

**Ministère de l'enseignement supérieur et de la recherche scientifique**

Université Saad DAHLAB Blida 1



**Faculté des Sciences**

**Département d'Informatique**

Mémoire Présenté par :

**KHIRI Tarek**

**En vue de l'obtention du diplôme de master**

**Domaine :** Mathématiques et informatique

**Filière :** Informatique

**Spécialité :** Informatique

**Option :** Ingénierie de logiciels

**Thème :**

**Développement d'un système d'évacuation guidée en utilisant  
la réalité virtuelle**

Soutenu le : xx/xx/2019, devant le jury composé de :

**Pr<sup>e</sup>. Abed H.**

Encadreur

**Mr. Lorem Ipsum**

Président

**Mr. Lorem Ipsum**

Examineur

**2018/2019**



# Déclarations

*Je déclare que cet ouvrage n'a été composé que par moi-même et qu'il n'a pas été soumis entièrement ou partiellement à une candidature précédente. Sauf indication contraire par référence ou par reconnaissance, le travail présenté ici est entièrement le mien.*

# Remerciements

*Je tiens à remercier toutes les personnes qui ont contribué au succès de mon travail et qui m'ont aidé lors de la rédaction de ce mémoire.*

*Tout d'abord, je tiens à rendre grâce à dieu tout puissant. Je voudrais dans un second temps remercier mon directeur de mémoire Mme. Abed, professeur à l'université de Blida 1, pour sa patience, sa disponibilité et surtout ses conseils, qui m'ont aidé au processus de ce travail.*

*Mes parents, je vous remercie pour votre soutien constant et vos encouragements.*

*A tous ceux et à toutes celles dont les acronymes n'apparaissent pas sur cette page, ils sont nombreux, qu'ils demeurent convaincus, que je ne vous ai pas oublié et que vous soyez assurés de ma profonde gratitude.*

*Merci.*

*KHIRI Tarek.*

# Résumé

Le développement de l'architecture des bâtiments a apporté de nouveaux niveaux de conception, les rendant robustes et efficaces. Cependant, ils sont devenus plus grands et plus complexes et surtout inefficaces pour les exercices d'évacuation en cas d'urgences tel que les incendies des bâtiments. La modélisation des informations du bâtiment (BIM) offre une multitude de couches d'informations numériques pouvant enrichir les entrées utilisées pour les procédures d'évacuation. La formation à l'évacuation traditionnelle n'est efficace que dans le cas simple et ne parvient pas à identifier et à résoudre les problèmes pouvant survenir lors du processus d'évacuation. Les exercices d'évacuation n'ont pas la capacité de faire face à l'évolution dynamique de la situation d'urgence. Ils sont donc moins dynamiques et plus enclins à échouer face à une tournure inattendue des événements.

Ce travail vise à développer une solution d'évacuation guidée cohérente basée sur les données de modélisation des informations du bâtiment, les systèmes de réalité virtuelle (RV) et de nouvelles pratiques de modélisation, soit le système « EgressVR ». Il a pour but d'offrir à la fois une solution dynamique en temps réel pour le processus d'évacuation et un environnement virtuel pour l'entraînement. Il présente une amélioration aux modèles de réseaux de flots dynamiques. L'objectif est de proposer une meilleure méthode de représentation graphique pour les bâtiments et une solution événementielle pour la défaillance des plans d'évacuation. La solution proposée est davantage basée sur une approche événementielle orientée-objet contrairement à la définition mathématique conventionnelle et la résolution linéaire des problèmes de flots.

**Mots clés :** Modélisation de l'évacuation, dynamique de la foule, dynamique piétonne, simulation du comportement de la foule, Serious Games, internet des objets, réalité virtuelle.

# Abstract

Building architecture development has brought new levels of building design, making them more robust and efficient, but also bigger and more complex to grasp from an occupant point of view, thus making them inefficient for evacuation drills. Building Information Modeling (BIM) offers a multitude of digital information layers which can help enrich data inputs used for evacuation procedures. Traditional evacuation training is only effective in simple case scenarios and fails to identify and solve real time problems that could occur in the evacuation process such as detecting congestion areas in the building and eventually evade and resolve congestion. Evacuation drills lack the ability to cope with the dynamic evolution of the emergency as well as that of the population. Thus, evacuation drills are less dynamic and more prone to fail under unexpected turn of events.

This research aims at developing a coherent guided evacuation solution based on Building Information Modeling data, Virtual Reality (VR) Systems, and new modeling practices to both offer a dynamic real time solution for the evacuation process, and provide a virtual environment for training, namely “EgressVR”. This work presents an enhancement over the dynamic network flow evacuation models. It aims at offering a better graphical representation method for buildings, and an event-aware solution for evacuation plan defiance. The proposed solution is based more on an event-driven object-oriented approach rather than the conventional mathematical definition and resolution of network flow problems.

**Keywords:** Evacuation modeling, egress modeling, indoor evacuation, crowd behavior simulation, crowd dynamics, pedestrian dynamics, Serious Games, internet of things, virtual reality.

# Table des matières

<b>Déclarations</b> .....	<b>I</b>
<b>Remerciements</b> .....	<b>II</b>
<b>Résumé</b> .....	<b>III</b>
<b>Abstract</b> .....	<b>IV</b>
<b>Table des matières</b> .....	<b>V</b>
<b>Liste des figures</b> .....	<b>VIII</b>
<b>Liste des tableaux</b> .....	<b>X</b>
<b>Introduction</b> .....	<b>1</b>
1. Contexte et motivation .....	2
2. Enoncé du mémoire .....	2
2.1. Problématique .....	2
2.2. Postulats .....	3
2.3. Buts et objectifs.....	3
3. Plan du mémoire .....	3
<b>Chapitre I : Etat de l'art</b> .....	<b>5</b>
1. Introduction.....	6
2. Building Information Modeling .....	6
2.1. Environnement bâti .....	6
2.2. Les solutions BIM.....	8
3. Aspects de recherche de la thématique de l'évacuation.....	10
3.1. La théorie d'évacuation.....	11
3.2. La modélisation de l'évacuation .....	12
3.3. Le modèle magnétique .....	14
3.4. Le modèle des forces sociales .....	16
3.5. Le modèle d'optimisation .....	17
3.6. Le modèle multi-agent .....	19
3.7. Le modèle de réseaux de flots.....	21
3.8. Le modèle mixte .....	22
4. Algorithmes de navigation .....	23
4.1. Approche du champ potentiel .....	23
4.1.1. Algorithme de remplissage par diffusion .....	24
4.2. Approche d'optimisation .....	24
4.2.1. Algorithme de Dijkstra.....	25
4.2.2. L'Algorithme A* .....	26

5. Solutions existantes.....	27
6. Synthèse .....	30
6.1. Critiques.....	30
6.2. Direction .....	31
7. Conclusion .....	31
<b>Chapitre II : Serious Games &amp; Moteurs de Jeux.....</b>	<b>33</b>
1. Introduction.....	34
2. Serious Games .....	34
3. Moteurs de jeux.....	35
3.1. Définition .....	35
3.2. Architecture d'un moteur de jeu .....	36
3.3. Les concepts de base pour la création d'un jeu vidéo.....	37
3.3.1. La Boucle du jeu .....	38
3.3.2. Evénements et gestionnaires d'événements .....	38
3.3.3. Objets d'un jeu vidéo .....	39
3.4. Intelligence artificielle dans l'industrie des jeux vidéo .....	40
3.4.1. Arbres de comportement .....	40
3.4.2. Blackboards & Services .....	42
4. Réalité virtuelle.....	43
4.1. Définition .....	43
4.2. Développement d'applications de la RV .....	44
4.3. Contribution scientifique .....	46
5. Conclusion .....	46
<b>Chapitre III : Internet des Objets .....</b>	<b>47</b>
1. Introduction.....	48
2. Définition .....	48
3. Radio-identification (RFID).....	50
4. Architecture de l'internet des objets .....	50
5. Conclusion .....	51
<b>Chapitre IV : Conception &amp; Implémentation .....</b>	<b>52</b>
1. Introduction.....	53
2. Conception .....	54
2.1. Architecture du système.....	54
2.1.1. Présentation du système EgressVR .....	54
2.1.2. Diagramme de classes .....	55
2.1.3. Processus proposé d'évacuation.....	56
2.2. Intégration BIM .....	58



2.2.1. Importation & mise en place de l'environnement BIM .....	58
2.2.2. Metadonnées BIM .....	59
2.3. Représentation du bâtiment.....	59
2.4. Solution proposée.....	61
2.4.1. Description de la solution.....	61
2.4.2. Fonction d'évaluation des coûts de transit .....	62
2.4.3. Algorithme d'évacuation.....	62
2.5. Comparaison par rapport au modèle de flots conventionnel .....	65
3. Implémentation .....	66
3.1. Environnement de développement.....	66
3.2. Principe de l'application EgressVR .....	67
3.2.1. Application Serveur.....	67
3.2.2. Entraînement & simulation .....	68
4. Tests & résultats.....	69
4.1. Approche d'évaluation.....	69
4.2. Scénario de test .....	70
4.3. Résultats & Observations.....	71
4.3.1. Choix de paramètres.....	71
4.3.2. Indice TEC .....	72
5. Conclusion .....	73
<b>Conclusion &amp; Perspectives .....</b>	<b>74</b>
1. Conclusion .....	74
1.1. Objectifs atteints .....	74
1.2. Difficultés rencontrées .....	75
2. Perspectives.....	75
<b>Bibliographie .....</b>	<b>1</b>

# Liste des figures

Figure 1 – Les systèmes BeIM (Tah J. et al., 2017) .....	7
Figure 2 – Définition du BIM (Yakami M., 2016) .....	8
Figure 3 – Exemple des détails techniques d’un modèle BIM (Eastman C., 2008) .....	8
Figure 4 – détection des problèmes de collision dans le BIM (Eastman C., 2008).....	9
Figure 5 – l’Analyse d’un modèle BIM (Eastman C., 2008).....	10
Figure 6 – Synthèse sur l’état des composants d’un Modèle BIM (Eastman C., 2008).....	10
Figure 7 – Cycle de vie d’un désastre (Alexander D., 2002).....	12
Figure 8 – Le vecteur VA résultant de l’effet de l’accélération a sur A pour éviter la collision avec B (Okazaki et al., 1993).....	15
Figure 9 – Exemple de simulation d’évacuation du modèle magnétique (Okazaki et al., 1993) .....	15
Figure 10 – Force résultante des piétons à proximité (Zeng W. et al., 2014).....	17
Figure 11 – Comportement d’un agent modélisant un évacué (Wang J. et al., 2014).....	20
Figure 12 – Champ de vision d’un agent (Zhang L. et al., 2014).....	21
Figure 13 – Exemple d’un réseau de flots d’évacuation (Pyakurel U. 2017).....	22
Figure 14 – Solution générée par le champ potentiel (Richardson O. et al., 2019).....	23
Figure 15 – Définition des Serious Games (Marer P. et al., 2015).....	35
Figure 16 – Unity runtime game engine architecture (Gregory J., 2009).....	37
Figure 17 – Classes Principales de UnrealEngine (Extraite de wiki.unrealengine.com/ue4_class_diagram).....	39
Figure 18 – Encodage de l’arbre de comportement pour <i>Exemple 1</i> . (Extrait de : Colledanchise M. et al., 2018). .....	42
Figure 19 – Blackboard pattern (Lalanda, P., 1997).....	43
Figure 20 – Équipements de la réalité virtuelle .....	44
Figure 21 – Exemple d’un rendu dans la RV (Makransky G. et al., 2017). .....	44
Figure 22 – Les six degrés de liberté (Muhanna A. M., 2015).....	45
Figure 23 – Khronos Open XR pour la RV/RA multiplateforme.....	46
Figure 24 – Skin Embedded RFID tag (Sethi P. et al., 2016).....	49
Figure 25 – Web Connected RFID (Srinidhi N. et al., 2018).....	49
Figure 26 – RFID Communication (Mashal I. et al., 2015) .....	50
Figure 27 – IoT 5-Layer Architecture (Mehta R. et al., 2018) .....	51

Figure 28 – Architecture du système EgressVR.....	55
Figure 29 – Diagramme de classes EgressVR.....	56
Figure 30 – Processus d'évacuation EgressVR.....	57
Figure 31 – Modèle de réseaux de flots amélioré.....	60
Figure 32 – Les maillages de navigation.....	61
Figure 33 – L'arbre de comportement de l'agent guide.....	64
Figure 34 – L'interaction agent guide / agent de contrôle de flux.....	65
Figure 35 – L'environnement de développement de Unreal Engine 4.....	66
Figure 36 – Exemple de Unreal Engine 4 Blueprint Visual Script pour la RV.....	67
Figure 37 – Interface Serveur EgressVR.....	68
Figure 38 – Plateforme de réalité virtuelle EgressVR pour l'entraînement à l'évacuation.....	69
Figure 39 – Le plan 2D du modèle BIM relatif au scénario des tests.....	71
Figure 40 – La variation du temps d'évacuation par rapport à la variation de $\varepsilon$ .....	72

# Liste des tableaux

Table 1 – Les Axes de recherche de l'évacuation humaine.....	11
Table 2 – Solutions informatiques de l'évacuation indoor (Kuligowski E. et al. 2010) .....	28
Table 3 – Types de nœuds d'un arbre de comportement (Extraite de : Colledanchise M. et al., 2018) .....	41
Table 4 – Comparaison de la solution proposée par rapport au modèle de flots.....	65
Table 5 – L'indice de performance TEC pour la solution proposée EgressVR .....	73

# **Introduction**

# **1. Contexte et motivation**

Depuis quelques années, la recherche dans le domaine de la gestion des foules et de l'évacuation se concentre sur l'utilisation d'anciennes méthodes et outils de modélisation, tels que la formulation mathématique du problème et l'utilisation d'algorithmes procéduraux discrets. Dans le cadre de cette étude, nous visons une autre voie pour aborder et résoudre ce problème. Ce travail vise de simplifier la perception du processus de modélisation de l'évacuation et mettre l'accent sur le rôle de nouvelles sources d'informations tel que Building Information Modeling, et offrir une expérience immersive du système via la réalité virtuelle. Ce travail est motivé par la mise en œuvre des outils de modélisation et de programmation offerts par les moteurs de jeux afin de profiter de leur efficacité dans la résolution de problèmes complexes. L'intelligence artificielle étant une solution révolutionnaire de nos jours, nous inspirons à l'utilisation des agents de l'intelligence artificielle en nous basant sur les arbres de comportement pour réussir la simulation comportementale relative à l'évacuation des personnes.

## **2. Enoncé du mémoire**

### **2.1. Problématique**

L'évacuation des personnes coincées dans un bâtiment a présenté un problème de recherche à plusieurs dimensions. Le but étant de formaliser le phénomène d'évacuation en un processus afin d'optimiser et de réussir la gestion des foules, et éventuellement réduire ou éviter les pertes humaines. De nombreux modèles d'évacuations proposés par les chercheurs présentent des formulations mathématiques et des solutions algorithmiques procédurales ainsi que des solutions de l'intelligence artificielle pour la résolution de ce problème. Toutefois, aucun d'entre eux ne représente réellement le comportement de la population évacuée lors de l'évènement et a tendance à être discursif en se basant sur des formules mathématiques bien précises et des données statistiques, les rendant vulnérables aux tournures inattendues des événements et la population défiant le plan d'évacuation.

Ce travail vise à explorer les différents travaux existants dans ce cadre, et d'en inférer les propositions les plus pertinentes pour la proposition d'un nouveau modèle et la mise en place d'un système d'évacuation guidée basée sur la simulation comportementale en temps réel. Nous proposons l'utilisation des méthodes de modélisation et paradigmes de

programmation nouvelles pour simplifier, décrire et résoudre le problème de l'évacuation humaine.

## 2.2. Postulats

La formation à l'évacuation nécessite des connaissances de la discipline d'évacuation. Ainsi, la présence d'un expert du domaine lors des exercices d'évacuation est indispensable. Il est supposé aussi que dans le cas réel, un system de positionnement (Indoor Positioning System) aura lieu pour lire les positions des évacués en temps réel.

## 2.3. Buts et objectifs

Ce travail a comme objectif un ensemble de buts à atteindre, les principaux jalons à achever étant comme suit :

- Simplifier le problème d'évacuation et offrir un modèle compréhensif du processus d'évacuation.
- Trouver une solution dynamique au problème d'évacuation capable de faire face à l'évolution de l'événement d'urgence ainsi qu'à celle de la population concernée.
- Créer un environnement virtuel d'entraînement à l'évacuation.
- Gérer la dynamique de la foule et résoudre les problèmes de congestion
- Optimiser et évaluer la solution proposée par rapport aux solutions existantes.

## 3. Plan du mémoire

- **Chapitre 1 : Etat de l'art**

Ce chapitre vise d'identifier les axes de recherche de la thématique d'évacuation. Cerner les différents modèles d'évacuation mis à l'œuvre et de décrire les algorithmes concernés. Une étude sur les solutions existantes d'évacuation est présentée. Enfin, notre direction par rapport à l'existant est mise en exergue.

- **Chapitre 2 : Serious Games & Moteurs de Jeux**

Le chapitre 2 introduit le concept de « Serious Games » (SG) en tant qu'une solution efficace pour l'acquisition de connaissances. Les SG sont à la base de la formation virtuelle d'évacuation. Une vision sur les moteurs de jeux serve de décrire leur structure, et d'identifier les éléments de base pour la création d'un jeu sérieux inspiré par la réalité virtuelle. On montre

aussi la contribution des moteurs de jeux à la recherche scientifique et au processus de développement.

- **Chapitre 3 : Internet des objets**

Ce chapitre introduit le concept de l'internet des objets (IoT) comme un outil de collecte d'informations. Il met l'accent sur l'architecture en couches de l'IoT et la technologie sensorielle pour l'identification de l'état du monde réel.

- **Chapitre 4 : Conception & Implémentation**

Le chapitre 4 est consacré à la conception du Système d'évacuation EgressVR. Il vise de présenter une vue générale sur le processus d'évacuation ainsi qu'une vue détaillée de l'architecture du système et ses composants de fondation. Il comprend une partie qui est destinée aux Tests relatifs au bon fonctionnement du système. Elle propose un framework, soit TEC, qui est destiné à décrire la performance du système par une métrique calculée d'un certain nombre de paramètres proposés.

- **Conclusion & Perspectives :**

Une conclusion sur le travail réalisé. Un rappel d'objectifs atteints et les difficultés rencontrées ont lieu. Cette partie comprend également une discussion sur les futures perspectives pour la thématique d'évacuation indoor.



# **Chapitre I : Etat de l'art**

## **1. Introduction**

Le phénomène d'évacuation et ses sujets de recherche allant d'études empiriques à la simulation et l'optimisation, en passant par la modélisation se sont multipliés au cours de la dernière décennie (Vermuyten H. et al., 2016). Les chercheurs s'intéressent principalement à l'analyse du comportement de la foule, ainsi qu'à la résolution du problème de l'évacuation en tenant compte des deux facteurs de l'accompagnement des occupants vers la sécurité, ces derniers étant la gestion des foules et l'optimisation du temps d'évacuation. Par conséquent, la principale préoccupation de ce chapitre est d'explorer la manière dont les travaux précédents ont défini et traité le comportement de la foule dans le contexte du processus d'évacuation en citant différents modèles utilisés, en mentionnant leurs caractéristiques uniques, et éventuellement, montrant leurs inconvénients. Il commence également par une brève introduction à la modélisation des informations sur le bâtiment (BIM) et met l'accent sur ses caractéristiques et les données disponibles. Une étude sur les solutions existantes d'évacuation a été mise en place afin d'identifier les principaux modèles. Enfin, une conclusion afin de préciser notre direction principale par rapport aux travaux existants et montrant la solution choisie pour la conception et l'implémentation est donnée.

## **2. Building Information Modeling**

### **2.1. Environnement bâti**

Le monde, tel qu'il est connu, subit des changements rapides en temps réel. Nous nous entendons, le point de vue architectural où l'information est importante pour le stockage à des fins de représentation et d'abstraction. Le but est de faciliter l'accès et la modification de celle-ci, ainsi que la création et l'intégration de nouveaux éléments. Cette information étant parfois introuvable, implicite, difficile à comprendre ou à extraire implique le besoin à un modèle de données global, cohérent et riche en information pour la représentation du monde réel. Le BIM (Building Information Modeling) ou la modélisation d'informations des bâtiments est une partie de la modélisation d'informations de l'environnement bâti (voir Figure 1), appelée aussi BeIM (Built environment Information Modeling) (Tah J. et al., 2017). Elle fournit la possibilité de représenter cette information à travers la collaboration entre les différents acteurs et individus. Elle encourage la collaboration grâce à son modèle de représentation qui améliore la productivité et la performance des projets courants en créant un environnement de travail connexe entre les membres agissant dans le projet. La croissance de l'utilisation du BeIM offre

la possibilité de l'intégration des systèmes indépendants associés aux différents aspects et échelles de l'environnement bâti pour délivrer des services et des solutions intelligentes pour l'amélioration du bien-être des citoyens (Tah J. et al, 2017). De nombreuses applications du BeIM ont apparues, y compris les moteurs de modélisation des villes en 3D à des fins techniques et artistiques, et le BIM sémantique qui formalise le modèle pour qu'il soit utilisé par la machine et les agents de l'intelligence artificielle.

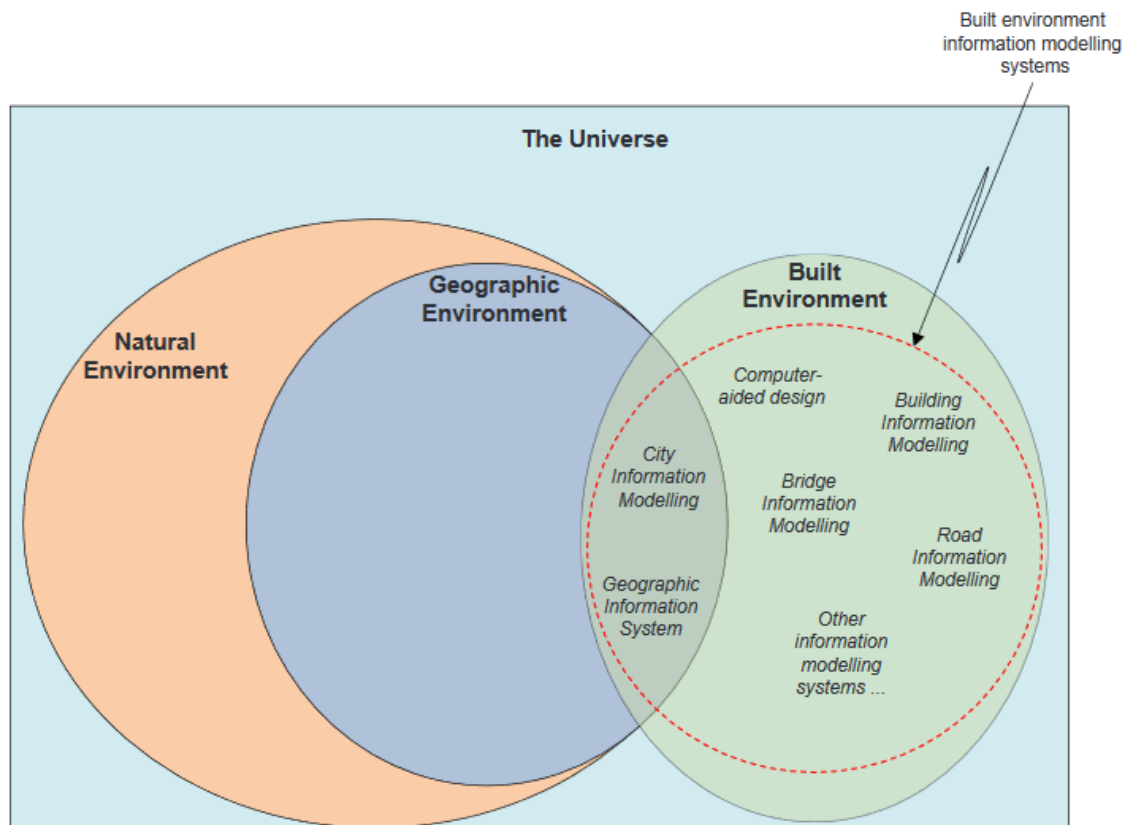


Figure 1 – Les systèmes BeIM (Tah J. et al., 2017)

L'environnement bâti (Built Environment) tel que Roof K. le définit de point de vue social, est l'espace créé par l'humain (Roof K, 2008). L'environnement bâti touche une grande partie du domaine d'ingénierie, y compris l'architecture, le design industriel, la géographie et l'urbanisme. Ainsi, toute information portée par les instances, qu'elles soient architecturales (tel que les bâtiments), industrielles (machines, entités smart, etc.) ou urbaines (routes, l'éclairage, canaux, réseaux, etc.) fait partie du BeIM. Le secteur industriel concerné par ces aspects interdisciplinaires inclut l'AEC (Architecture, Engineering and Construction) qui utilise le BIM qui, grâce à sa représentation paramétrique orientée-objet, participe à

l'augmentation de l'efficacité et réduit les coûts de livraison du projet en question (Tah J. et al, 2017). D'autres utilités ont été proposés, notamment, l'intégration du BIM à l'éducation, et aux règlements de la loi de l'état (Violeta N. et Arta B., 2017).

## 2.2. Les solutions BIM

Plusieurs définitions du BIM existent, l'une des plus populaires est celle qui le définit en tant que verbe ou phrase adjectif pour décrire les outils, processus et technologies facilités par une documentation numérique lisible par machine, concernant un bâtiment et ses performances, sa conception, sa construction et son fonctionnement (Eastman C. et al., 2011, p.467). Ainsi, Le BIM est un modèle 3D pour les architectes et ingénieurs (Levy S. M., 2010). Il est un modèle orienté-objet, intelligent, riche en information et en détails techniques sur le bâtiment représenté (Eastman C., 2008) (voir Figure 3). Il représente un processus de planification, conception, construction et de gestion efficace des bâtiments et leurs infrastructures dans le cycle de vie du projet (Yakami M., 2016) (voir Figure 2).

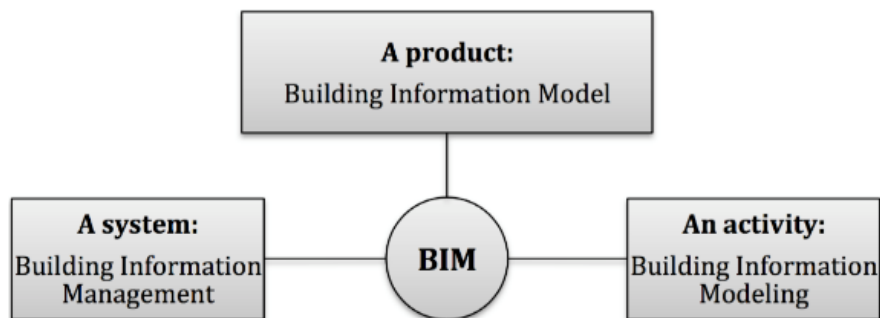


Figure 2 – Définition du BIM (Yakami M., 2016)

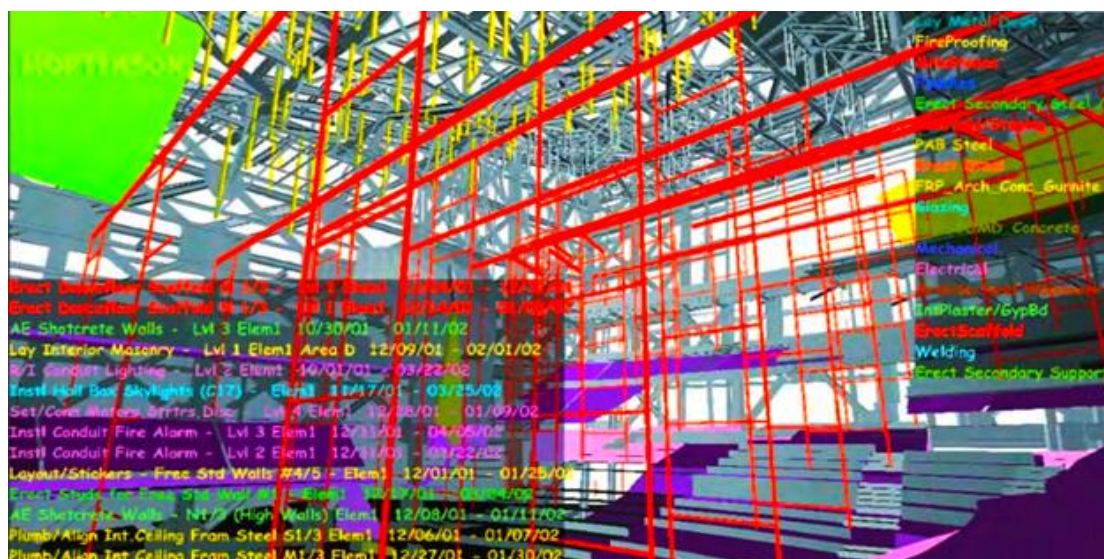


Figure 3 – Exemple des détails techniques d'un modèle BIM (Eastman C., 2008)

Le BIM ressemble beaucoup aux techniques de CAD (Computer-Aided Design) utilisés pour concevoir des produits mécaniques et électriques de point de vue de la représentation 3D, mais lui est différent de point de vue de l'objet conçu (n'est pas limité à l'architecture bâtiments) et la méthode de travail. Il peut aussi faire des analyses et détecter les problèmes fonctionnels dans les divers éléments de conception (voir Figure 4, 5 et 6) (Kia S. et al., 2013). La modélisation paramétrique orientée-objet développée dans les années 80 du 20ème siècle était la base du BIM. Elle représente des objets à l'aide des paramètres et des règles qui déterminent la géométrie ainsi que certaines caractéristiques et des propriétés non-géométriques. Ces paramètres et règles permettent la mise à jour des objets selon les changements apportés par l'utilisateur ou ceux du contexte (Eastman C. et al., 2008, p.25).

De nombreux formats d'échange existent pour le modèle BIM, où les plus populaires sont, IFC (buildingSmart Industrial Foundation Class) utilisé par la majorité des applications dans le domaine de l'AEC, et XML dont on cite CityGML, un moteur de création et de modélisation de villes (Liu X. et al, 2017).

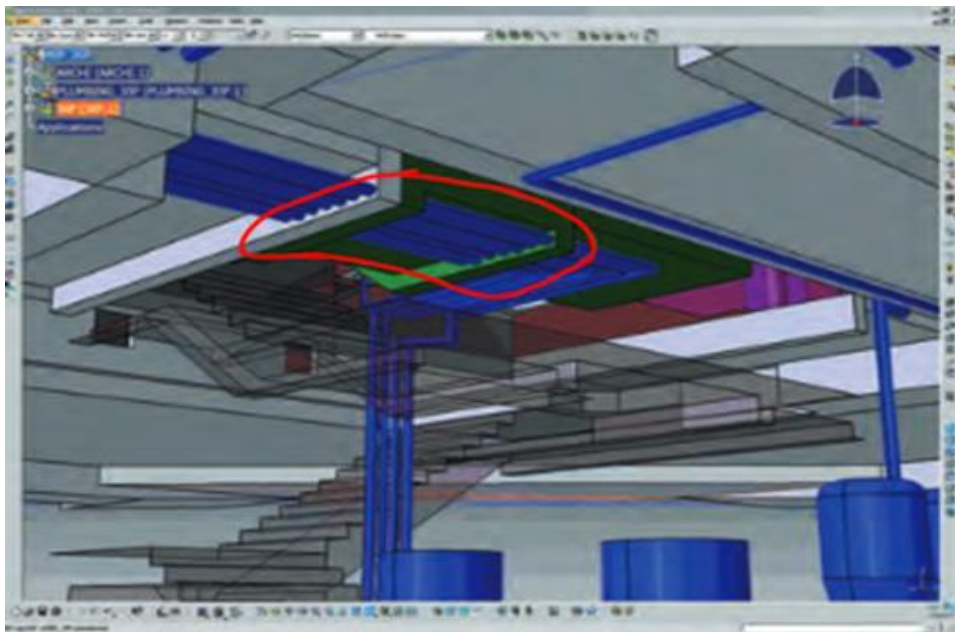


Figure 4 – détection des problèmes de collision dans le BIM (Eastman C., 2008)

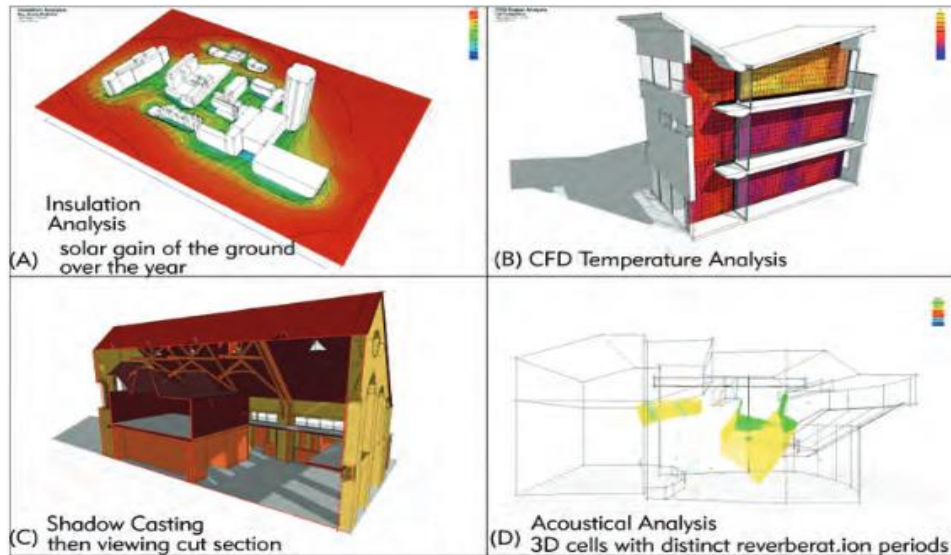


Figure 5 – l'Analyse d'un modèle BIM (Eastman C., 2008)

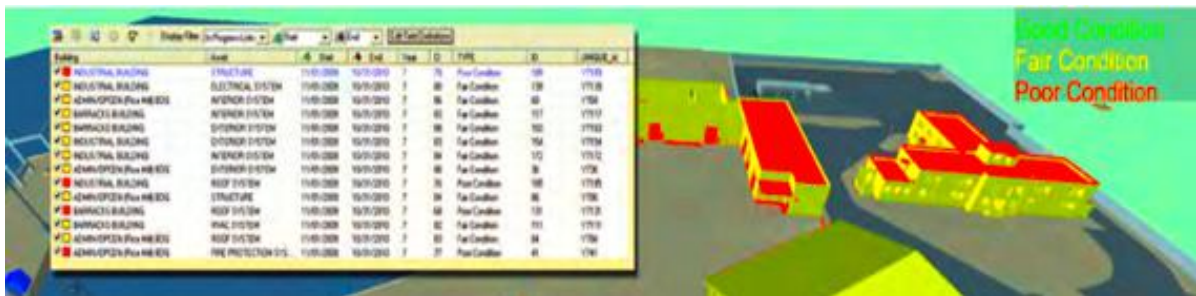


Figure 6 – Synthèse sur l'état des composants d'un Modèle BIM (Eastman C., 2008)

Les applications basées sur le BIM sont largement répandues, dont la plupart sont des environnements de travail BIM (BIM tools) tel que Autodesk® Revit, Tekla BIMsight®, et GRAPHISOFT® ARCHICAD. Il existe des applications qui utilisent l'information du BIM ou l'intègre avec d'autres types d'informations pour achever la finalité du système en question, des mentions honorables comprennent Bentley® Hevacomp, qui se spécialise à la simulation et la création des infrastructures économes en énergie. Relativement à la thématique de l'évacuation, on mentionne BIM-VE, un environnement virtuel d'évacuation en cas d'incendie (Wang B., 2014).

### 3. Aspects de recherche de la thématique de l'évacuation

Le problème d'évacuation peut être découpé en quatre principaux aspects de recherche, la théorie d'évacuation, la modélisation d'évacuation, la prise de décision d'évacuation, et l'évaluation du risque d'évacuation (Wang J. & Sun J., 2014) comme le montre la Table 1.



Research aspects	Main contents	Some typical work
Evacuation theory	Correlations of evacuation parameters	Fruin et al. ; Daamen et al. ; Pauls et al.
	Laws of human evacuation behavior	Helbing et al.; Ma;
Evacuation modeling	Mathematical models	Okazaki et al. ; Helbing et al. ; Cruz et al. ; Lu et al.
	Simulation models	Kisko et al. ; Sheffi et al. ; Hobeika et al. ; Tufekci et al. ; Owen et al. ; Lo et al. ;
Evacuation decision-making	Evacuation route selection	Cova et al. ; Tuydes et al.
	population assignment	Gao et al.;
	resource allocation	Fiedrich et al.
Evacuation risk evaluation	Gathering risk	Fraser
	Traffic risk	Church et al.; Qiang et al.

Table 1 – Les Axes de recherche de l'évacuation humaine.

### 3.1. La théorie d'évacuation

Elle concerne l'étude de plusieurs paramètres associés au mouvement de personnes lors de l'évacuation. Elle se base sur l'observation ou tests et simulations afin de visualiser l'impact de celles-ci sur le mouvement de la foule ainsi que les lois du comportement d'évacuation humaine. Ces lois visent de décrire le type du comportement de l'individu et le groupe, de point de vue de la psychologie comportementale et à des fins de reproduction ou de la simulation des mouvements des particules dans un environnement virtuel (Wang J. & Sun J., 2014).

Les théories existantes ne prennent pas en considération tous les paramètres. Parfois, elles sont basées sur des hypothèses contradictoires comme celles de la théorie de la prise de décision et la théorie de panique qui supposent la capacité ou non des évacués à agir rationnellement lors d'un évènement (Wang J. & Sun J., 2014). Cependant, avec le développement rapide de la technologie et la disponibilité de données et d'informations, un

nombre de chercheurs travaillent sur l'enrichissement des théories existantes dont Helbing et al., Li J. et al. (Tel que cité dans Wang J. & Sun J., 2014).

### 3.2. La modélisation de l'évacuation

Qu'il s'agit d'un désastre, sinistre, une calamité ou d'une catastrophe, ces concepts sont considérés comme des événements qui nécessitent une réaction rapide et des mesures permettant d'éviter tout dégât humain ou corporel. Une situation d'urgence est donc en jeu. Ces événements étant répétitifs génèrent un cycle de vie perceptible muni de quatre phases, qui sont respectivement, la mitigation ou l'atténuation, la préparation (preparedness), la réponse et la reprise (Alexander D., 2002) (voir Figure 7).

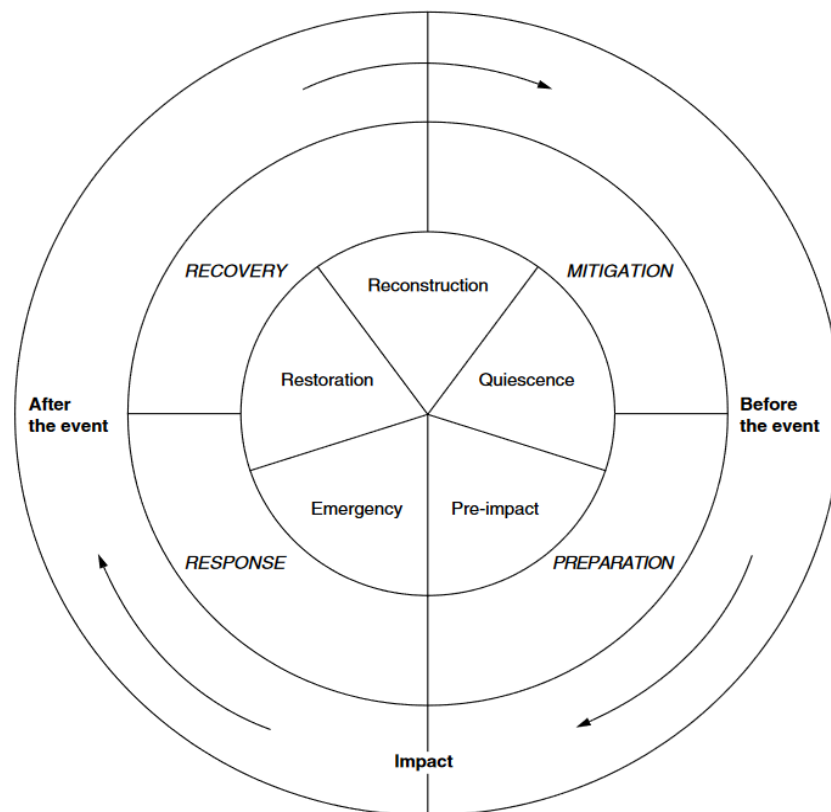


Figure 7 – Cycle de vie d'un désastre (Alexander D., 2002)

La mitigation comprend toutes les actions et mesures préventives pour réduire ou éviter les catastrophes futures. Ces premiers incluent les solutions structurelles tel que celles de l'ingénierie de la sécurité des bâtiments, et des solutions non-structurelles tel que l'assurance et le plan d'évacuation (Alexander D., 2002). Cependant le plan d'évacuation étant fait au préalable rend ce premier statique et ne répond pas aux tournures imprédictibles des événements. Ceci dit, il est nécessaire d'avoir l'avis d'un expert sur les chemins supposés d'être



sûrs. Ce dernier peut ne pas être disponible lors de l'événement ou n'a pas les ressources nécessaires pour analyser rapidement la situation et d'escorter l'ensemble des occupants à l'abri. Une solution informatique capable de gérer efficacement ce flux d'information est indispensable. Elle permet la mise en place d'un plan d'évacuation dynamique à la phase de réponse.

Les modèles d'évacuation sont généralement découpés en deux parties par utilité: Les modèles relatifs à l'évacuation civile de masses, et ceux relatifs à l'évacuation de personnes dans les bâtiments en danger (Wang J. & Sun J., 2014). L'auteur mentionne aussi un découpage en deux modèles, l'un mathématique (description quantitative pour prédire le mouvement des occupants), l'autre dit de simulation, qui utilise des outils de simulation informatique développés à partir des modèles mathématiques. Quelques travaux sur les modèles mathématiques comprennent, le modèle magnétique (Okazaki et al., 1993), le modèle de forces sociales (Helbing D. et al., 1995), le modèle de file d'attente dans l'agenda de Cruz F.R.B., le modèle d'optimisation dans l'agenda de Cova, et le modèle de flux de foule continu dans l'agenda de Henderson L. (cités dans Wang J. & Sun J., 2014).

Autres auteurs ont découpé les modèles selon quatre dimensions, le niveau de granularité de représentation des évacués (Modèle microscopique: par individu; Modèle macroscopique: en masse; Modèle mésoscopique: traite à la fois des individus et des groupes d'individus), la structure du modèle (continu: espace 2D ou 3D; discontinue: graphe), le comportement de l'occupant (implicite; conditionnel; probabiliste), et la méthode de modélisation (Comportementale complète: incorpore toutes les actions de l'occupant; Partielle: calcule seulement le mouvement de l'occupant), l'agenda de Kuligowski E.D. (tel que cité dans Tan K.Y., 2011; Manley M.T., 2012).

Les modèles macroscopiques, tels que le modèle de réseaux de files d'attentes, concentrent seulement sur la perspective de la foule. Ils concernent l'allocation d'espace pour les personnes dans les installations piétonnes, et ignorent les interactions entre les évacués, et donc, ne prennent pas en compte l'hétérogénéité des évacués et le comportement individuel. Ceci, a donné naissance aux modèles microscopiques, appelés aussi modèles microscopiques de simulation, qui se découpent en deux modèles en considérant leur structure, (1) continue, comme celui du modèle magnétique et de forces sociales, (2) discret, comme celui du modèle d'automates cellulaires (cellular automata) dans l'agenda de Blue V. (Chen X. et al., 2017). Ce type de modèle prend en charge toutes les interactions entre les évacués résultant en une

meilleure circulation de la foule. Cependant, le modèle de réseaux de files d'attentes et les modèles macroscopiques en général restent utiles pour la détection de points de congestion dans un plan d'évacuation (Cruz F.R.B et al., 2005). Ceci dit, les modèles macroscopiques sont plus performants mais ne peuvent pas remplacer certains avantages offerts par les modèles microscopiques. Ainsi, le modèle le plus répandu de nos jours est bien le modèle mésoscopique qui combine les deux visées (totale et individuelle) sur la dynamique de la foule.

Du point de vue de la recherche et la thématique de la théorie d'évacuation, on distingue trois axes de recherche concernés par le problème en question, Le premier volet est consacré à l'étude empirique du comportement des piétons et de la dynamique des foules. Le second est concerné par le développement de modèles mathématiques pour décrire de manière aussi réaliste que possible les mouvements et les interactions des piétons (l'agenda de Teknomo K. tel que cité dans Vermuyten H. et al., 2016). Enfin, le troisième flux de la recherche utilise une méthodologie basée sur l'optimisation pour développer des modèles qui déterminent des plans d'évacuation optimaux ou des solutions de conception (l'agenda de Abdelghany cité dans Vermuyten H. et al., 2016). (voir plus loin dans la section 3.5).

### 3.3. Le modèle magnétique

Dans ce modèle chaque piéton est considéré comme un objet magnétique doté d'une charge positive. Il est de même pour les murs et les obstacles, alors que le point objectif ou la sortie est dotée d'une charge négative. Chaque piéton se déplace vers la sortie la plus proche de lui grâce à la force d'attraction entre les deux charges. Il évite les obstacles grâce à la force répulsive entre ces deux derniers (Okazaki et al., 1993). La formule de la force exercée est calculée par la loi de Coulomb (1).

- $F = (k \cdot q_1 \cdot q_2 / r^3) \cdot r$  ... (1)
- $F$  : vecteur de force magnétique
- $k$  : constante
- $q_1$  : charge du piéton
- $q_2$  : charge d'un pôle magnétique
- $r$  : distance entre  $q_1$  et  $q_2$

Le modèle lit d'abord les informations liées à l'environnement ainsi que celles des piétons. Les données et paramètres qui représentent l'environnement incluent le positionnement des murs, coins, fenêtres et sorties. Les coins sont parfois considérés comme des cibles

transitoires lorsque le plan d'évacuation est complexe. Les données relatives au piéton comprennent la destination, vitesse, et orientation. Une méthode de marche décrite par des fonctions mathématiques, détermine le plan de mouvement du piéton, qui est soit une route indiquée (dans le cas des coins), le chemin le plus court (une visée linéaire vers le but), ou la découverte (but indéfini).

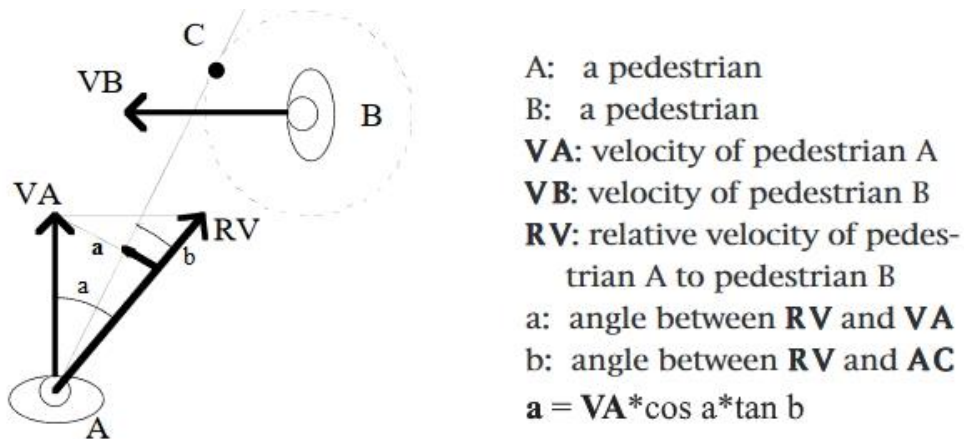


Figure 8 – Le vecteur VA résultant de l'effet de l'accélération a sur A pour éviter la collision avec B (Okazaki et al., 1993)

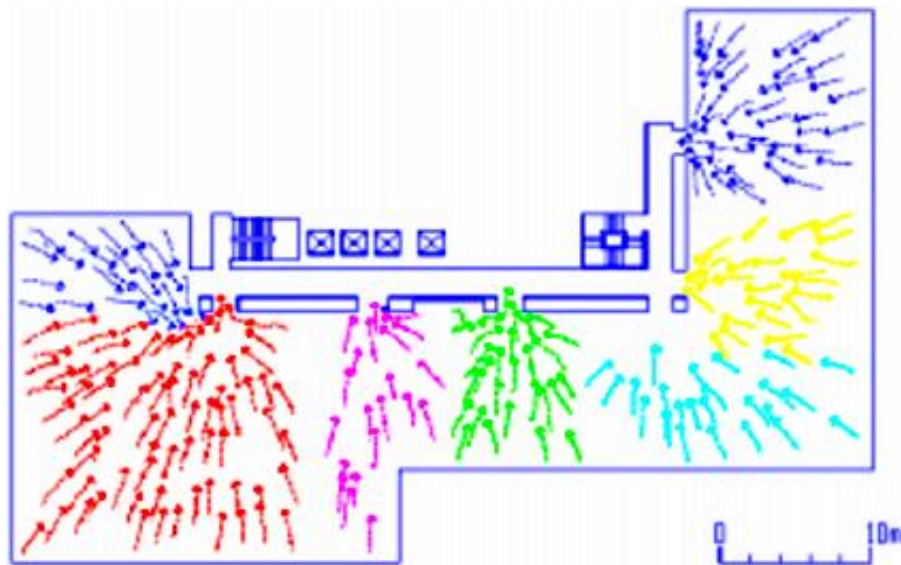


Figure 9 – Exemple de simulation d'évacuation du modèle magnétique (Okazaki et al., 1993)

Le modèle magnétique décrit ci-dessus est conçu dans le but de simuler un plan d'évacuation qui consiste à attirer les gens vers les sorties et à les éloigner les uns des autres ainsi que des obstacles à l'intérieur du bâtiment. Cependant, le problème des modèles mathématiques est le manque de pouvoir exprimer tous les phénomènes et les situations pouvant résulter en conséquence. Les formules mathématiques et les contraintes ne suffisent

pas pour simuler tous les cas réels du scénario. Par exemple, étant donné que l'orientation de l'occupant vers la sortie est affectée par la force des autres obstacles et par la sortie elle-même. Si un occupant est orienté perpendiculairement vers un obstacle et vers la sortie en même temps, pour ainsi dire sur la même ligne, la composition de tous les autres vecteurs de force étant nuls. L'occupant entre dans un état d'équilibre qui le pousse à rester immobile et à ne pas bouger. De plus, l'injection d'une force légère au voisinage de l'occupant dans une direction qui n'est pas maintenue par la même droite que celle portée par l'orientation de l'occupant pourrait s'avérer utile pour rompre l'état d'équilibre. Toutefois, une décision sur l'étendue et la direction de la force ne pourrait pas être faite parce qu'une mauvaise décision pourrait conduire l'occupant à être pris à une impasse ou peut prolonger le temps d'évacuation, résultant ainsi en une solution d'évacuation non optimale. Enfin, on pourrait dire la même chose pour les cas de bâtiments complexes où l'occupant pourrait être entraîné dans une impasse, résultant de l'attraction irrésistible de la sortie et de la force de répulsion des obstacles et autres occupants, rendant ainsi le modèle magnétique viable uniquement pour des scénarios de cas simples.

### 3.4. Le modèle des forces sociales

Ce modèle est inspiré par les mêmes hypothèses suggérées par le modèle magnétique. Il suppose que chaque personne se comporte agressivement envers d'autres évacués, et donc une force répulsive dite sociale, est prise en considération. Cependant, il dénote que l'évacué peut subir des forces attractives envers les personnes qu'il connaît ou admire. Cette force est sujet à décroître sur l'abscisse du temps. L'intensité des forces attractives et répulsives est relative à l'orientation de l'évacué, et donc, l'influence d'autres personnes se trouvant hors du champ de vision de l'évacué a un impact faible. Le mouvement de l'évacué est supposé de prendre la forme d'un polygone (dans le cas complexe), où il se va déplacer d'un sommet à l'autre sur le bord de ce dernier pour arriver à la sortie. En prenant en compte les paramètres proposés et les variations aléatoires du comportement de l'évacué, un vecteur de mouvement appelé de motivation qui est la somme de tous les vecteurs de forces (voir Figure 10), et une valeur de fluctuation, forment le modèle de forces sociales, qui est défini dans l'équation (1) :

$$\frac{d\vec{\omega}_\alpha}{dt} = \vec{F}_\alpha(t) + \text{fluctuations} \quad (1)$$

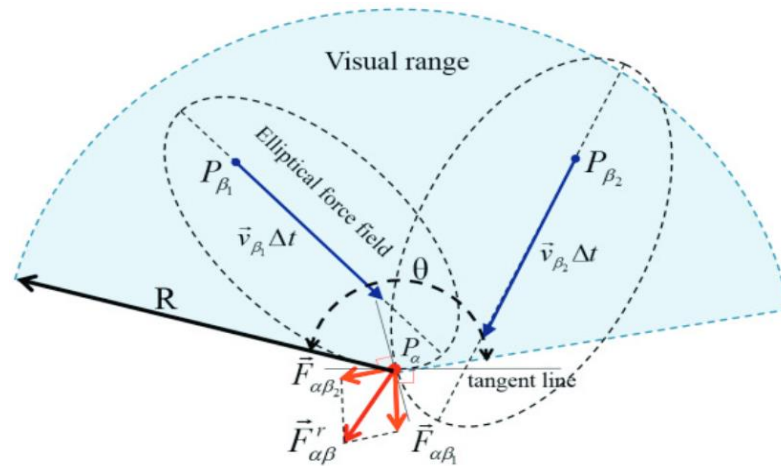


Figure 10 – Force résultante des piétons à proximité (Zeng W. et al., 2014)

Des exemples pour l'élaboration de ce modèle inclut les travaux de Zeng W. et al. (2014) pour la simulation du comportement de piétons dans les passages cloutés, afin de minimiser les accidents de circulation (collision des piétons avec les véhicules). Helbing D. et al. (2005) ont incorporé le concept de nervosité au modèle pour inclure le changement de vitesse des piétons influencé par l'environnement, l'agenda de Helbing (tel que cité dans Chen X. et al., 2017).

Le modèle de forces sociales est basé sur les mêmes contraintes du modèle magnétique, en ajoutant seulement les forces nouvellement définies pour exprimer les relations entre les individus à l'intérieur du bâtiment. Les mêmes anomalies d'état d'équilibre et d'optimalité peuvent être notées. Elles rendent ainsi le modèle vulnérable aux situations complexes du monde réel. De plus, l'identification des relations entre des individus au sein d'une population dynamique pourrait s'avérer très difficile du point de vue de la mise en œuvre. Les gens se sentent plus inquiets à propos de leurs vie privée, de sorte que les forces sociales exprimées par ce modèle deviennent moins précises et peuvent amener le cas réel à s'écarter du comportement attendu de la simulation, tel que prévu par un plan d'évacuation. Enfin, avoir besoin à identifier ces relations entre tous les évacués entraîne une augmentation exponentielle du volume de données et de la complexité des calculs, entraînant de mauvaises performances lorsque la population augmente (Millà-Gutiérrez A., 2016).

### 3.5. Le modèle d'optimisation

Le modèle d'optimisation est le plus abordable aux solutions informatiques d'évacuation. Contrairement aux modèles mathématiques qui sont destinés à l'étude et la

simulation du comportement de la foule, il vise à proposer une solution (chemin) optimale d'évacuation à suivre au lieu de se contenter avec le chemin prédictible par la foule. Il décrit un ensemble d'instances (sous-modèles) dont l'intérêt de chaque instance est l'optimisation d'un critère spécifique (ou une multitude) relatif à l'évacuation. Ces critères sont bien la distance parcourue (chemin plus court), le temps total d'évacuation (chemin plus rapide), et le taux de réussite de l'évacuation (chemin plus sûr). Bayram V. (2016) a examiné la littérature sur les modèles d'optimisation et a identifié quatre approches de base dans ce contexte: l'allocation la plus proche (Nearest Allocation), équilibre d'usagers (User Equilibrium), optimalité du système (System Optimal), et optimalité restreinte du système (Constrained System Optimal).

Le modèle d'allocation la plus proche (Nearest Allocation) est un modèle microscopique essentiellement utilisée pour la planification et la gestion du trafic d'évacuation. Ce modèle s'inspire du comportement individuel égoïste humain, et ainsi, chaque évacué tente d'atteindre l'abri sûr le plus proche. Dans une représentation graphique  $G(A, N)$ , la même que celle utilisée par le modèle d'automate cellulaire (où  $G$  est un graphe doté d'un ensemble de nœuds  $N$  et un ensemble d'arcs  $A$ ), qui représente une abstraction du site d'évacuation. Le plan coïncide avec la recherche du chemin le plus court (pour chaque personne) que certains algorithmes visent de résoudre tel que celui de Dijkstra et l'algorithme  $A^*$  (voir plus loin à la section 4.2).

Le modèle d'équilibre d'usagers (UE) est basé sur le premier principe de Wardrop qui dit que « le temps de trajet sur tous les itinéraires réellement utilisés sont égaux, et inférieurs à ceux qu'un seul véhicule subirait s'il choisit une autre route ». Le système est dit dans un état d'équilibre d'usagers (Wardrop J. G., 1952). Contrairement au modèle d'allocation la plus proche, ce modèle dépend de la "perception" (expérience et évaluation) de la personne pour le choix du chemin le plus court ou le moins coûteux. Il ne propose pas une solution réelle pour un ensemble d'évacués ne possédant pas toutes les informations sur le site d'évacuation. Ceci dit, Wardrop n'a pas fourni d'algorithmes pour résoudre l'UE. Le problème étant la complexité des calculs de l'action suivante pour chaque personne en prenant en compte les actions d'autres piétons. Ce qui ressemble bien au concept de Nash Equilibrium, proposé par Nash J.F. (1951). Cependant, Wardrop a défini ces connaissances comme désirables.

Une variante du modèle d'équilibre d'usagers appelée équilibre d'usagers stochastique (SUE) proposée par Daganzo C.F. et al. (1977) vise d'assouplir la contrainte d'avoir une

connaissance parfaite concernant les coûts de déplacement, en modélisant les erreurs de perception à l'aide des variables aléatoires (Hazelton M.L., 1998).

Le modèle System Optimal (SO) met l'accent sur l'optimisation du temps total d'évacuation, en utilisant le deuxième principe de Wardrop selon lequel «la durée moyenne du trajet est minimale ». Le bénéfice total de la population par rapport à l'individu augmente. Il propose par conséquent des itinéraires parfois plus longs à certains individus que ceux qu'ils auraient choisi probablement. En conséquence, le système sera vulnérable aux défaillances produites par le stress lors de l'évacuation. Ceci du fait que les gens peuvent ne pas avoir un comportement conscient d'accepter des itinéraires beaucoup plus longs que les plus courts qu'ils perçoivent (Bayram V., 2016).

Le modèle Constrained System Optimal (CSO) améliore le modèle d'optimalité du système en imposant des contraintes liées aux préférences individuelles des évacués, ces contraintes étant un facteur de tolérance permettant de proposer une issue de secours acceptable ne se trouvant pas très loin du chemin le plus court et minimisant la congestion.

### **3.6. Le modèle multi-agent**

Le modèle multi-agent basé sur la modélisation d'agents de l'intelligence artificielle, dont le but est de simuler la pensée et le comportement d'un individu lors de l'événement de l'évacuation. Comme son nom l'indique, ce modèle traite chaque individu comme une entité à part ayant ses propres actions, et donc, peut être classé comme un modèle microscopique. Dans certains cas, l'ajout d'un agent superviseur qui gère des dimensions spécifiques tel que la congestion, ou le fait que les actions des agents peuvent avoir une relation avec l'état de toute la population en question, le modèle sera mésoscopique (du fait d'avoir incorporé la vue macroscopique de la modélisation).

Les agents dans le domaine de l'intelligence artificielle sont principalement conçus avec des organigrammes qui décrivent le comportement dudit agent lors d'un événement donné. Il est rendu plus sensible à l'état du monde virtuel afin d'atteindre ses objectifs et sa raison d'être lorsque certains critères sont remplis. Wang J. et al. (2014) ont conçu un modèle de simulation d'évacuation des piétons qui prend en compte la panique des individus en introduisant un indice de panique qui conduira l'agent à s'écarter de son objectif précédent consistant à éviter les embouteillages au profit du chemin le plus court ou de la sortie la plus proche (voir Figure 11). Le modèle implémenté est un modèle d'évacuation multi-agent avec différentes classes

d'agents (selon l'âge et le sexe) pour représenter l'hétérogénéité de la population et simuler une variance dans la prise de décision.

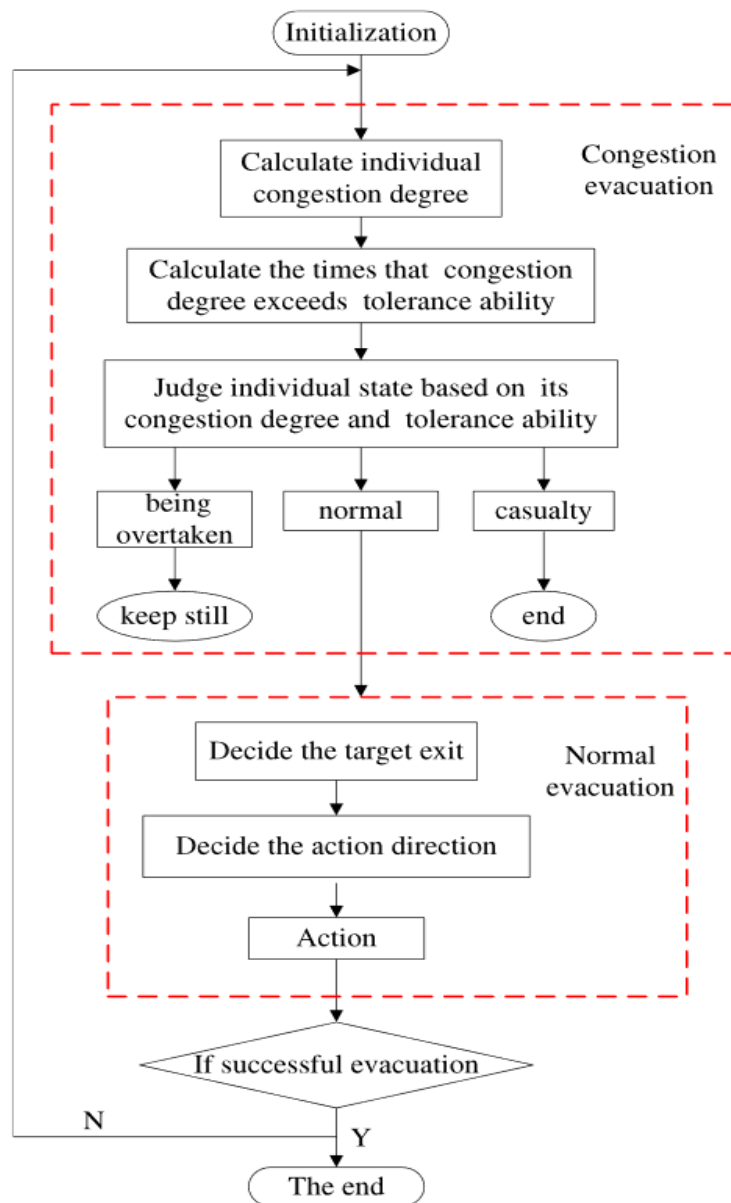


Figure 11 – Comportement d'un agent modélisant un évacué (Wang J. et al., 2014)

Tel que mentionné précédemment (section 3.1), la plupart des recherches sur l'évacuation se concentrent sur la compréhension de l'évacuation sous tous ses aspects, tels que l'évolution de l'urgence et le comportement de la foule (Helbing D., 2005). Elles tentent donc de simuler ces comportements soit en introduisant des données statistiques et des formules mathématiques, soit en concevant des agents intelligents artificiels qui se comportent exactement comme les occupants et font éventuellement les mêmes erreurs. La plupart des modèles, en plus des paramètres de simulation comportementale tels que la panique, prennent



également en compte le champ de vision de l'agent et les perturbations causées par le danger et les autres occupants à l'intérieur du bâtiment. Ce qui limite encore plus le champ de vision dudit agent (voir Figure 12).

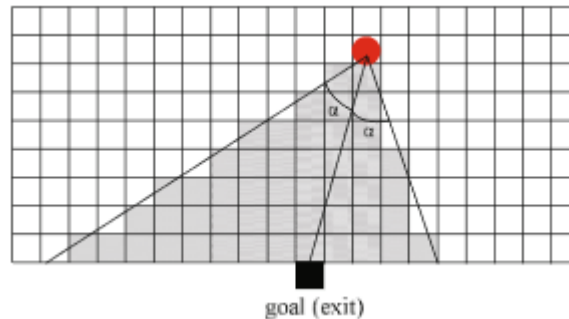


Figure 12 – Champ de vision d'un agent (Zhang L. et al., 2014).

Une autre limite viendrait du fait que l'agent est ici pour aider l'occupant du monde réel et non pas pour se comporter de la même manière que ce dernier, éventuellement en tant que guide. Ainsi, lorsqu'on envisage une solution guidée du problème d'évacuation basée sur l'intelligence artificielle, les agents modélisés à des fins de simulation sont jugés moins pratiques pour aider les occupants du monde réel. Les agents entièrement conscients de l'environnement qui ne sont pas limités par le champ de vision et les obstacles du monde réel sont plus pertinents pour aider les occupants à réussir efficacement le processus d'évacuation.

### 3.7. Le modèle de réseaux de flots

Le modèle de réseaux de flots est un modèle macroscopique qui décrit le problème d'évacuation en tant qu'un problème de flots (Pyakurel U., 2017). Les recherches sur le problème de maximisation de flot remontent à l'année 1955 et a été résolu par Harris T.E & Ross F.S. dans le cadre de la maximisation de flux des réseau ferroviaire (Schrijver A., 2002).

Le problème de flots dans l'évacuation suggère l'existence d'un graphe orienté doté d'une source de flux  $s$  (source node) et un point d'évacuation  $d$  (sink node) ou nœud d'évier. Le bâtiment est représenté par un graphe orienté  $G(V, A)$ . Un nœud de l'ensemble  $V$  représente une salle. Tandis que l'ensemble des arcs  $A$  représente les liens (portes, couloirs) entre les salles. On définit l'ensemble de nœuds sources  $S \subseteq V$ , l'ensemble de nœuds intermédiaires  $I \subseteq V$ , et l'ensemble de nœuds d'évier  $D \subseteq V$ . Soit  $v \in V$  et :  $A_v^{out} = \{(w, w) \in A\}$  l'ensemble d'arc sortant de  $v$ ,  $A_v^{in} = \{(w, v) \in A\}$  l'ensemble d'arc entrants. On suppose que :  $A_s^{in} = A_d^{out} = \emptyset$

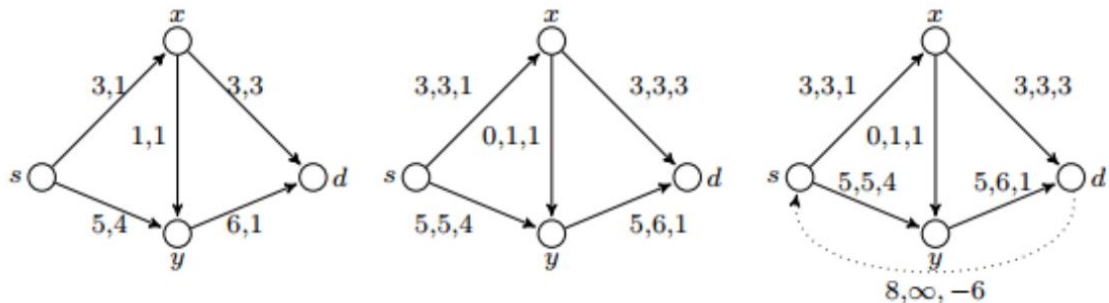
Le réseau de flots est constitué de fonctions non négatives :  $b_A : A \rightarrow \mathbb{R}^+$  dénotant les capacités d'arcs,  $b_V : V \rightarrow \mathbb{R}^+$  dénotant les capacités de nœuds.  $c_A : A \rightarrow \mathbb{Z}^+$  pour le coût des arcs et  $\tau_A : A \rightarrow \mathbb{Z}^+$  pour le temps de transit. Le temps total d'évacuation, appelé aussi temps d'horizon est noté  $T$ . L'ensemble d'évacués sont modélisés en motif de flots. Soit  $x^r(e)$  le débit du flux dans un arc  $e$ , alors :  $x^r(e) \leq b_A(e)$ . La collection de données  $N = (V, A, b_A, b_V, \tau_A, S, D, T)$  avec un temps d'horizon prédéterminé  $T$  représente un réseau de flots dynamique. Le réseau multi-terminal avec plusieurs nœuds sources  $S$  et nœuds d'évier  $D$  peut être transformé en un réseau à deux terminaux en ajoutant le super terminal (\*) et introduisant les arcs  $(*, s_i)$  et  $(d_i, *)$  au chaque nœud source  $s_i \in S$  et nœud d'évier  $d_i \in D$  respectivement.

La résolution du problème de réseaux de flot consiste à maximiser le flux des nœuds. En d'autres termes, la somme des flux des arcs d'entrée soit égale à la somme des flux des arcs de sortie. Ceci implique la résolution du problème linéaire (1) qui maximise la fonction  $val(x^r)$  sous les contraintes (2) et (3) :

$$val(x^r) = \sum_{e \in A_d^{in}} x^r(e) = \sum_{e \in A_s^{out}} x^r(e) \quad (1)$$

$$\sum_{e \in A_v^{in}} x^r(e) - \sum_{e \in A_v^{out}} x^r(e) = 0, \forall v \in V \setminus \{s, d\} \quad (2)$$

$$0 \leq x^r(e) \leq b_A(e) \quad \forall e \in A, t \in \tau \quad (3)$$



(a) (capacity, cost) (b) (flow, capacity, cost) (c) (Circulation)  
(a) Evacuation network (b) Maximum flow (c) Minimum cost circulation flow

Figure 13 – Exemple d'un réseau de flots d'évacuation (Pyakurel U. 2017).

Le modèle de flots suppose la conservation de flux d'occupants et ne tient pas compte du cas lorsque le nombre d'occupants excède la valeur permise. C'est-à-dire, dans le cas où les occupants dévient le plan d'évacuation dans un moment  $t \in ]0, T[$ . Le plan doit être refait et le programme linéaire doit être résolu pour un nouveau temps d'horizon  $T - t$ .

### 3.8. Le modèle mixte

Ce type de modèle a pratiquement deux sens. L'un désignant l'hétérogénéité des particules (piéton, véhicule, etc.) tel que les travaux de Hanawa Y. et al. (2016) dans le domaine d'évacuation de la foule à grande échelle piéton-véhicule. L'autre définition étant l'approche qui combine une multitude de modèles typiques afin de tirer parti de certains modèles et de compenser les inconvénients d'autres modèles.

## 4. Algorithmes de navigation

Le mouvement de la foule est un aspect important de la modélisation de l'évacuation qui affecte considérablement le processus ainsi que les résultats et les performances de l'évacuation. Deux approches de navigation dans les modèles cellulaires seront abordées dans cette section.

### 4.1. Approche du champ potentiel

L'approche de champ potentiel vise à inciter les occupants à suivre un plan de flux de particules statique prédéfini, le rendant ainsi moins dynamique et plus orienté utilisateur. La structure du bâtiment est représentée par une grille de cellules à laquelle sont affectés des poids de cellules et des points de cheminement statiques. Les distances entre les points de cheminement et le reste des cellules sont calculées avant la simulation et sont stockées dans un tableau de données. Ces des coûts de navigation orientant les occupants vers le flux de points de cheminement menant éventuellement à la sortie (Figure 14).

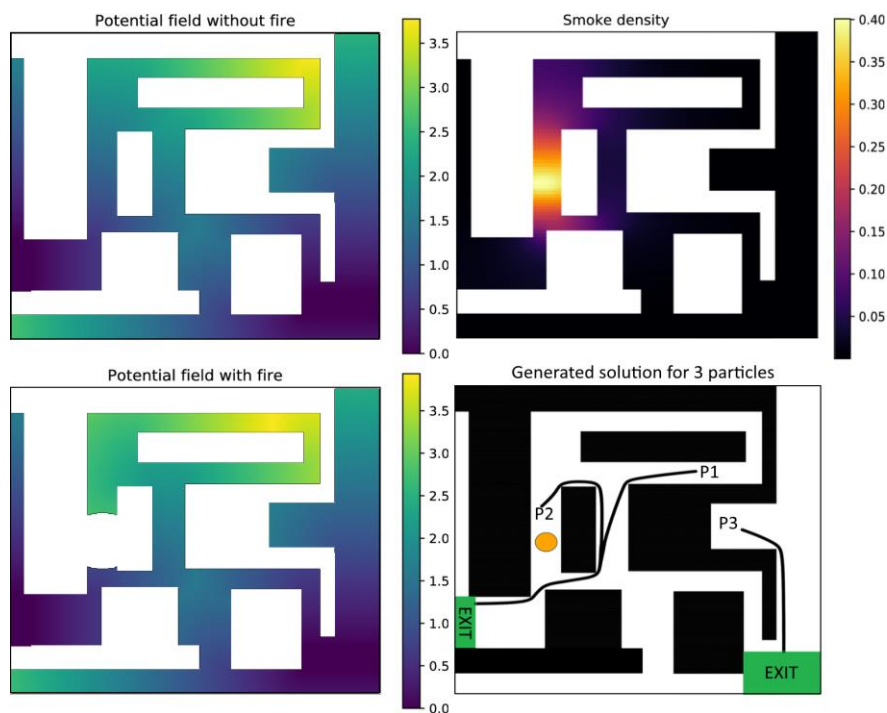


Figure 14 – Solution générée par le champ potentiel (Richardson O. et al., 2019)

Dans le cas où un seul point de cheminement représentant la sortie est établi, les valeurs de distance de la table de données seraient similaires à celles obtenues avec l'algorithme de Dijkstra (voir la section 4.2.1.).

#### **4.1.1. Algorithme de remplissage par diffusion**

L'algorithme de remplissage est basé sur une expansion sur tous les nœuds voisins, remplaçant éventuellement le coût de la distance parcourue de ce dernier si un chemin de déplacement moins coûteux est découvert. La simplicité de l'algorithme de remplissage par diffusion permet d'avoir plusieurs nœuds de départ ou d'objectifs et facilite la mise en place d'une implémentation récursive qui peut être décrite dans le pseudocode suivant :

- **Soit un entier : coût initial égal à -1**

**BEGIN**

- 1. Définir le nœud de départ en tant que nœud actuel**
- 2. Définissez le coût du déplacement étant égal au coût initial + 1**
- 3. Pour chaque nœud voisin : Appelez l'algorithme de remplissage à avec le coût initial égal au coût de déplacement actuel et le nœud voisin en question en tant que nœud de départ.**

**END**

- **Identifiez le chemin le plus court entre le nœud d'objectif et le nœud de départ en fonction de la valeur de déplacement du nœud voisin calculé précédemment.**

L'algorithme Flood Fill est principalement utilisé pour les applications de dessin et de peinture, permettant à l'utilisateur de peindre toute une zone sélectionnée avec la même couleur. Toutefois, comme l'indique le code ci-dessus, des valeurs de coût peuvent être utilisées pour indiquer un coût de déplacement. Elle permettrait de remonter d'un point d'objectif quelconque du graphe jusqu'au nœud de départ choisi par l'utilisateur au début de l'algorithme.

## **4.2. Approche d'optimisation**

L'approche du chemin optimisé met l'accent sur la réduction des coûts de navigation en tenant compte de la distance, du temps et d'autres paramètres souhaités (dans ce cas, l'évitement du danger), ce qui se traduit par une évacuation rapide des occupants par des voies sûres. Les coûts sont répartis sur différents espaces du bâtiment et sont donc représentés par un graphe pondéré qui ressemble à une grille avec des valeurs respectives des cellules. Deux algorithmes de navigation typiques et largement utilisés sont présentés, l'algorithme de recherche du chemin

le plus court de Dijkstra (1959) et l'algorithme A \* pour la recherche du chemin le plus rentable en termes de coûts.

#### **4.2.1. Algorithme de Dijkstra**

Le problème du chemin le plus court peut être résolu en utilisant l'algorithme de Dijkstra. Il est basé sur le marquage des nœuds d'un graphe comme étant visités si leur distance par rapport au nœud cible est inférieure au nœud actuellement occupé. Il renvoie l'ensemble des nœuds représentant la solution lorsque le nœud objectif est atteint. Ceci le rend plus performant que l'algorithme de remplissage par diffusion en termes de critères d'arrêt. Dans certains scénarios, l'algorithme de remplissage par diffusion est meilleur que l'algorithme de Dijkstra, car il donne la possibilité de s'étendre sur différents nœuds cibles lorsque plusieurs nœuds d'objectif ont lieu. Un exposé utile et détaillé de l'algorithme de Dijkstra et de son implémentation peut être vu dans le pseudocode suivant :

- 1) Marquez tous les nœuds comme non visités avec une distance provisoire infinie pour chaque nœud.**
- 2) Marquer le nœud actuel comme visité avec une distance provisoire de 0.**
- 3) Calculer toutes les distances provisoires des voisins non visités du nœud actuel et remplacer la valeur voisine si elle est inférieure (la distance provisoire d'un voisin est égale à la somme entre la distance du chemin emmenant vers le nœud actuel et la distance entre ce dernier et le nœud voisin en question).**
- 4) Marquer le nœud actuel comme visité et le supprimer de l'ensemble non visité**
- 5) Définir le nœud avec la distance provisoire la plus basse en tant que nœud actuel.**
- 6) Si le nœud actuel n'est pas le nœud objectif, retournez à l'étape 3.**

Cette méthode nous permet de trouver le nœud d'objectif ainsi que tous les chemins optimaux emmenant vers d'autres nœuds autre que le nœud initial (s'il existe) avec la méthode de recherche gloutonne. Cependant, une structure de données doit être utilisée pour stocker tous les chemins les plus courts vers tous les nœuds d'un graphe menant au nœud d'objectif. Une façon de procéder consiste à stocker les nœuds marqués dans un tableau appelé tableau prédécesseur, où chaque nouveau nœud marqué est lié à son prédécesseur précédent (nœud actuel à l'étape 3).

### 4.2.2. L'Algorithme A\*

La recherche A\* est assez similaire à celle de Dijkstra en ce sens qu'elle tient compte de la distance parcourue, mais ajoute également une valeur heuristique qui nous indique à quelle distance nous nous trouvons du but. Ainsi, contrairement à la recherche (Breadth First Search) de Dijkstra, A\* considère uniquement les chemins qui semblent moins coûteux en termes de coûts de déplacement anticipé. Les nœuds dont la valeur heuristique est inférieure à l'objectif, et les distances plus courtes jusqu'au point de départ, sont plus susceptibles d'être étendus par le processus de recherche. On définit les éléments de base suivants :

- **nœud de départ** : la position d'où on commence la découverte.
- **nœud d'objectif** : la position qu'on veut atteindre avec un coût minimal.
- **g(n)** : le coût effectif de transit du **nœud de départ** vers le nœud **n**
- **h(n)** : la valeur heuristique estimant la distance entre le **nœud courant** et le **nœud d'objectif**.
- **f(n)** : fonction d'évaluation du chemin optimal, retourne le coût estimé de transit du **nœud de départ** vers le **nœud d'objectif** en passant par le **nœud n**
- **f(n) = g(n) + h(n)**

Le pseudocode suivant montre le déroulement de l'algorithme A\* :

- 1) Soit **liste ouverte** et **liste fermée** deux listes vides
- 2) Mettre le nœud de départ dans la **liste ouverte**
- 3) Tant que (**liste ouverte** non vide)
  - a) Définir le **nœud courant** le nœud ayant la valeur **f** la plus petite
  - b) Enlever le **nœud courant** de la **liste ouverte**
  - c) Ajouter le **nœud courant** à la **liste fermée**
  - d) Si (**nœud courant** est le **nœud d'objectif**) **END**
  - e) Pour chaque (**nœud fils**)
    - i) Si (**nœud fils** appartient à la **liste fermée**) aller à e)
    - ii) Calculer **f, g, h**
    - iii) Si (**nœud fils** appartient à la **liste ouverte**)
      - Si (**nœud fils** dans **liste ouverte** admet **g** plus petit) aller à e)
      - Sinon remplacer la nouvelle valeur de **g**

iv) Ajouter **nœud fils** à la **liste ouverte**

**FIN**

La recherche A\* privilégie les voies prometteuses pour l'expansion de la recherche, alors que la recherche gloutonne de Dijkstra s'étend sur toutes les directions possibles du graphe. Cela rend l'algorithme de recherche A\* beaucoup plus optimisé en termes de complexité et de temps d'exécution tant qu'une valeur heuristique correcte existe. Les valeurs heuristiques doivent obéir à la contrainte suivante : Le coût heuristique admissible est inférieur ou égal au coût effectif.

Un bon exemple qui s'avérerait assez utile pour satisfaire la contrainte est la distance euclidienne (distance linéaire réelle) entre deux nœuds d'un graphe. Donc, à condition que nous ayons une bonne évaluation de la valeur heuristique des coûts entre différents nœuds dans un graphe, A\* est une meilleure solution pour la recherche du chemin optimal (Permana S. et al., 2018). Un scénario dans lequel la distance linéaire pourrait ne pas offrir les solutions optimales serait lorsque l'espace de recherche est composé de terrains irréguliers. Dans le cas de bâtiments à étages différents, cela pourrait conduire à choisir un chemin plus long que prévu. Des contraintes autre que la distance linéaire doit être prises en compte pour approcher ou satisfaire les critères d'optimalité.

## **5. Solutions existantes**

Une étude sur les solutions existantes d'évacuation informatisée a été faite par Kuligowski E. et al. (2010) montrant les modèles d'évacuation les plus adoptés et les critères relatifs à chaque solution. Les critères incluent le type du modèle adopté, sa structure (ici, tous les modèles utilisent soit un réseau de nœuds dont seulement le type du réseau diffère, soit un espace 3D non-délimité), la perspective (niveau de granularité) du modèle ainsi que celle de l'ensemble d'occupants, le plan d'évacuation (mouvement) et le type de la visualisation (s'il y'en a) de ce dernier (voir Table 2). Les informations extraites concernent seulement les solutions d'évacuation indoor (de l'intérieur vers l'extérieur).

	<b>Modeling Method</b>	<b>Structure</b>	<b>Perspective Model/Occ</b>	<b>Visualization</b>	<b>Movement</b>
<b>EVACNET4</b>	Optimization	Coarse	Global	Not Available	User Choice
<b>STEPS</b>	Behavioral	Fine	Individual	2D, 3D	Potential
<b>SIMULEX</b>	P.Behavior	Coarse	Individual	2D, 3D	Social Force
<b>GridFlow</b>	Part.Behavior	Coarse	Individual	2D, 3D	Density
<b>FDS+Evac</b>	Part.Behavior	Coarse	Individual	2D, 3D	Social Force
<b>Pathfinder 2009</b>	Partial Behavior	Coarse	Individual/Global	2D, 3D	Density, Social Force
<b>SimWalk</b>	Part.Behavior	Coarse	Individual	2D, 3D	Potential, Conditional
<b>PEDFLOW</b>	Behavioral	Coarse	Individual	2D, 3D	Social Force
<b>PedGo</b>	Partial Behavior /Behavioral	Fine	Individual/Individual, Global	2D, 3D	Cellular Automata, Conditional
<b>ASERI</b>	Behavioral, Risk Assessment	Coarse	Individual	2D, 3D	Social Force
<b>BldEXO</b>	Behavioral	Fine	Individual	2D, 3D	Potential
<b>Legion</b>	Behavioral	Coarse	Individual	2D, 3D	Social Force, Conditional
<b>EPT</b>	Behavioral	Fine	Individual	2D, 3D	User Choice, Conditional
<b>MyriadII</b>	Behavioral	Continuou s, Coarse, Fine	Individual	2D, 3D	Mixed
<b>MassMotion</b>	Behavioral	Coarse	Individual/Individual, Global	2D, 3D	Conditional

Table 2 – Solutions informatiques de l'évacuation indoor (Kuligowski E. et al. 2010)

La méthode de modélisation et le type de mouvement associé concernent le plan d'évacuation en question. Ils marquent des critères qui indiquent la présence de certains paramètres pris en considération (temps d'évacuation, congestion, comportement des



occupants, etc.) pour le choix de la solution d'évacuation. Ainsi, deux modèles peuvent être inclus l'un dans l'autre si l'un de ces premiers traite tous les paramètres présents dans l'autre modèle, et présente d'autres dimensions d'analyses afin d'enrichir la base des entrées et éventuellement offrir un meilleur résultat. Nous définissons ainsi chaque type noté par :

- **User choice** : Dans ce modèle, l'utilisateur assigne au préalable des valeurs de vitesse, de débit et de densité à certains espaces du bâtiment.
- **Potential** : La carte du bâtiment est divisée en une grille de cellules auxquelles sont attribuées des valeurs potentielles qui créent un certain flux pondéré qui guidera les occupants vers la sortie.
- **Social Force** : une bulle à 360 degrés est attribuée à chaque occupant, qui correspond à la distance minimale entre ce premier et les autres occupants ainsi que les obstacles dans le bâtiment. Un algorithme d'acheminement est donc associé tout en respectant la contrainte de distance minimale.
- **Density** : Les occupants se voient attribués des valeurs de vitesse et de vélocité en fonction de leur position dans le bâtiment. Les chambres et les couloirs sont attribués avec des valeurs de densité en fonction de leur structure et des données de congestion recueillies au préalable.
- **Conditional** : le déplacement des évacués dépend entièrement des conditions de l'environnement, y compris la structure, l'évolution de la situation dangereuse et l'état des autres occupants. La congestion n'est pas prise en considération.
- **Cellular Automata** : La carte du bâtiment est divisée en une grille de cellules auxquelles les évacués occupent dans un pas de temps, ceci forme un état global, le prochain état est décidé à l'aide d'une fonction de transition qui représente le plan d'évacuation.
- **Mixed** : Des diverses entrées, paramètres et solutions sont utilisés.

En regardant le tableau présenté ci-dessus, on note que les modèles les plus adoptés sont des modèles de comportement dont la modélisation du comportement a lieu et varie en termes de règles et de propositions d'un modèle à l'autre. Le tableau montre que :

- La structure sous-jacente des modèles suggérés (représentation) dépend des automates cellulaires et des réseaux basés sur des nœuds dans lesquels différents objets et personnes évacuées occupent pour représenter l'état du monde simulé dans un moment donné.

- La perspective du modèle et celle des occupants varient de microscopiques, macroscopiques et mésoscopiques, où les modèles microscopiques et mésoscopiques sont observés de nombreuses fois. Ceci indique que le comportement individuel et global de la population représente une dimension d'analyse importante tout au long de la procédure d'évacuation.
- La visualisation joue un rôle important. La plupart des modèles perçus ci-dessus présentent des méthodes de représentation allant de la visualisation d'espace bidimensionnelle à l'expérience immersive tridimensionnelle. Il signifie que la perception visuelle de la procédure d'évacuation est vitale pour le processus de développement ainsi que pour l'exécution en temps réel.
- L'attribut du mouvement présente l'aspect décisionnel de ces modèles. Il dénote les paramètres qui affectent le mouvement des occupants et les décisions prises dans l'ensemble du bâtiment jusqu'au point de sortie.

## 6. Synthèse

### 6.1. Critiques

Le problème avec les modèles mentionnés précédemment et les solutions existantes est qu'ils ne gèrent pas les changements au sein de la population et ne tiennent pas compte du fait que les occupants se conformeront ou pas à la solution traitée du plan d'évacuation, rendant ainsi le système dépourvu de l'aspect décisionnel de point de vue de la simulation. La majorité des travaux antérieurs sur la prise de décision d'évacuation simulent ce que les agents décideraient lorsque certains critères sont remplis. Les règles d'exécution sont basées sur des études statistiques, ce qui rend la simulation moins fiable. Par conséquent, les agents ne représentent pas la population réelle. Certains échantillons de la population dégénérés ayant un comportement non déterministe peuvent donc être observés, ce qui éloigne l'ensemble des règles de comportement de l'agent de la réalité et conduit finalement à des mauvais résultats basés sur des hypothèses faibles.

Pour pallier le manque de paramètres, fonctions et contraintes précis et représentatifs présentés par les modèles de comportement, des mesures doivent avoir lieu. Il est nécessaire de simuler en temps réel la population et l'environnement sans avoir recours à l'expérience antérieure ni à prendre de valeurs basées sur des études statistiques, car celles-ci peuvent générer de faux résultats concernant l'aspect simulation comportementale.

Du point de vue des données requises, deux valeurs d'entrée sont nécessaires en temps réel : la position de toutes les personnes évacuées à l'intérieur du bâtiment et l'état de toutes les voies d'évacuation à l'intérieur du bâtiment. En considérant le temps de traitement requis et l'impact des performances sur les ressources disponibles, la plupart des solutions précédentes, en particulier celles basées sur les automates cellulaires, requièrent un rafraîchissement de l'état du monde virtuel d'un moment  $t$  donné à l'autre à travers une fonction de transition. Cette fonction représente le plan d'évacuation, où des paramètres suggérés pourraient être introduits pour simuler le comportement non déterministe et la panique manifestée par les évacués en période d'urgence (tel que les travaux de Helbing D. et al. (2005)). Ceci implique un traitement et des calculs continus jusqu'à ce que toutes les particules de la simulation soient escortées vers la sécurité. Les données du plan d'évacuation sont ensuite envoyées aux occupants du monde réel qui pourraient bien se comporter d'une manière non conforme au plan, comme mentionné précédemment.

Enfin, étant donné que la simulation en temps réel nécessite des données en temps réel, le plan d'évacuation est considéré comme dynamique, et un traitement constant aurait lieu jusqu'à ce que toutes les personnes soient escortées, ce qui prend plus de temps. Sous cet aspect, le plan d'évacuation dans la simulation en temps réel ressemble à des suggestions dynamiques d'itinéraires au lieu d'un ordre d'évacuation selon une trajectoire inchangée qui pourrait être rejetée pour une raison ou une autre dans des situations réelles.

## **6.2. Direction**

Après avoir étudié les modèles existants, nous sommes arrivés à la conclusion de mettre à l'œuvre un modèle mixte multi-agent pour satisfaire l'aspect évacuation guidée et l'aspect décisionnel. Ce modèle est basé sur la conformité des agents aux actions des occupants tout en suggérant des solutions optimales. La sécurité des occupants étant une partie essentielle du processus, l'aspect optimisation est gardé en esprit. La structure spatiale d'automates cellulaires nous facilite la mise en place d'un algorithme de navigation efficient (soit l'algorithme A\*) et s'avère utile pour l'intégration avec la représentation BIM.

## **7. Conclusion**

Ce chapitre a présenté les principaux aspects théoriques de la thématique d'évacuation. Une visée sur les différents axes de recherche, les modèles existants, les solutions BIM et les algorithmes navigation a eu lieu. Une synthèse montrant notre direction est présenté.

Le chapitre suivant présente le concept des Serious Games et la réalité virtuelle. Il étudie également les outils de création des jeux sérieux, soit les moteurs de jeux.

# **Chapitre II : Serious Games & Moteurs de Jeux**

## 1. Introduction

De nos jours, les jeux vidéo sont en plein essor à cette époque (Chatfield T., 2009). Ils gagnent de plus en plus en popularité chaque jour, offrant de nombreux détails complexes et une simulation du comportement du monde réel pour faire plonger les joueurs dans une expérience réaliste. Des moteurs de jeu ont été introduits sur le marché pour aider les développeurs dans le processus de création de jeux en fournissant un ensemble de bibliothèques utiles telles que celles qui simulent les lois physiques, ce qui s'avère être très efficace pour satisfaire les délais dudit processus. De plus, avec la popularité croissante de l'industrie des jeux vidéo, les gens développent un esprit d'apprentissage favorisant les jeux contrairement à l'enseignement traditionnel avec des documents et les livres, menant finalement à l'émergence de jeux sérieux (Serious Games). Les jeux sérieux sont un type de jeux qui se concentre sur l'acquisition de connaissances en utilisant des moyens interactifs numériques entre l'utilisateur et la source d'informations, tels que des capteurs de mouvement et des environnements de réalité virtuelle. Ils maximisent l'expérience de la courbe d'apprentissage et minimisent la difficulté du processus d'apprentissage.

Ce chapitre vise à offrir une brève introduction aux moteurs de jeux, leur architecture, et à l'ensemble des outils qui seront utilisés par la suite pour construire l'environnement du processus d'évacuation.

## 2. Serious Games

L'intérêt dans les jeux a augmenté au cours de la dernière décennie d'une manière exponentielle, et maintenant l'industrie des jeux vidéo est plus populaire et rentable que celle de Hollywood et des films en général (Chatfield T., 2009). Les théories modernes sur l'apprentissage efficace montrent que ceci est à son apogée lorsqu'il est actif, expérientiel, basée sur la résolution des problèmes et fournit une rétroaction immédiate (Connolly T. et al., 2012). Tels efforts ont conduit à l'émergence des serious games (SG), une nouvelle forme de jeux qui combine l'aspect divertissement avec des aspects éducationnels et d'acquisition de connaissances. La simulation (que ça soit d'une partie du monde réel ou dans son ensemble) joue un rôle important dans la partie expérience de l'utilisateur (UX). Elle simplifie l'interaction pour le joueur afin de familiariser son état d'esprit avec l'interface du SG. Ainsi, on définit les serious games comme étant l'intersection entre les trois dimensions, divertissement, apprentissage et simulation (Marer P. et al., 2015) (voir Figure 15).

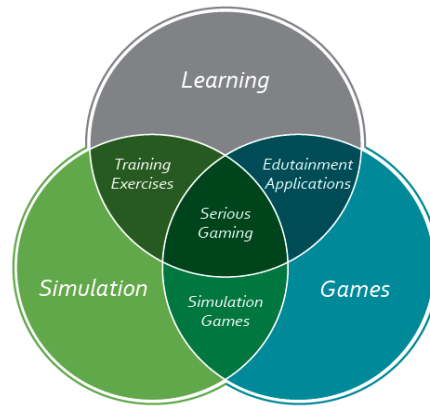


Figure 15 – Définition des Serious Games (Marer P. et al., 2015)

Le développement des Serious Games suit pratiquement le même cycle que celui des jeux vidéo. Il nécessite les mêmes approches de modélisation, concepts et ressources (hardware : processeur graphique ; software : bibliothèques). Dit Autrement, les jeux ont la même infrastructure (en termes d'objets et de techniques utilisées) et donc, à tout nouveau projet de réalisation de jeu, une duplication de travail aura lieu. Ceci a créé le besoin à un système (framework) possédant cette structure de base ou environnement de travail facilitant la tâche de création, suivi, et maintien du projet, offrant une variété d'outils (libraries, debugging tools, etc.) à la disposition du programmeur, d'où le recours aux moteurs de jeux (Game Engines) (Gregory J., 2009).

### 3. Moteurs de jeux

#### 3.1. Définition

Un moteur de jeu est un cadre logiciel (Framework) pour construire et créer des jeux vidéo (Lewis M. et Jacobson J., 2002). L'ensemble d'outils aidant à fournir le rendu (2D ou 3D), le son, la simulation (lois physiques), la compatibilité (par rapport à la plate-forme et au hardware), et la mise en réseau (réplication réseau) de l'application en question. Cela nous permet de se focaliser sur une thématique bien définie (expérience, implémentation, simulation, etc.) sans se soucier de la recréation de l'infrastructure à chaque nouveau projet. Quant aux paradigmes de programmation utilisés pour le développement, la majorité (voir la totalité) de moteurs de jeux existants sont basés sur la programmation orientée-objet (d'où la flexibilité de l'élaboration et la réutilisation des composants existants). Ils utilisent parfois des structures de modélisation (qui facilitent la compréhension et la génération automatique du code ainsi que

la compatibilité multiplateforme) tels que les langages de programmation visuelle (VPL) et les arbres de comportement (Behavior trees).

Le terme moteur de jeux (Game Engine) a apparu au milieu des années 90 dénotant un ensemble d'outils créés par les développeurs d'un jeu spécifique, mis à la disposition de ces derniers ou encore la communauté, pour faciliter la tâche d'extension du jeu existant ou la création d'un autre qui lui ressemble (Gregory J., 2009). Cependant, il était perceptible pour une grande partie de développeurs que la commercialisation de moteurs de jeux fournit une excellente source de revenu, ce qui par la suite, a engendré un nouveau type de moteurs de jeux basés sur la généralisation de l'architecture d'un jeu pour qu'ils soient utilisables pour la création de n'importe quel type de ce dernier. Cependant solution générale engendre un problème de performance causé par le manque de l'optimisation (Gregory J., 2009). Ce phénomène d'abstraction et de généralisation de l'architecture de moteurs de jeux a permis l'élaboration de l'ensemble d'outils fournis dans des domaines autres que l'industrie des jeux COTS (Commercial Off-The-Shelf), tel que l'infographie, la publicité, l'animation, la simulation et l'enseignement. Il est à noter l'utilisabilité de moteurs de jeux dans la recherche scientifique (Lewis M. et Jacobson J., 2002).

### **3.2. Architecture d'un moteur de jeux**

La structure des moteurs suit un motif en couches (voir Figure 16 ci-dessus). Une architecture à base de composants est au noyau, c'est-à-dire que l'orienté-composant est au cœur du moteur de jeu (Gregory J., 2009). Des comportements spécifiques sont encapsulés dans des composants et sont conçus pour être utilisés via des interfaces publiques, à savoir l'API du moteur de jeu. Les couches du niveau inférieur incluent l'ensemble des bibliothèques essentielles utilisées par les composants, le gestionnaire de ressources et des éléments spécifiques au système d'exploitation. Au niveau supérieur, des composants spécifiques au jeu sont rendus possibles via l'héritage et des interfaces publiques.



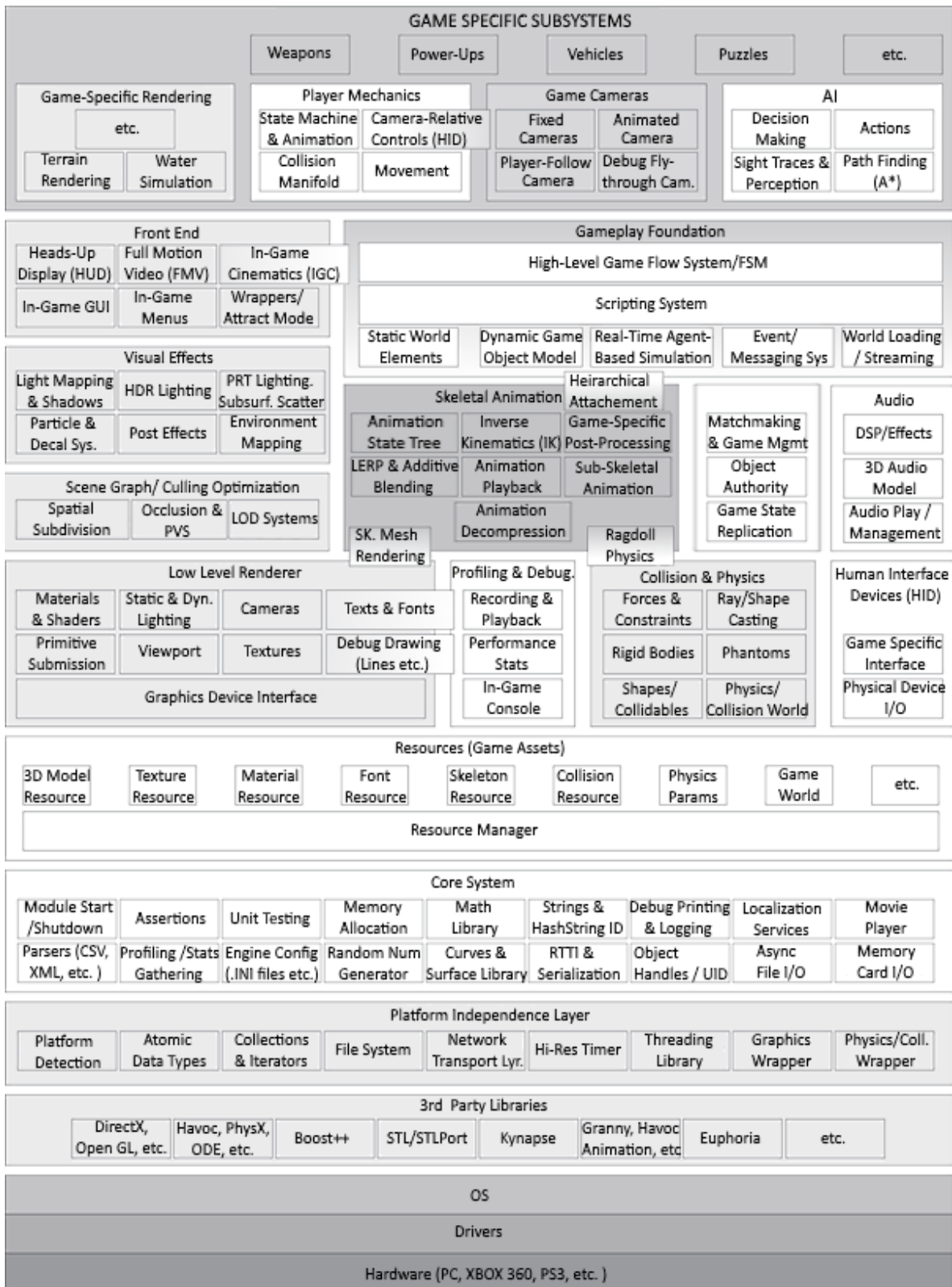


Figure 16 – Unity runtime game engine architecture (Gregory J., 2009).

### 3.3. Les concepts de base pour la création d'un jeu vidéo

### 3.3.1. La Boucle du jeu

La boucle du jeu (game loop) maintient le jeu toujours dans un état actif au lieu de l'exécuter séquentiellement et éventuellement terminer l'exécution. L'attribut spécial du game loop est la méthode de mise à jour (update) qui conserve tous les objets de jeu et les met à jour à tout moment. Elle exécute certaines tâches critiques telles que le rendu graphique du jeu à l'écran en fonction de la durée d'image (frame time), l'écoute des événements d'entrée/sortie, ou les évènements personnalisés.

**Frame time** : représente le temps nécessaire à l'unité de traitement pour afficher une autre image à l'écran, donnant ainsi l'illusion d'une lecture vidéo d'animation fluide.

Le nombre d'images rendues à l'écran par seconde est appelé trames par seconde (FPS), il est principalement utilisé comme référence pour la segmentation du lancement des évènements, ainsi que pour mesurer les performances du jeu. Un exemple de processus qui utilise la segmentation du frame time pour contrôler la qualité d'un service est la bibliothèque Physics Library, qui simule et met à jour le comportement physique des objets pour les correspondre à la physique du monde réel.

**Double buffering** : La livraison des trames est réalisée avec un pattern de programmation appelé tampon double (double buffering), où une mémoire tampon aura lieu pour satisfaire les demandes d'affichage vers l'écran. La mémoire tampon actuelle représente une image qui a déjà été traitée et attend d'être affichée à l'écran. Le tampon suivant indique le tampon prochain en cours de traitement.

### 3.3.2. Événements et gestionnaires d'événements

Les événements représentent une partie essentielle à l'exécution d'un jeu. La programmation événementielle est au cœur du développement des jeux vidéo (Nystrom R., 2014). Les événements indiquent que certains critères ont été remplis, des tâches et des comportements planifiés auront donc lieu. Les événements sont exécutés par un type de méthode appelée gestionnaire d'événement (event handler). Il contient des instructions pour une tâche planifiée. Un certain nombre d'événements importants sont les suivants : événement Begin play, événement Tick, événements de chevauchement, événements personnalisés, et événements d'entrée utilisateur :

- **Begin play** : indique qu'un certain niveau de jeu (activité) a commencé son exécution. Cela déclenchera la création des objets relatifs au jeu et l'initialisation des variables.

- **Tick events** : déclenche sur un très petit pas de temps pour donner l'illusion d'une mise à jour en temps réel des propriétés, attributs, et comportements des objets. Une faible valeur de pas de temps pour ce type d'événements rend l'exécution du processus de jeu continue, réactive et interactive.
- **Événement de chevauchement (overlap event)** : une requête de chevauchement qui vérifie à l'aide de volumes de collision, les raycasts ou de traces linéaires (line trace) si deux objets ou plus se chevauchent, et renvoie les objets en question. Utile pour la communication inter-objet dans le jeu.
- **Événements personnalisés** : événement d'objet défini par l'utilisateur qui peut être lié à d'autres événements, appelé directement ou en interne.

### 3.3.3. Objets d'un jeu vidéo

Les classes conventionnelles dans la programmation font la base des classes définies dans un jeu vidéo. Cependant, il existe déjà dans le cœur d'un moteur de jeux une hiérarchie de classes prête à l'emploi qui facilite la conception du jeu, où certains objets sont considérés principaux, et les autres complémentaires. Dans cette partie, la nomenclature et la représentation du moteur de jeux Unreal Engine sera considéré (voir Figure 17).

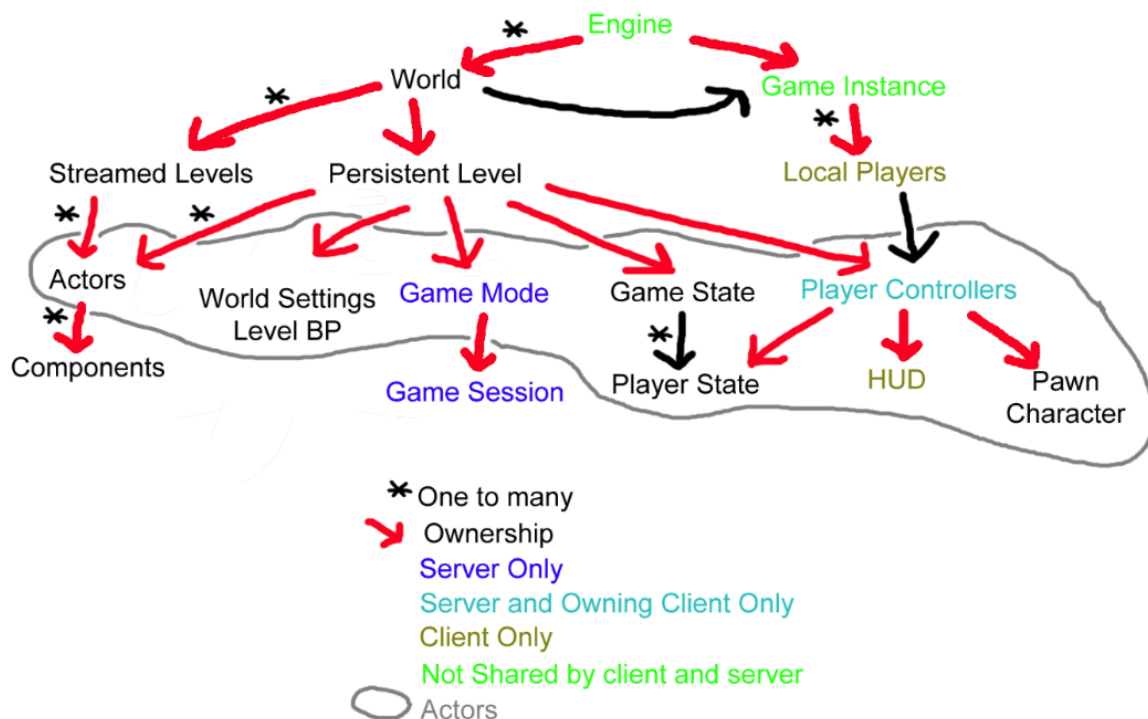


Figure 17 – Classes Principales de UnrealEngine (Extraite de [wiki.unrealengine.com/ue4\\_class\\_diagram](http://wiki.unrealengine.com/ue4_class_diagram)).

On mentionne les objets de base suivants :

- **Object** : La classe directe à partir de laquelle tous les objets du moteur de jeux héritent. Ne peut pas lancer un événement de mise à jour en temps réel (tick). Les instances qui n'admettent pas un événement tick ne peuvent pas exister dans le monde virtuel du jeu sans être attachés aux objets qui ont un événement tick.
- **World** : Cette classe représente le monde virtuel qui encapsule les niveaux du jeu (Game level), les acteurs (Actors) ayant rapport avec ces derniers, et sont éventuellement rendu (optionnellement) à l'écran.
- **Actor** : La classe de base pour les objets pouvant être représentés dans le monde du jeu sans être attachés à un propriétaire. Ils peuvent lancer des événements tick, et peuvent avoir des composants attachés.
- **Actor Component** : une classe composante qui peut lancer des événements tick mais ne peuvent pas exister sans acteur propriétaire.
- **Game Mode** : décrit les règles d'exécution du jeu (conditions d'arrêt ou de la fin)
- **Player Controller** : dénote l'objet référençant l'acteur possédé par le joueur.
- **HUD (Head up display)** : représente l'interface du joueur.
- **Widget** : La classe de base qui encapsule les éléments de l'interface utilisateur.

### 3.4. Intelligence artificielle dans l'industrie des jeux vidéo

#### 3.4.1. Arbres de comportement

Un arbre de comportement (Behavior Tree) est à la fois un langage de modélisation et un paradigme de programmation permettant de structurer et de basculer entre différentes tâches dans un agent autonome, étant donné qu'une activité peut être divisée en tâches ou en actions (Colledanchise M. & Ogren P., 2018). Les arbres de comportement étaient initialement utilisés comme un moyen d'exprimer la modularité dans la structure de contrôle d'un personnage non jouable (NPC) dans un jeu (Klönker A., 2013).

Un arbre de comportement est représenté par un arbre orienté où les nœuds internes sont appelés nœuds de flux de contrôle et les nœuds feuilles sont appelés nœuds d'exécution. Tous les nœuds ne peuvent pas avoir plus d'un parent alors que le nœud racine est sans parent. Un BT commence son exécution à partir du nœud racine à une fréquence donnée en utilisant des événements tick, puis se propage aux enfants. Il existe quatre catégories de nœuds de flux de contrôle :

- **Sequence** : Les nœuds de séquence exécutent leurs enfants de gauche à droite. Si un nœud enfant ne parvient pas à démarrer, l'exécution s'arrête, retourne au nœud parent et s'exécute à nouveau au prochain événement tick.
- **Selector** : Les nœuds sélecteurs exécutent leurs enfants de gauche à droite et s'arrêtent avec succès au premier nœud enfant qui termine son exécution avec succès.
- **Parallel** : Les nœuds parallèles exécutent simultanément des nœuds fils et ne retournent avec succès que si l'exécution d'un certain seuil minimum défini par l'utilisateur est atteinte.
- **Decorator** : Les nœuds décorateurs, également connus sous le nom de « conditions » dans certains systèmes d'arbres de comportement sont attachés à des nœuds composites pour définir si une branche de l'arborescence, peut être exécutée.

Les feuilles de l'arbre de comportement sont désignées par des actions. Ils ne peuvent pas avoir des fils. Les actions sont détaillé et décrit par le langage de programmation choisi.




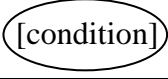
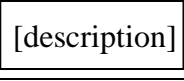
Type du nœud	Symbol
Séquence	
Sélecteur	
Parallèle	
Décorateur	
Action	

Table 3 – Types de nœuds d'un arbre de comportement (Extraite de : Colledanchise M. et al., 2018)

*Exemple 1.* Un exemple d'un arbre de comportement peut être vu dans la figure ci-dessous (voir Figure 18) pour le comportement suivant : Tant que la balle n'est pas à la disposition de l'agent, chercher la balle et la mettre dans le sac. Si pour n'importe quelle raison on ne peut pas le faire, alors demander de l'aide.

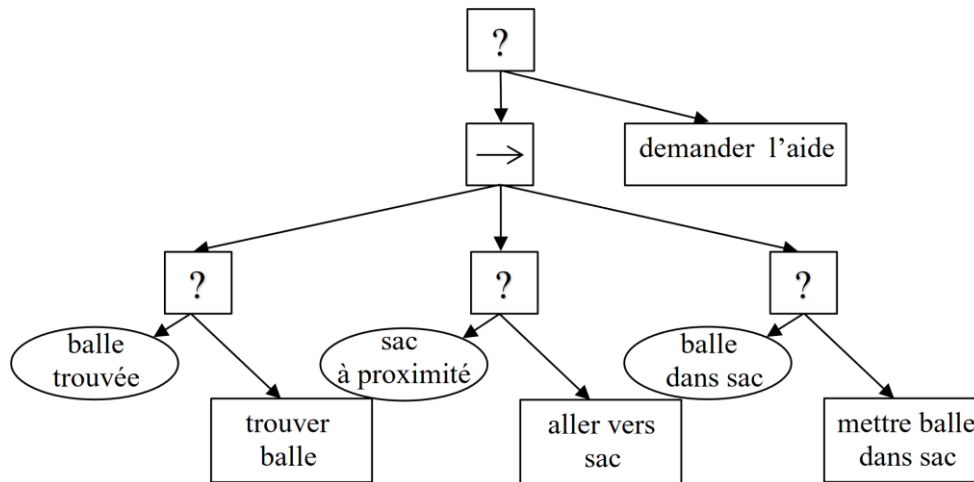


Figure 18 – Encodage de l'arbre de comportement pour *Exemple 1*. (Extrait de : Colledanchise M. et al., 2018).

### 3.4.2. Blackboards & Services

Le système à tableaux noirs (Blackboard pattern) est un modèle de conception comportementale. Il s'applique aux domaines d'application de l'IA impliquant une recherche multidimensionnelle complexe dans laquelle la connaissance du contrôle joue un rôle important pour la résolution de problèmes (Carver N. & Lesser V., 1994).

Les tableaux noirs aident à la création des comportements événementiels efficaces. Ils cachent les calculs et réduisent les frais de traitement en centralisant les données pour un accès et une réutilisation facile. Il agit également comme un bloc-notes pour les données qui n'appartiennent pas à une certaine catégorie mais qui sont nécessaires pour suivre le contexte et le comportement (Carver N. & Lesser V., 1994). Suivant l'exemple précédent (*Exemple 1*), un vecteur 3D peut être stocké pour garder une trace de l'emplacement actuel de la balle sans avoir besoin de le recalculer à mesure que sa valeur change. Les valeurs stockées sont appelées clés. Un tableau noir donc représente un ensemble de clés.

Les services maintiennent le tableau à jour car les tableaux noirs sont statiques et ne contiennent aucun comportement (voir Figure 19). Dans la figure ci-dessous, on voit comment les services (appelées « Knowledge Sources » dans ce cas) font pour mettre à jour le tableau. Les services peuvent être attachés aux nœuds d'un arbre de comportement. Cela permet de garder à jour les clés (nodes) essentielles du tableau lorsqu'elles sont attachées à la racine. Tandis que les associer à des branches contrôlées par des nœuds de flux omet certaines mises à jour de clés lorsque leur sous-comportement respectif n'est pas utilisé. Cela augmente le degré de la gestion et l'optimisation des données.

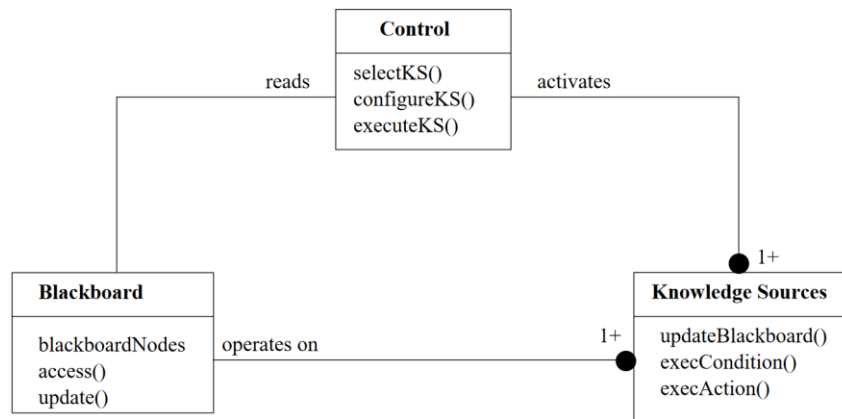


Figure 19 – Blackboard pattern (Lalanda, P., 1997)

## 4. Réalité virtuelle

### 4.1. Définition

La réalité virtuelle (VR) a connu de tellement de changement vu son évolution, allant de l'exploration d'images panoramiques statiques jusqu'à l'expérience de la simulation 3D en temps réel à l'aide d'un matériel de virtualisation dédié (HMD), que sa définition diffère d'un auteur à l'autre (LaValle S.M., 2017). De nos jours, la réalité virtuelle est considérée comme l'acte qui consiste à induire un comportement ciblé (expérience désignée par le créateur) dans un organisme (l'utilisateur) en utilisant une stimulation sensorielle artificielle (imagerie et son 3D). Alors que l'organisme n'a que peu ou pas de conscience sur l'interférence (son cerveau donne des coordonnées pour des objets qui n'existent pas réellement) (LaValle S.M., 2017).

Le terme « réalité virtuelle » a été utilisé pour la première fois au milieu des années 80, lorsque Jaron Lanier, fondateur de VPL Research (Virtual Programming Languages) en 1984, également appelé “père de la réalité virtuelle”, a commencé à développer les équipements, y compris des lunettes et des gants, nécessaires pour faire l'expérience de ce qu'il appelle « réalité virtuelle » (Botella C. et al., 2017).

La stimulation sensorielle artificielle, si réussie à engager l'utilisateur avec l'environnement créé en suspendant la vision sur le monde réel, elle rend alors l'expérience immersive (IVR). Elle est achevée à l'aide d'un matériel dédié pour la réalité virtuelle appelé Head Mounted Display (HMD) ou des accessoires téléphoniques qui simulent le comportement du HMD (Cardboard) (voir Figure 20). Ainsi que des solutions de suivi du mouvement (motion tracking) tel que Leap Motion, et peuvent s'étendre jusqu'à l'utilisation d'autres stimulants

(pouvant être réels) tels que les charges électriques (pour simuler la douleur) et aussi bien d'autres objets qui réagissent aux événements venant du monde virtuel (programme) pour introduire d'autres sensations (la température, l'odeur, etc.).



Figure 20 – Équipements de la réalité virtuelle

Autres termes proches de la RV incluent, la réalité augmentée (AR), qui dénote une vision directe du monde réel dont les éléments sont enrichis (augmentés) par des informations visuelles générées par l'ordinateur. La virtualité augmentée (AV) qui se caractérise par la vision du monde réel via un support d'affichage, et la réalité mixte (MR) qui consiste en une vision sur le monde réel mêlée avec des éléments générés par ordinateur (Lovreglio R., 2018).

#### 4.2. Développement d'applications de la RV

L'étape principale est bien d'offrir un rendu pour chaque œil (side-by-side rendering) à l'aide de deux caméras virtuelles (Makransky G., 2017) (voir Figure 21).

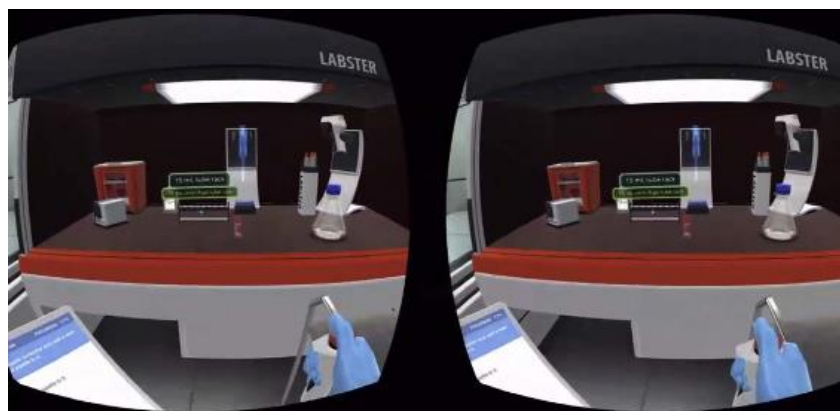


Figure 21 – Exemple d'un rendu dans la RV (Makransky G. et al., 2017).

Lors du développement pour VR, le choix du matériel cible et de la plate-forme cible est affecté par les fonctionnalités souhaitées de l'application, soit le degré de liberté (DOF) (Crespo R. et al., 2015). Le système DOF (Degree of Freedom) désigne l'ensemble des contraintes définies par le développeur pour orienter l'expérience utilisateur dans une certaine



direction (Mazuryk T. et al., 1999). L'ensemble de contraintes comprend les descripteurs de mouvement suivants :

- **Heave** : soulèvement (montée ou descente)
- **Sway** : balancement (glisser à gauche ou à droite).
- **Surge** : se glisser vers l'avant ou l'arrière.
- **Yaw** : rotation vers la gauche ou la droite.
- **Pitch** : tangage (inclinaison vers le haut ou vers le bas).
- **Roll** : se rouler (pivote d'un côté à l'autre).

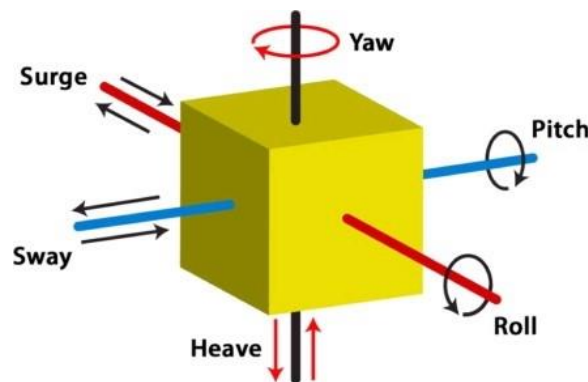


Figure 22 – Les six degrés de liberté (Muhanna A. M., 2015).

Les interactions utilisateur-objet et les actions pour sélectionner et attraper des objets dans le monde virtuel sont gérées par la cinématique inverse (Inverse Kinematics) et des requêtes générées par les diffusions de rayons de collision (Raycast). La cinématique inverse (IK) étudie la recherche de solutions concernant les règles appliquées à un acteur associé au système 6DOF pour approcher et interagir avec l'objet cible (Tolani D. et al., 2000).

Le développement d'applications de la réalité virtuelle obéit généralement à deux impératifs : la connaissance de l'API (Application Programming Interface) fourni avec le matériel, et la conception d'une expérience utilisateur interactive (l'absence d'éléments d'interface utilisateur statiques) pour assurer l'expérience immersive attendue de l'application (Alger M.W., 2015). Les appareils cibles Android doivent inclure la prise en charge de Google VR SDK (Google VR Software Development Kit).

Cependant, une interface de programmation d'applications multiplateformes appelée Open XR, est en cours de développement par un consortium regroupant 100 grandes entreprises de logiciels et de matériel informatique, le groupe Khronos (Takahashi D., 2019). Elle est destinée à améliorer la prise en charge matérielle et la portabilité du code.

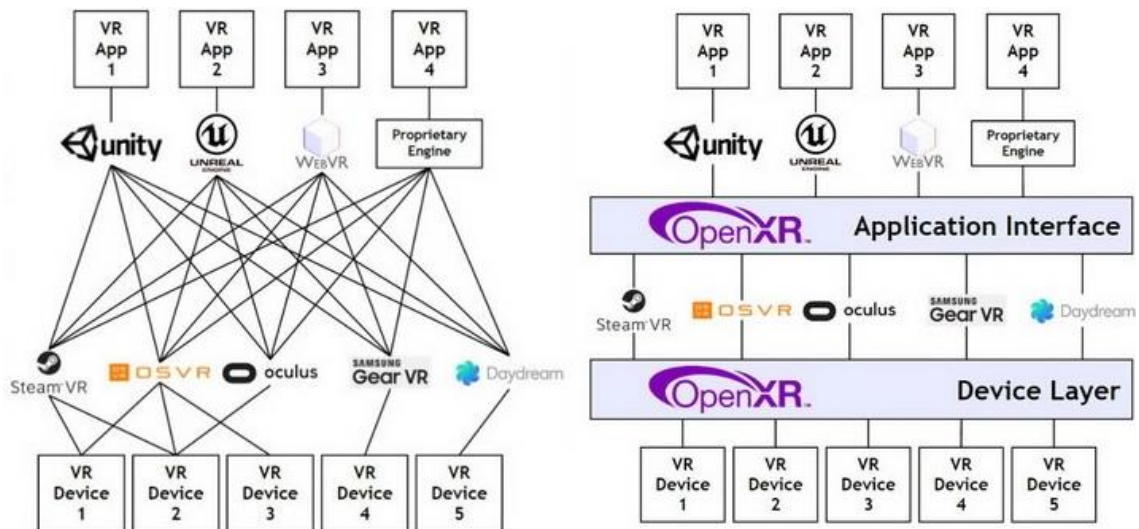


Figure 23 – Khronos Open XR pour la RV/RA multiplateforme

### 4.3. Contribution scientifique

La réalité virtuelle vise en premier lieu d'améliorer l'expérience de l'utilisateur (UX). De nombreuses utilisations dans les domaines scientifiques aussi bien que dans la vie quotidienne ont lieu. Elle est également utilisée pour l'étude du comportement de la foule (Olivier A.H. et al., 2014). La recherche dans le domaine du comportement psychologique profite de manière sensible de la réalité virtuelle pour faire des expériences qui n'étaient pas possibles dans les laboratoires de recherche auparavant (Botella C. et al., 2017). Autres domaines incluent l'enseignement et la formation des employés (Dávideková M. et al., 2017), ainsi que l'élaboration des techniques de la réalité virtuelle pour la simulation et l'entraînement d'évacuation sans se soucier des contraintes exposées dans les laboratoires d'études et le terrain dans la vie réelle tels que les limitations éthiques (Bernardes S.M.F. et al., 2015).

## 5. Conclusion

Ce chapitre a présenté le concept de jeux sérieux. Une visée sur les moteurs de jeux et leur structure de base a eu lieu. Nous avons discuté le concept d'arbres de comportement et l'application de l'intelligence artificielle dans le contexte de moteurs de jeux. Eventuellement, on a cerné les éléments de base pour le développement des applications de la réalité virtuelle.

Le chapitre suivant présente le concept d'internet des objets et son architecture. Il mentionne également les technologies sensorielles mises à l'œuvre pour connecter les objets et faciliter la communication entre eux, soit la technologie RFID.

# **Chapitre III : Internet des Objets**

## 1. Introduction

Les estimations pour le nombre d'appareils connectés vont de 25 à 50 milliards d'ici l'année 2020 (Lee I. et al., 2015). Ceci dit, on peut utiliser ces appareils dont nous disposons pour atteindre le niveau de conscience requis pour recueillir des données du monde en temps réel. Ils vont à la suite servir à la création et mise en œuvre des entités intelligentes aidant à faciliter la vie humaine en minimisant l'interaction homme-machine par rapport à la collecte de données et les directives.

Ce chapitre vise d'introduire le concept de l'internet des objets comme un outil de collecte d'information qui va nous faciliter la virtualisation du monde réel via la technologie de RFID.

## 2. Définition

La définition de l'internet des objets (IoT) n'est pas explicite et ne connaît pas une qui est précise, cependant, plusieurs définitions descriptives et par utilité existent. Dorsemaine B. et al. ont fait une recherche taxonomique sur l'IoT et proposent la définition suivante : un groupe d'infrastructures interconnectant des objets connectés, permettant leur gestion, exploration de données et accès aux données qu'ils génèrent, où les objets connectés sont des capteurs ou des actionneurs effectuant une fonction spécifique et capables de communiquer avec d'autres équipements (Dorsemaine B. et al., 2015). Autres définitions existent, tel que celle qui le définit comme étant, une extension de la connectivité réseau et de la capacité informatique aux objets, dispositifs, capteurs et éléments qui ne sont pas généralement considérés comme des ordinateurs. Ces objets intelligents nécessitent une intervention humaine minimale pour générer, échanger et consommer des données. Ils offrent souvent une connectivité aux fonctions de collecte, d'analyse et de gestion de données à distance (Kenneth A. et al., 2015).

Le terme "Internet des objets" (internet of things) a été introduit en 1999 par Kevin Ashton quand il travaillait sur l'optimisation de la chaîne d'approvisionnement en utilisant la RFID (Radio Frequency Identification) et la technologie sensoriale pour le suivi des marchandises (Ashton K., 2009). Il explique que les ordinateurs ont besoin de l'intervention de l'humain pour l'acquisition de l'information sur le monde réel et la réalisation de tâches, tandis que les gens ont peu ou pas de temps à investir. La société, l'économie et la survie est basée sur des "objets" (Ashton K., 2009).

Les concepts associés à l’IoT incluent : M2M (Machine to Machine), WoT (Web of Things) et IoE (Internet of Everything). Le concept de M2M (communication intermachines) est apparu dans les années 70 dénotant un ensemble de machines connectées entre elles et pouvant échanger de l’information sans l’intervention de l’humain, à l’aide des solutions de télécommunication PPP (liaison point à point) ou des moyens spécifiques aux fabricants. Elles peuvent faire du reporting (envoyer des rapports) vers un serveur M2M dédié au monitoring (surveillance) (Srinidhi N. et al., 2018). Contrairement à l’IoT, où les objets connectés sont hétérogènes et la connexion étant standardisée par un protocole (TCP/IP), les entités connectées doivent soit être homogènes (même protocole de communication : HDLC, PPP, SLIP, etc.), soit être construites par le même fabricant pour assurer la compatibilité et la compréhension des messages échangés. On en déduit que l’IoT est une forme évoluée, plus générale de la M2M.

Le terme IoE (Internet of Everything) été introduit par la société spécialisé dans le matériel réseau Cisco pour lancer un nouveau domaine du marketing, il s’agit de non seulement identifier et d’interconnecter les objets, y compris éventuellement les individus (voir Figure 24), mais aussi les connecter à l’internet (Srinidhi N. et al., 2018). Ce domaine de marketing se matérialise par les compagnies de collecte d’information qui traitent les Big Data chargées d’en retirer des bénéfices en la revendant à des entités de production ou des services, afin de répondre aux attentes des consommateurs et des citoyens. Un synonyme pour le terme IoE est le Web of Things (WoT) où les objets connectés utilisent une même base (le web) pour le stockage de l’information, ce qui permet aux objets de communiquer les uns avec les autres (voir Figure 25).

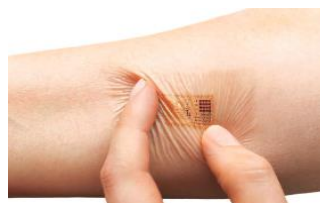


Figure 24 – Skin Embedded RFID tag (Sethi P. et al., 2016)

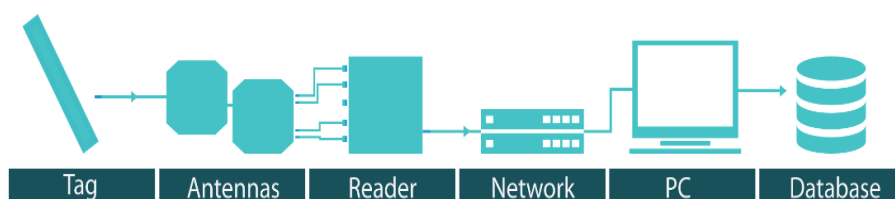


Figure 25 – Web Connected RFID (Srinidhi N. et al., 2018)

### 3. Radio-identification (RFID)

La technologie qui a révolutionné l'internet des objets est connue sous le nom de RFID (Radio Frequency Identification), où chaque objet est doté d'une puce électronique appelée étiquette RFID (RFID tag) lui permettant de communiquer via le protocole IP avec d'autres objets ou des serveurs. Les tags RFID se décomposent en trois types : passif, semi-passif et actif. Le premier étant capable seulement de répondre aux requêtes de l'émetteur (appelé Lecteur RFID, doté d'un émetteur et un récepteur) est alimenté par le signal du lecteur (voir Figure 26). Les tags semi-passifs et actifs sont alimentés par une batterie, cependant le tag semi-passif tout comme les tags passifs utilise la rétrodiffusion (backscatter) pour l'envoi de messages (Roberti M., 2011). Les tags RFID peuvent aussi être catégorisés en deux types selon leur portée, la première étant à proximité (jusqu'à 10 mètres) et l'autre pour la transmission lointaine (2 à 3 kilomètres). Des technologies autres que la RFID existent, il y en a également le NFC, Wi-Fi, et ZigBee, où la seule différence étant le moyen, la portée, le débit et le coût (Mehta R. et al., 2018).

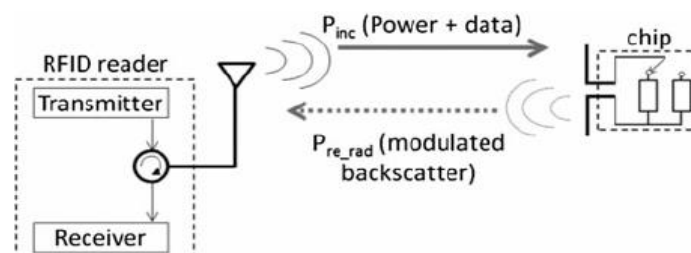


Figure 26 – RFID Communication (Mashal I. et al., 2015)

### 4. Architecture de l'internet des objets

Plusieurs architectures de l'IoT ont été proposées (RAMI 4.0, IIRA, IoT-A), la plus adoptée est celle qui le décrit comme étant construit dans un motif en couches (Antão L. et al., 2018). Ainsi on trouve deux architectures proposées par les chercheurs du domaine : l'architecture à 3 couches et l'architecture à 5 couches (Wu M. et al., 2010). La première est considérée la plus simple, elle comporte une couche de perception où se trouvent les capteurs d'identification d'objets smart et de collecte de l'information de l'environnement, une couche réseau responsable de connecter les objets et de la transmission de l'information, et une couche application destinée à offrir des services et satisfaire des requêtes de l'utilisateur (Sethi P. et al., 2017). L'architecture à 5 couches y intègre 2 autres couches, business et processing, la première responsable de gérer l'ensemble d'applications IoT, bien que la couche processing

(middleware layer) est responsable d'analyser et de traiter les données venant de la couche de transport, elle est considérée comme un pont entre la couche perception et celle du processing (voir Figure 27).

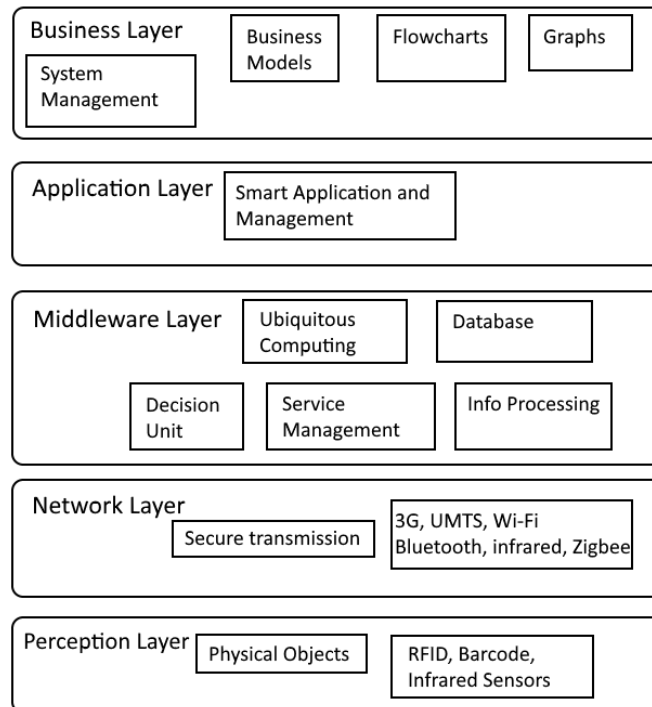


Figure 27 – IoT 5-Layer Architecture (Mehta R. et al., 2018)

L'Internet des objets est une nouvelle technologie qui est nécessaire de nos jours avec la croissance continue de données et de dispositifs électroniques. L'avantage de l'IoT est bien l'utilisation de ces ressources pour la fourniture de services. De point de vue de l'évacuation, Gokceli S. et al. (2017) ont proposé la mise en place d'un modèle d'évacuation IoT. Un réseau RFID / WSN (Wireless Sensor Network) a été mis en place à des fins d'identification et de positionnement des individus. Des instructions pour le plan d'évacuation sont transmises via des messages vocaux ou des appels téléphoniques.

## 5. Conclusion

Ce chapitre avait comme but d'introduire le concept d'internet des objets pour aider à réaliser la simulation en temps réel de l'évacuation via la technologie de RFID.

Le chapitre suivant rentre dans les détails de conception, de réalisation du système proposé d'évacuation,

# **Chapitre IV : Conception & Implémentation**



# 1. Introduction

Le processus d'évacuation nécessite des informations détaillées sur l'environnement réel afin de réussir la phase de prise de décision du plan d'évacuation. Le système doit être parfaitement au courant de toutes les données pouvant être utilisées pour gérer l'ensemble du processus. Ainsi, l'utilisation de la modélisation des informations du bâtiment (BIM) est essentielle pour atteindre notre objectif. Elle fournit une carte sémantique du bâtiment qui peut être utilisée par l'ordinateur sans autre prétraitement. En outre, l'utilisation mise en œuvre de l'internet des objets (IoT) permet au système de détecter l'état des objets à l'intérieur du bâtiment, y compris les évacués, en les étiquetant avec des étiquettes RFID afin de déterminer leur emplacement exact dans le bâtiment. Les systèmes basés sur la réalité virtuelle fournissent une meilleure interaction et une meilleure cognition à l'utilisateur. L'utilisation d'une telle solution peut donc faciliter l'apprentissage et la compréhension du système d'évacuation en question.

La phase de conception de ce chapitre vise à introduire et à concevoir une architecture d'un système d'évacuation guidée. Elle tient compte de l'évolution de l'environnement : y compris l'état du bâtiment, la gravité de la situation d'urgence et la situation de la population à l'intérieur de ce bâtiment. Elle intègre les solutions logicielles et matérielles mentionnées précédemment afin de fournir un environnement virtuel pour l'entraînement à l'évacuation. Elle comprend l'introduction de solutions techniques permettant d'atteindre les comportements souhaités, à savoir la navigation, le comportement de la foule, et la gestion de la congestion.

La phase de l'implémentation met l'accent sur les différents outils informatiques et composants du système EgressVR et montre le principe d'exécution des application relatives au système (l'application de l'entraînement à l'évacuation en utilisant la RV) et leurs interfaces graphiques respectives.

## **2. Conception**

La conception du modèle d'évacuation guidée choisi repose sur une méthode de modélisation mésoscopique, dans laquelle les deux visées sur la population (microscopique et macroscopique) sont utilisées pour réussir la gestion de la foule. La vue macroscopique englobe le système de détection des zones encombrées en temps réel via un graphe de congestion reflétant une vue globale de la structure du bâtiment en question, de la densité, et de la distribution de la population à l'intérieur de ce bâtiment (voir Section 2.4). La vue microscopique est utilisée à travers de la modélisation d'agents de l'intelligence artificielle qui servent de guides. Ces guides représentent des avatars pour chaque occupant, ce qui permet au système de détecter l'état, la vitesse et la position de l'occupant, ainsi que de simuler les actions des évacués à l'intérieur de la population.

Les agents ne prennent pas les mesures que l'occupant devrait prendre pour évacuer efficacement, mais suggèrent plutôt les meilleures actions et itinéraires en fonction de leur vision du monde virtuel et des données de congestion. Ils imitent les actions choisies par l'occupant et mettent à jour l'itinéraire d'évacuation en conséquence. En plus de la vue graphique globale du bâtiment, une vue plus détaillée de la structure du bâtiment est rendue possible via l'intégration de la modélisation des informations sur le bâtiment (BIM). Elle fournit ainsi une vue 3D dudit bâtiment où, à la fois l'espace continu et la structure du modèle d'automates cellulaires discrets sont mis à l'œuvre pour améliorer le système de navigation en cherchant les zones navigables et non occupées dans le bâtiment.

L'approche IoT avec l'information BIM ont été mis en œuvre afin de rendre vivant le bâtiment (Smart Building). Elle fournit une source d'information pertinente sur l'environnement et ses composants, facilitant par la suite la proposition d'une solution d'évacuation orienté-objet plus compréhensible, extensible, et moins complexe que les formulations et contraintes proposées par les modèles mathématiques ainsi que les formules et fonctions stochastiques relatives aux modèles de simulation.

### **2.1. Architecture du système**

#### **2.1.1. Présentation du système EgressVR**

L'architecture du système d'évacuation proposé qui décrit les principaux éléments fonctionnels avec leurs sous-éléments respectifs et la répartition de la charge du travail impliqué. Les données d'environnement initiales sont extraites d'une source de données BIM

et les modifications apportées au monde virtuel sont déclenchées par un ensemble d'actionneurs représentant l'espace IoT, ce qui fait du modèle BIM une représentation visuelle du bâtiment tandis que les actuateurs fonctionnent en tant que des détecteurs rapportant les mêmes changements subis par le bâtiment réel au monde virtuel.

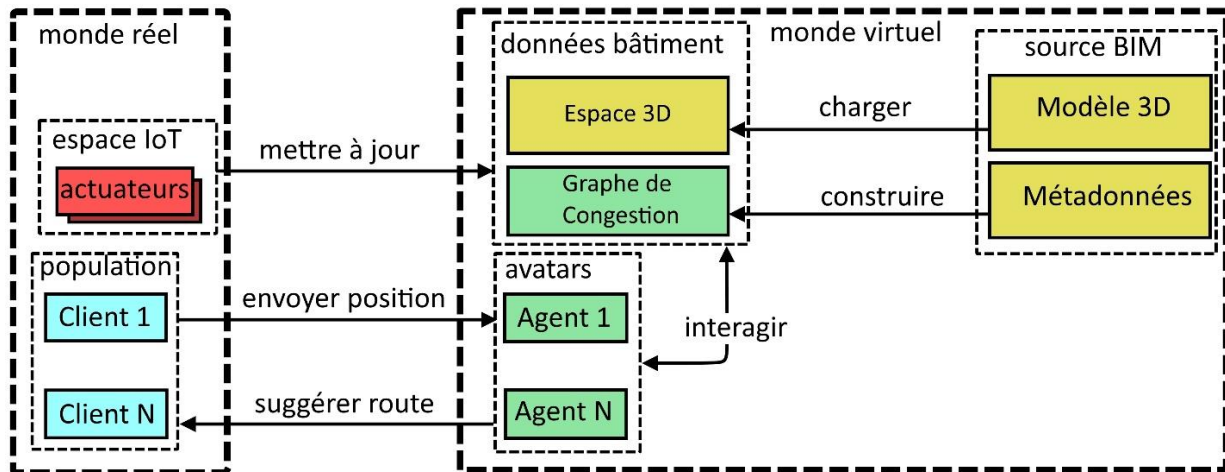


Figure 28 – Architecture du système EgressVR

Le système EgressVR est décrit par les acteurs de base suivants :

- **Client** : la personne évacuée dans le monde réel.
- **Agent  $i$**  : agent virtuel qui représente à la fois un guide et un avatar pour un client donné. Il récupère le plan d'évacuation transmis par le réseau de congestion et l'envoie au client.
- **Congestion graph** : représente la structure macroscopique du bâtiment et les informations relatives aux portes encombrées décrites par des agents de contrôle de flux (door agents) qui gèrent la redirection de flux d'évacués.

### 2.1.2. Diagramme de classes

Le diagramme de classes suivant (voir Figure 29) montre les différents acteurs et classes qui représente la solution d'évacuation proposée (voir plus loin à la section 2.4). Ce découpage de classes est motivé par l'approche orientée Acteur-Composant. Cette approche se repose sur l'encapsulation de comportements dans des composants. Ainsi, un acteur est une entité représentée par un composant racine (root component) et une hiérarchie de sous-composants fils. Ces composants coopèrent pour décrire le comportement (actions) global d'un acteur.

L'acteur Guide Agent utilise un composant AI Controller Guide qui décrit le comportement de l'agent guide à travers un arbre de comportement (voir Section 3.4 du



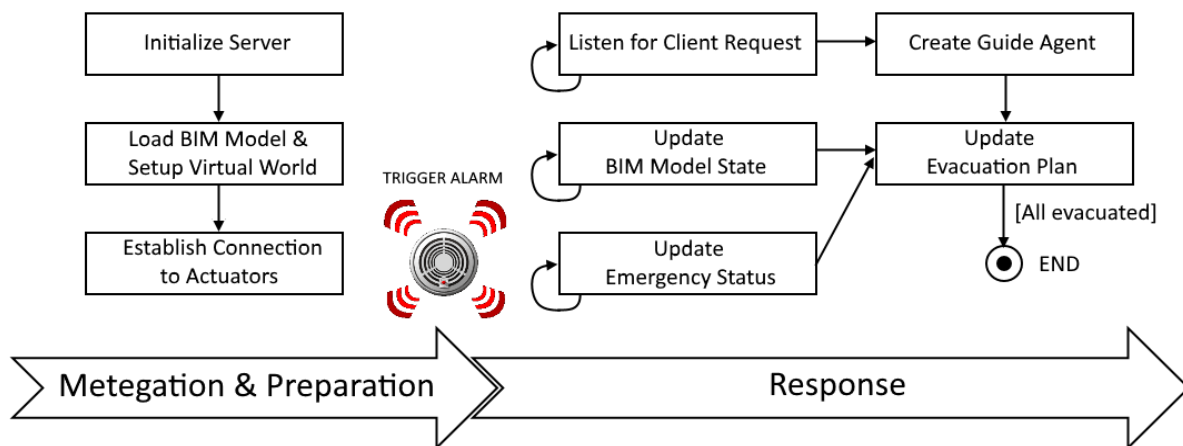


Figure 30 – Processus d'évacuation EgressVR

Comme le montre la figure ci-dessus, le plan d'évacuation est conçu et mis à jour en fonction de certains événements dans la phase de réponse au lieu d'être établi de manière statique dans les phases de mitigation et de préparation. Cela permet au système de prendre conscience de la tournure inattendue des événements, de faire face rapidement aux situations dans lesquelles des changements de l'état du bâtiment ont lieu, ou lorsque les occupants défont le plan d'évacuation établi en conséquence de comportements de panique. Le plan d'évacuation comprend la somme de tous les plans d'évacuation établis par chaque agent guide en fonction de l'état de l'occupant respectif et de l'ensemble de la population. Ce qui traduit les aspects micro et macro du modèle d'évacuation proposé.

Les différentes activités du processus d'évacuation et leurs propres tâches relatives sont répertoriées comme suit :

- **Mise en place de l'environnement virtuel**

- 1) Chargement de la représentation 3D du modèle BIM dans l'espace virtuel
- 2) Récupération de la structure du bâtiment (étages, salles, passages) à partir des métadonnées du bâtiment.
- 3) Mise en place du réseau de congestion en définissant des points de passage pondérés en fonction des coûts reliant différents étages, portes et passages aux points de sortie du bâtiment et en gardant une trace de la bande passante.
- 4) Etablir une connexion avec les actionneurs présents dans le bâtiment, pouvant inclure des détecteurs de chaleur, les états des portes et d'autres dispositifs de sécurité.
- 5) Calibrer les positions des actionneurs dans l'environnement virtuel pour les faire correspondre à celles du monde réel.

- 6) Mise à jour de l'état du danger à l'aide des détecteurs à l'intérieur du bâtiment pour suivre l'évolution du danger et anticiper les scénarios à venir.
- **Configuration des clients et des agents guides**
    - 1) Ecoute des demandes d'inscription des clients dans le processus d'évacuation.
    - 2) Récupérer la position du client à l'intérieur du bâtiment
    - 3) Créer un agent d'évacuation à la position respective du client en tant que représentant et guide pour ledit client.
  - **Guider les évacués**
    - 1) Chaque agent traite le chemin d'évacuation le plus rentable en termes de paramètres de distance et de congestion.
    - 2) L'agent suggère le plan d'évacuation traité au client concerné.
    - 3) Maintenir le plan d'évacuation à jour en fonction de l'état du monde virtuel, les actions des occupants, et de l'ensemble de la population.

## 2.2. Intégration BIM

Cette partie décrit comment l'échange de données BIM se fait pour importer la représentation 3D du bâtiment ainsi que décrire le format associé aux métadonnées du bâtiment en question.

### 2.2.1. Importation & mise en place de l'environnement BIM

Les moteurs de jeux, offrent un niveau élevé de privilèges d'accès aux ressources, car le principal public visé par ces premiers est bien des développeurs de solutions informatiques. Il leur permet d'intégrer et d'échanger des informations à partir d'autres logiciels en temps réel en fournissant un pont de communication, où il suffit juste d'implémenter l'interface fournie par le moteur de jeux afin d'établir la connexion cohérente entre les deux applications. Cela nous permet d'avoir une ligne de communication constante entre l'application actuelle et le logiciel BIM.

Ainsi, toute modification apportée au modèle BIM est répercutée sur le modèle faisant partie de l'application, offrant une vue dynamique sur le modèle BIM et les changements qu'il peut subir. Nous optons dans notre travail pour la même approche de conception pour charger le modèle en premier lieu. Les changements répercutés sur le BIM se fait seulement au niveau de l'application.

### 2.2.2. Metadonnées BIM

Les modèles BIM sont riches d'information grâce à la possibilité de représenter les objets BIM par des attributs et des propriétés (voir Section 2.2 du chapitre 1). Après avoir importé le modèle BIM à l'aide du module Data Smith de Unreal Engine on utilise le l'acteur Data Smith Actor pour avoir une liste d'objets BIM et d'en filtrer les salles et les portes (voir Section 2.1.2). La position de ces derniers est utilisée pour mettre en place un réseau de congestion (graphe) qui décrit l'aspect macro du modèle d'évacuation proposé (voir section 2.3).

La structure des métadonnées utilisées est la suivante :

- **BIMObject** représente un objet BIM
- **MetaData** est un composant de **BIMObject**
- **MetaData** : Ensemble de couples de chaînes de caractères (clé, valeur)
- **RoomType** est un sous type de **BIMObject** et a les métadonnées :
  - **Instance ID** : identificateur d'instance d'une salle
  - **Surface** : la surface de la salle
- **DoorType** est un sous type de **BIMObject** et a les métadonnées :
  - **Instance ID** : identificateur d'instance d'une porte
  - **Attached Room 1** : Identificateur de la salle attachée 1
  - **Attached Room 2** : Identificateur de la salle attachée 2
  - **Door Width** : la largeur de la porte

### 2.3. Représentation du bâtiment

Il est nécessaire que les agents guides ainsi que les agents de contrôle de flux soient conscients de l'environnement qui les entoure pour pouvoir mettre un plan d'évacuation. Deux représentations du bâtiment en question ont lieu. L'une macroscopique qui représente la structure logique du bâtiment. Elle est utilisée par l'agent de contrôle de flux pour rediriger le flux d'agents guides vers d'autres portes voisines en cas de congestion. L'autre est microscopique. Elle est réalisée à l'aide de la représentation 3D du modèle BIM et les maillages de navigation.

- **Représentation macroscopique :**

La structure du bâtiment est représentée par un graphe orienté. On définit deux types de nœuds : le nœud salle « Room Node » qui représente une salle, une chambre ou un couloir. Le nœud de contrôle de flux « Door Agent » qui représente une porte. Contrairement à la

définition mathématique proposée par le modèle de réseaux de flots, notre définition permet d'avoir plusieurs liens entre deux nœuds de type « salle ». Ceci nous permet de pouvoir représenter la structure de deux salles ayant plusieurs portes entre eux (voir Figure 31).

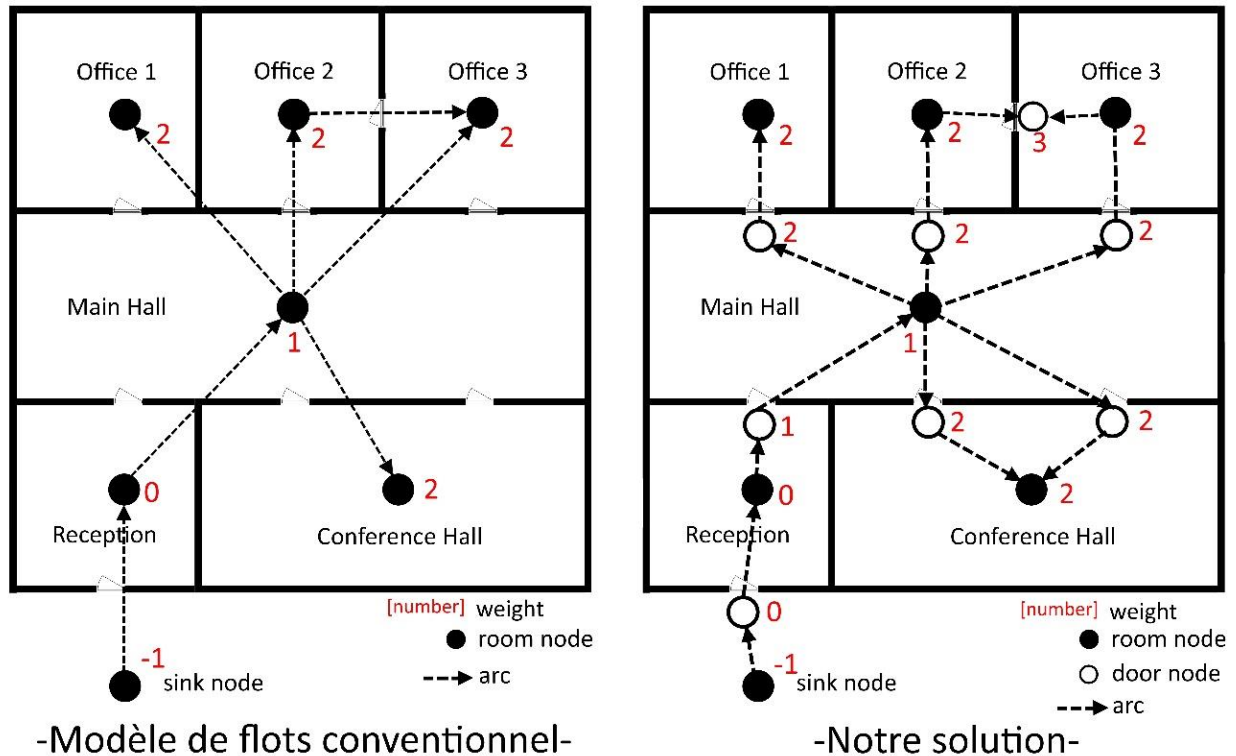


Figure 31 – Modèle de réseaux de flots amélioré

Par conséquent, un niveau élevé de gestion de flux entre salles est atteint. Tel que montré dans la figure ci-dessus (Figure 31), on peut affecter les évacués à des portes moins encombrées. Contrairement aux modèles conventionnels qui expriment seulement une seule valeur de flux entre deux salles, notre solution considère toute la structure du bâtiment lors de traitement de la solution. Ceci déminue l'ambiguïté de la solution envoyée aux évacués.

- **Représentation microscopique :**

L'agent guide pendant son cycle de vie tente de trouver un chemin optimal d'une salle à l'autre vers la sortie. Dans le modèle de réseau de flots la manière dont un agent se comporte pour naviguer dans une salle n'est pas prise en considération. Nous proposons une vue micro pour augmenter la conscience de l'agent de l'environnement qui l'entoure. La représentation 3D du modèle BIM est mise en place avec les maillages de navigation pour mettre à l'œuvre une structure d'automates cellulaires. Les modèles d'automates cellulaires conventionnels se basent sur le découpage du plan du bâtiment en une grille de cellules carrées interconnectées.



Cette représentation nécessite une grille de haute résolution (nombre élevé de cellules) pour pouvoir détecter les zones navigables dans un plan de bâtiment. Cependant, nous nous inspirons des jeux vidéo et d'autres travaux de recherche (voir section 3, chapitre 1) pour mettre en place des maillages de navigation qui optimisent la représentation cellulaire à l'aide de la résolution variable de cellules et l'introduction des cellules triangulaires (voir Figure 32).

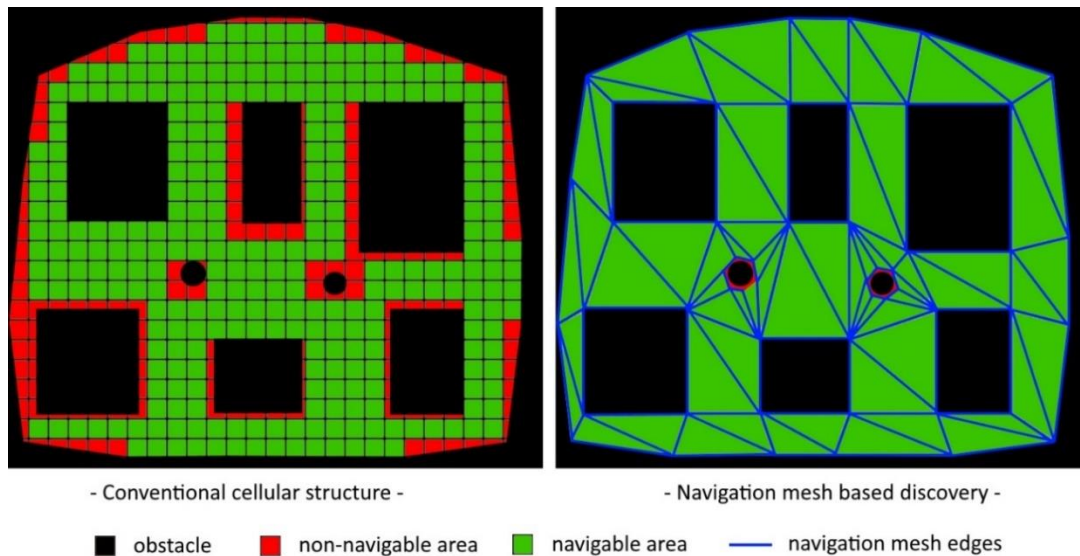


Figure 32 – Les maillages de navigation

Dans le cas des maillages de navigation, la recherche du chemin optimal se fait sur les centres des bords (edges) de ce premier. Toutefois, c'est à nous de choisir l'implémentation de la recherche optimale (voir Section 2.4). L'optimisation du chemin résultant se fait en utilisant une méthode de relaxation (String Pulling) pour satisfaire l'optimalité (Demyen D., 2007).

## 2.4. Solution proposée

### 2.4.1. Description de la solution

Notre solution est basée sur la gestion de flux d'occupants dans le bâtiment. Elle propose une amélioration sur les modèles de réseaux de flots pour le problème de flot le plus rapide (voir section 3.7, chapitre 1). Elle vise d'optimiser le temps d'évacuation en fournissant une représentation plus détaillée du bâtiment (voir Section 2.3) et en utilisant le critère du chemin le plus rapide.

Le modèle de flots conventionnel ne supporte pas l'excès de flux d'occupants au niveau des arcs et suppose leur coopération avec le plan proposé. Cependant, notre solution propose

l'introduction de files d'attentes pour gérer l'excès de flux. Le flux dans les portes se calcule par rapport à la vitesse des évacués inscrits à passer par la porte en question dans un moment  $t$  donné, contrairement aux modèles de flots qui supposent l'homogénéité des évacués. A l'aide d'une fonction d'évaluation  $f_c$  qu'on propose et la valeur de flux dans une porte  $n$ , on évalue le coût de transit pour un évacué  $e$  dans un moment  $t$  afin de satisfaire le critère d'optimalité, soit le chemin le plus rapide.

### 2.4.2. Fonction d'évaluation des coûts de transit

La fonction d'évaluation de coût de transit notée  $f_c(e, n)$  évalue le coût de transit en termes de temps pour un agent  $e$  de sa position actuelle vers le nœud d'évier  $d$  en passant par le nœud  $n$  et admettant une vitesse moyenne  $v(e)$ . On note  $\varepsilon$  l'espace minimal sans collision en termes de temps (secondes) entre deux particules  $e$  et  $e'$  (sa valeur exacte est discutée plus tard à la section 4.3.1). Elle indique le temps minimal requis pour un évacué  $e$  pour passer par une porte  $p$ . On définit ainsi un domaine de collision  $C$  qui coïncide avec l'ensemble pouvant arriver à une porte  $n$  dans le même temps en considérant la marge  $\varepsilon$  par :

$$C = \{e' / e' \in file(n) \wedge \left| \frac{distance(e',n)}{v(e')} - \frac{distance(e,n)}{v(e)} \right| < \varepsilon \}$$

La fonction d'évaluation de coût  $f_c(e, n)$  pour le temps de transit en considérant le temps d'attente à cause de la congestion en un moment  $t$  est donc défini par :

$$f_c(e, n) = \frac{distance(e,n)+distance(n,d)}{v(e)} + (\min(Card(C) - capacité(n), 0)) \times \varepsilon$$

### 2.4.3. Algorithme d'évacuation

Nous proposons une autre approche pour la résolution du programme linéaire compact (voir section 3.7, chapitre 1), soit la programmation orientée-événement. Les événements nous permettent de gérer l'attribution de flux à un moment  $t$  donné et évaluer le coût de transit à l'aide de la fonction d'évaluation  $f_c$ . Pour ce faire, nous définissons les éléments suivants :

- **D** : un ensemble de nœuds d'évier du graphe de congestion (section 2.3)
- **Nœud porte** : une structure contenant un poids, le flux courant dans ce premier, le flux maximum, une file d'attente pour les agents guides inscrits.
- **Nœud salle** : représente une salle, décrit le contexte pour les nœuds portes voisins.
- **Agent guide** : agent se déplaçant d'un nœud à l'autre, possédant une procédure d'envoi de requêtes d'acheminement.

Notre solution proposée est décrite par l'algorithme suivant :

- 1) Initialiser les nœuds portes :
  - a) Pour chaque (noeud porte  $p$ )
    - Affecter un poids infini à  $p$
    - Initialiser le flux courant à 0
    - Définir la *capacité*
  - b) Pour chaque (nœud d'évier  $d_i \in D$ )
    - Exécuter l'algorithme de remplissage par diffusion (poids = distance)
- 2) Redirection de flux :
  - Événement : OnReceiveRequest :
  - Contexte : nœud porte courant  $p$
  - Déclencheur : requête d'un agent guide  $e$
  - a) Soit un nœud porte  $m = p$
  - b) Chercher la porte **voisine** avec le moindre flux en utilisant la fonction  $f_c$
  - c) Affecter ce nœud à  $m$
  - d) Ajouter  $e$  à la file de  $m$
  - e) Changer la cible de  $e$  vers  $m$
- 3) Le prochain pas :
  - Événement : OnStartOverlap :
  - Contexte : nœud porte courant  $p$
  - Déclencheur : agent guide  $e$  a passé par la porte  $p$
  - a) Soit un nœud porte  $m = p$
  - b) Chercher la porte **prédécesseur** ayant le moindre flux en utilisant la fonction  $f_c$
  - c) Affecter ce nœud à  $m$
  - d) Ajouter  $e$  à la file de  $m$
  - e) Changer la cible de  $e$  vers  $m$

L'agent guide est responsable de la découverte du chemin le plus proche. Il coopère ensuite avec les agents de contrôle de flux (les nœuds portes) pour éviter les nœuds encombrés. Les nœuds portes se chargent désormais de la recherche du chemin optimal, soit le chemin le plus rapide. En utilisant la structure d'arbres de comportement (défini dans la section 3.4.1, chapitre 2) on décrit le comportement suivant de l'agent guide:

- L'agent cherche d'abord le chemin le plus proche en utilisant les maillages de navigation et la recherche A\*.
- En utilisant le résultat des points de navigation, on lance une requête d'acheminement vers un nœud porte.
- Si l'agent admet un rôle de simulation, il se déplace vers la cible envoyée par l'agent de contrôle de flux. Sinon, il imite l'action (nouvelle position) choisie par le client respectif.

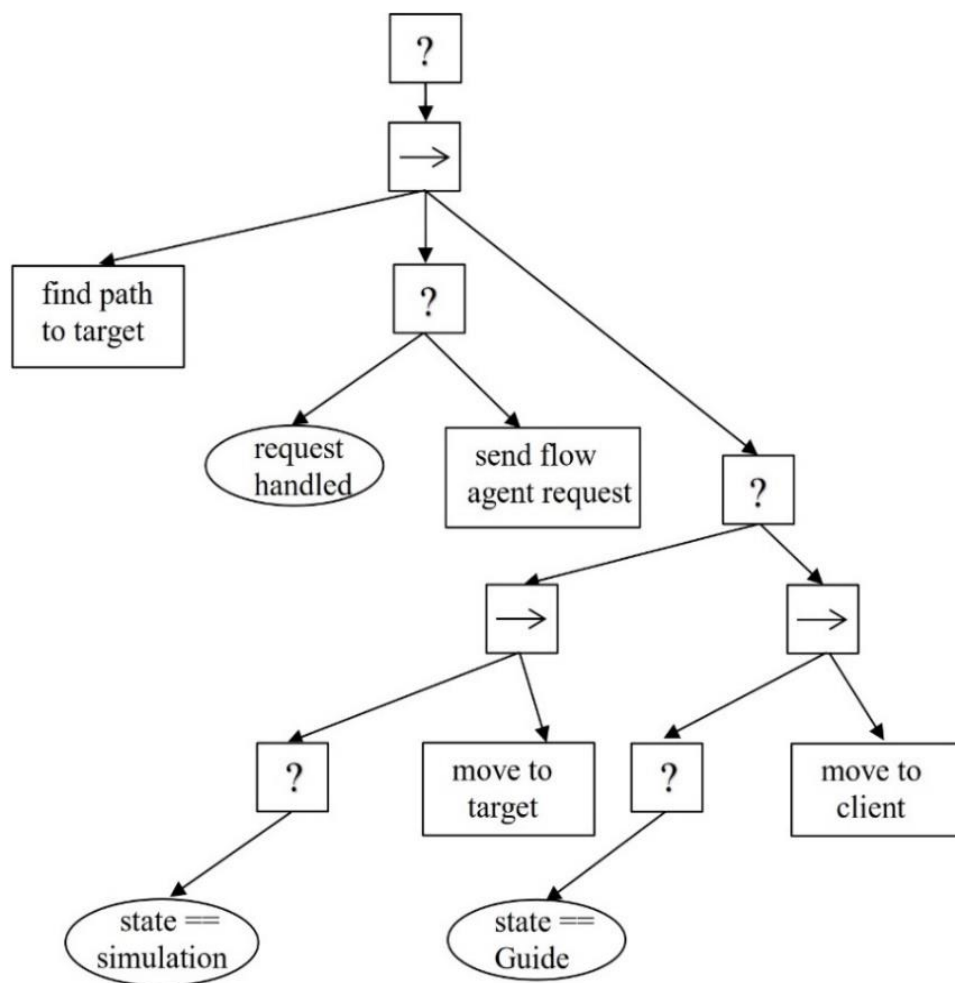


Figure 33 – L'arbre de comportement de l'agent guide.

La requête est réalisée à l'aide des événements de chevauchement entre le rayon (raycast) émis par l'agent et un composant boîte de collision appartenant au nœud de contrôle de flux. Le rayon utilise l'orientation de l'agent (entre deux points du plan) pour traduire son intention (voir Figure 34).

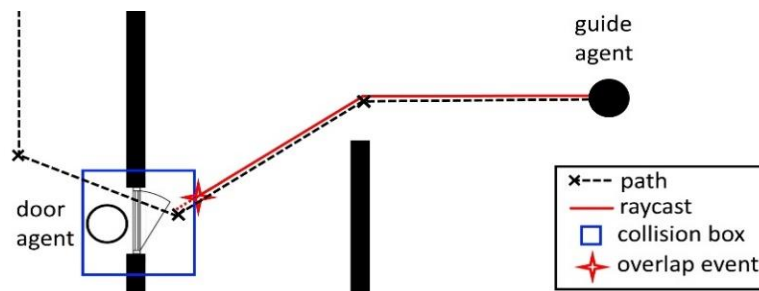


Figure 34 – L’interaction agent guide / agent de contrôle de flux.

## 2.5. Comparaison par rapport au modèle de flots conventionnel

Notre solution présente une amélioration sur les modèles de réseaux de flots. Elle fait ça en offrant une représentation plus détaillée du bâtiment en question. Elle prend en considération des paramètres qui ne sont pas prise en charge par le modèle de flots conventionnel. La solution proposée déminue l’effort d’évacuation par rapport au principe de conservation de flots et le flot le plus rapide. Ceci en favorisant le chemin le plus rapide par rapport à chaque évacué en tenant compte de l’état de la population, contrairement au modèle conventionnel qui ne considère que le temps total de l’évacuation. Le tableau suivant (voir Table 4) montre les principales différences entre le modèle conventionnel de réseaux de flot et la solution proposée et met l’accent sur les points forts de cette dernière :

Critère	Modèle de flots conventionnel	Notre Solution
Proposition / Principe	<ul style="list-style-type: none"> <li>• Conservation de flot / flot le plus rapide</li> </ul>	<ul style="list-style-type: none"> <li>• Gérer les excès de flot par les files d’attente</li> </ul>
Représentation	<ul style="list-style-type: none"> <li>• Globale / ambiguë</li> <li>• Nœud source doit exister/ difficile à décider son emplacement.</li> <li>• Temps de transit pour arcs est prédéfini</li> </ul>	<ul style="list-style-type: none"> <li>• Détaillée</li> <li>• Seulement les nœuds d’éviers doivent exister / facile à mettre en œuvre</li> <li>• Temps variable de transit basé sur la vélocité des évacués</li> </ul>
Plan d’évacuation	<ul style="list-style-type: none"> <li>• Optimal : temps global d’évacuation</li> <li>• Non-optimal : temps individuel d’évacuation</li> <li>• Suppose la coopération des évacués avec le plan (plan statique)</li> <li>• Plan entier fourni</li> </ul>	<ul style="list-style-type: none"> <li>• Optimal : temps global d’évacuation</li> <li>• Optimal : temps individuel d’évacuation</li> <li>• Compatible avec la défaillance du plan (plan dynamique)</li> <li>• Prochain pas seulement</li> </ul>
Evacué	<ul style="list-style-type: none"> <li>• Vélocité constant</li> </ul>	<ul style="list-style-type: none"> <li>• Vélocité variable</li> </ul>

Table 4 – Comparaison de la solution proposée par rapport au modèle de flots.

## 3. Implémentation

### 3.1. Environnement de développement

Nous utilisons le moteur de jeux UnrealEngine4 dans le processus de développement du système EgressVR et l'application mobile de réalité virtuelle (voir Figure 35)



Figure 35 – L'environnement de développement de Unreal Engine 4

Le logiciel source pour la manipulation des données BIM s'appelle Autodesk Revit. Les modules offerts par Unreal Engine 4 que nous utilisons sont :

- **Data Smith Plug-in** : Permet l'importation des modèles BIM et supporte une variété d'extensions pour les logiciels compatibles BIM (soit. rvt ce cas).
- **Google VR & Google VR Motion Controller** : Offre le support RV pour les dispositifs mobiles.
- **Recast Library** : Outils de construction de maillages de navigation et la recherche optimale A\* native.
- **Character AI** : Interface pour la création d'arbres de comportement.

Le langage de programmation utilisé au cours de développement est un langage de script visuel (Blueprint Visual Scripting) qui se traduit en code c++ à l'étape de compilation (voir Figure 36). Le choix de plateforme cible affecte le code généré pour qu'il soit compatible

avec cette première. Ceci dit, Notre solution est compatible avec plusieurs plateforme connues (Windows, Linux, MacOS).

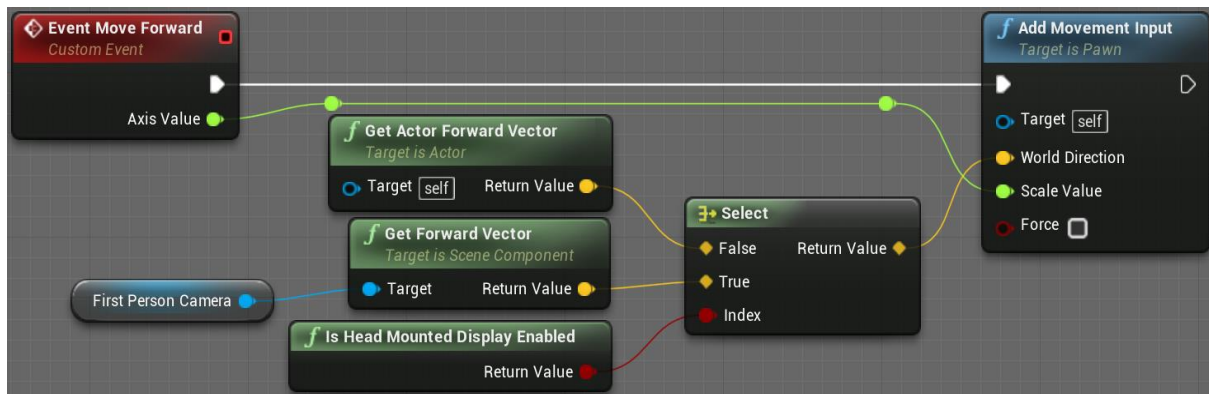


Figure 36 – Exemple de Unreal Engine 4 Blueprint Visual Script pour la RV

## 3.2. Principe de l'application EgressVR

### 3.2.1. Application Serveur

Une application serveur hébergeant le monde virtuel et les agents qui interagissent avec ce dernier prend en charge la totalité du traitement de simulation, de calcul et la génération du plan d'évacuation. L'application serveur est dédiée à la création de l'environnement BIM virtuel, à l'hébergement des agents guides et à leur liaison avec leurs clients respectifs, ainsi qu'au traitement des plans d'évacuation via le réseau de congestion construit. On pourrait aussi faire des simulations et des tests afin de mesurer la performance du plan établi (voir section 3.2.2).

Pour lancer le serveur, l'utilisateur clique sur le bouton « Initialize Server » après avoir choisi un modèle BIM préexistant (voir Figure 37). Le serveur se lance en mode « listen », c'est-à-dire qu'initialement il fonctionne seulement comme un serveur d'écoute qui satisfait les requêtes client à rejoindre le serveur. Un serveur d'écoute ne participe pas dans la session afin d'optimiser l'utilisation de ressources. On peut rejoindre la session en mode spectateur pour visualiser le processus d'évacuation en temps réel en cliquant sur le bouton « Spectate Session ».



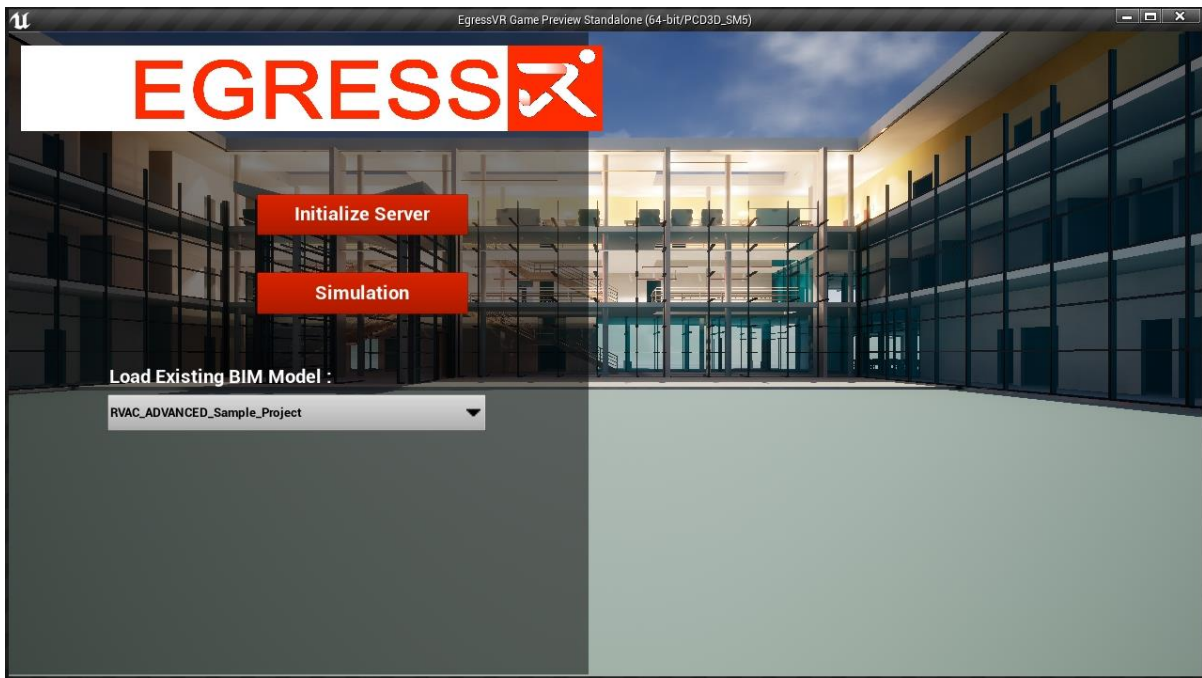


Figure 37 – Interface Serveur EgressVR

### 3.2.2. Entraînement & simulation

Les aspects formation et simulation de ce projet suivent des flux d'exécution légèrement différents de ceux du processus d'évacuation principal en termes d'entrées requises. Dans le processus de simulation, tous les agents-guides n'ont plus besoin de connexion établie avec les clients et leurs positions respectives. Ils se guident eux-mêmes en fonction des solutions optimales traitées. Bien que cela ne traduise pas le comportement des occupants du monde réel, cela permet de visualiser et d'affirmer l'optimalité des solutions traitées, ainsi que d'évaluer la fonctionnalité et les performances du système.

Le processus de formation à l'évacuation et l'entraînement en tant que jeu sérieux ressemble à la simulation dans presque tous les aspects. Cependant, les agents dans ce cas agissent en tant que pions contrôlables pouvant être utilisés par de vrais utilisateurs dans une application mobile autonome, interagissant avec le monde virtuel via des périphériques d'entrée sortie, y compris le matériel de la réalité virtuelle. Des événements déclencheurs de danger auront lieu pour immerger les utilisateurs à l'expérience et les aider à faire face aux situations de panique. La possibilité de plusieurs utilisateurs simultanés encourage la collaboration et le travail en équipe pour réussir le processus d'évacuation de manière optimale sans conflits. Le processus de formation est destiné à remplacer les exercices d'évacuation traditionnels. Il joue donc un rôle important dans la phase de préparation relative au cycle de vie d'une catastrophe



(voir section 3.2, Chapitre 1). Cela permet finalement aux utilisateurs de se familiariser avec le système.

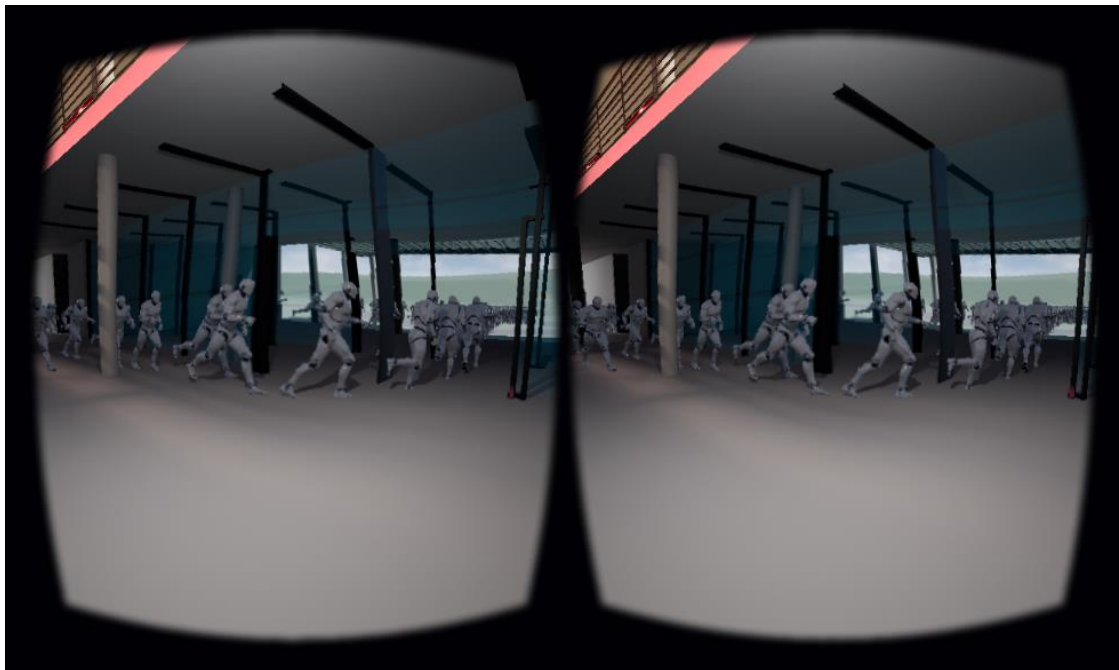


Figure 38 – Plateforme de réalité virtuelle EgressVR pour l’entraînement à l’évacuation.

Comme le montre la figure ci-dessus (voir Figure 38), le système EgressVR offre une expérience immersive pour s’entraîner à l’évacuation. Les utilisateurs se voient comme des agents évacués dans le monde virtuel où ils doivent suivre le plan proposé par la solution pour s’évacuer d’une manière optimale.

## 4. Tests & résultats

Dans cette partie, on propose une méthode d’évaluation pour mesurer la performance du système EgressVR en termes de paramètres d’optimalité (temps d’évacuation, effort, degré de congestion), soit le framework TEC. Le point commun entre ces paramètres est bien d’assurer la sécurité des évacués en : minimisant le temps d’évacuation, réduisant l’effort, et évitant la congestion et les conflits. Un scénario de test est mis en place pour mesurer l’indice de performance par rapport aux modèles d’optimisation pour le chemin le plus court.

### 4.1. Approche d’évaluation

La mesure de performance pour un système d’évacuation  $i$  est décrite par une métrique d’évaluation sujet à trois paramètres de d’optimalité, soit :

- **Temps d’évacuation :**

Noté  $T$ , calculé en faisant le rapport entre le temps optima  $T_o$  qui décrit le temps optimal d'évacuation (soit le chemin le plus court en supposant l'absence de collisions entre les évacués) et le temps d'évacuation pour la solution d'évacuation en question, soit  $T_i$ .

Ainsi :  $T = \frac{T_o}{T_i}$  avec  $0 \leq T_o \leq T_i$  et  $0 < T \leq 1$

- **Economie d'effort :**

Noté  $E$ , elle argumente l'optimalité du chemin généré. Elle est calculée en faisant le rapport entre l'effort optima  $E_o$  (décrit par la distance optimale parcourue par l'ensemble de la population sur le chemin le plus court en négligeant les collisions entre les évacués) et l'effort engendré par la solution d'évacuation en question, soit  $E_i$ .

Ainsi :  $E = \frac{E_o}{E_i}$  avec  $0 \leq E_o \leq E_i$  et  $0 < E \leq 1$

- **Valeur inverse de congestion :**

Notée  $C$ , calculé en faisant le rapport entre  $C_m - C_i$  et la congestion engendrée par la solution d'évacuation en question, soit  $C_i$ . Avec  $C_i = \sum_t^{T_i} Card(C_t)$  et  $C_m = \frac{n}{2} \times p$ . ( $C_m$  étant le nombre maximum de collisions pouvant survenir,  $p$  le nombre de portes dans le bâtiment,  $n$  le nombre d'évacués pour le scénario de test, et  $C_t$  les collisions entre les évacués en un moment  $t$  pour l'ensemble de portes).

Ainsi :  $C = \frac{C_m - C_i}{C_m}$  avec  $0 \leq C_i \leq C_m$  et  $0 \leq C \leq 1$

La métrique de performance pour un système d'évacuation  $i$  est définie donc par :

$$P_i = \frac{e_1 T + e_2 E + e_3 C}{e_1 + e_2 + e_3} \quad \text{Avec } 0 < P \leq 1$$

Les paramètres  $e_1$ ,  $e_2$  et  $e_3$  sont des facteurs de préférence pour les critères  $T$ ,  $E$  et  $C$  dans le cas où on favorise un critère par rapport à un autre.

## 4.2. Scénario de test

Le scénario de test est construit à partir d'un modèle BIM de  $1641m^2$  comportant 2 étages, 76 salles (y compris les escaliers et les halls), 5 escaliers, 101 portes (y compris les entrées d'escaliers), dont 5 sorties, et deux points d'évacuation (voir Figure 38).

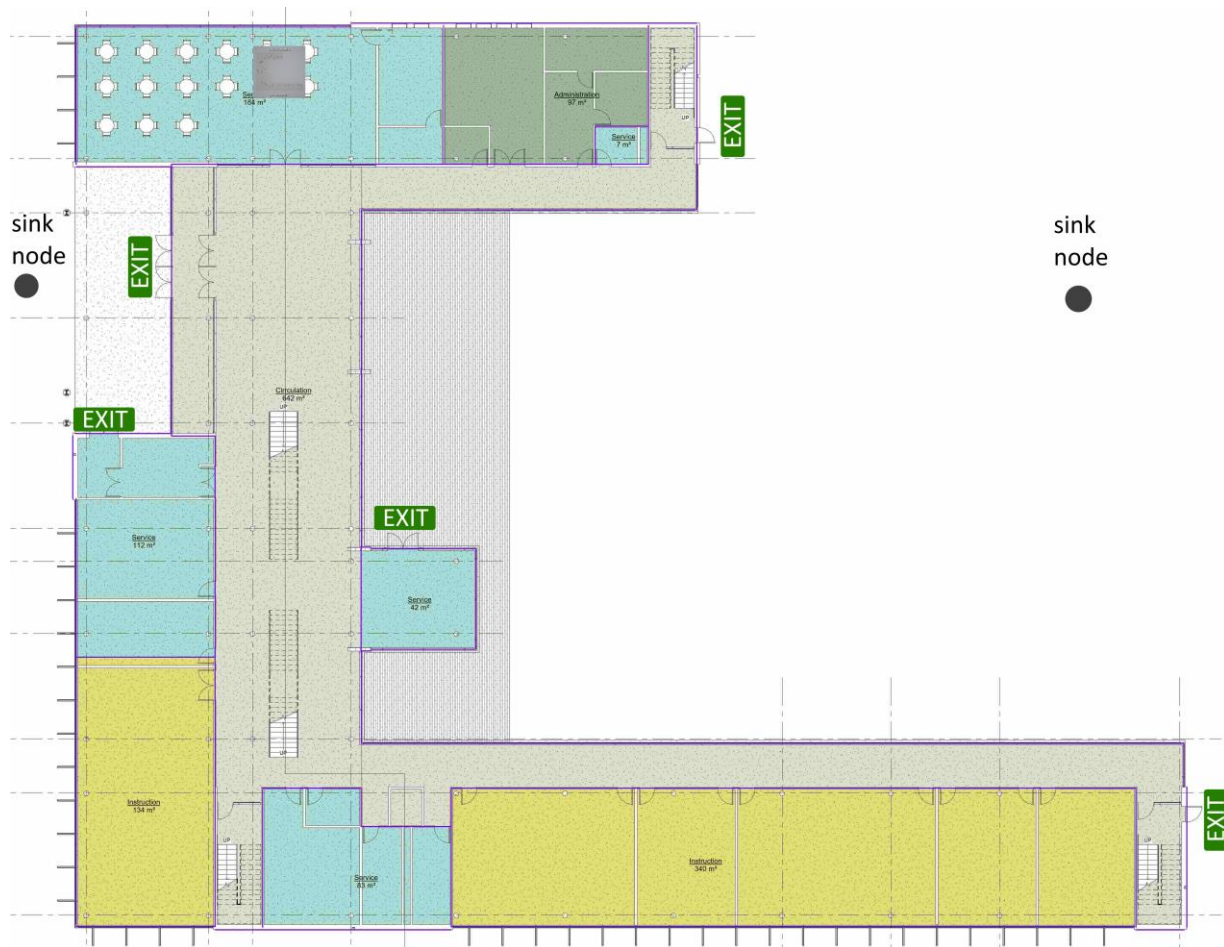


Figure 39 – Le plan 2D du modèle BIM relatif au scénario des tests

Le nombre d'évacués est de 180 personnes avec des vitesses variable et une vitesse maximale de  $3,5 \text{ m/s}$  qui est équivalent à  $12 \text{ km/h}$  pour un homme de 50-54 ans (kettle100.com, 2019).

### 4.3. Résultats & Observations

Cette partie est consacrée à la discussion du choix de paramètres pour le système EgressVR et le calcul d'indice TEC pour les deux méthodes d'évacuation : modèle d'automates cellulaires pour le chemin le plus court, et la solution proposée de gestion de flux pour le chemin le plus rapide du Système EgressVR.

#### 4.3.1. Choix de paramètres

Avant de passer aux détails et résultats de la mesure de performance TEC, on doit d'abord fixer la valeur de  $\varepsilon$  qui coïncide avec la valeur minimale sans collision en termes de temps (secondes) entre deux particules  $e$  et  $e'$  (voir section 2.4.2). Cela nous permet de

représenter le temps d'attente d'un évacué pour une porte  $t$  donnée lorsque cette dernière atteint sa capacité maximale à un moment  $t$  appartenant au temps d'évacuation.

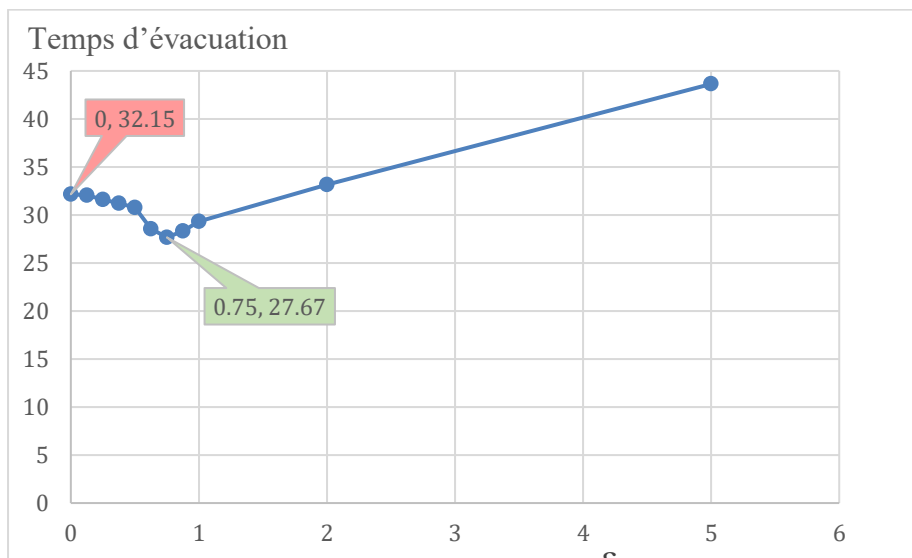


Figure 40 – La variation du temps d'évacuation par rapport à la variation de  $\varepsilon$

Suivant les résultats ci-dessus, on déduit qu'une valeur très basse de  $\varepsilon$  ainsi qu'une valeur très élevée ne contribue pas à l'optimisation du temps d'évacuation par rapport au cas du chemin le plus court ( $\varepsilon = 0$ ). Cependant vers les valeurs rapprochant  $\varepsilon = 0.75s$  on note une amélioration de 16,19%. En fait, la valeur de  $\varepsilon$  est équivalente au temps minimal pour qu'une personne passe par une porte de capacité 1. Une étude faite par Daamen et al. (2012) se concentre sur la relation entre la capacité de la porte, la composition de l'utilisateur et le niveau de stress, argumentant en moyenne 2,8 personnes par seconde pour une porte d'un mètre de largeur, qui est équivalent à 0.35 secondes pour une seule personne. Cependant, pour notre étude, la valeur de  $\varepsilon = 0.75s$  offre un meilleur résultat.

Les paramètres  $e_1 = 2, e_2 = 1$  et  $e_3 = 1$  sont choisis respectivement pour les indices  $T, E$  et  $C$ . On calcule en premier lieu l'indice  $P_{\text{sans préférences}}$ , après on calcule  $P_f$  l'indice pour la mesure préférée, soit  $T$ , le temps d'évacuation dans ce cas.

### 4.3.2. Indice TEC

Nous commençons d'abord par le calcul des paramètres de verdict qui sont : les valeurs optimales d'effort  $E_o$ , de temps d'évacuation  $T_o$ , et la valeur maximale de congestion  $C_m$  pour le scénario de test en question. Tel que mentionné précédemment (section 4.1), les collisions entre les évacués sont désactivées afin de fournir le climat théorique optimal pour l'évacuation. Le chemin d'évacuation généré pour chaque évacué est celui du chemin le plus court.

$$T_o = 21.35sec$$

$$E_o = 6571,47m$$

$$C_m = \frac{180}{2} \times 101 = 9090 \text{ collisions}$$

Indice	Modèle d'optimisation	EgressVR
$T$	$\frac{21.35}{32.15} = 0.664$	$\frac{21.35}{27.67} = 0.771$
$E$	$\frac{6571,47}{6945,96} = 0.946$	$\frac{6571,47}{7162,12} = 0.917$
$C$	$\frac{9090 - 438}{9090} = 0.951$	$\frac{9090 - 324}{9090} = 0.964$
$P$	0.853	0.884
$P_f$	0.806	0.855

Table 5 – L'indice de performance TEC pour la solution proposée EgressVR

Comme le montre la table ci-dessus (Table 5), la solution proposée réduit le temps d'évacuation tout en minimisant l'effort de transit et la congestion causée par la distribution de flux d'évacuées lors de l'évacuation.

## 5. Conclusion

Ce chapitre a présenté la conception et implémentation d'un système d'évacuation basé sur la gestion de flux d'occupants dans un bâtiment pour réussir l'évacuation en un temps minimal, et offre une plateforme d'entraînement à l'évacuation en utilisant la réalité virtuelle soit le système EgressVR. Il présente une amélioration sur les modèles de réseaux de flots. Une méthode pour la mesure de la performance du système EgressVR été proposée, soit la méthode TEC.

La partie suivante qui est la dernière présente une conclusion sur le travail réalisé. Elle mentionne un rappel d'objectifs, les objectifs atteints, ainsi que les difficultés rencontrées. Une perspective sur la thématique d'évacuation et la convergence entre le BIM et l'IoT aura lieu.

# Conclusion & Perspectives

## 1. Conclusion

Ce travail nous a permis de découvrir les aspects de recherche dans la thématique de l'évacuation et de cerner les différents modèles mis à l'œuvre. On a pu également découvrir l'importance du BIM dans le développement de solutions qui se reposent sur la virtualisation du monde réel. Cette partie est donc consacrée à citer les objectifs atteints et discuter les difficultés rencontrées.

### 1.1. Objectifs atteints

Les objectifs atteints avec succès au cours de ce travail sont :

- Nous avons simplifié le problème d'évacuation en citant les différents aspects de recherche dans la thématique d'évacuation et offrir un modèle compréhensif du processus d'évacuation en mettant à l'œuvre une structure d'arbres de comportement et une approche événementielle pour décrire ce processus.
- Nous avons conçu une solution dynamique au problème d'évacuation qui présente une amélioration sur les modèles de réseaux de flots capable de faire face aux tournures inattendues d'événements et à la défaillance engendrée par la population concernée suivant une approche orientée événement et la simulation en temps réel.
- Nous avons conçu un environnement virtuel de sessions collaboratives pour l'entraînement à l'évacuation.
- La gestion de la dynamique de la foule et la distribution de flux en mettant à l'œuvre une fonction d'évaluation de coût du transit en termes de temps et les collisions pouvant survenir à un moment  $t$  offrant par conséquent et résoudre les problèmes de congestion
- Nous avons pu optimiser le temps d'évacuation tout en minimisant l'effort impliqué dans le processus et réduisant la congestion. Un framework pour la mesure de performance d'une solution évacuation selon les critères : d'optimalité de temps d'évacuation, l'effort engendré, et la valeur de congestion est mis en place, soit la méthode proposée TEC.

## 1.2. Difficultés rencontrées

Malgré tout le progrès et les objectifs qu'on a pu atteindre avec succès, il y avait des obstacles qui ont affecté la qualité de travail réalisé, nous citons :

- L'absence de matériel et des éléments IoT nous a conduit de se contenter juste avec la simulation et l'entraînement pour les tests de validation. Le facteur de temps disponible pour réaliser ce travail joue un rôle important dans ce propos.
- La non-disponibilités de données relatives à la nature et la propagation de la situation d'urgence qui est supposé d'être fourni par les actuateurs IoT, et ainsi, la modélisation de la situation d'urgence est absente.
- L'absence d'un modèle BIM riche en métadonnées affecte la qualité du comportement d'agents guides et limite les potentiels du BIM dans l'aspect virtualisation du monde réel et la sémantique BIM. La raison étant que les modèles BIM gratuits disponibles sur le web sont de qualité basse.
- La complexité des calculs pour la mise en œuvre d'une solution qui prend en charge la détection de zones de congestion nous a conduit de se limiter à faire les calculs de collisions seulement au niveau des portes et des escaliers. En conséquence, des conflits au niveaux intermédiaires des salles peuvent avoir lieu.
- Vu la disponibilité du temps, la comparaison de la solution proposée se limite seulement aux modèles d'optimisation.

## 2. Perspectives

Ce travail impliqué dans la thématique de l'évacuation nous a ouvert de nouveaux horizons pour mieux comprendre le processus et la modélisation de l'évacuation, ainsi que la charge et la complexité de calcul impliqué. Malgré qu'on s'est limité à l'utilisation du concept de l'IoT seulement au niveau de la conception, on envisage mettre l'accent sur la convergence entre le BIM et l'IoT dans l'aspect de la virtualisation du monde réel et création des solutions intelligentes. Nous espérons mettre à l'œuvre la sémantique BIM pour renforcer l'aspect sémantique de l'évacuation. Il est notable de dire que l'intégration de modélisation de la situation d'urgence est une partie indispensable pour la réussite du processus d'évacuation sans pertes humaines, ainsi on espère introduire cette partie à ce travail existant.

# Bibliographie

- Alger M.W., 2015. Visual Design Methods for Virtual Reality. (n.p) : *September (2015)*.
- Antão L. et al., 2018. Requirements for Testing and Validating the Industrial Internet of Things. *10.1109/ICSTW.2018.00036*.
- Ashton K., 2009. That internet of things thing. *RFiD J. 22 (7) (2009) (p.97–114)*.  
*www.rfidjournal.com/article/view/4986*.
- Bayram V., 2016. Optimization models for large scale network evacuation planning and management: A literature review. *Surveys in Operations Research and Management Science, Volume 21, Issue 2, (p.63-84)*. Elsevier B.V. (2016). DOI : *10.1016/j.sorms.2016.11.001*.
- Bernardes S.M.F. et al., 2015. Methodological Approaches for Use Virtual Reality to Develop Emergency Evacuation Simulations for Training in Emergency Situations. *Procedia Manufacturing, Volume 3, (p.6313-6320)*, Elsevier B.V. (2015). DOI: *10.1016/j.promfg.2015.07.946*.
- Botella C. et al., 2017. The Science of Cognitive Behavioral Therapy. Chapter 22 - Virtual Reality and Other Realities (p.551-590). *Academic Press, Elsevier, Ltd. (2017)*. DOI: *10.1016/B978-0-12-803457-6.00022-2*.
- Carver N. & Lesser V., 1994. Evolution of blackboard control architectures. *Expert Systems with Applications 7(1), (p.1-30)*, Elsevier B.V. (1994). DOI: *10.1016/0957-4174(94)90023-X*.
- Chatfield T., 2009. Video Games now outperform Hollywood Movies. *The Observer*.  
*www.theguardian.com/technology/gamesblog/2009/sep/27/videogames-hollywood*.  
*27 Septembre 2009*.
- Colledanchise M. & Ogren P., 2018. Behavior Trees in Robotics and AI: An Introduction. *CRC Press (2018)*. ISBN: 9780429489105. DOI: *10.1201/9780429489105*
- Connolly T. et al., 2012. A systematic literature review of empirical evidence on computer games and serious games. *Elsevier Ltd*.
- Chen X. et al. 2017. Social force models for pedestrian traffic – state of the art. *Transport Reviews*, DOI: *10.1080/01441647.2017.1396265*.
- Crespo R. et al., 2015. Virtual Reality Application for Simulation and Off-line Programming of the Mitsubishi Movemaster RV-M1 Robot Integrated with the Oculus Rift to



- Improve Students Training. *Procedia Computer Science*. 75. DOI: 107-112. 10.1016/j.procs.2015.12.226.
- Cruz F.R.B et al., 2005. Approximate Analysis of M/G/c/c State Dependent Queueing Networks. *Preprint submitted to Computers & Operations Research*.
- Daamen W, Hoogendoorn SP, 2010. Emergency door capacity: influence of door width, population composition and stress level. *Fire Technology* 48, 1 (2012) (p.55-71), Springer (2012). DOI: 10.1007/s10694-010-0202-9.
- Daganzo C.F. et al., 1977. On Stochastic Models of Traffic Assignment. *Transportation Science* 11(3) (p.253-274). INFORMS, Maryland, USA (1997). DOI: 10.1287/trsc.11.3.253.
- Dávideková M. et al., 2017. Utilization of Virtual Reality in Education of Employees in Slovakia. *Procedia Computer Science Volume 113* (p.253-260), Elsevier B.V. (2017).
- Demyen D., 2007. Efficient Triangulation-Based Pathfinding. *Accepted dissertation submitted for the degree of Master of Science. Alberta University, Canada* (2007).
- Dorsemaine B. et al., 2015. Internet of Things: A Definition & Taxonomy. *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, Cambridge, 2015, pp.72-77. DOI: 10.1109/NGMAST.2015.71*
- Eastman C. et al., 2011. BIM Handbook: A guide to building information modelling for owners, managers, designers, engineers and contractors. *John Wiley & Sons*.
- Gokceli S. et al., 2017. IoT in Action: Design and Implementation of a Building Evacuation Service. *Journal of Computer Networks and Communications, vol. 2017, Article ID 8595404, 13 pages, Hindawi* (2017). DOI: 10.1155/2017/8595404.
- Gregory J., 2009. Game Engine Architecture. *A K Peters, Ltd. Taylor and Francis Group, LLC*.
- Hanawa Y. et al., 2016. The mixed evacuation problem. *In Combinatorial Optimization and Applications - 10th International Conference, COCOA 2016, Proceedings* (pp. 18-52). (Vol. 10043 LNCS). Springer Verlag. DOI: 10.1007/978-3-319-48749-6\_2
- Hazelton M.L., 1998. Some Remarks on Stochastic User Equilibrium. *Transportation Research Part B: Methodological, Volume 32, Issue 2, (p.101-108). Elsevier Science, Ltd. DOI: 10.1016/S0191-2615(97)00015-5*.
- Helbing D. et al., 1995. Social Force Model for Pedestrian Dynamics. *Institute of Theoretical Physics, University of Stuttgart, 70550 Stuttgart, Germany*.
- Helbing D. et al., 2005. Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions. *Transportation Science* 39(1):1-24, Institute for

*Operations Research and the Management Sciences (INFORMS)*. DOI: 10.1287/trsc.1040.0108.

- Kia S. et al., 2013. Review of Building Information Modeling (BIM) Software Packages Based on Assets Management. *Amirkabir University of Technology*.
- Kenneth A.R. et al., 2015. The Internet of Things: An Overview Understanding the Issues and Challenges of a More Connected World. *Internet Society (October 2015)*. ISOC (2015).
- Kuligowski E. et al., 2010. A Review of Building Evacuation Models. Technical Note 1680, 2nd Edition, Fire Research Division Engineering Laboratory, National Institute of Standards and Technology, (November 2010).
- Lalanda P., 1997. Two complementary patterns to build multi-expert systems. *PloP (1997)*, Monticello, Illinois.
- LaValle S. M., 2017. Virtual Reality. *Cambridge University Press (2017)*.
- Lee I. et al., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons (2015)*, Volume 58, Issue 4, (p.431-440).
- Levy S. M., 2010. Green and sustainable buildings. *Construction Process Planning and Management: An Owner's Guide to Successful Projects* (p. 223-249). *Butterworth-Heinemann (2010)*. DOI: 10.1016/B978-1-85617-548-7.00009-4.
- Lewis M. & Jacobson J., 2002. Game engines in scientific research - Introduction. *Commun. ACM*. 45. (p.27-31). DOI: 10.1145/502269.502288.
- Liu X. et al, 2017. A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS). *International Journal of Geo-Information, February 2017*.
- Lovreglio, R., 2018, A Review of Augmented Reality Applications for Building Evacuation. *17th International Conference on Computing in Civil and Building Engineering, ICCCBE 2018, Tampere, Finland*.
- Manley M.T., 2012. Exitus: An Agent-Based Evacuation Simulation Model for Heterogeneous Populations. *A dissertation submitted in partial fulfillment of the requirements for the degree of PhD*.
- Marer P. et al., 2015. Missing Link Discovered: Planting Csikszentmihalyi's Flow Theory into Management and Leadership Practice by using FLIGBY, the official Flow-Leadership. *ALEAS Simulations, Inc. 12121 Wilshire Blvd, Suite 525 Los Angeles, CA 90025, USA*.
- Mashal I. et al., 2015. Choices for interaction with things on Internet and underlying issues. *Ad Hoc Networks* 28, (p.68-90). Elsevier B.V. May 2015. DOI: 10.1016/j.adhoc.2014.12.006.

- Mazuryk T. et al., 1999. Virtual Reality - History, Applications, Technology and Future.
- Makransky G. et al., 2017. Adding immersive virtual reality to a science lab simulation causes more presence but less learning. *Learning and Instruction* 60, (p.225-236), Elsevier (2019). DOI: 10.1016/j.learninstruc.2017.12.007
- Mehta R. et al., 2018. Internet of Things: Vision, Applications and Challenges. *International Conference on Computational Intelligence and Data Science (ICCIDS 2018), Procedia Computer Science* 132 (p.1263–1269), Elsevier Ltd. (2018). DOI: 10.1016/j.procs.2018.05.042.
- Millà-Gutiérrez A., 2016. Crowd Modeling and Simulation on High Performance Architectures. *Computer Architecture and Operating Systems department. A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy University of Autònoma de Barcelona, Cerdanyola del Vallès, Barcelona (2016).*
- Muhanna A. M., 2015. Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions. *Journal of King Saud University - Computer and Information Sciences*, 27(3), (p.344-361). Elsevier B.V. (2015). DOI: 10.1016/j.jksuci.2014.03.023
- Nash, J. (1951). Non-Cooperative Games. *Annals of Mathematics*, 54(2), second series, (p.286-295). DOI :10.2307/1969529.
- Nystrom R., 2014. Game Programming Patterns. (n.p): Nystrom (2014). ISBN: 978-0-9905829-2-2.
- Okazaki et al., 1993. Study of Simulation Model for Pedestrian Movement with Evacuation and Queuing. *Department of Architecture and Environmental Design, Architecture and Environmental Engineering, Kyoto University, Yoshida Honmachi, Kyoto, 606-01 Japan.*
- Olivier A.H. et al., 2014. A Virtual Reality Platform to Study Crowd Behaviors. *Transportation Research Procedia, Volume 2* (p.114-122), Elsevier B.V. (2014). DOI: 10.1016/j.trpro.2014.09.015.
- Permana, S et al., 2018. Comparative Analysis of Pathfinding Algorithms A \*, Dijkstra, and BFS on Maze Runner Game. *IJISTECH (International Journal of Information System & Technology)*. DOI: 10.30645/ijistech.v1i2.7.
- Pyakurel, Urmila. 2018. Network Flow Models for Evacuation Planning. *Journal of the Institute of Engineering*, 13(1) (p.90-101). TUTA/IOE/PCU (2017), Nepal. DOI: 10.3126/jie.v13i1.20353

- Richardson O. et al., 2019. Effects of Environment Knowledge in Evacuation Scenarios Involving Fire and Smoke: A Multiscale Modelling and Simulation Approach. *Fire Technology* 55 (2) (p.415–436), Springer (2019). DOI: 10.1007/s10694-018-0743-x.
- Roberti M., 2011. What Is a Semi-passive RFID Tag. Retrieved from: [www.rfidjournal.com/blogs/experts/entry?8117](http://www.rfidjournal.com/blogs/experts/entry?8117)
- Roof K. et al., 2008. Public Health: Seattle and King County’s Push for the Built Environment. *Journal of Environmental Health*, July 2008.
- Sethi P. et al., 2017. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 9324035, 25 pages, (2017). DOI: 10.1155/2017/9324035.
- Srinidhi N. et al., 2018. Network optimizations in the Internet of Things: A review. *Engineering Science and Technology, an International Journal* (2018), Elsevier B.V.
- Sydänheimor L. et al., 2005. Effects of size and shape of metallic objects on performance of passive radio frequency identification. *International Journal of Advanced Manufacturing Technology* 30(9) (p.897-905), Springer-Verlag London Ltd. (2005). DOI: 10.1007/s00170-005-0133-7.
- Tah J. et al., 2017. A state-of-the-art review of built environment information modelling BeIM. *University of Zagreb*.
- Takahashi D., 2019. Khronos Group releases early OpenXR spec for AR and VR hardware standards. Retrieved from: [www.venturebeat.com/2019/03/18/khronos-group-releases-early-openxr-spec-for-ar-and-vr-hardware-standards/](http://www.venturebeat.com/2019/03/18/khronos-group-releases-early-openxr-spec-for-ar-and-vr-hardware-standards/)
- Tan K.Y., 2011. Evacuation Timing Computations Using Different Evacuation Models. *thesis report submitted in partial fulfilment of the requirements for the degree of Master of Engineering in Fire Engineering*.
- Tolani D. et al., 2000. Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. *Graphical Models* 62(5) (p.353-388), Elsevier Ltd. (2000). DOI : 10.1006/gmod.2000.0528.
- Vermuyten H. et al., 2016. A Review of Optimization Models for Pedestrian Evacuation and Design Problems. *Safety Science* 87 (p.167-178) (2016), Elsevier Ltd. DOI : 10.1016/j.ssci.2016.04.001.
- Violeta N. et Arta B., 2017. The integration of BIM in education: a literature review and comparative context. *Global Journal of Engineering Education*, Volume 19, Number 3, 2017.

- Wang B., 2014. End user Oriented BIM enabled Multi-functional Virtual Environment Supporting Building Emergency Planning and Evacuation. *A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy, Cardiff University.*
- Wang J. & Sun J., 2014. Principal Aspects regarding to the Emergency Evacuation of Large-scale Crowds: A Brief Review of Literatures until 2010, Elsevier Ltd.
- Wang J. et al., 2014. Multi-Agent Based Modeling and Simulating for Evacuation Process in Stadium. *Journal of System Science and Complexity* (2014) 27 (p.430–444), Springer-Verlag Berlin Heidelberg (2014). DOI: 10.1007/s11424-014-3029-5.
- Wu M. et al., 2010. Research on the Architecture of Internet of Things. *In Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 5. IEEE, 2010, pp. V5–484.*
- Yan W. et al., 2011. *Integrating BIM and gaming for real-time interactive architectural visualization. Automation in Construction* 20 (2011) (p.446–458), Elsevier B.V. (2010). DOI: 10.1016/j.autcon.2010.11.013.
- Zeng W. et al., 2014. A Modified Social Force Model for Pedestrian Behavior Simulation at Signalized Crosswalks, *Elsevier L.*