get('speed'))
                else: self.Blocked = True
```

La formulation d'un comportement global de recherche d'objectifs motive les piétons à marcher dans la direction de l'objectif. Le décideur réduit sa vitesse et se tourne brusquement vers la cible sans la manquer lorsque le piéton s'approche de la sortie mais ne lui fait pas face. Au contraire, face à la cible, il se déplace librement dans la direction de la cible à une vitesse élevée (rapide).

Avec cette méthode, le décideur scanne d'abord les secteurs, qui indiquent les obstacles internes éloignés, les obstacles externes (limites) et les piétons bloqués, auquel cas il ajoute à la liste des décisions "gsb" (comportement de recherche d'objectifs) pour se déplacer vers une direction spécifiée par l'expression *abs (self.obtenir_angle(soi-même.position, sp) - soi.obtenir_angle(soi-même.position, goal_position))* avec une vitesse de déplacement rapide.

Si l'obstacle est de quelque nature que ce soit, le décideur vérifie si ce type va dans la direction opposée et ajoute ensuite aux décisions le même angle de direction de mouvement mais arrête de bouger sinon il ajoutera rapidement la vitesse de déplacement.

Si le décideur ne trouve pas de position de secteur (la liste *new_positions* est vide) pour rechercher le but, ce qui signifie qu'il est entouré d'obstacles de toutes les directions définies par les secteurs de son champ visuel, alors le comportement d'évitement d'obstacles doit être invoqué, qui est défini avec la méthode *obstacle_avoiding_behavior (self, distance_perceptions)*, pour éviter les obstacles frontaux car les distances entre le décideur et les obstacles sont proches à la portée locale.

Cette méthode est appelée parce que tous les obstacles sont proches du piéton, auquel cas il vérifie si l'un des obstacles est quelconque. Dans le code, le décideur ajoute à la liste "*pab*" (comportement d'évitement des piétons (obstacles)), la décision de suivre la direction, qui est calculée par l'expression suivante. La valeur absolue de l'angle entre la position du secteur (à

partir de laquelle le décideur a observé une sorte de marche dans sa direction) et le but moins l'angle entre le décideur et le but, à une vitesse lente en mouvement.

Si le type va dans la direction opposée, la résolution à ajouter à la même direction que ci-dessus mais avec un arrêt en mouvement jusqu'aux prochaines étapes de simulation.

```
def goal_seeking_behavior(self, distance_perceptions):
    gsb = [(sp, sp.distance(goal_position),
        abs(self.get_angle(self.position, sp) -
        self.get_angle(self.position, goal_position)), "Fast")
        for (x, (sp, od, k)) in enumerate(distance_perceptions) if od
        == "Far" and (k == -1 or (k != -1 and Pedestrians[k].Blocked))]
    gsb = gsb + [(sp, sp.distance(goal_position),
        abs(self.get_angle(self.position, sp) -
        self.get_angle(self.position, goal_position)), "Slow")
        for (x, (sp, od, k)) in enumerate(distance_perceptions)
        if od == "Far" and k != -1 and not Pedestrians[k].Blocked
        and self.opposite(self.perceiveDirection(self.get_angle
        (sp, goal_position)), Pedestrians[k].direction)]
    gsb = gsb + [(sp, sp.distance(goal_position),
        abs(self.get_angle(self.position, sp) -
        self.get_angle(self.position, goal_position)), "Fast")
        for (x, (sp, od, k)) in enumerate(distance_perceptions)
        if od == "Far" and k != -1 and not Pedestrians[k].Blocked
        and not self.opposite(self.perceiveDirection
        (self.get_angle(sp, goal_position)),
        Pedestrians[k].direction)]
    return [{'index': (alpha * d + beta * a) / (alpha + beta), 'sp': sp,
        'speed': speed}
        for (i, (sp, d, a, speed)) in enumerate(gsb)]
```

Si aucune sorte n'est un obstacle, alors tous les obstacles sont des obstacles internes ou externes, ce qui met le décideur dans un état de blocage (voir la méthode de comportement) et ne peut alors pas être évacué (c'est-à-dire l'accès à la sortie).

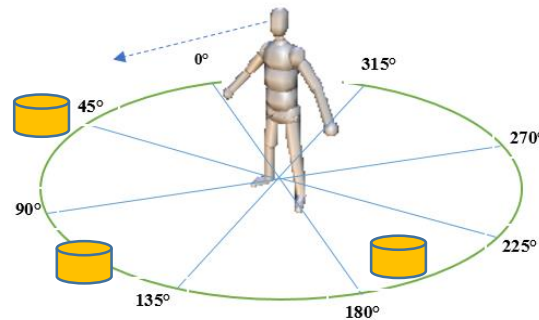


Figure 4.15 : comportement d'évitement des obstacles.

La définition du comportement d'évitement d'obstacles est la tendance d'un piéton à changer de direction progressivement et en douceur plutôt que brusquement. Pour cette raison, s'il y a une distance piéton-obstacle identique dans chaque secteur, le code est construit de manière à ce que les piétons aillent généralement dans la même direction (voir figure 4.15).

Ce modèle d'évacuation des foules est décrit de manière à faciliter l'analyse de ses résultats de simulation en utilisant une méthode permettant de modifier des paramètres tels que les

dimensions de l'espace, le nombre de secteurs du champ visuel, les paramètres de pondération, le nombre d'obstacles, la forme de chaque obstacle, nombre de piétons, etc. Certaines valeurs de ces paramètres peuvent rendre le modèle plus complexe et sa simulation plus longue et encombrante.

```
def obstacle_avoiding_behavior(self, distance_perceptions):
    oab = [(sp, k) for (x, (sp, od, k)) in
            enumerate(distance_perceptions)
            if od == "Near" and k != -1 and not Pedestrians[k].Blocked]
    new_positions = []
    if oab != []:
        pab = [(sp, sp.distance(goal_position),
                                abs(self.get_angle(self.position, sp) - self.get_angle(
                                    self.position, goal_position))), "Slow"]
        for (x, (sp, k)) in enumerate(oab) if not
        Pedestrians[k].GoalReached and
        not self.opposite(self.perceiveDirection(self.get_angle(sp,
                                                                goal_position)),
                          Pedestrians[k].direction)]
        pab = pab + [(sp, sp.distance(goal_position),
                                abs(self.get_angle(self.position, sp) - self.get_angle(
                                    self.position, goal_position))), "Stop"]
        for (x, (sp, k)) in enumerate(oab) if not
        Pedestrians[k].GoalReached and
        self.opposite(self.perceiveDirection(self.get_angle(sp, goal_position)),
                      Pedestrians[k].direction)]
        new_positions = [{'index': (alpha * d + beta * a) / (alpha + beta),
                          'sp': sp, 'speed': speed}
                          for (i, (sp, d, a, speed)) in enumerate(pab)]
    return new_positions
```

4. Conclusion

Nous avons illustré dans ce chapitre notre modèle avec une description détaillée sur tous les composants et les attributs de notre simulateur. Nous avons essayé de mettre en évidence une modélisation d'animation comportementale de piétons composant la foule qui utilise l'intelligence artificielle, où des connaissances sur le comportement des piétons ont été extraites à l'aide de fouille de données (KDD) à partir des données synthétiques, sur lequel repose notre étude, dont le but est de diminuer la complexité de navigation d'un ensemble de piétons dans un environnement totalement inconnu. Chaque piéton est doté de faculté d'autonomie et d'intelligence de réagir aux événements perçus, et de s'interagir dans le groupe. Pour Ces raisons, nous avons choisi d'adopter pour un piéton une architecture à base d'agent et assurer les quatre modules: module de perception, module de réaction, et un dernier module pour l'action.

Chapitre 5 : Analyse de résultats de simulation

1. Introduction

Ce chapitre a pour but de valider expérimentalement, à travers un ensemble de simulations, les résultats théoriques du chapitre précédent. Nous disposons des piétons autonomes muni d'un moteur d'inférence intelligent capable de percevoir l'environnement par le piéton sous forme de connaissances qu'il peut par la suite utiliser pour étendre son registre comportemental dans le cadre de la sélection de l'action. Nous avons réalisé plusieurs expérimentations pour illustrer les différentes caractéristiques du modèle. Pour cela, nous avons soumis des piétons à des situations simples et dangereuses que nous avons définies au préalable. Ce cinquième et dernier chapitre a pour objectif de valider et simuler les comportements de piétons avec différentes échelles (nombre de piétons), puis faire des analyses sur les résultats de simulation mais avant tout nous avons fait une étude sur la relation entre la vitesse de marche et la densité ainsi que quelques notions sur les comportements de piétons. A la fin de ce chapitre plusieurs types de graphiques en tant que résultats d'analyses des données de sortie sont présentés.

2. Vitesse de mouvement d'individu

La vitesse de mouvement de piéton a fait l'objet de différents travaux scientifiques. D'après les travaux de McNeil [116], la vitesse de marche préférentielle d'un piéton (appelée vitesse de marche de confort ou vitesse de marche libre) est celle qui permet de minimiser l'énergie métabolique dépensée au cours du mouvement. Cette énergie métabolique peut être calculée grâce à des modèles mathématiques prenant en compte les propriétés physiologiques des muscles et des tendons, mais aussi la masse de piéton ou encore la force exercée par ses pieds sur le sol.

Continent	vitesse moyenne de marche
Europe	1,41 m/s
États-Unis	1,34 m/s
Australie	1,44 m/s
Asie	1,24 m/s

Tableau 5.1 : Vitesses moyennes de marche dans différents Continents [130]

Ainsi, la vitesse de marche libre est une grandeur qui peut varier de façon significative d'un piéton à l'autre, en fonction :

- Du sexe : les hommes se déplacent en moyenne plus vite que les femmes.
- De l'âge : un adulte se déplace en moyenne plus rapidement qu'un enfant ou qu'une personne âgée.
- De sa destination et de sa familiarité avec son trajet : une personne se rendant à son travail se déplace plus rapidement qu'un touriste ou qu'un promeneur.
- De sa culture : les vitesses de marche moyennes varient beaucoup selon les pays.

3. Vitesse de mouvement et densité

Il semble clair que, dans le cas d'un mouvement de foule, les piétons se trouvent obligés de réduire leur vitesse du fait de la proximité immédiate des autres piétons. Dans ce cas, l'occupation d'un compartiment peut être quantifiée par une grandeur macroscopique : la

densité de piétons, qui s'exprime généralement en piétons par mètre carré (m^{-2}). Il convient alors de s'intéresser aux variations de la vitesse individuelle de marche en fonction de la densité de piétons. Ce point a fait l'objet de nombreuses études. Fruin [117] estime que la vitesse de marche libre n'est atteignable que dans le cas où la densité environnante est inférieure à la valeur de $0,4 \text{ m}^{-2}$ [118].

L'évolution du débit de piétons en fonction de la densité est généralement représentée sur un diagramme fondamental comme celui de la figure 5.1.

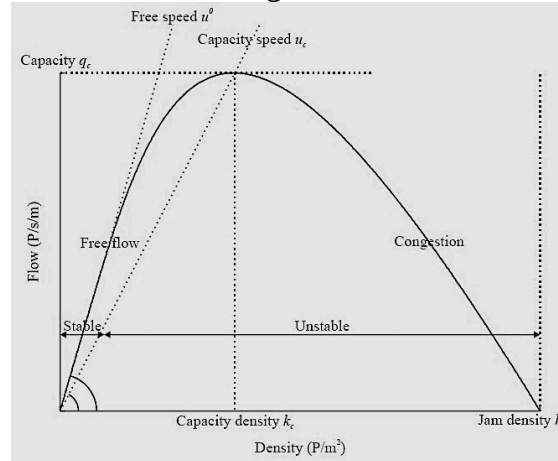


Figure 5.1 : Relation débit-densité pour la marche de piéton

On observe bien sur cette courbe la présence d'une densité seuil ρ_c au-delà de laquelle le débit (flow) est ralenti. On observe également l'existence d'une densité critique ρ_c qui annule le débit.

Cette densité critique correspond à une situation de saturation dans laquelle les individus ne bougent plus. Les valeurs de ρ_c trouvées dans la littérature varient de $3,9 \text{ m}^{-2}$ [119] à $6,25 \text{ m}^{-2}$. On retiendra la valeur de Weidmann [113], égale à $5,4 \text{ m}^{-2}$ et qui représente donc un bon compromis dans cet intervalle de valeurs. On notera également que cette valeur ρ_c est plus faible dans un escalier, et ne peut guère y dépasser 3 m^{-2} [20].

En terme de vitesse, les différentes études menées aboutissent à des lois densité-vitesse décroissantes se rapprochant de lois linéaires du type :

$$V(\rho) = V_0 - a \rho$$

où V_0 est la vitesse individuelle libre et $V(\rho)$ la vitesse individuelle effective. La figure 5.2 représente différentes lois densité-vitesse issues de la littérature. On y trouve notamment les lois écrites par Fruin [117], Weidmann [113], Virkler et Elayadath [121], Older [119], Sarkar et Janardhan [122] et Tanaboriboon [123].

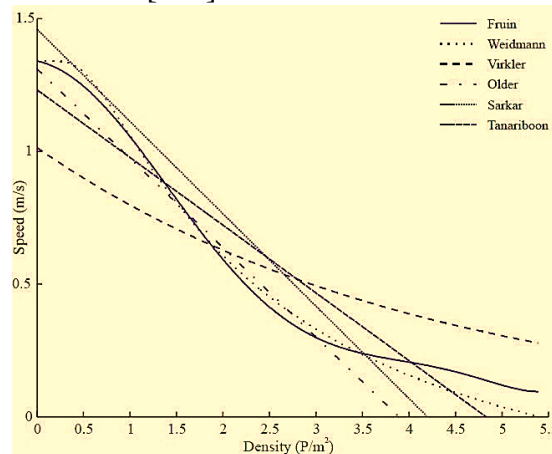


Figure 5.2 : Relations vitesse-densité

4. Notions de comportement

La notion de comportement est fondamentale dans la marche des piétons, car elle définit les décisions des individus et influence par conséquent fortement les mouvements individuels et collectifs. Ceci est encore plus vrai dans le cas de l'urgence et la panique qu'il engendre sont multipliés. Différentes études sur le comportement des piétons (généralement dans les milieux encombrés) ont permis de dégager des caractéristiques générales sur le comportement.

1. Un piéton se déplace de préférence en ligne droite afin de minimiser sa distance à parcourir. Il évite les changements de direction quand cela est possible.
2. Un piéton adapte son mouvement (trajectoire et vitesse) en fonction de son objectif à atteindre d'une part, et de la densité de personnes qu'il peut percevoir dans son champ de vision [124]. Ainsi, si le chemin le plus court est encombré (forte densité), il préférera contourner la zone encombrée et maintenir sa vitesse de confort plutôt que ralentir (si cette option lui semble plus rapide).
3. Un piéton cherche à conserver autour de lui un certain espace et tend à maintenir avec les autres individus une distance sociale minimale, qui dépend de la densité.
4. Dans le cas d'un engorgement ou d'une forte densité, une personne seule peut chercher à se faufiler entre les foules pour avancer plus vite [125].
5. Un piéton évite de faire demi-tour et de marcher à contrecourant dans un lieu encombré. Toutefois, lorsque plusieurs piétons ou plusieurs groupes se déplacent à contrecourant, on observe la formation spontanée de lignes de piétons se dirigeant dans la même direction, ce qui contribue à la fluidification du trafic : on parle de phénomène d'auto-organisation. Ce phénomène a été mis en évidence dans plusieurs études [126].
6. Lorsqu'une foule dense tente de traverser un passage étroit (une porte par exemple), les individus ont tendance à contourner le bouchon pour passer sur le côté, tout en exerçant une pression sur le bouchon.
7. Une foule n'est pas constituée uniquement d'entités individuelles, mais également de groupes d'individus hiérarchisés (famille) ou non (collègues, amis). Ces groupes ont tendance à se comporter comme une entité à part entière, et à tenter de se regrouper quand ils se trouvent séparés [127].
8. Dans le cas où une ouverture est traversée par deux foules se déplaçant en sens contraires, une alternance des sens de circulation à travers la porte se forme spontanément, ce qui engendre un phénomène d'oscillation [128].
9. Lorsque des piétons tournent dans un angle, ils ralentissent leur allure, ce qui accroît la densité de piétons au niveau de l'angle. Ce phénomène est connu sous le nom de "corner hugging" [129]. Ces différents traits de comportement listés ici sont assez généraux. En particulier, il est important de vérifier que l'on retrouve ces différents comportements dans les modèles d'évacuation (notamment les points 1, 2, 4 et 6).

5. Validation du modèle et simulation

Ce modèle d'évacuation de foule doit être validé avant d'exécuter la simulation (c'est-à-dire d'exécuter le code Python). Le modèle est validé en animant le comportement d'un petit nombre de piétons dans l'espace physique pour comprendre leurs mouvements.

La fonction `do_animation()` ci-dessous peut être appelée pour générer des données d'animation à partir de la simulation du modèle, puis générer une animation en appelant la fonction partielle `animate()` à l'aide des outils fonctionnels Python.

Le résultat de l'animation est enregistré sous forme de fichier GIF animé (Graphics Interchange Format) à afficher pour vérifier les comportements des piétons.

```
def do_animation():
    fig, ax = plt.subplots(1,1)
    plt.ylim(-SpaceLength/2-200,SpaceLength/2+200)
    plt.xlim(-SpaceWidth/2-200,SpaceWidth/2+200)
    #fig.set_size_inches(5,5)
    animation_data = generate_animation_data()
    ani = FuncAnimation(fig, functools.partial(animate, ax = ax,
                                              animation_data =
                                              animation_data), frames=len(animation_data[0]), interval=500,
    repeat=False)
    # Save the animation as an animated GIF
    ani.save("animation.gif", dpi=100, writer=PillowWriter(fps=1))
    fig.savefig("animationEnd.png")
```

La fonction *generate_animation_data* (*n*) est utilisée pour générer des données d'animation de l'état initial d'un certain nombre de piétons à leur dernier état, dans lequel tous les piétons ont atteint la position cible (sortie) ou sont entrés dans l'état de blocage.

```
def generate_animation_data():
    animation_data = [[] for _ in Pedestrians]
    for p in range(len(Pedestrians)):
        animation_data[p].append(Pedestrians[p].position)
    plot_for_animation(animation_data, "animationStart.png")
    while not pedestrians_behavior_done():
        for p in range(len(Pedestrians)):
            Pedestrians[p].behavior()
            animation_data[p].append(Pedestrians[p].position)
    return animation_data
```

La fonction d'animation ci-dessous peut être appelée pour effacer l'état précédent et tracer le nouvel état après une étape pour déplacer tous les piétons. Dans les tracés, tous les piétons représentés par leurs points de localisation sont annotés avec les numéros associés.

```
Def animate(I, ax, animation_data):
    ax.clear()
    for p in range(len(Pedestrians)):
        ax.plot(animation_data[p][i].x, animation_data[p][i].y,
        color='green', label='original', marker='o')
        ax.annotate(f"({p})",
        (animation_data[p][i].x,animation_data[p][i].y),
        textcoords="offset points", xytext=(0,10), ha='center')
    plot_physical_space(ax)
```

La figure 5.3 (a) montre l'état initial de la répartition de tous les piétons (en colleur verte) annotée par leur nombre dans l'espace physique. Les obstacles sont colorés en rouge et la sortie (but) est une ligne verte en bas.

Lorsque nous ouvrons le fichier GIF, nous voyons l'animation des mouvements des piétons de cet état initial au dernier état illustré à la figure 5.3 (b). Dans ce dernier cas, on observe que tous les piétons ont atteint l'objectif. L'environnement physique des simulations est simple à l'extrême. Il s'agit d'une surface dans laquelle tous les piétons de la simulation sont libres de se déplacer.

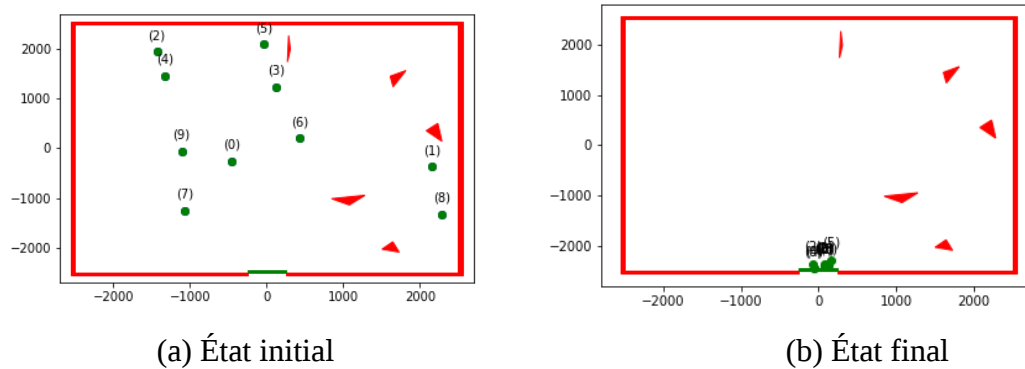


Figure 5.3 : Animation avec 8 piétons

Ce modèle d'évacuation des foules est testé de manière approfondie pour vérifier comment les piétons évitent les obstacles et les frontières, leurs types, et comment ils changent de direction et actualisent leur vitesse pour atteindre le but. Après ce processus de vérification, qui nous a permis de corriger de nombreux bugs notamment des erreurs de comportement des piétons, une simulation peut être lancée avec un grand nombre de piétons et d'obstacles.

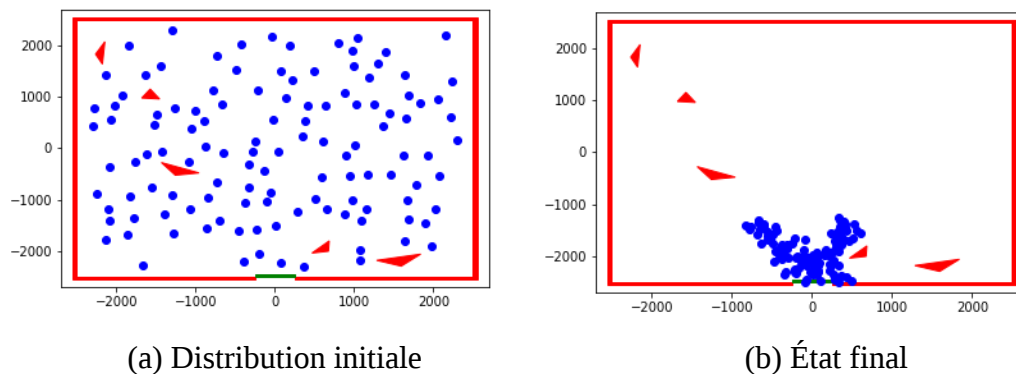


Figure 5.4 : Simulation du modèle d'évacuation des foules

Le code Python a été écrit pour exécuter la simulation, dans laquelle plusieurs résultats peuvent être suivis pour l'évaluation des performances dans la section 6. La figure 5.4 (a) montre la randomisation initiale de la foule de 100 piétons en bleu et les obstacles intérieurs et extérieurs en rouge. Pour cette simulation, chaque piéton se déplace en direction de la sortie à la vitesse requise jusqu'à l'arrêt final illustré à la figure 5.4 (b) une fois la simulation terminée. Pour l'évaluation des performances, ce modèle d'évacuation des foules a été simulé sur différentes valeurs de paramètres.

6. Expériences et évaluation des performances

L'évaluation de la performance de cette analyse d'évacuation des foules à l'aide de la simulation du modèle décrit à le chapitre 4, est basée sur une évaluation de l'évolutivité sur des foules de grandes tailles, qui doivent adhérer à la randomisation piétonne composée de foules et

d'obstacles. Par conséquent, l'évaluation mesurera les compromis entre l'évolutivité et la randomisation des piétons.

L'évaluation des performances est basée sur un ensemble de mesures ou des paramètres. La première mesure est l'efficacité de l'évacuation de la foule, qui est la capacité d'évacuer autant de piétons que possible. Le degré d'efficacité de l'évacuation des foules est mesuré en termes de pourcentage de piétons R qui atteignent l'objectif par simulation de modèle sur le nombre total de piétons N ($Efficacité = R/N * 100\%$).

L'évacuation de la foule est effective si tous les piétons qui la composent ont atteint le but (sortie).

La deuxième mesure est l'efficacité de l'évacuation des foules, qui est la capacité d'évacuer un nombre prévu de piétons avec le moins d'énergie piétonnière, qui s'exprime comme suit :

Si nous exprimons le nombre total de mouvements de piétons par NM (le mouvement des piétons est le mouvement après mise à jour de son état en direction et en vitesse, utilisé pour mesurer la dynamique des piétons et la relation vitesse-densité), la distance totale parcourue par tous les piétons pour atteindre le but par D.

Ensuite, le degré d'efficacité est $Efficacité = ((N/NM + ID/D) * Densité) * 100\%$, où ID est la distance totale idéale entre les positions initiales des piétons et la position cible, et la Densité est la densité du modèle, qui est $Densité = (N + \sum PO)/NP$, où $\sum PO$ est le nombre total de positions occupées par des obstacles internes et NP est le nombre total de positions associées par espace physique.

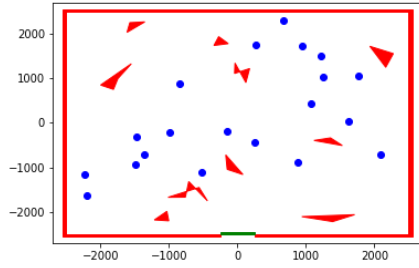
Si nous exprimons le nombre total de mouvements de piétons par NM (un mouvement de piéton est le mouvement après la mise à jour de son état de direction et de vitesse, il est utilisé pour mesurer la dynamique des piétons et la relation vitesse-densité), la distance totale parcourue par tous les piétons pour atteindre le but par D, alors le degré d'efficacité est $Efficacité = ((N/NM + ID/D) * C) * 100\%$, où ID est la distance totale idéale entre les positions initiales des piétons et la position du but, et c'est la complexité du modèle, qui est $C = (N + \sum PO)/P$, où $\sum PO$ est le nombre total de positions occupées par les obstacles internes et P est le nombre total de positions associées à l'espace physique.

La troisième mesure est le temps total T passé par tous les piétons pour atteindre l'objectif ou arrêter de bouger. En plus de ces mesures, il existe une mesure commune de la performance d'évacuation qui mesure le temps d'évacuation et l'efficacité/efficacité [31] ensemble pour aider à évaluer les compromis potentiels entre le temps d'évacuation et la capacité d'évacuation, ainsi que la performance globale d'évacuation des foules.

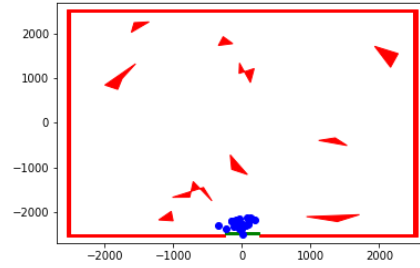
Pour mesurer ces paramètres de performance, plusieurs expériences doivent être effectuées à l'aide d'une simulation de modèle pour évacuer une grande foule. Nous devons donc pouvoir faire varier le nombre de piétons et voir comment cela évolue. Nous avons créé au hasard un espace physique (une salle carrée), composé de 10 obstacles internes de formes différentes.

Nous avons réalisé huit expériences ($e=1, \dots, 8$), dans lesquelles les huit foules diverses ont été réparties au hasard dans la salle composée de 20, 40, 60, 80, 100, 120, 140 et 160 piétons, respectivement, à évacuer d'une seule sortie (but) avec une largeur spécifiée.

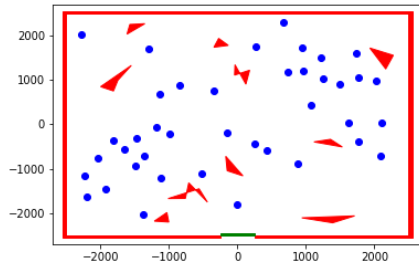
Les simulations du modèle ont été exécutées sur un ordinateur portable exécutant Windows 10 (64 bits) avec un processeur Intel (R) Core (TM) i5-4210U cadencé à 1,70 GHz et 8 Go de RAM.



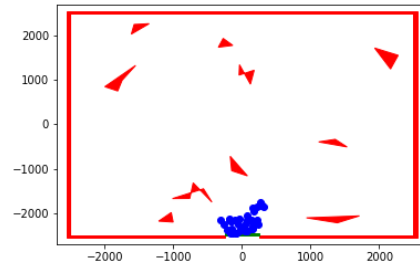
(a1) Distribution initiale d'une expérience avec 20 piétons



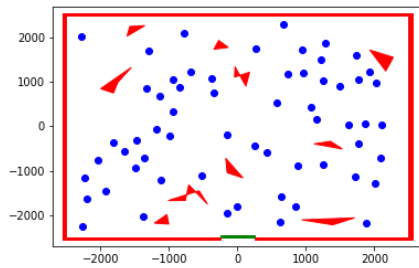
(b1) État final d'une expérience avec 20 piétons ($t=08.46s$)



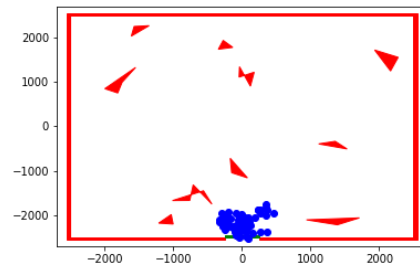
(a2) Distribution initiale d'une expérience avec 40 piétons



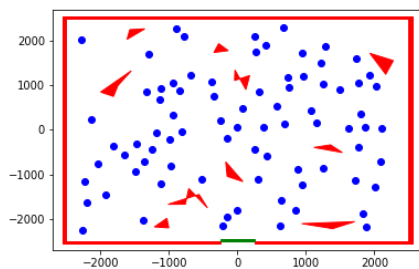
(b2) État final d'une expérience avec 40 piétons ($t=21.64s$)



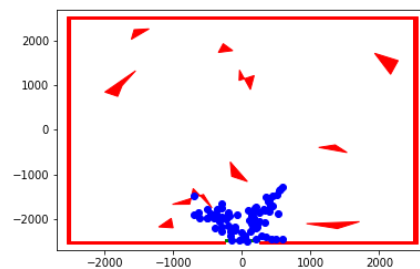
(a3) Distribution initiale d'une expérience avec 60 piétons



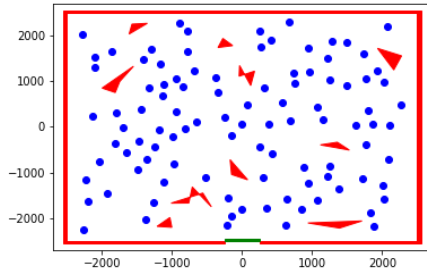
(b3) État final d'une expérience avec 60 piétons ($t=32.87s$)



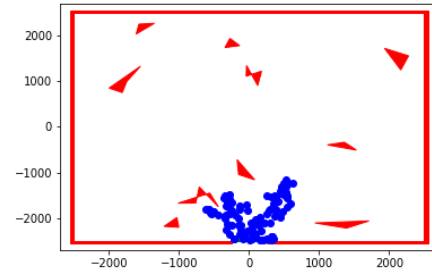
(a4) Distribution initiale d'une expérience avec 80 piétons



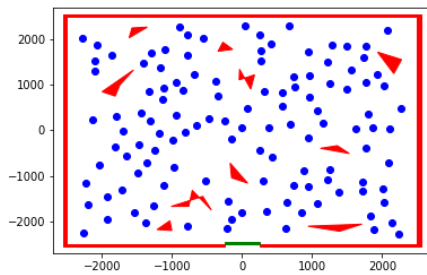
(b4) État final d'une expérience avec 80 piétons ($t=48.80s$)



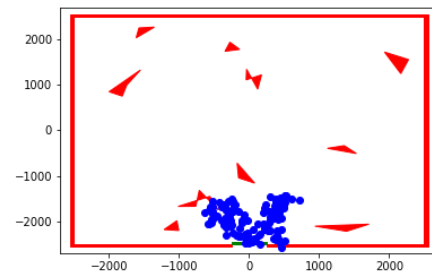
(a5) Distribution initiale d'une expérience avec 100 piétons



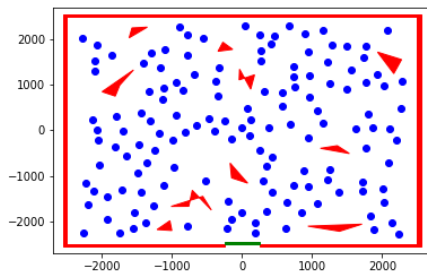
(b5) État final d'une expérience avec 100 piétons ($t=65.93s$)



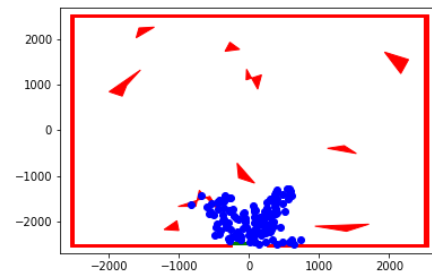
(a6) Distribution initiale d'une expérience avec 120 piétons



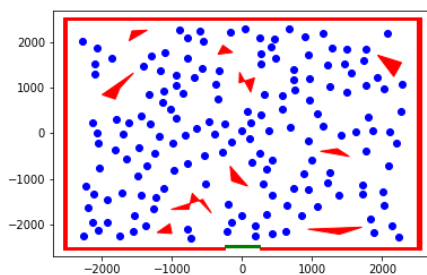
(b6) État final d'une expérience avec 120 piétons ($t=92.19s$)



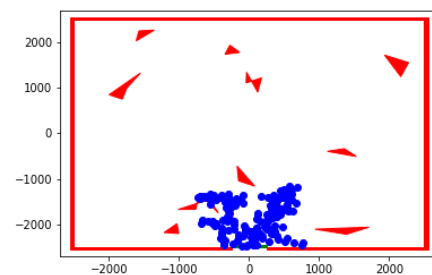
(a7) Distribution initiale d'une expérience avec 140 piétons



(b7) État final d'une expérience avec 140 piétons ($t=122.42s$)



(a8) Distribution initiale d'une expérience avec 160 piétons



(b8) État final d'une expérience avec 160 piétons ($t=155.36s$)

Figure 5.5 : Simulation de foule avec différentes situations nombre des obstacles et nombre de piétons

Les expériences par simulations de modèles visent à voir si nous pouvons évacuer ces foules à grande échelle de l'espace physique et à mesurer son efficacité et son efficacité.

Tableau 5.2 : Résultats expérimentaux

experiment	N. of Pedestrians	complexity	Reached the goal	N. of movements	Distance travelled	Ideal distance	T(seconds)	Effectiveness	Efficiency
E=1	20	12.0%	20	359	71554.5	61784.67	7.65	100.0%	85.94%
E=2	40	24.0%	39	680	135547.5	109938.96	18.52	97.5%	80.73%
E=3	60	36.0%	60	1015	198974.0	150560.28	31.31	100.0%	75.31%
E=4	80	48.0%	80	1260	245035.0	193382.25	44.46	100.0%	78.55%
E=5	100	60.0%	100	1600	315699.5	236088.38	67.29	100.0%	74.44%
E=6	120	72.0%	119	1825	360660.0	256611.76	80.78	99.17%	70.83%
E=7	140	84.0%	138	1985	390559.0	291358.11	99.03	98.57%	74.26%
E=8	160	96.0%	159	2258	441845.0	334681.13	121.38	99.38%	75.4%

Le tableau 5.2 montre pour chaque expérience ‘e’, le nombre de piétons N composant la foule, la densité du modèle, le nombre total R de piétons ayant atteint l'objectif, le nombre total de déplacements de piétons NM , la distance totale D parcourue par les piétons avant d'atteindre l'objectif ou d'arrêter le mouvement, la distance idéale ID , le temps total passé par tous les piétons lors de leurs déplacements T en secondes, l'efficacité d'évacuation de la foule ES et l'efficacité EY .

Les résultats ont montré que presque tous les piétons atteignaient l'emplacement de but (sortie) et qu'il y avait une augmentation proportionnelle du nombre d'étapes de simulation (nombre de mouvements) et du temps de simulation avec l'augmentation du nombre de piétons (intensité du modèle). Ces résultats sont cohérents avec ceux des chercheurs précédents [32, 33].

La figure 5.6 montre que le nombre total de mouvements et la distance totale parcourue augmentent évidemment avec l'augmentation de la densité (le nombre de piétons dans l'espace physique), entraînant une diminution de la vitesse moyenne jusqu'à ce que la foule atteigne presque un état stationnaire lorsque la densité dépasse une condition critique.

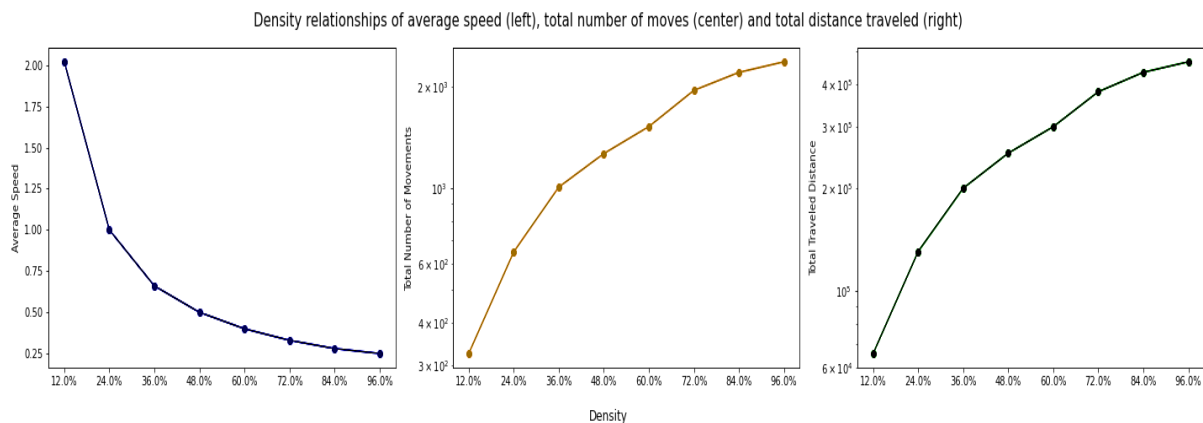


Figure 5.6 : Relations de densité entre la vitesse moyenne (à gauche), le nombre total de mouvements (au centre) et la distance totale parcourue (à droite).

Cependant, si la densité est à un niveau acceptable, le piéton se déplacera à la vitesse souhaitée en raison d'un espace suffisant et de légères influences des autres piétons. Ces résultats sont conformes à ce qui était attendu de la description du modèle.

Il est important d'évaluer la performance globale du modèle d'évacuation des foules en combinant les résultats métriques ci-dessus. La métrique F_e (Score F de combinaison de l'efficacité et de l'efficacéité) est utilisée pour calculer le compromis entre l'efficacité d'évacuation des foules ES et l'efficacité EY (Formule 1).

$$F_e = \frac{(\beta^2 + 1) \times EY_e \times ES_e}{\beta^2 \times EY_e \times ES_e} \quad (1)$$

Dans la formule précédente, EY_e et ES_e ($\in [0,1]$) représentent respectivement l'efficéité et l'efficacité de la simulation du modèle expérimental. Par conséquent, la pondération relative entre eux est déterminée à l'aide de cette formule. Nous utilisons la métrique en plus de calculer la performance du modèle d'évacuation de foule P_e (Précision de l'expérience) pour l'expérience e (Formule 2).

$$P_e = \frac{1}{1 + \exp^{\gamma \times T_e / N_e - \delta}} \quad (2)$$

Dans la formule, T_e est le temps de simulation (secondes) pour l'expérience e . Le nombre N_e est le nombre total de piétons. Pour permettre la comparaison des valeurs métriques entre des expériences de différentes tailles, nous utilisons le temps de simulation par nombre de piétons (c'est-à-dire T_e / N_e dans le calcul).

Pour distinguer les résultats de l'expérience sur différentes tailles, deux paramètres de contrôle γ (appelé la pente) et δ (appelé le décalage) sont utilisés. Leurs valeurs sont choisies pour que P_e (la simulation du modèle rapide par rapport à N_e) soit suffisamment proche de un (au-dessus de 0,99). Pour ces expériences, $\gamma = 2$ et $\delta = 5$.

La métrique de performance globale OP (Score F de la combinaison P et F) récompense la simulation du modèle lorsqu'elle permet aux piétons d'atteindre l'objectif plus rapidement, plus efficacement et plus efficacement et elle est définie comme une métrique composite de temps de simulation, d'efficéité et d'efficacité pour chaque expérience e , où $e = 1, \dots, 8$ comme dans ce qui suit (Formule 3).

$$OP_e = \frac{(\alpha^2 + 1) \times P_e \times F_e}{\alpha^2 \times P_e + F_e} \quad (3)$$

Les valeurs de β et α sont définies sur les formule 1 et 3, ce qui signifie que les poids de l'efficacité et de l'efficacité de l'évacuation des foules sont égaux, et que le poids du temps de simulation du modèle par rapport au poids de l'efficacité et de l'efficacité est également égal.

La figure 5.4 montre les résultats de la métrique de performance globale OP de la simulation du modèle qui est calculée pour chaque expérience. Le compromis entre l'efficacité et l'efficacité de l'évacuation des foules est supérieur à 80%, avec des performances globales supérieures à 90% et les performances d'évacuation sont presque parfaites, soit plus de 99%.

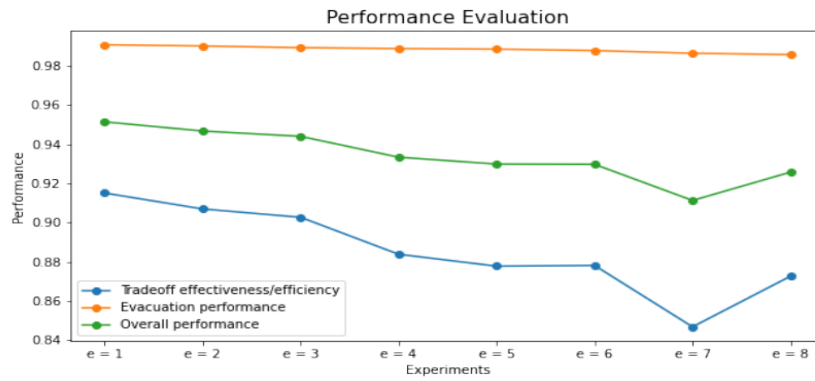


Figure 5.7 : Évaluation des performances

Les résultats agrégés montrent que la simulation du modèle se déroule de manière extrêmement fluide dans tous les domaines pour chaque expérience. Ces résultats numériques, illustrés à la figure 5.7, nous aident à comprendre l'efficacité globale de notre modèle en matière d'évacuation des foules.

7. Etude comparative

Nous avons évalué et comparé nos résultats de simulation précédente avec d'autres simulateurs de foules de piétons et dans les mêmes conditions au niveau d'environnement (espace physique) de simulation comme illustrée dans le tableau suivant :

Tableau 5.3 : comparaison entre notre simulation et quelques simulateurs de foule (facteur de temps /s)

Expérience	Nombre de piétons	SimLing Notre simulation	Simulex	KAUST	Helbing
01	20	7.65	17.2	14	9
02	40	18.52	22.4	32	36
03	60	31.31	26.5	48	48
04	80	44.46	30.2	108	61
05	100	67.29	35.1	122	80.5
06	120	80.78	37.9	136	94.3
07	140	99.03	40.3	148	120
08	160	121.38	43.4	155	161

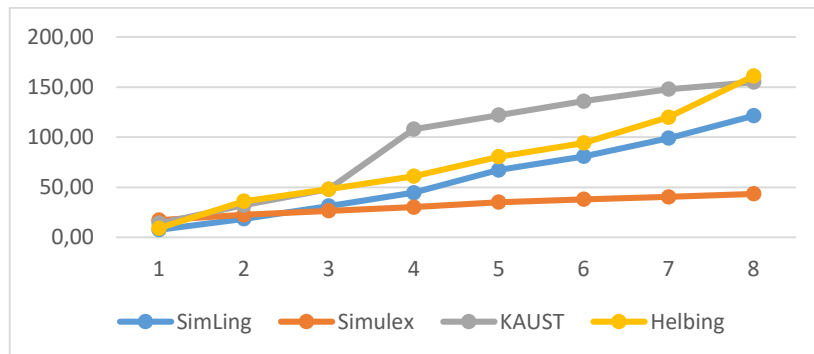


Figure 5.8 : comparaison entre les vitesses d'évacuation (notre simulation nommée par SimLing)

La figure 5.8 montre une comparaison entre notre simulateur ‘SimLing’ et les simulateurs Simulex, KAUST et Helbing où une foule de 20, 40, 60, 80, 100, 120, 140 et 160 piétons entités utilisées. Cette comparaison montre que notre simulateur ‘SimLing’ est bien compétitive avec ces simulateurs. Sa performance est dépassée seulement par le simulateur Simulex.

8. Conclusion

Nous avons décrit dans ce chapitre un modèle pour simuler les mouvements et les comportements d’une foule de piétons autonomes qui respecte le principe de la boucle perception-décision-action.

Nous avons simulé des scénarios avec différents nombres de piétons en déplacement dans un espace physique qui contient des obstacles statiques et dynamiques. L’idée originale de ce travail est de présenter la possibilité de simuler plusieurs niveaux de comportement d’individus et de groupes avec la technique de l’intelligence artificielle.

Nous avons comparé nos résultats de simulation avec d’autres simulateurs existants au point de vue temps d’évacuation et nous avons obtenue des résultats très satisfaisants.

Conclusions et perspectives

Conclusions et perspectives

Dans cette thèse, nous avons présenté un modèle d'animation comportementale de foule de piétons qui a donné lieu à un certain nombre d'expérimentations de comportement des piétons et la traduction de ce comportement en un outil de simulation opérationnelle. Dans un premier temps, nous avons présenté le problème étudié dans cette thèse et l'approche utilisée pour développer le programme de simulation, y compris quelques études de cas réalisées avec ce programme. Ceci est suivi d'une discussion sur les résultats de cette recherche et les conclusions sur la modélisation des piétons dans l'espace physique. Ensuite, des recommandations pour de futures recherches sont données.

Avant d'entamer notre recherche, nous avons présenté une étude approfondie sur l'état de l'art où les caractéristiques de l'animation comportementale, le comportement de foule de piétons puis les techniques de modélisation de foule avec une synthèse sur les divers simulateurs. L'étude de la littérature a permis d'identifier des points concernant les influences de cas de panique, de direction et de vitesse souhaitées de mouvement des piétons et les relations fondamentales de circulation des piétons (relation entre vitesse, direction et densité) qui identifient les comportements d'évacuation.

Nous avons présenté une description avec le langage Python d'un modèle de simulation d'évacuation de foules situées dans un espace physique avec des situations différentes. Le comportement des piétons composant ce modèle est décrit à l'aide de la technique de l'intelligence artificielle, notamment la perception de grandeurs numériques telles que les distances, les angles de direction et les vitesses de déplacement qui se transforment par nature en grandeurs qualitatives pour le raisonnement humain.

La description de ce modèle de simulation de l'évacuation de foule de piétons d'un espace physique en face de situations dangereuses, est basée sur des données synthétiques extraites des résultats de simulation de comportement de foules. Nous avons extrait des connaissances sur ces données en appliquant la tâche de clustering de la technique de fouille de données. Ainsi, nous avons développé un programme Python pour implémenter l'algorithme de clustering afin de générer les classes nécessaires pour les données acquises à partir de simulation tel que la distance, la vitesse et la direction.

Avant la simulation, ce modèle a été validé à l'aide d'une animation (une exécution animée) qui nous a permis de corriger plusieurs erreurs, notamment dans la partie de description du comportement des piétons. Les résultats de simulation sont analysés pour évaluer la performance de ce modèle. Nous avons obtenu un bon compromis entre son efficacité et son efficience. Ces résultats d'analyse de performance sont très prometteurs pour les perspectives d'avenir.

Le principal résultat de cette thèse est un outil de simulation opérationnelle validé pour le comportement des piétons durant les cas de panique, qui non seulement modélise le comportement de mouvement des piétons, mais inclut également le choix de l'itinéraire, la performance des activités et l'interaction des piétons entre eux. Mais peut également être appliqué dans d'autres zones piétonnes, telles que les villes, les centres commerciaux, les manifestations, le pèlerinage et les stades, etc...

D'un point de vue scientifique, la performance des expériences est la plus pertinente. En particulier, les expériences de goulots d'étranglement donnent un nouveau regard sur le comportement des piétons en ce qui concerne la formation de couches et l'effet de "fermeture éclair", ainsi que l'auto-organisation dans deux flux directionnels et de croisement. Une autre

implication est la répartition libre de la vitesse des piétons, indiquant une plage de vitesses de conception. De plus, les emplacements des piétons en attente en amont d'un goulot d'étranglement sont différents, comparés aux résultats de la plupart des modèles de simulation. Des résultats rapportés par d'autres chercheurs, tels que la formation de voies et l'auto-organisation des flux de piétons, émergent également des expériences.

Les modèles en ce qui concerne le comportement de choix sont tous microscopiques, c'est-à-dire que les piétons font des choix individuels, en fonction du chemin d'évacuation. De plus, la performance des activités est modélisée au microscope, de même que l'interaction avec les autres piétons en commun. Le comportement de mouvement des piétons, quant à lui, est modélisé de manière quasi macroscopique, en appliquant des relations entre vitesse et densité pour divers types d'espaces physiques (infrastructures).

Bien que cette recherche ait donné de nombreuses nouvelles perspectives sur le comportement des piétons et ait également produit un outil de simulation dans lequel différents aspects du comportement des piétons sont couverts, en particulier en ce qui concerne le comportement dans les cas de panique, la simulation de manifestations, etc....

Les expériences décrites dans cette thèse ne concernaient que le comportement des foules de piétons. De plus, seules quelques variables influençant ce comportement de mouvement ont pu être explorées. Des expériences supplémentaires sont nécessaires sur différentes compositions du flux, telles que le sexe, l'âge, s'ils ont des maladies, la culture, etc... Par conséquent, les observations du monde réel sont essentielles, non seulement pour observer le comportement des piétons sur divers types d'infrastructures, mais aussi pour exclure le comportement artificiel des piétons dû aux conditions de l'expérience (répétition des aspects et négociation avec les mêmes piétons tout au long des expériences, météo, température, comportement de regroupement, pression temporelle).

La deuxième perspective est de générer automatiquement la description des espaces physiques. L'entrée de cette automatisation sera des images sur les espaces physiques et grâce à la technique de détection d'objets, nous pourrions avoir leurs descriptions.

La troisième perspective est d'utiliser l'intelligence artificielle pour prédire le comportement des piétons.

Une quatrième perspective est de développer un logiciel de détection et de suivi dédié pour transformer automatiquement les images vidéo en données de trajectoire décrivant l'emplacement et la vitesse de chaque piéton. Parmi les nouvelles découvertes de ces expériences figuraient les relations vitesse-densité, les paramètres du comportement de suivi des piétons et les capacités dans le cas de panique.

Références

- [1] Youngchul Shin, Ilkyeong Moon, Robust building evacuation planning in a dynamic network flow model under collapsible nodes and arcs, *Socio-Economic Planning Sciences*, Volume 86, 2023.
- [2] Adrian, J., A. Seyfried, and A. Sieben. 2020. "Crowds in Front of Bottlenecks at Entrances from the Perspective of Physics and Social Psychology." *Journal of the Royal Society Interface* 17 (165): 1–12
- [3] [Bakar, Noor Akma Abu](#); [Majid, Mazlina Abdul](#); [Ismail, Khalid Adam](#), An Overview of Crowd Evacuation Simulation, [Advanced Science Letters](#), Volume 23, Number 11, November 2017, pp. 11428-11431
- [4] Zheng, L. J., X. L. Peng, L. L. Wang, and D. H. Sun. 2019. "Simulation of Pedestrian Evacuation Considering Emergency Spread and Pedestrian Panic." *Physica A-Statistical Mechanics and Its Applications* 522: 167–181.
- [5] Jinbin Zhang, Jun Zhu, Pei Dang, Jianlin Wu, Yuxuan Zhou, Weilian Li, Lin Fu, Yukun Guo & Jigang You (2023) An improved social force model (ISFM)-based crowd evacuation simulation method in virtual reality with a subway fire as a case study, *International Journal of Digital Earth*, 16:1, 1186-1204
- [6] Dong Zhang, Wenhong Li, Jianhua Gong, Guoyong Zhang, Jiantao Liu, Lin Huang, Heng Liu & Haonan Ma (2023) Deep reinforcement learning and 3D physical environments applied to crowd evacuation in congested scenarios, *International Journal of Digital Earth*, 16:1, 691-714
- [7] Zhou M , Dong H , Wang F-Y , Wang Q , Yang X . Modeling and simulation of pedestrian dynamical behavior based on a fuzzy logic approach. *Inf Sci* 2016;360:112–30
- [8] M. Zhou, H. Dong, D. Wen, X. Yao and X. Sun, "Modeling of Crowd Evacuation With Assailants via a Fuzzy Logic Approach," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2395-2407, Sept. 2016,
- [9] Chris Baker, *Artificial Intelligence: Learning automation skills with Python*, Kindle Edition, September 12, 2019
- [10] Teso S, Alkan Ö, Stammer W and Daly E (2023) Leveraging explanations in interactive machine learning: An overview. *Front. Artif. Intell.* 6
- [11] Helbing D , Molnr P . Social force model for pedestrian dynamics. *Physreve* 1995;51(5):4282 .
- [12] A. Varas et al., "Cellular automaton model for evacuation process with obstacles," *Phys.A,Stat.Mech.Appl.*, vol. 382, no. 2, pp. 631–642, Aug. 2007.
- [13] S. Liu, L. Yang, T. Fang, and J. Li, "Evacuation from a classroom considering the occupant density around exits," *Phys. A, Stat. Mech. Appl.*, vol. 388, no. 9, pp. 1921–1928, May 2009.
- [14] D. Helbing, M. Isobe, T. Nagatani, and K. Takimoto, "Lattice gas simulation of experimentally studied evacuation dynamics," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.*, vol. 67, no. 6, Jun. 2003, Art. ID 067101.
- [15] R. Guo and H. Huang, "A mobile lattice gas model for simulating pedestrian evacuation," *Phys. A, Stat. Mech. Appl.*, vol. 387, no. 2, pp. 580–586, Jan. 2008.
- [16] D. Shi and B. Wang, "Evacuation of pedestrians from a single room by using snowdrift game theories," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.*, vol. 87, no. 2, Feb. 2013, Art. ID 022802.
- [17] S. Bouzat and M. Kuperman, "Game theory in models of pedestrian room evacuation," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.*, vol. 89, no. 3, Mar. 2014, Art. ID 032806.

- [18] A. Shende, M. P. Singh, and P. Kachroo, "Optimization-based feedback control for pedestrian evacuation from an exit corridor," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1167–1176, Dec. 2011.
- [19] H. Frantzich and D. Nilsson, "Evacuation experiments in a smoke filled tunnel," in *Proc. 3rd Int. Symp. Human Behav. Fire*, 2004, pp. 229–238.
- [20] M. Kobes et al., "Way finding during fire evacuation; an analysis of unannounced fire drills in a hotel at night," *Build. Environ.*, vol. 45, no. 3, pp. 537–548, Mar. 2010.
- [21] J. Ma, S. M. Lo, and W. Song, "Cellular automaton modeling approach for optimum ultra high-rise building evacuation design," *Fire Safety J.*, vol. 54, pp. 57–66, Nov. 2012.
- [22] Y. Song et al., "Crowd evacuation simulation for bioterrorism in micro spatial environments based on virtual geographic environments," *Safety Sci.*, vol. 53, pp. 105–113, Mar. 2013.
- [23] M. Nasir, C. P. Lim, S. Nahavandi, and D. Creighton, "A genetic fuzzy system to model pedestrian walking path in a built environment," *Simul. Model. Pract. Theory*, vol. 45, pp. 18–34, Jun. 2014.
- [24] M. DellOrco, M. Marinelli, and M. Ottomanelli, "Simulation of crowd dynamics in panic situations using a fuzzy logic-based behavioural model," in *Computer-Based Modelling and Optimization in Transportation*. Cham, Switzerland: Springer-Verlag, Feb. 2014, pp. 237–250.
- [25] Bahamid, A., Mohd Ibrahim, A. A review on crowd analysis of evacuation and abnormality detection based on machine learning systems. *Neural Comput & Applic* **34**, 21641–21655 (2022)
- [26] Lee J, Won J, Lee J (2018) Crowd simulation by deep reinforcement learning.
- [27] Wang K, Shi X, Goh APX, Qian S (2019) A machine learning based study on pedestrian movement dynamics under emergency evacuation. *Fire Saf J* 106:163–176.
- [28] Yao Z, Zhang G, Lu D, Liu H (2019) Data-driven crowd evacuation: a reinforcement learning method. *Neurocomputing* 366:314–327.
- [29] Li X, Liang Y, Zhao M, Wang C, Bai H, Jiang Y (2019) Simulation of evacuating crowd based on deep learning and social force model. *IEEE Access* 7:155361–155371.
- [30] Raschka S, Patterson J, Nolet C. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*. 2020; 11(4):193
- [31] [Zidane, Y.J.-T.](#) and [Olsson, N.O.E.](#) (2017), "Defining project efficiency, effectiveness and efficacy", *International Journal of Managing Projects in Business*, Vol. 10 No. 3, pp. 621-641
- [32] D. Helbing, A. Johansson, and H. Z. Al-Abideen, "Dynamics of crowd disasters: An empirical study," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.*, vol. 75, no. 4, Apr. 2007, Art. ID 046109.
- [33] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, Sep. 2000.
- [34] Cherif FOUJIL, "Animation comportementale : simulation de foule d'humains virtuels"; université de Biskra. (12/12/2006).
- [35] PANZOLI David. "Proposition de l'architecture « Cortexionist » pour l'intelligence comportementale de créatures artificielles", Thèse de Doctorat, Université de Toulouse, (30/09/2008).
- [36] Stéphane Donikian, "Modélisation, contrôle et animation d'agents virtuels autonomes évoluant dans des environnements informés et structurés", Université de Rennes 1. (26/08/2004).
- [37] Mallot (H.A). "Behavior-oriented approaches to cognition: theoretical perspectives". *Theory in biosciences*, pp: 196-200. (1997).

- [38] Craig Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model", Proc. SIGGRAPH, Computer Graphics, v.21, n.4, Symbolics Graphics Division, Los Angeles, (1987). <http://www.red3d.com/cwr/>
- [39] D.Thalmann, "A New Generation of Synthetic Actors: the Interactive Perceptive Actors", Proc. Pacific Graphics'96 Taipeh, Taiwan, pp: 200-219. (1996).
- [40] D.Thalmann, "Using Virtual Reality Techniques in the Animation Process in: Virtual Reality Systems". (Earnshaw et al. eds), Academic Press, pp:143-159. (01/12/1998)
- [41] Robert G. Lord et P. E. Levy. "Moving from cognition to action: A control theory perspective". Applied Psychology: an international review, pp: 335-398. (1994).
- [42] Allen Newell (1990). "Unified theories of cognition". Harvard University Press, Cambridge, Massachusetts.
- [43] Turner (R.H.). Handbook of modern sociology, chap. Collective Behavior, pp. 382-425. Rand McNally, 1964.
- [44] Yamori (K.). "Going with the flow: Micro-macro dynamics in the macrobehavioral patterns of pedestrian crowds". Psychological Review, vol. 105, n3, pp. 530-557.(1998).
- [45] Boles (W.). "Planning pedestrian environments: A computer simulation model". Man-Environment Systems, vol. 11, pp. 41-56. (1981).
- [46] Milgram (S.) et Toch (H.). The handbook of social psychology, chap. Collective Behavior: Crowds and social movements, pp. 507-610. Addison-Wesley, 1969.
- [47] Henderson, L. F. "The Statistics of Crowd Fluids". Nature. 229, pp: 381-383. (1971).
- [48] Paris (S.), "Caractérisation des niveaux de services et modélisation des circulations de personnes dans les lieux d'échanges", Thèse de Doctorat, Université de Rennes I, (10/10/2007).
- [49] Peschl (I.). "Flow capacity of door openings in panic situations". BAUN, vol. 26, n2, pp. 62-67. (1971).
- [50] Goldenstein (S.), Karavelas (M.), Metaxas (D.), Guibas (L.), Aaron (E.) et Goswami (A.). "Scalable nonlinear dynamical systems for agent steering and crowd simulation". Computer & Graphics, vol. 25, pp. 983-998. (2001)
- [51] Ulicny (B.) et Thalmann (D.). – Towards interactive real-time crowd behavior simulation. Computer Graphics Forum, vol. 21, n4, pp. 767-775. (2002)
- [52] Hall, E. T. The Silent Language. Garden City: Anchor Books. "Emerging Social Conventions: personal Space and Privacy in Shared Virtual Worlds", pp: 27-430 Project, (12/1998).
- [53] Reynolds, C. "Steering Behaviors for Autonomous Characters". In. Proc. Game Developers Conference, pp: 763-782. (1999).
- [54] O'Hara (N.). "K-d tree neighbour finding for the flocking algorithm". Dans: Pacific Graphics. (2000).
- [55] Ashida (K.), Lee (S.), Allbeck (J.), Sun (H.), Badler (N.) et Metaxas (D.). "Pedestrians: Creating agent behaviors through statistical analysis of observation data". Dans : Computer Animation, éd. par Society (IEEE Computer), pp. 84-92. Seoul, Korea, (2001).
- [56] Tsutsuguchi (Ken), Sugiyama (Kazuhiro) et Sonehara (Noboru). "The motion generation of pedestrians as avatars and crowds of people". Dans : Digital Cities. pp. 275-287. Springer-Verlag, (2000).
- [57] Moulin (B.), Chaker (W.), Perron (J.) et Pelletier (P.). "Mags project: Multiagent geosimulation and crowd simulation". Dans : COSIT 2003, éd. par Kuhn (W.), Worboys (M.F.) et Timpf (S.), pp. 151-168. Springer-Verlag, 2003.
- [58] LAMARCHE Fabrice, "Humanoïdes virtuels, réaction et cognition : une architecture pour leur autonomie" ; Université de Rennes 1. (19/12/2003).
- [59] Aiello, J.R. Human Spatial Behaviour. In D. Stokols & I. Altman, "Handbook of environmental Psychology": Volume 1. New York: Wiley & Sons, pp: 389-504. (1987).

- [60] Bowers, J., Benford, S. D., Craven, M., Morphet, J., Wyver, J., Greenhalgh, C., "Evaluating Out of This World: An Experiment in Inhabited Television", eRENA deliverable D7a.1, ESPRIT Long Term Research Project 25379. (1999).
- [61] Jeffrey, P., Mark, G., "Constructing Social Spaces in Virtual Environments: A Study of Navigation and Interaction", Hook, K., Munro, A., Benyon, D., Workshop on Personalised and Social Navigation in Information Spaces, pp: 16- 17, (03/1998).
- [62] Belnesch Hellmuth. Atlas de la Psychologie, Encyclopédies d'Aujourd'hui, (1995).
- [63] Feldman, D. C. "The development and enforcement of groups norms. Academy of management Review", pp. 47-53. (1984).
- [64] Kauppinen, K.; Kivimäki, A.; Era, T. and Robinson, M. "Producing identity in Collaborative Virtual Environments". In Proceedings of ACM Symposium on Virtual Reality Software and Technology, pp: 35-42. (1998).
- [65] Jeffrey, P. "Emerging Social Conventions: personal Space and Privacy in Shared Virtual Worlds", CS, pp: 27-430 Project, (12/1998).
- [66] Stephen Chenney, "Flow Tiles", Eurographics/ACM SIGGRAPH Symposium on Computer Animation R. Boulic, D. K. Pai (Editors) University of Wisconsin, Madison, (2004). <http://www.cs.wisc.edu/~schenney/>
- [67] Nuria Pelechano, Jan M. Allbeck, and Norman I. Badler. "Virtual crowds: methods, simulation, and control". Book. University of California–Berkeley. www.morganclaypool.com. (2008).
- [68] Dijkstra, J., Timmermans, H. J. P. and Jessurun, A. J. A "Multi-Agent Cellular Automata System for Visualizing Simulated Pedestrian Activity". In. Proc. Theoretical and Practical Issues on Cellular Automata. Cellular Automata for research and Industry, Springer-Verlag. Berlin, pp: 29-36. (2000).
- [69] Kirchner, A., Namazi, A., Nishinari, K. and Schadschneider, A. "Role of Conflicts in the Floor Field Cellular Automaton Model for Pedestrian Dynamics". In. Proc. 2nd International Conference on Pedestrians and Evacuation Dynamics. (PED), London, UK, pp: 51-62. (2003).
- [70] Tecchia, F., Loscos, C., Conroy, R. and Chrysanthou, Y. "Agent Behavior Simulator (ABS): A Platform for Urban Behavior Development". In. Proc. ACM/EG Games Technology Conference 55(3), pp: 601-644. (2001).
- [71] Tecchia, F., Loscos, C., Conroy, R. and Chrysanthou, Y. "Agent Behavior Simulator (ABS): A Platform for Urban Behavior Development". In. Proc. ACM/EG Games Technology Conference. (2001).
- [72] Torrens, P. M. "Behavioral intelligence for geospatial agents in urban environments". In. Proc. IEEE Intelligent Agent Technology, Los Alamitos, CA, IEEE, pp: 63-66. (2007).
- [73] Loscos, C., Marchal, D. and Meyer, A. "Intuitive Crowd Behaviour in Dense Urban Environments using Local Laws". IEEE Theory and Practice of Computer Graphics, pp: 122. (2003).
- [74] Craig Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model", Proc. SIGGRAPH, Computer Graphics, v.21, n.4, Symbolics Graphics Division, Los Angeles, (1987). <http://www.red3d.com/cwr/>
- [75] Musse, S. R. and Thalmann, D. "From One Virtual Actor to Virtual Crowds: Requirements and Constraints". In. Proc. (2000).
- [76] David C. Brogan, Ronald A. Metoyer et Jessica K. Hodgins, "Dynamically Simulated Characters in Virtual Environments", Georgia Institute of Technology, Atlanta, (1998).
- [77] Nuria Pelechano, Kevin O'Brien, Barry Silverman et Norman Badler, "Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication", Center for Human Modeling and Simulation, University of Pennsylvania, Philadelphia, (2005). <http://www.seas.upenn.edu/~npelecha/>

- [78] Milazzo, J. S., Roupail, N. M., Hummer, J. E. and Allen, D. P. "The Effect of Pedestrians on the Capacity of Signalized Intersections". *Transportation Research Record*. pp: 37-46. (1998).
- [79] Hoogendoorn, S. P. "Pedestrian Travel Behavior Modeling". In. *Proc. Travel Behavior Research Lucerne*, Elsevier, pp: 10-15. (2003).
- [80] Soraia Raupp Musse et Daniel Thalmann, "Hierarchical Model for Real Time Simulation of Virtual Human Crowds", *Computer Graphics Lab, Federal Institute of Technology, Lausanne*, (2000).
- [81] Henderson, L. F. "The Statistics of Crowd Fluids". *Nature*. 229, pp: 381-383. (1971).
- [82] J. Archea. The evacuation of non-ambulatory patients from hospital and nursing home fires: A framework for a model. Technical report. NBSIR 79-1906. 1.3.2 (11/1979).
- [83] Lovas, G. C. "Modeling and Simulation of Pedestrian Traffic Flow". *Transportation Research* 28(6), pp: 429-443. (1994).
- [84] Paris (S.), "Caractérisation des niveaux de services et modélisation des circulations de personnes dans les lieux d'échanges", *Thèse de Doctorat, Université de Rennes I*, (10/10/2007).
- [85] Galea, E. R., Perez Galparsoro, J. M. and Pearce, J. A "Brief Description of the EXODUS Evacuation Model". In. *Proc. International Conference on Fire Safety, San Francisco, USA*, pp: 149-162. (1993).
- [86] Owen, M., Galea, E. R., Lawrence and Filippidis, L. "The Numerical Simulation of Aircraft Evacuation and its Applications to Aircraft Design and Certification". *The Aeronautical Journal*. 102(1016), pp: 301-312. (1998).
- [87] Buckmann, L. T. and Leather, J. "Modelling Station Congestion the PEDROUTE Way. *Traffic Engineering and Control*". 35(6) pp : 373-377. (1994).
- [88] Ulicny, B. and Thalmann, D. "Crowd Simulation for Interactive Virtual Environments and VR Training Systems". In. *Proc. Eurographics Workshop on Animation and Simulation*. Springer-Verlag. Berlin, pp: 163-170. (2001).
- [89] Thompson, P. A. and Marchant, E. W. "Testing and Application of the Computer Model Simulex". *Fire Safety Journal* 25, pp: 149-166. (1995).
- [90] Bouvier, E. and Cohen, E. "Simulation Of Human Flow With Particles Systems". In. *Proc. Simulators International XII*, Phoenix. (1995).
- [91] Ahn, J, Oh, S. and Wohn, K. "Optimized Motion Simplification for Crowd Animation. *Computer Animation and Virtual Worlds*". pp: 155-165. (2006).
- [92] Mott-MacDonald. "STEPS Simulation of Transient Evacuation and Pedestrian Movements User Manual, Mott MacDonald". (2003).
- [93] Musse, S. R., Babski, C., Capin, T. and Thalmann, D. "Crowd Modelling in Collaborative Virtual Environments". In. *Proc. ACM Virtual Reality Software and Technology (VRST)*, Taipei, Taiwan, pp: 115-123. (1998).
- [94] Reynolds, C. "Steering Behaviors for Autonomous Characters". In. *Proc. Game Developers Conference*, pp: 763-782. (1999).
- [95] Thomas, G. and Donikian, S. "Virtual Humans Animation in Informed Urban Environments". In. *Proc. Computer Animation, IEEE Computer Society*. Washington, DC, USA, pp: 112-121. (2000).
- [96] Allbeck, J. and Badler, N. "Embodied Autonomous Agents. *Handbook of Virtual Environments: Design, Implementation and Applications*". K. Stanney. Lawrence Erlbaum Associates, Philadelphia, pp: 313-332. (2002).
- [97] Bindiganavale, R., Schuler, W., Allbeck, J., Badler, N., Joshi, A. and Palmer, M. "Dynamically Altering Agent Behaviors Using Natural Language Instructions". In. *Proc. Autonomous Agents*, ACM New York, NY, USA: 293-300. (2000).

- [98] Chi, D., Costa, M., Zhao, L. and Badler, N. "The EMOTE Model for Effort and Shape". In. Proc. ACM SIGGRAPH, New Orleans, LA, ACM Press. New York, USA, pp: 173-183. (2000).
- [99] Maletic, V. "Body, Space, Expression: The development of Rudolf Labans Movement and Dance Concepts". Mouton de Gruyter, New York. (1987).
- [100] Shao, W. and Terzopoulos, D. "Autonomous Pedestrians". In. Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Los Angeles, California ACM Press. New York, USA, pp: 19-28. (2005).
- [101] Shao, W. and Terzopoulos, D. "Autonomous Pedestrians". Graphical Models. 69, pp: 246-274. (2007).
- [102] Yu, Q. and Terzopoulos, D. "A Decision Network Framework for the Behavioral Animation of Virtual Humans". In. Proc. ACM SIGGRAPH/Eurographics, symposium on Computer animation, San Diego, California Eurographics Association, pp: 119-128. (2007).
- [103] Torrens, P. M. "Behavioral intelligence for geospatial agents in urban environments". In. Proc. IEEE Intelligent Agent Technology, Los Alamitos, CA, IEEE, pp: 63-66. (2007).
- [104] Turner, A. and Penn, A. "Encoding Natural Movement as an Agent-based System: an Investigation into Human Pedestrian Behaviour in the Built Environment". Environment and Planning B: Planning and Design. Pion Limited, London. 29, pp: 473-490. (2002).
- [105] Eric Darnell and Tim Johnson, PDI/DreamWorks, 1998.
http://www.dreamworksanimation.com/dwa/opencms/movies/movie_antz.html
- [106] John Lasseter, Andrew Stanton, "A Bug's Life". (1001 Pattes), Pixar, (1999).
<http://www.pixar.com/featurefilms/abl/>
- [107] Thompson (P.A.) et Marchant, (E.W.), "Simulex: Developing new computer modeling techniques for evacuation". Unit of Fire Safety Engineering, University of Edhimburg. (1995).
- [108] Steels (L.). "The artificial life roots of artificial intelligence". In: Artificial Life Journal, Vol. 1(1), pp: 75-110, MIT Press, (1994).
- [109] Pelechano, N., Allbeck, J. and Badler, N. "Controlling Individual Agents in High-Density Crowd Simulation". In. Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA), San Diego, CA, ACM Press. New York, USA, pp: 99-108. (2007).
- [110] Pelechano, N. and Badler, N. "Modeling Crowd and Trained Leader Behavior during Building Evacuation". IEEE Computer Graphics and Applications. 26(6), pp: 80-86. (2006).
- [111] Hanspeter A. Mallot. "Spatial cognition: Behavioral competences, neural mechanisms and evolutionary scaling". pp 40-48. (1999).
- [112] Braun, A., Raupp Musse, S., and L.O.B. Bodmann. Modeling Individual Behaviors in Crowd Simulation. Computer Animation and Social Agents, New Brunswick (USA), 2003, pages 143-148.
- [113] U. Weidmann. Transporttechnik der fussgänger. 1992.
- [114] L.F. Henderson. The statistics of crowd fluids. Nature, 229 :381-383, 1971.
- [115] J. Pauls. Calculating evacuation times for tall buildings. Fire Safety Journal, 12(3): 213-236, 1987
- [116] R. McNeill Alexander. Energetics and optimization of human walking and running: the 2000 raymond pearl memorial lecture. American Journal of Human Biology, 14(5) :641-648, 2002.
- [117] J.J. Fruin. Designing for pedestrians: A level of service concept. Highway research record, (355) :1-15, 1971.
- [118] J.J. Fruin. Designing for pedestrians. Public Transportation United States, 1992.

- [119] S.J. Older. Movement of pedestrians on footways in shopping streets. Traffic engineering & control, 1968.
- [120] W.M. Predtetschenski and A.I. Milinski. Personenströme in gebäuden. berechnungsmethoden für die projektierung. verlagsgesellschaft rudolf müller, köln-braunsfeld, 1971. isbn: 3-481-16844-6. German, translation from Russian.
- [121] M.R. Virkler and S. Elayadath. Pedestrian speed-flow-density relationships. Number HS-042 012. 1994.
- [122] A.K. Sarkar and K.S.V.S. Janardhan. A study on pedestrian flow characteristics. In The 76th TRB annual meeting, National Research Council, Washington, DC, pages 169–192, 1997.
- [123] Y. Tanaboriboon, S.S. Hwa, and C.H. Chor. Pedestrian characteristics study in singapore. Journal of transportation engineering, 112(3) :229–235, 1986.
- [124] M. Moussaïd, D. Helbing, and G. Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. Proceedings of the National Academy of Sciences, 108(17) :6884–6888, 2011.
- [125] M. Fukamachi and T. Nagatani. Sidle effect on pedestrian counter flow. Physica A: Statistical Mechanics and its Applications, 377(1) :269–278, 2007.
- [126] G.K. Still. Crowd dynamics. PhD thesis, University of Warwick, 2000.
- [127] S.D. Reicher. The st. pauls’ riot: An explanation of the limits of crowd action in terms of a social identity model. European journal of social psychology, 14(1) :1–21, 1984.
- [128] D. Helbing. Traffic and related self-driven many-particle systems. Reviews of modern physics, 73(4) :1067, 2001.
- [129] P. Pecol. Modélisation 2D discrète du mouvement des piétons : application à l’évacuation des structures du génie civil et à l’interaction foule-passerelle. PhD thesis, Université Paris-Est, 2011.
- [130] W. Daamen. Modelling passenger flows in public transport facilities. TU Delft, Delft University of Technology, 2004.
- [131] Nesrine Masmoudi, Thèse doctorat, 2018, Modèle bio-inspiré pour le clustering de graphes : application à la fouille de données et à la distribution de simulations.
- [132] N. Labroche. Modélisation du système de reconnaissance chimique des fourmis pour le problème de la classification non-supervisée : application à la mesure d’audience sur Internet. Thèse de doctorat, Laboratoire d’Informatique, Université de Tours, décembre 2003. (Cité en pages 35, 40, 62, 70, 92 et 152.)
- [133] C. Charles. SearchXQ : une méthode daide à la navigation fondée sur O-means, algorithme de classification non-supervisée. Application sur un corpus juridique Français. Thèse pour obtenir le grade de docteur de l’Ecole des Mines de Paris, spécialité "Informatique Temps Réel, Robotique et Automatique", 2004. (Cité en page 40.)
- [134] A.K. Jain et R.C. Dubes. Algorithms for clustering data. Prentice Hall Advanced Reference Series, 1988. (Cité en pages 3, 41, 42, 44, 46 et 127.)
- [135] J.B. MacQueen. Some methods of classification and analysis of multivariate observations. In Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability, pages 281–297, 1967. (Cité en pages 44, 90 et 96.)
- [136] R.H. Bonner. On some clustering techniques. IBM Journal, 1964. (Cité en page 44.)
- [137] G. Celeux et E. Diday. Classification automatique de données environnement statistique et informatique. Dunod, Informatique, 1989. (Cité en pages 3, 44, 46 et 47.)
- [138] Mohamed Chatra : Animation des phénomènes collectifs cellulaires par modèle physique particulière. Mémoire magister, Université Mohamed Khider Biskra, 2010. <http://thesis.univ-biskra.dz/id/eprint/863>

- [139] Chatra, M, Bourahla, M. (2024). Agent-based simulation of crowd evacuation through complex spaces. *Ingénierie des Systèmes d'Information*, Vol. 29, No. 1, pp. 83-93. <https://doi.org/10.18280/isi.290110>