

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider Biskra
Faculté des Sciences Exactes et des sciences de la Nature et de la Vie
Département d'Informatique

N° d'ordre :

Série :



Mémoire

Présenté en vue de l'obtention du diplôme de Magister en Informatique

Option: **Synthèse d'images et vie artificielle**

Titre :

Simulation réaliste de Croisement de flux de piétons par intégration de la capture de mouvement

Par :

Amar DELLABANI

DJEDI Nouredine

CHERIF Foudil

BABAHENINI Med Chaouki

MELKEMI Kamel Eddine

Professeur Université de Biskra

MCA Université de Biskra

MCA Université de Biskra

MCA Université de Biskra

Président

Rapporteur

Examineur

Examineur

Juin 2012

Remerciement

Je remercie "Dieu" de nous avoir donné la patience, la santé, le courage d'avoir arrivé jusque là.

Je tiens à remercier le docteur. Foudil CHERIF qui m'a honoré par son encadrement, pour sa direction, son orientation, ses conseils et toutes ses remarques constructives pour le bon déroulement de notre travail.

Je tiens à remercier Pr. DJEDI Noureddine, Dr.

BABAHENINI Mohammed Chaouki et MELKEMI Kamel Eddine d'avoir accepté d'être les examinateurs.

Merci à mes parents de m'avoir amené jusqu'ici, merci également à toute ma grande famille, frères, sœurs, neveux et nièces.

Enfin, je remercie ma femme qui m'a encouragé et qui a su me supporter, pour le meilleur et pour le pire pendant ces années. A ma fille Ilham.

En fin, nous remercions, avec toute la suprême sincérité, tous ceux qui nous ont aidés de près ou de loin à terminer et à réaliser ce travail.

Amar DELLABANI

المتحركة السلوكية هي طريقة الأكثر استخداما لمحاكاة حركة البشر في العالم الافتراضي. هذا الأسلوب يعطي للإنسان الافتراضي حكم الذاتي (صناعة القرار). الإنسان الافتراضي يتصور بيئته، وعلى ذلك يتصرف ويقوم باتخاذ قرارات من تلقاء نفسه (قرارات مستقلة) تتعلق بالحالة أو البيئة المتصورة، من أجل إظهار سلوك متناسقة على مقربة من الكائنات الحية المحاكاة.

كجزء من هذه المذكرة، نود أن نقدم نموذج يحاكي تدفق عبور حشود المشاة واقعيا. اقترحنا نموذج عام لمحاكاة الحشد. نقترح نموذجين للمحاكاة عبور المشاة، في أول نموذج الإنسان الافتراضي عبارة عن أشكال هندسية بسيط مع الحركة إنسحابية، والنموذج الثاني الإنسان الافتراضي هو نموذج ثلاثي الأبعاد والحركة استخدمنا الحركة الملتقطة من الواقع.

اهتمنا بتطوير الجوانب التالية: تخطيط المسار (البحث عن مسار) في بيئة افتراضية، تجنب التصادم بين المشاة أثناء الاحتشاد، إدراج الحركات الملتقطة على الإنسان الافتراضي، وأخيرا استخدام القواعد الاجتماعية للحركة في الحشود.

كلمات البحث:

الحشود، وتخطيط المسار ، والسلوك، وتجنب الاصطدام، والملاحة، والتقاط الحركة.

Résumé

L'animation comportementale est une méthode la plus utilisée pour simuler le mouvement des humains dans un monde virtuel. Cette méthode donne de l'humain virtuel l'aspect d'autonomie. Les humains virtuels perçoivent leur environnement, agissent sur ce dernier et surtout prennent d'eux-mêmes des décisions en rapport avec la situation perçue dans le but d'exhiber un comportement cohérent proche de l'organisme vivant simulé.

Dans le cadre de ce mémoire, nous souhaitons apporter de contributions à la simulation de croisement de flux de piétons de manière réaliste. Nous proposons un système générique de simulations de foules. nous proposons deux modèles de simulation du croisement de piétons, dans le premier modèle l'humain virtuel est modélisé par de forme géométrique simple et l'animation par translation, et dans le deuxième modèle l'humain virtuel est modélisé par un modèle 3D et l'animation par la capture de mouvement.

Nous nous sommes intéressés à développer les aspects suivants : la planification de chemin (la recherche de chemin) dans un environnement virtuel; l'évitement de collisions entre les piétons pendant le croisement; l'intégration des mouvements capturés à nos humanoïdes et enfin l'utilisation des règles sociales de mouvement de la foule.

Mots-clés :

Foules, planification de chemin, comportement, évitement de collision, navigation, capture de mouvement.

Abstract

The behavioral animation is a preferred method to simulate the movement of humans in a virtual world. This method gives the appearance of virtual human autonomy. The Virtual humans perceive their environment, act on it and above all they are taking independent decisions related to the perceived situation in order to exhibit consistent behavior close to the simulated living organism.

As part of this thesis, we wish to make contributions to the simulation of pedestrian crossing flow realistically. We propose a generic framework for crowd simulation. we propose two simulation models of pedestrian crossing, in the first virtual human model is modeled by a simple geometric form and animation translation, and in the second model the virtual human is modeled by a 3D model and animation by motion capture.

We were interested to develop the following aspects: path planning (search path) in a virtual environment, the avoidance of collisions between pedestrians during the crossing, the integration of our movements captured humanoids and finally the use of social rules of movement of the crowd.

Key words:

Crowds, path planning, behavior, collision avoidance, navigation, motion capture.

Sommaire

1	L'ENVIRONNEMENT VIRTUEL	7
1.1	INTRODUCTION	7
1.2	REPRESENTATION DE L'ENVIRONNEMENT	7
1.2.1	REPRESENTATION EXACTE DE L'ENVIRONNEMENT	8
1.2.1.1	Triangulation de Delaunay	8
1.2.1.2	Décomposition en trapèzes	11
1.2.2	REPRESENTATION APPROXIMATIVE DE L'ENVIRONNEMENT	12
1.2.2.1	Modèles à base de grilles	12
1.2.2.1.1	Grilles uniformes	12
1.2.2.1.2	Grilles hiérarchiques	13
1.2.2.1.3	Décomposition par rectangles	14
1.2.2.2	Cartes de cheminement	15
1.2.2.2.1	Graphe de visibilité	15
1.2.2.2.2	Diagramme de Voronoï généralisé	16
1.2.2.2.3	Cartes de cheminement probabilistes	17
1.3	RECHERCHE DE CHEMIN	18
1.3.1	CALCUL SUR LES GRAPHES	18
1.3.1.1	Discrétisation de l'espace et construction du graphe	18
1.3.1.2	Les algorithmes de parcours	19
1.3.1.3	Dijkstra	20
1.3.1.4	A étoile(A*)	20
1.3.1.5	Évolutions du A*	21
1.3.2	CRITERES DE RECHERCHE DE CHEMIN	22
1.3.3	METHODES A CHAMPS DE POTENTIEL	23
1.4	ENVIRONNEMENT VIRTUEL INFORME	24
1.4.1	SMART OBJECTS	25
1.4.2	ANNOTATIONS	26
1.4.3	ONTOLOGIES	26
1.5	INTEGRATION DES OBJETS INTERACTIFS	27
1.5.1	LES OBJETS STATIQUES	27
1.5.2	LES OBJETS DYNAMIQUES	27
1.6	CONCLUSION	27
2	NAVIGATION D'HUMAINS VIRTUELS	30
2.1	INTRODUCTION	30
2.2	LES INDIVIDUS	30
2.2.1	MODELISATION DES PROPRIETES DES HUMAINS VIRTUELS	30
2.2.1.1	La perception	31
2.2.1.2	L'émotion	31
2.2.1.3	Le comportement	32
2.2.1.4	L'action	33
2.2.1.5	La Mémoire	34
2.2.2	LE COMPORTEMENT D'INDIVIDU	34
2.2.2.1	Comportement de recherche et fuite	35

2.2.2.2	Comportement d'arrivée.....	35
2.2.2.3	Comportement d'évitement d'obstacles	36
2.2.2.4	Comportement de suivi de chemin.....	37
2.2.2.5	Evitement de collision non-alignée.....	37
2.3	LE GROUPE DES INDIVIDUS.....	38
2.3.1	LE COMPORTEMENT DE GROUPES	39
2.3.1.1	La séparation.....	40
2.3.1.2	La cohésion.....	41
2.3.1.3	L'alignement.....	41
2.3.1.4	Le suivi de leader.....	42
2.4	LA FOULE	43
2.4.1	SIMULATION DE FOULE.....	43
2.4.2	RELATIONS ENTRE LES INDIVIDUS	46
2.5	PROCESSUS MENTAUX DE NAVIGATION.....	46
2.5.1	PLANIFICATION D'UN CHEMIN	47
2.5.2	LOCOMOTION	48
2.6	CONCLUSION	49
3	<u>CAPTURE DE MOUVEMENTS</u>	<u>52</u>
3.1	INTRODUCTION	52
3.2	LES SYSTEMES DE CAPTURE DU MOUVEMENT	53
3.2.1	LES SYSTEMES D'ACQUISITION OPTO-ELECTRONIQUES.....	55
3.2.1.1	Marqueurs passifs.....	55
3.2.1.2	Marqueurs actifs	56
3.2.2	LES SYSTEMES D'ACQUISITION MAGNETIQUES	58
3.2.3	LES SYSTEMES D'ACQUISITION MECANIQUES	59
3.2.4	LES SYSTEMES D'ACQUISITION VIDEOS.....	60
3.2.5	LES SYSTEMES D'ACQUISITION SONORES.....	62
3.3	MELANGE DE MOUVEMENTS CAPTURES	63
3.4	ADAPTATION DE MOUVEMENTS CAPTURES	64
3.5	POSITIONNEMENT DES MARQUEURS	65
3.6	CONCLUSION	67
4	<u>MODELE DE CROISEMENT DE FLUX DE PIETONS.....</u>	<u>70</u>
4.1	INTRODUCTION	70
4.2	L'OBJECTIF DU SYSTEME.....	70
4.3	L'ARCHITECTURE DE NOTRE SYSTEME	70
4.3.1	L'ANIMATION	71
4.3.1.1	Perception	71
4.3.1.2	Décision	72
4.3.1.3	Action	72
4.4	STRUCTURE DE NOTRE SYSTEME.....	72
4.4.1	MODELISATION DE LA SCENE.....	75
4.4.2	NOTRE MODELE DE FOULE	75
4.4.3	DISCRETISATION LA SCENE	76
4.4.4	RECHERCHE DE CHEMIN	77
4.4.5	BOUCLE D'ANIMATION	77
4.4.5.1	La prévision de collision	78

4.4.5.2	Evitement de collision	80
4.4.5.2.1	Collision face à face	81
4.4.5.2.2	Collision en arrière	81
4.4.5.2.3	Collision de côté	81
4.4.5.3	Comportements d'évitement	82
4.4.5.3.1	Croisement de groupes	85
4.4.5.3.2	Goulots d'étranglement	85
4.4.5.3.3	Alignement dans deux directions	86
4.4.5.3.4	Cas des passages étroits	87
4.4.5.4	Formation de files	87
4.4.5.5	Recalcul du chemin	89
4.4.5.6	La procédure de déplacement les acteurs	89
4.5	CONCLUSION	90
5	<u>IMPLEMENTATION ET RESULTATS</u>	<u>92</u>
5.1	INTRODUCTION	92
5.2	SIMCROI	92
5.3	L'ENVIRONNEMENT DE DEVELOPPEMENT	94
5.3.1	3DS MAX	94
5.3.2	OGRE 3D	95
5.3.2.1	L'entité	97
5.3.2.2	Le SceneManager	97
5.3.2.3	Ogre Exporter	98
5.4	L'ARCHITECTURE SIMCROI V1	98
5.4.1	RESULTATS	99
5.5	L'ARCHITECTURE SIMCROI V2	103
5.5.1	L'EXPORTATION LES INDIVIDUS	103
5.5.2	RESULTATS	104
5.6	CONCLUSIONS	107

Liste des figures

Fig.I.1.	Discrétisation de l'espace avec une triangulation Delaunay contrainte.....	9
Fig.I.2.	Discrétisation de l'espace avec une triangulation Delaunay filtrées.	10
Fig.I.3.	Discrétisation en trapèzes.....	11
Fig.I.4.	Discrétisation de l'espace avec une grille régulière.....	13
Fig.I.5.	Discrétisation de l'espace avec une grille hiérarchique type quadtree.	14
Fig.I.6.	Discrétisation par rectangle.....	15
Fig.I.7.	Graphe de visibilité calculé sur l'environnement.	16
Fig.I.8.	Graphe Voronoï généralisé.....	17
Fig.I.9.	Exemple de cartes de cheminement générées à partir d'une discrétisation par triangulation de Delaunay contrainte.	19
Fig.I.10.	Fonctionnement de l'algorithme de Dijkstra.....	20
Fig.I.11.	Fonctionnement de l'algorithme de A*.....	21
Fig.I.12.	Une carte de champ de potentiel.	24
Fig.I.13.	Exemple de Smart Object : interaction avec un casier.....	25
Fig.II.1.	Structure du Modèle comportemental.	31
Fig.II.2.	Le comportement d'arrivée.....	35
Fig.II.3.	L'évitement d'obstacle.	36
Fig.II.4.	Le comportement de suivre de chemin.....	37
Fig.II.5.	Le comportement d'évitement de collisions non-alignées.	38
Fig.II.6.	Notion de voisinage.	40
Fig.II.7.	Le comportement de séparation.....	40
Fig.II.8.	Le comportement de cohésion.....	41
Fig.II.9.	Le comportement de d'alignement.	42
Fig.II.10.	Comportement de suivi de leader.	42
Fig.II.11.	Architecture de contrôle des foules issue des travaux de Raupp Musse.....	45
Fig.II.12.	Représentation mentale hiérarchique pour la planification de chemin selon J.Wiener et H. Mallot.....	48
Fig.II.13.	Représentation de l'aire de Goffman (en vert).	49
Fig.III.1.	Principe de système de capture de mouvement.....	53
Fig.III.2.	Exemple de système d'acquisition opto-électronique.	55
Fig.III.3.	Principe du système d'acquisition optique.	56
Fig.III.4.	Système Optotrak Smart Markers.	57
Fig.III.5.	Exemple de système d'acquisition magnétique.....	58
Fig.III.6.	Exemple de système d'acquisition mécanique.	60
Fig.III.7.	Système d'acquisition vidéo par cycles de prédiction/correction.....	61
Fig.III.8.	A gauche, les deux positions à mélanger. A droite, l'animation de transition entre les ces deux positions.....	63
Fig.III.9.	Set de 37 marqueurs disposés sur le corps du sujet.	65
Fig.IV.1.	L'architecture du système.	71
Fig.IV.2.	Boucle d'animation comportementale.	71
Fig.IV.3.	Organigramme du système.....	74
Fig.IV.4.	Modèle de foule.	76
Fig.IV.5.	Discrétisation la scène.....	77
Fig.IV.6.	Prévision de collision.	78
Fig.IV.7.	Les trois types de collisions.	79

Fig.IV.8.	Evitement d'obstacle par un groupe.....	80
Fig.IV.9.	Croisement de groupes.....	85
Fig.IV.10.	Goulot d'étranglement.....	86
Fig.IV.11.	Alignement dans deux.....	86
Fig.IV.12.	Passage étroit.....	87
Fig.IV.13.	Lane formation in pedestrians.....	88
Fig.IV.14.	Procédure de déplacement d'acteur.....	90
Fig.V.1.	L'interface de première version d'application.....	93
Fig.V.2.	L'interface de deuxième version d'application.....	93
Fig.V.3.	Modèle de l'humanoïde 3D.....	95
Fig.V.4.	L'architecture de première version d'application.....	98
Fig.V.5.	Résultat de l'algorithme A*.....	99
Fig.V.6.	Croisement entre deux groupes.....	100
Fig.V.7.	L'alignement dans deux directions.....	101
Fig.V.8.	Passage étroit.....	101
Fig.V.9.	Croisement entre de flux.....	102
Fig.V.10.	L'architecture de deuxième version d'application.....	103
Fig.V.11.	Croisement entre deux groupes.....	104
Fig.V.12.	L'alignement dans deux directions.....	105
Fig.V.13.	Passage étroit.....	106
Fig.V.14.	Passage dans un couloir.....	107

Liste des Tableaux

Tab II.1. Comparaisons des différentes caractéristiques comportementales des foules en fonction du type de contrôle qui leur est imposé.	45
Tab III.1. Liste des marqueurs disposée sur des points anatomiques superficiels de référence	66
Tab IV.1. Tableau de traitement de l'évitement de collision	82

Introduction générale

Introduction générale

La vie quotidienne de l'homme est conditionnée par ses mouvements qui représentent son principal moyen d'interaction avec l'environnement (déplacements, atteintes et manipulations d'objets, etc.). La performance et le succès de ces interactions dépendent de ses capacités motrices. En cherchant à comprendre le fonctionnement du contrôle moteur humain, il est possible d'identifier les facteurs influant sur l'efficacité de la réalisation d'une tâche, et ainsi de savoir pourquoi certaines personnes ont de meilleures performances que d'autres. Les applications de l'étude du mouvement sont nombreuses, dans des domaines variés : sport, musique, handicap, robotique, ergonomie, etc.

La simulation de la foule d'humains virtuels est un sujet très récent dans la recherche en infographie, et a pour but de générer facilement les mouvements, actions et comportements d'un grand nombre d'individus. Il existe plusieurs types d'applications de simulation de foule. Parmi elles, nous pouvons citer le peuplement d'environnement virtuel, en assurant à l'animateur un moyen facile de programmer des foules, décrire les mouvements de foule pour étudier les flux de personnes durant une simulation, et donner de l'intelligence et de l'autonomie aux foules pour obtenir des acteurs automatiquement animés.

L'animation d'un grand nombre de personnes est importante dans beaucoup de situations. La gestion des grandes foules évoque plusieurs problèmes, par exemple le problème de croisement de deux foules de piétons aux deux directions opposées, les travaux ont été effectués sur ce problème basé sur comment croiser de flux de piétons et assurer l'évitement de collision entre les piétons. Il dispose ainsi d'un grand nombre de modèles et d'acquis, mais reste tout de même très ouvert encore aujourd'hui.

L'animation d'humains virtuels est un problème récurrent en synthèse d'images. Deux approches ont été développées au cours des dernières années. La première vise à améliorer les techniques de cinématique inverse et de manipulation d'images clés pour augmenter le contrôle de l'infographiste sur l'animation. La seconde approche se focalisait sur la génération automatique de mouvements à partir de la simulation des forces animant un corps humain.

L'animation comportementale donne à l'humain virtuel l'aspect d'autonomie. Les humains virtuels perçoivent leur environnement, agissent sur ce dernier et surtout prennent

d'eux-mêmes des décisions en rapport avec la situation perçue dans le but d'exhiber un comportement cohérent proche de l'organisme vivant simulé.

Contributions

Dans le cadre de ce mémoire, nous souhaitons apporter de contributions à la simulation de croisement de flux de piétons de manière réaliste, en tenant en compte des situations réelles.

Nous nous sommes intéressés de développer les aspects suivants : la planification de chemin (la recherche de chemin) dans un environnement virtuel ; l'évitement de collisions entre les piétons pendant le croisement ; l'intégration des mouvements capturés à nos humanoïdes et enfin l'utilisation des règles sociales de mouvement de la foule.

Organisation du mémoire

Pour répondre à cette problématique, nous organiserons ce document en deux parties :

La première partie est un développement de l'état de l'art sur tous les aspects qui entrent dans le domaine de l'animation comportementale et se décompose en trois chapitres :

Dans le *chapitre 1*, « L'environnement virtuel », nous présentons les différents types des représentations d'environnement virtuel, et les différentes méthodes de recherche le chemin, et l'environnement virtuel informé, et enfin l'intégration des objets dans l'environnement.

Dans le *chapitre 2*, « Navigation d'humains virtuels », nous présentons les propriétés des humains virtuels et les comportements des individus, des groupes et de la foule, et enfin le processus mental de navigation.

Dans le *chapitre 3*, « Capture de mouvements », nous présentons les différents systèmes de capture de mouvements, et les méthodes de mélange et adaptation de mouvements capturés, et enfin le positionnement des marqueurs sur le corps du sujet.

La seconde partie du mémoire est dédiée à la présentation des travaux effectués dans le cadre de ce mémoire. Elle se divise en deux chapitres :

Dans le *chapitre 4*, « Modèle de croisement de flux de piétons », nous proposons deux modèles de simulation du croisement de piétons, dans le premier modèle l'humain virtuel est modélisé par de forme géométrique simple et l'animation par translation, et dans le

deuxième modèle l'humain virtuel est modélisé par un modèle 3D et l'animation par la capture de mouvement.

Dans le *chapitre 5*, « L'implémentation et résultats », nous exposerons les résultats obtenus.

La conclusion mettra en avant les perspectives d'évolution du modèle proposé.

État de l'art

Chapitre I

L'environnement Virtuel

1 L'environnement Virtuel

1.1 Introduction

L'évolution des modèles informatiques avec l'approche microscopique s'est intéressée à l'aspect individuel de la décision relative au déplacement. C'est à cet aspect que s'intéresse la planification de chemin. Le but de la planification de chemin n'est plus, comme avec la navigation réactive, de gérer les interactions locales entre individus, mais de trouver globalement les endroits par lesquels passer pour atteindre une destination.

On voit donc apparaître avec ces modèles la nécessité d'une description topologique, permettant de représenter et d'organiser l'environnement de simulation. Cette description topologique peut être plus ou moins fine, exacte ou approximative, et peut éventuellement contenir des informations sémantiques.

Cette description est ensuite exploitée par des algorithmes de recherche de chemin. Ceux-ci peuvent se baser sur différentes informations issues de l'environnement, qu'elle soit géométriques comme les distances, ou qualitatives comme la charge cognitive associée au chemin.

Nous allons présenter, dans le présent chapitre, les différentes méthodes de décomposition de l'environnement, afin de choisir la meilleure représentation de notre environnement. En cite deux types de représentation, la décomposition approximative de l'espace libre, et la décomposition exacte. Dans la deuxième partie, en présente quelques méthodes utilisées pour générer la carte de cheminement. Dans la troisième partie, en présente les environnements virtuels informés.

1.2 Représentation de l'environnement

L'environnement dans lequel les entités évoluent est représenté par une géométrie statique. Cette géométrie traduit la structure de l'environnement et donc les contraintes imposées lors de la navigation. De manière générale, cette géométrie représente les obstacles sous la forme de polygones en **2D** et sous la forme de polyèdres en **3D**. Cette hypothèse sert de base à un certain nombre de méthodes de représentation de l'espace et s'avère être corrélée avec les méthodes de modélisation d'environnement. Que cet environnement ait été produit avec un logiciel de modélisation **3D** ou un outil dédié, pour permettre une interprétation

rapide du point de vue des entités le peuplant, il doit être représenté dans une structure de données appropriée. Dans ce domaine, plusieurs méthodes ont été proposées dans le monde de la robotique [Lat91]. Ces méthodes consistent à discrétiser l'espace de manière à identifier les zones dans lesquelles l'entité peut naviguer. Cette discrétisation peut ensuite être utilisée pour représenter la topologie des lieux au travers de la notion d'accessibilité entre zones. Cette propriété s'avère nécessaire pour la recherche d'un chemin entre deux points.

Nous allons étudier deux méthodes : la représentation exacte de l'environnement, et la représentation approximative de l'environnement.

1.2.1 Représentation exacte de l'environnement

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

1.2.1.1 Triangulation de Delaunay

La triangulation de Delaunay [BY98] crée un ensemble de triangles en réunissant des points fournis en entrée. L'algorithme d'unification respecte pour contrainte que le cercle circonscrit à un triangle ne contienne aucun autre point que les trois sommets qui le composent. Une conséquence de cette propriété est que l'angle minimum d'un triangle produit est maximisé. La complexité de l'algorithme d'unification est en $O(n \cdot \ln n)$, avec n le nombre de points fournis en entrée.

La propriété la plus intéressante de cette triangulation est que chaque point y est relié à son plus proche voisin par l'arête d'un triangle. Ainsi, cette triangulation peut être utilisée pour représenter l'espace navigable, les points d'entrée étant issus des obstacles. Une autre utilisation possible de cette triangulation est cette fois dynamique lors de la simulation [LD04], pour calculer un graphe de voisinage entre les entités mobiles, qui serviront cette fois-ci de points d'entrées.

La triangulation de Delaunay contrainte [Che87] permet de modérer les contraintes de la version standard afin de conserver certaines arêtes de la définition graphique d'origine. Pour se faire, la contrainte du cercle circonscrit à un triangle est modifiée, pour spécifier que

tout point inclus dans ce cercle ne peut être relié à tous les points du triangle sans intersecter un segment contraint.

Cette triangulation étend la propriété du plus proche voisin en permettant d'y associer une notion de visibilité. Si l'on considère que les arêtes contraintes coupent la visibilité d'un point à l'autre, la propriété de plus proche voisin devient : chaque point est relié à son plus proche voisin visible.

La triangulation de Delaunay contrainte est aussi utilisée pour obtenir une subdivision spatiale en triangles [KBT03], en introduisant les obstacles à la navigation sous la forme d'autant de segments contraints (Figure I.1). Il a été prouvé [BY95] que le nombre de triangles produits lors d'une telle subdivision est linéaire en fonction du nombre de points, indiquant que la discrétisation est donc directement proportionnelle à la complexité géométrique de l'environnement. Cette propriété présente un avantage certain comparativement aux méthodes approximatives, où la discrétisation est fonction de la précision désirée de la représentation.

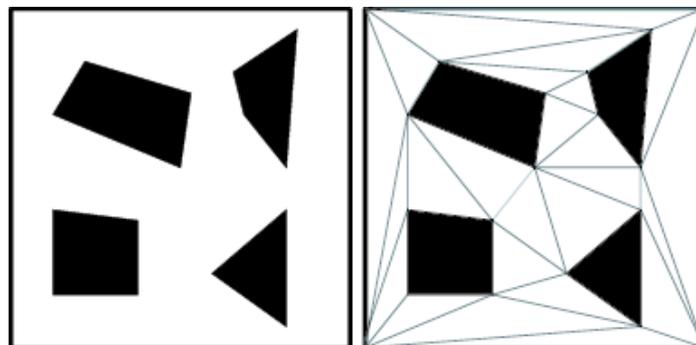
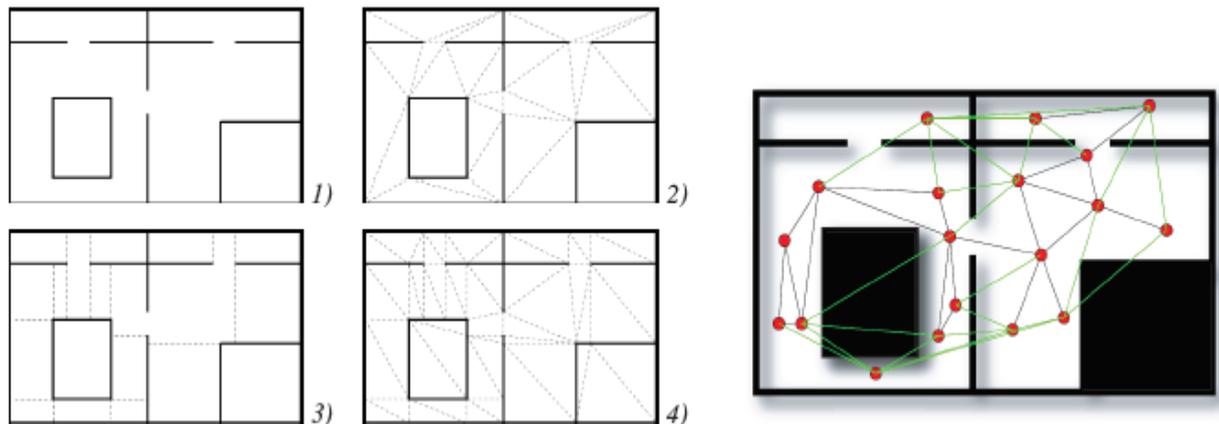


Fig.I.1. Discretisation de l'espace avec une triangulation Delaunay contrainte.

La triangulation de Delaunay filtrée [LD04] est une extension de la version contrainte. Deux types de filtrages sont proposés, l'un ayant pour effet d'augmenter le nombre de triangles produits afin d'affiner la représentation de l'environnement, l'autre de le diminuer afin de tenir compte de la visibilité pour l'élaboration d'un graphe de voisinage.

Premièrement, concernant son application à la subdivision spatiale, la triangulation de Delaunay est filtrée par l'ajout progressif de contraintes représentant les goulets d'étranglement (Figure I.2(a)). Ainsi, en considérant l'ensemble des arêtes produites, on est sûr de représenter tous les rétrécissements présents dans l'environnement.

Deuxièmement, concernant son application aux graphes de voisinage, la triangulation de Delaunay est filtrée sur un principe équivalent à la triangulation contrainte, en supprimant les arêtes intersectant des obstacles de l'environnement (Figure I.2(b)). Ainsi, cette triangulation peut servir à déterminer trivialement quels sont les voisins directs visibles d'une entité, et avec un simple parcours de graphe peut sélectionner les entités visibles à une certaine distance.



- (b) Filtrage pour subdivision spatiale.
- 1) Environnement d'origine.
 - 2) Triangulation de Delaunay contrainte.
 - 3) Calcul des plus courtes distances des sommets aux segments voisins.
 - 4) Triangulation filtrée.

- (a) Filtrage pour graphe de voisinage. Seules les arêtes noires seront conservées, n'entrant pas en collision avec les obstacles de l'environnement.

Fig.I.2. Discretisation de l'espace avec une triangulation Delaunay filtrées.

Synthèse

Pour conclure concernant triangulation Delaunay, nous allons retenir deux points forts. Tout d'abord, ces représentations sont totalement indépendantes d'une quelconque précision désirée, reproduisant exactement l'environnement d'origine tout en l'organisant de manière plus accessible. Ainsi, il sera possible par ces représentations d'extraire la définition graphique source, pour son éventuelle utilisation dans des processus plus précis. Ensuite, ces représentations exposent plusieurs propriétés, comme le calcul des plus proches voisins, permettant par la suite d'automatiser un certain nombre de calculs à moindre coûts.

Néanmoins, un point faible est associé à ces techniques, résidant dans la profusion des cellules produites. En effet, pour des environnements à géométrie complexe, notamment contenant des courbes, le nombre de triangles obtenus augmente très rapidement. Même si l'effectif des triangles reste linéaire en fonction du nombre d'obstacles traités, cela réduira

tout de même les performances d'algorithmes se basant sur cette définition. Une solution proposée par F. Lamarche et al. [Lam03, LD04] consiste à abstraire la subdivision obtenue, par des regroupements successifs, permettant de hiérarchiser les processus et ainsi de réduire le nombre de cellules à manipuler à chaque étape. Il faut tout de même noter que pour rester avantageuse en terme de réalisme, une telle abstraction doit permettre de conserver un maximum d'informations. Dans le cas contraire, elle risque de fausser les décisions prises par des algorithmes l'exploitant directement (comme la planification de chemin).

1.2.1.2 Décomposition en trapèzes

Cette décomposition permet de décomposer l'environnement sous forme de cellules trapézoïdales [Lat91]. Elle est basée sur un algorithme de balayage [BY95]. Les points délimitant la géométrie de l'environnement sont triés suivant l'axe des ordonnées. Puis un maximum de deux segments est généré, ayant pour origine le point sélectionné et pour extrémité la prochaine intersection avec le bord d'un polygone délimitant un obstacle (Figure I.3).

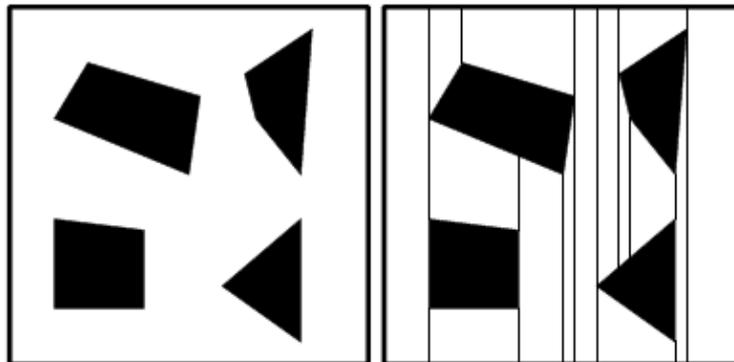


Fig.I.3. Discretisation en trapèzes.

L'algorithme de subdivision possède une complexité de $O(n \ln n)$, n étant le nombre de points utilisés pour décrire la géométrie des obstacles. Le nombre de cellules alors générées est en $O(n)$, tout comme le nombre de segments reliant ces cellules.

La principale qualité des méthodes de discrétisation exactes réside dans le nombre de cellules qu'elles génèrent. Ce nombre est linéaire en fonction du nombre de points utilisés pour décrire la forme des obstacles. Il ne dépend donc pas de la précision de représentation voulue, mais uniquement de la complexité intrinsèque de la géométrie représentant l'environnement. La deuxième propriété se situe dans l'adaptation automatique de la taille des

cellules à la densité des obstacles. Ces méthodes combinent donc les avantages des discrétisations hiérarchiques et par rectangle tout en conservant l'information géométrique exacte [Lam03].

1.2.2 Représentation approximative de l'environnement

Les représentations approximatives de l'environnement sont sans doute les plus utilisées en animation comportementale, du fait de leur facilité de mise en œuvre. Ces représentations vont décrire l'espace libre – de navigation – avec des formes géométriques simples telles que carré ou rectangle en **2D** ou sous la forme de cubes ou parallélépipèdes en **3D**. Deux modèles entrent dans cette catégorie : les modèles à base de grilles, et les cartes de cheminement.

1.2.2.1 Modèles à base de grilles

1.2.2.1.1 Grilles uniformes

La méthode de décomposition en grille uniforme se base sur la notion de cellules carrées en deux dimensions et cubiques en trois dimensions. Le principe est donc de quadriller l'espace, de manière uniforme, avec ces cellules. Les cellules sont alors vides si elles appartiennent à la zone de navigation, partiellement obstruées si elles contiennent une partie d'un obstacle et obstruées si elles sont incluses à l'intérieur d'un obstacle (Figure I.4).

La précision de ce mode de représentation de l'environnement dépend de la taille de la cellule de base. Plus les cellules sont grandes moins la représentation est précise et inversement. Supposons que l'environnement soit un carré de n cellules de côté, la taille mémoire nécessaire pour le stockage de la grille est en $O(n^2)$ plus généralement, en dimension d , le nombre de cellules est n^d . La complexité mémoire constitue l'un des problèmes majeurs de cette méthode pour laquelle il faut trouver un compromis précision/taille mémoire dépendant de chaque environnement à représenter. Outre son impact sur le coût mémoire, l'augmentation de précision grossit d'autant la taille du graphe topologique et possède donc un impact non négligeable sur le coût de la recherche d'un chemin [TB96].

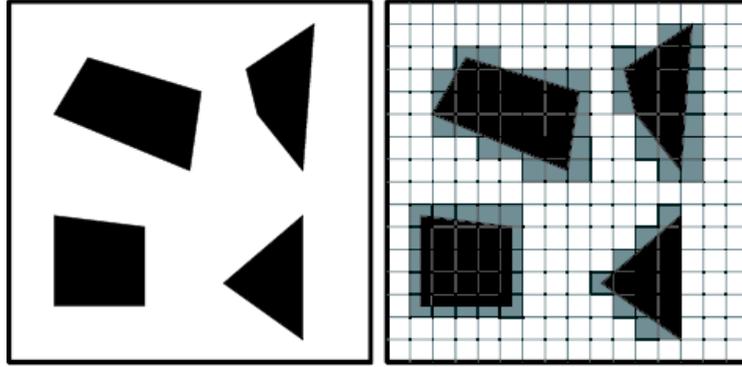


Fig.I.4. Discretisation de l'espace avec une grille régulière.

Dans le cas d'une représentation $2d/2d^{1/2}$ les possibilités des cartes graphiques actuelles peuvent être utilisées pour calculer cette carte en une passe de rendu, en utilisant peu de temps processeur [Kuf98]. La méthode consiste à effectuer un rendu de la scène en vue de dessus avec une projection orthogonale. L'image rendue, couplée à l'information de *Z-Buffer* permet alors d'obtenir une représentation de l'environnement sous la forme d'une grille régulière en $2d^{1/2}$ en identifiant rapidement les zones de navigation et les obstacles. Son utilisation en $2d/2d^{1/2}$ dans le cadre de l'animation comportementale est assez répandue [Kuf98, TC00] dans la mesure où elle est simple d'utilisation et très rapide en terme d'accès. En trois dimensions, elle permet à des humanoïdes de synthèse de naviguer dans des environnements assez complexes [BTh98] en prenant automatiquement en compte les informations de hauteur dans la planification. Elle simplifie le problème de la gestion des pieds d'appui car la sélection des cellules représentant le sol lors de la planification fournit toute l'information nécessaire.

1.2.2.1.2 Grilles hiérarchiques

Le principe des grilles hiérarchiques consiste à décrire l'espace sous forme d'un arbre de cellules qui subdivisent récursivement l'espace. Dans le cas en deux dimensions, le *quad-tree* (Figure I.5) subdivise récursivement l'espace en décrivant une cellule carrée à un niveau comme la réunion de quatre cellules carrées disjointes recouvrant tout l'espace de la cellule mère. Dans le cas en trois dimensions, l'*octree* fait de même avec une forme de base cubique subdivisée en huit sous cubes. Lors de la phase d'initialisation de cette structure, une subdivision s'arrête à un niveau donné de l'arbre si la profondeur de la cellule est égale à la profondeur maximale de l'arbre (directement corrélée à la taille maximale de la cellule la plus précise) ou si toutes les sous cellules sont libres pour la navigation ou toutes à l'intérieur d'un obstacle. En terme d'occupation mémoire, au pire cas il est du même ordre de grandeur que

pour les grilles régulières. Dans les faits, elle s'avère souvent moins coûteuse pour les environnements peu denses en obstacles, ce qui explique, en partie son utilisation en robotique [YSS+98].

Le graphe topologique associé à ce type de représentation est du même type que pour la grille régulière mais est uniquement constitué des feuilles de l'arbre (*quad-tree/octree*). Au pire, la taille du graphe est donc égale à celle du graphe associé à la grille régulière. Cependant, dans un très grand nombre de cas, la précision de la grille régulière est augmentée pour obtenir une meilleure approximation de la géométrie des obstacles. Cette augmentation du nombre de cellules est uniforme sur toute la carte, la même augmentation de précision dans une grille hiérarchique, d'après le mode de construction n'augmente la précision que sur le bord des obstacles en conservant des cellules les plus grandes possibles pour des zones homogènes. L'utilisation de cette structure permet donc de réduire considérablement, par rapport à la grille régulière, le nombre de cellules et donc la taille du graphe topologique associé [Lam03].

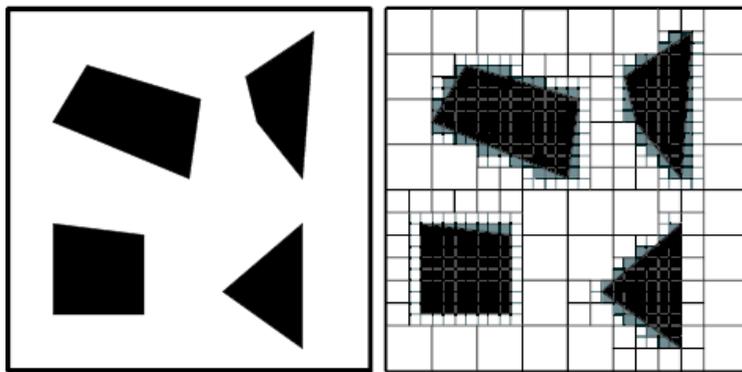


Fig.I.5. Discretisation de l'espace avec une grille hiérarchique type quadtree.

1.2.2.1.3 Décomposition par rectangles

Cette décomposition représente l'espace sous la forme d'une collection de rectangles, alignés sur les axes, de taille différente [Lat91]. Ces rectangles recouvrent tout l'espace et ne partagent que des frontières, sans se recouvrir. Au même titre que dans les méthodes précédentes, ces rectangles (ou cellules) représentent une zone de navigation libre, une zone contenue à l'intérieur d'un obstacle ou bien une zone comprenant à la fois une zone de navigation et une partie d'obstacle. La figure I.6 montre cette décomposition sur l'environnement déjà utilisé précédemment, pour la comparaison, cette subdivision a été générée avec la même précision que pour le quadtree de la figure I.5.

Par rapport aux méthodes précédentes, ce type de décomposition apporte l'avantage de permettre de créer des cellules de taille variable en fonction de la complexité géométrique de l'environnement. La grille régulière force une discrétisation en cellules de taille fixe, la décomposition hiérarchique permet d'utiliser un facteur d'échelle (fois deux sur chaque dimension à chaque niveau de l'arbre) sur la taille de la cellule alors que la décomposition en rectangle permet de régler la taille en x et en y de chacune des cellules. Elle permet donc d'essayer de maximiser la taille des cellules représentant un espace homogène, permettant ainsi de diminuer la taille de la structure de données utilisée pour représenter l'espace [Lam03].

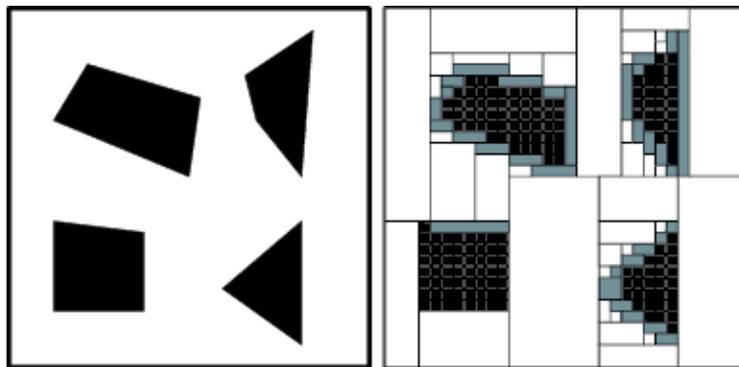


Fig.I.6. Discretisation par rectangle.

1.2.2.2 Cartes de cheminement

Les méthodes à base de cartes de cheminement discrétisent l'espace sous la forme d'un réseau de chemins permettant aux entités de naviguer en évitant les obstacles. Pour ce faire, en certain nombre de points clefs sont répartis à l'intérieur de l'environnement et connectés s'il existe un chemin en ligne droite les reliant sans rencontrer d'obstacle. Différents méthodes, basées sur ce concept existent. Elles diffèrent principalement par le mode de génération des points clefs et par la façon de les relier [Lam03].

1.2.2.2.1 Graphe de visibilité

Le graphe de visibilité utilise les sommets des polygones décrivant les obstacles de l'environnement, comme points clefs de la carte de cheminement. Par la suite, deux points clefs sont reliés si et seulement s'ils sont mutuellement visibles. Autrement dit si et seulement si il existe un chemin les reliant en ligne droite et exempt d'obstacle (Figure I.7).

Le réseau de chemins ainsi créé possède la propriété de maximiser la longueur des parcours en ligne droite, permettant ainsi de minimiser la distance parcourue [ACF01].

Cependant, la taille du graphe généré est directement dépendante du nombre de paires de points mutuellement visibles. Considérons que dans l'environnement, n points soient mutuellement visibles. Chacun de ces points est relié aux $n-1$ autres points. En considérant que le graphe est non orienté, ces n points gèrent donc $n(n-1)/2$ liens de visibilité. Un environnement ouvert, de grande taille, avec des obstacles ponctuels disparates, possédant donc un grand nombre de relations de visibilité, constitue l'un des pires cas pour la construction de ce type de graphe de cheminement.

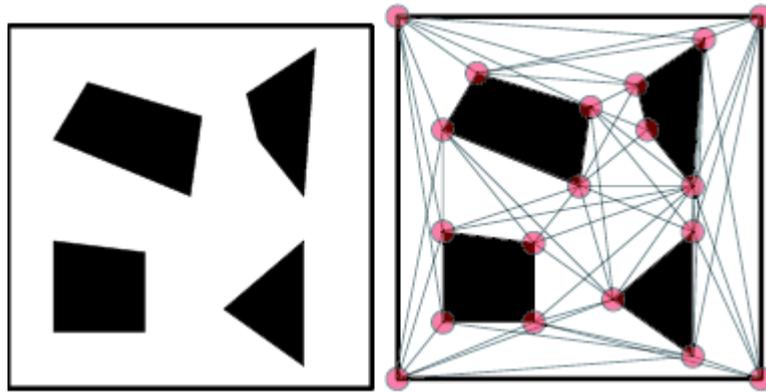


Fig.I.7. Graphe de visibilité calculé sur l'environnement.

1.2.2.2 Diagramme de Voronoï généralisé

Cette méthode est basée sur une notion d'équidistance aux obstacles de l'environnement. Pour se faire, un ensemble de sites sont évalués au sein de l'environnement, correspondant aux obstacles, dont les intersections vont former les points clefs. Les chemins ainsi générés maximisent la distance aux obstacles. Ce calcul s'avère complexe, mais son approximation peut être obtenue rapidement par des méthodes utilisant les cartes graphiques [HKL+99], où le calcul est obtenu directement en effectuant le rendu des sites. Une autre méthode utilise la triangulation de Delaunay pour obtenir les points clefs du diagramme de Voronoï généralisé, en se servant du centre des triangles calculés. La carte de cheminement ainsi produite peut être considérée comme un condensé des informations de la triangulation, ne contenant plus la définition géométrique des obstacles de l'environnement.

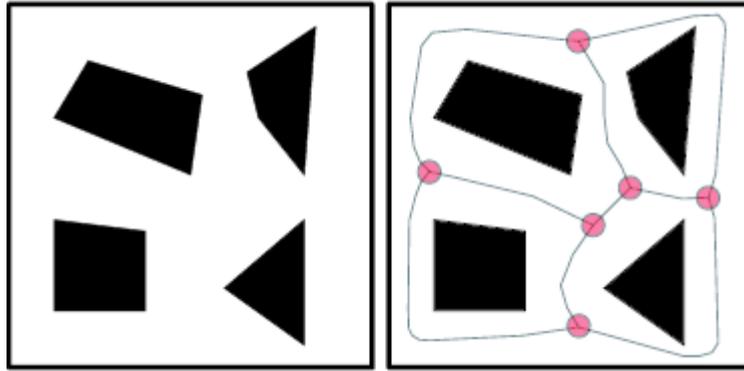


Fig.I.8. Graphe Voronoï généralisé.

1.2.2.2.3 Cartes de cheminement probabilistes

Plutôt que de se baser directement sur les informations géométriques pour construire une carte de cheminement et identifier les points clefs, les cartes de cheminement probabilistes [Ove02] s'appuient sur une génération aléatoire de points clefs pour couvrir l'espace libre. Deux points peuvent ensuite être reliés s'il existe, entre eux, un chemin libre de collision. Ce test constitue l'opération la plus coûteuse. Il exploite la géométrie de l'environnement et de l'entité en déplacement pour vérifier que le chemin généré ne provoque pas de collision.

Cette méthode est très utilisée dans le cadre de la planification de chemin pour des objets polyarticulés tels que des bras [Kuf99] par exemple. Dans le cadre de la locomotion cette technique a permis la génération de mouvements très complexes [CL03, PL03] adaptés à la difficulté de l'environnement, tels que des sauts, marcher sur des piliers, se baisser pour éviter une branche... .

Pour conclure, les cartes de cheminement présentent le net avantage de fournir une description très condensée de l'environnement, ne nécessitant une prise de décision qu'au niveau des points clefs. Néanmoins, leur définition seule n'est pas suffisante pour gérer la complexité de la locomotion humaine. Par exemple, leur exploitation devient très difficile lorsqu'il s'agit de gérer le croisement des entités le long d'un même chemin. Certaines méthodes [PCM+06] proposent de régler ce problème en subdivisant chaque chemin dans sa largeur, les gérant ainsi comme un ensemble de rails parallèles entre lesquels vont pouvoir transiter les entités en mouvement. Mais, même si une telle méthode produit des animations visuellement plausibles, son fonctionnement est trop éloigné du comportement humain pour être exploité dans des simulations réalistes. Les représentations sous forme de cartes de cheminement semblent donc bien adaptées à un processus de planification de chemin général,

ne tenant pas compte des autres entités, mais beaucoup moins à un processus de navigation, gérant lui les mouvements locaux.

1.3 Recherche de chemin

Le déplacement des humanoïdes à l'intérieur d'un environnement virtuel est, le plus souvent, lié à la volonté d'atteindre une position particulière. Il s'agit du problème traité par la recherche de chemin : comment trouver un chemin depuis une position donnée allant vers une position voulue, en évitant les obstacles statiques de l'environnement. Pour générer un chemin plausible, l'approche la plus communément adoptée est de rechercher un chemin minimisant certains critères comme la distance, un coût énergétique... Même si cette recherche d'un chemin optimal n'est pas forcément corrélée avec la navigation humaine [WM03], elle permet de générer des chemins plausibles et évitant des détours excessifs qui s'avèrent beaucoup moins réalistes. Deux grands types d'approches peuvent être distingués : les approches à base de graphe et les approches à base de champs de potentiel.

1.3.1 Calcul sur les graphes

L'utilisation de l'algorithmique des graphes pour le calcul d'un chemin nécessite l'utilisation d'une discrétisation de l'espace car un nœud du graphe est assimilé à un point de l'environnement. Les arcs représentent alors une notion d'accessibilité entre deux points et sont values par une estimation de l'effort à fournir (il s'agit le plus souvent de la distance) pour relier ces points. La recherche d'un chemin va donc se résumer à la recherche d'une suite de nœuds, reliés par des arcs dont la somme des valeurs associées minimise le coût global du chemin.

1.3.1.1 Discrétisation de l'espace et construction du graphe

Avant de pouvoir appliquer les algorithmes de calcul sur les graphes, il faut disposer d'une représentation de l'environnement sous cette forme. Les cartes de cheminement exposées précédemment fournissent directement cette information : chaque point clef de l'environnement devient un nœud alors que chaque arc représente la relation d'accessibilité entre ces nœuds. Dans le cas d'une discrétisation de l'espace sous la forme de cellules, les informations de cellules et de connexité sont utilisées pour générer une carte de cheminement. Traditionnellement, les points clefs sont choisis soit comme étant le centre de la cellule, soit comme appartenant aux bords franchissables de cette même cellule. Les arcs

du graphe sont alors construits en fonction des relations de connexité entre les cellules. La figure I.9 montre les deux formes de graphes de cheminement qui peuvent être construites en utilisant comme base une triangulation de Delaunay contrainte ; la figure de gauche utilise les centres de gravité des triangles comme points clefs ; la figure de droite utilise le milieu des segments non contraints reliant deux triangles [Lam03].

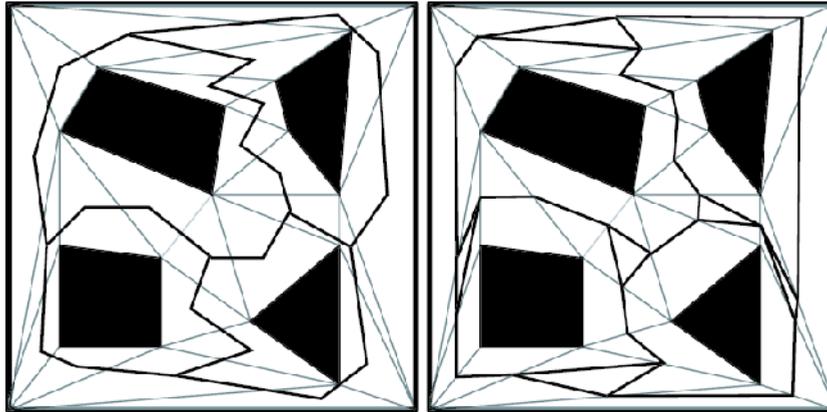


Fig.I.9. Exemple de cartes de cheminement générées à partir d'une discrétisation par triangulation de Delaunay contrainte.

Une autre approche, utilisée dans le cadre d'environnement intérieur, consiste à travailler sur le graphe d'adjacence des cellules (extraites de la subdivision spatiale) pour caractériser des espaces (pièces, couloirs, portes) [Lau83, Lau89]. Chaque espace étant un agrégat de cellules, le graphe topologique des espaces est plus abstrait et donc moins complexe que le précédent. Cela permet de définir une stratégie de navigation hiérarchique s'appuyant sur le graphe topologique et raffinant les calculs, à posteriori, à l'intérieur de chaque espace.

1.3.1.2 Les algorithmes de parcours

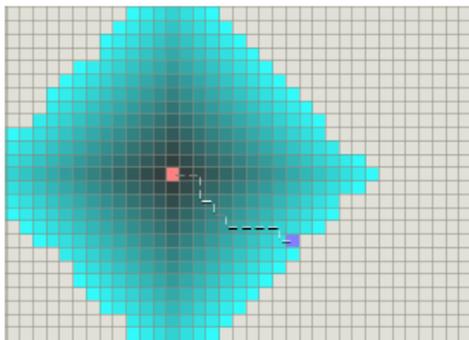
Différents algorithmes de parcours de graphe peuvent être utilisés dans le cadre de la recherche de chemin. Ceux que nous décrivons ici se basent sur un critère de minimisation de coût pour extraire le plus court chemin. Ce coût représente souvent une notion de distance spatiale, mais nous verrons dans la section suivante que des heuristiques plus complexes peuvent être utilisées.

1.3.1.3 Dijkstra

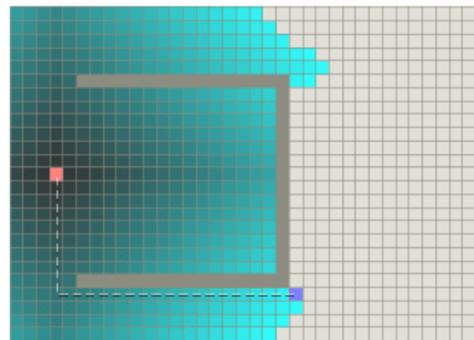
L'algorithme de **Dijkstra**, aussi dénommé *flood-fill* [LRD+90], permet de trouver l'ensemble des meilleurs chemins entre deux nœuds du graphe. Cet algorithme utilise une file de priorité où il stocke pour chaque nœud exploré deux informations :

- la distance parcourue depuis le nœud de départ jusqu'au nœud courant ;
- le nœud précédent, i.e. parcouru avant le nœud courant.

L'algorithme commence avec un nœud source correspondant généralement à la position courante de l'entité, qui n'a donc aucun prédécesseur et un compteur de distance nul. Cet algorithme fonctionne directement sur le graphe topologique, qu'il soit explicite (par exemple avec les cartes de cheminement) ou implicite (par exemple avec les grilles). Afin de trouver un chemin entre deux nœuds, en admettant que la longueur de ce chemin soit L , l'algorithme explorera par propagation circulaire l'ensemble des nœuds se trouvant à une distance inférieure à L (Figure I.10).



(a) Environnement non contraint



(b) Environnement avec obstacle concave

Fig.I.10. Fonctionnement de l'algorithme de Dijkstra.

Les intérêts majeurs de cet algorithme sont sa robustesse (si un chemin existe il sera forcément trouvé) et son abstraction totale de la destination, si ce n'est comme condition de fin. Son point faible est sa complexité calculatoire, étant de l'ordre de $O(A + N \cdot \log N)$ où A est le nombre d'arcs du graphe, et N son nombre de nœuds. Malgré tout, ce point faible peut aussi être un avantage, l'algorithme supportant des recherches guidées par un but plus conceptuel, associé à la sémantique des nœuds parcourus.

1.3.1.4 A étoile(A*)

Dans les cas où la performance de la recherche est primordiale, comme en animation comportementale (dans le cas des jeux vidéos par exemple), l'algorithme A^* [Nil82] est plus

utilisé. Celui-ci fonctionne d'une façon similaire à l'algorithme de **Dijkstra**, mais en ajoutant un coût prédictif aux nœuds correspondant au reste du chemin à parcourir. Le coût total est donc la longueur réelle du chemin jusqu'au nœud, incrémentée par une valeur prédite pour le chemin menant au but, calculée par une heuristique. L'algorithme va ainsi parcourir les nœuds dans l'ordre croissant de leurs coûts associés (Figure I.11). La rapidité de convergence de cet algorithme est directement dépendante de la qualité de l'heuristique employée. Dans le cas de la recherche de plus court chemin, l'heuristique employée est une estimation de distance, généralement la norme géométrique séparant le nœud courant du but.

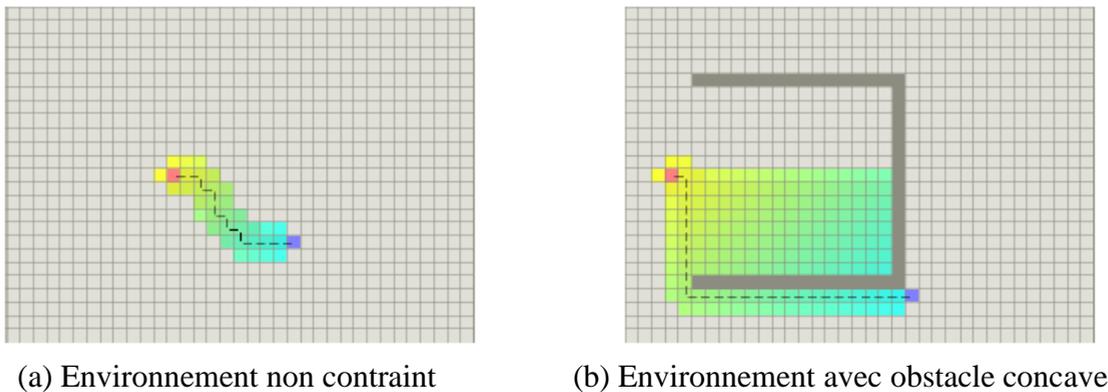


Fig.I.11. Fonctionnement de l'algorithme de A^* .

L'avantage de cet algorithme réside dans sa rapidité calculatoire. Son inconvénient majeur réside dans le recours à une heuristique. En effet, plus l'évaluation de la longueur de chemin sera complexe, pouvant faire intervenir d'autres paramètres que la distance (par exemple un coût associé à la sémantique), plus cette heuristique sera difficile à expliciter. Ainsi, il est difficile d'utiliser cet algorithme pour des recherches de chemin dont le but n'est pas clairement identifié dans le graphe topologique, comme pour des recherches exploratoires où à buts multiples.

1.3.1.5 Évolutions du A^*

Un certain nombre d'évolutions ont été proposées pour améliorer les propriétés du A^* . B. Logan et N. Alechina [LA98] proposent l'algorithme *ABC* (pour A^* with Bounded Cost), permettant d'ajouter des contraintes molles à respecter lors de la recherche, comme des limitations de temps ou d'énergie.

L'algorithme **IDA*** [Kor85] commence par effectuer une recherche en profondeur dans le graphe. Tout d'abord, une distance maximale de parcours D est estimée comme étant la distance au but. L'exploration commence alors avec le nœud d'origine, puis évalue récursivement les voisins jusqu'à ce que l'une des deux conditions suivantes soit vérifiée : soit un chemin est trouvé, il est alors minimal, soit un nœud est parcouru dont le coût est supérieur à D . Dans ce dernier cas, la recherche continue avec une borne D remise à jour par la plus petite estimation de la distance au but évaluée précédemment. Dans les faits, cet algorithme est plus lent que le **A*** traditionnel, mais nécessite moins de mémoire.

L'algorithme **HPA*** [BMS04] (pour Hierarchical Pathfinding **A***) consiste à redécouper–abstraire– le graphe de l'environnement afin de hiérarchiser la recherche. Ensuite, l'algorithme propose d'associer des pré-calculs de plus courts chemins entre des points clefs de chaque partie abstraite. Ainsi, cet algorithme va pouvoir évaluer un chemin de haut niveau en ne manipulant que peu de nœuds.

1.3.2 Critères de recherche de chemin

Différents critères peuvent être évalués pour le coût des algorithmes précédents. Les plus anciennes méthodes, ainsi que la plupart de celles utilisées en animation, se basent uniquement sur la distance séparant les nœuds. Ainsi, dans des simulations se voulant plus réalistes, d'autres critères d'évaluation sont proposés.

H.H. Hochmair et V. Karlsson [HHK05] introduisent la notion de déviation minimale, en tenant compte des changements de direction dans le coût associé à un chemin. Ils prennent aussi en compte des facteurs qualitatifs via des préférences, ainsi qu'une connaissance relative à l'environnement. M. Duckham et L. Kulik [DK03] prennent en compte le concept de charge cognitive associée à la recherche d'un chemin, avec une stratégie de chemin le plus simple. Un autre paramètre, fondamentalement lié aux foules d'individus, est la congestion d'une zone de l'environnement [ST05]. Un facteur de stress [Osa04] peut aussi être appliqué, lié au nombre de personnes dans l'environnement immédiat ainsi qu'à la déviation par rapport au chemin le plus court.

On peut ainsi voir que de nombreuses méthodes sont proposées afin de quantifier et qualifier un chemin dans l'environnement. Malgré tout, peu de ces méthodes associent plusieurs critères à la fois. Même s'il semble que la fusion de ces informations soit nécessaire pour se rapprocher d'une procédure réaliste, le problème de trouver une base de comparaison entre des données de nature différentes n'a pas encore, aujourd'hui, été résolu. Le mélange

des critères paraît donc essentiel pour considérer l'application des méthodes de recherche de chemin à la simulation d'humains virtuels.

1.3.3 Méthodes à champs de potentiel

Le modèle à base de champs de potentiels consiste en une définition de l'environnement permettant directement de résoudre les déplacements de personnes. Les champs de potentiels caractérisent les obstacles de l'environnement par des forces de répulsion, et les buts par des forces d'attraction [RKB+94, TCP06]. Un gradient de forces, ou champ de potentiel, est alors déduit en chaque point de l'environnement comme étant une somme pondérée, le plus souvent par la distance, des potentiels de répulsion et du potentiel lié au but (Figure I.12). La navigation d'une entité est alors simulée par une descente de gradient depuis sa position dans l'environnement.

Les méthodes basées sur les champs de potentiels s'avèrent simples et efficaces, mais posent le problème des minima locaux : zones de l'environnement où un potentiel minimal isolé apparaît (Figure I.12(b)). Ainsi, la méthode de navigation associée va pousser l'entité à se déplacer vers le minimum local le plus proche, qui ne représente pas forcément son but. Pour pallier ce type de problème, des méthodes à base de marche aléatoire sont utilisées. Par exemple, dans *RPP* (RandomPathPlanner) [BL91], lorsqu'un minimum local est atteint, un ensemble de configurations aléatoires sont tirées puis testées avec une phase de sortie du minimum et une phase de convergence vers le prochain minimum. Les informations sont alors stockées dans un graphe dont les nœuds sont les minima locaux et les arcs traduisent des chemins entre deux minima. D'autres méthodes existent à base de tirage aléatoire de direction à suivre [CRR01], plutôt que de configuration, pour échapper au minimum local. Ces méthodes ne sont cependant pas très adaptées à l'animation comportementale. Les comportements induits par les tirages aléatoires, s'ils sont acceptables pour des robots, ne le sont pas pour des humanoïdes car ils sont en dehors de la logique de navigation humaine qui comporte une part importante de planification.

De manière générale, les méthodes à base de champs de potentiels ne sont pas applicables directement à la simulation d'humains, du fait du manque de souplesse dans la méthode de contrôle. On peut néanmoins trouver quelques cas d'application à grande échelle [GSN04], où le comportement individuel est noyé dans la caractérisation globale des mouvements de milliers d'individus. Notons tout de même que cette technique a deux grands avantages qu'il serait bon de conserver dans une évolution permettant une décision

individuelle plus importante. Premièrement, cette méthode permet la fusion de l'information au sein de l'environnement, traduisant tout ce qui intervient dans la navigation par des potentiels qui sont mélangés. Deuxièmement, et c'est une conséquence directe du premier point, ces méthodes introduisent une abstraction très forte de l'environnement, grâce à laquelle l'individu ne raisonne plus sur des entités perçues indépendantes (que ce soit la topologie des lieux ou les autres individus), mais directement sur une abstraction spécialisée pour sa tâche de déplacement.

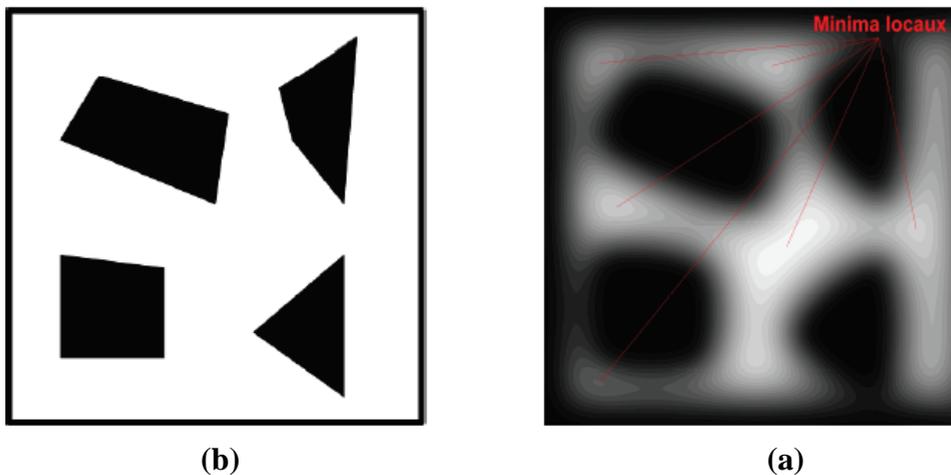


Fig.I.12. Une carte de champ de potentiel.

1.4 Environnement virtuel informé

Un environnement virtuel informé est un environnement virtuel dont les modèles 3D contiennent non seulement la géométrie de la scène mais aussi toutes les informations pertinentes pour les entités comportementales à simuler comme les éléments symboliques ou sémantiques qui peuvent permettre à ces entités de percevoir, décider et agir [Don4]. Thouvenin propose d'élargir cette définition en y incluant la notion de modèles de représentation des connaissances : un environnement virtuel informé est un environnement virtuel doté de modèles à base de connaissance dans lequel il est possible à la fois d'interagir et de permettre des comportements par interprétation de représentations dynamiques ou statiques [Mou09].

Les environnements virtuel informé intègrent sur les objets des connaissances sur le savoir (description de l'objet) et le savoir-faire (interactions possibles avec l'objet) et ce de plusieurs manières [Sep07].

1.4.1 Smart Objects

Les Smart Objects (les objets intelligents), introduits par M. Kallmann et D. Thalmann [KAL98], sont basés sur la théorie écologique de la perception de J.J. Gibson. En effet, un Smart Object contient l'ensemble des informations nécessaires à une interaction. Ces informations sont ensuite transmises à l'individu lorsqu'il effectue l'interaction.

Quatre types de données constituent un Smart Object [KAL02]:

1. Les propriétés intrinsèques de l'objet, i.e. physiques, comme la taille, la forme, ou le centre de masse de l'objet.
2. Les informations concernant l'interaction, servant à l'acteur pour accomplir cette interaction, comme l'endroit où se positionner, ou encore l'identification des éléments nécessaires (comme une poignée ou un bouton).
3. Les comportements internes à l'objet, pouvant être déclenchés par une interaction, et pouvant être conditionnés par des variables internes à l'objet, comme la fonction d'impression d'une imprimante.
4. Les comportements attendus de l'acteur lors de chaque interaction avec l'objet.

Les comportements de plus haut niveau sont ensuite décrits grâce à des automates à états finis. Les états sont ici constitués par les actions de base pouvant être accomplies par ou avec l'objet. Une animation est alors pourvue, en se basant sur un ensemble de gestes de base. Ces gestes sont adaptés en temps réel par un processus de cinématique inverse et de la planification de mouvement [KAL03], afin de prendre en compte la position relative de l'agent vis à vis de l'objet (Figure I.13).

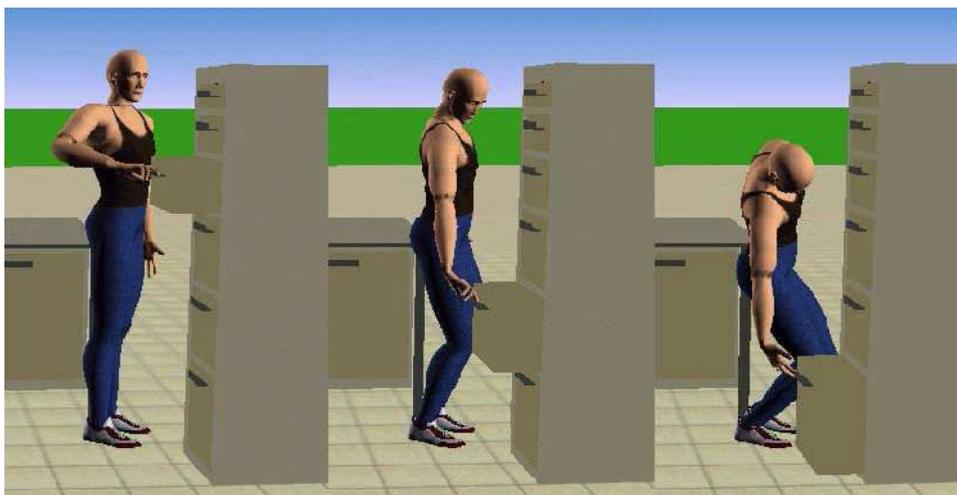


Fig.I.13. Exemple de Smart Object : interaction avec un casier.

Abaci et al [ACT06] étendent les *Smart Objects* en rajoutant les informations de pré-conditions et les effets des actions. Ce rajout permet de fournir à un planificateur les actions réalisables sur les objets et donc de créer des comportements complexes dirigés par des buts.

1.4.2 Annotations

Doyle [Doy02] propose d'« annoter » les objets avec les connaissances spécifiques liées à l'environnement et de ne laisser dans la personne que les compétences abstraites. Il présente un *framework* où chaque information provenant de l'environnement est annotée.

Le système d'annotation permet aux personnages d'évoluer dans des environnements non familiers en étant crédibles et compétents. Il permet aussi de faciliter la mise en place de plusieurs personnages avec des ressources limitées.

Aubry [Aub07] utilise lui aussi la notion d'annotation d'environnements virtuels, afin d'avoir à la fois un environnement réaliste et des données abstraites liées à celui-ci. Contrairement aux travaux de Doyle, les annotations servent ici de moyen de communication avec l'utilisateur, dans le cadre, par exemple, de travail collaboratif.

1.4.3 Ontologies

D'autres travaux ont été réalisés pour exprimer le contenu d'un environnement, qu'il soit visuel ou non avec un seul langage de description.

Otto [Ott05] parle d'environnements virtuels sémantiques où des techniques du Web sémantique sont utilisées, le but étant de trouver une base indépendante du logiciel de description d'environnements virtuels.

Gutiérrez [Gut05] propose lui aussi une représentation sémantique de la géométrie de l'objet, ainsi que de ses fonctionnalités intrinsèques et des différentes interactions qu'il offre.

Les différentes techniques que nous venons de présenter apportent des informations supplémentaires à l'environnement. Nous voulons des personnages crédibles dans un environnement virtuel en trois dimensions. L'ajout d'informations non seulement sémantiques mais aussi spatiales pour améliorer les interactions des personnages avec les objets est particulièrement utile pour modéliser des environnements différents et y faire évoluer des personnages autonomes.

1.5 Intégration des objets interactifs

Les objets définis dans l'environnement est géo-localisé, i.e. il connaît ses positions dans l'espace, mais aussi le nœud du graphe topologique qui lui correspond. Le but premier de cette géo-localisation est de rendre les objets accessibles aux algorithmes basés sur notre définition topologique, dont notamment la planification de chemin et l'évitement de collision. Il y a deux catégories d'objets interactifs sont ici à prendre en compte : premièrement, les objets statiques dont la localisation spatiale est figée au cours d'une simulation, deuxièmement, les objets dynamiques pouvant se déplacer au cours d'une simulation [Seb07].

1.5.1 Les objets statiques

Les objets statiques sont fixes dans l'environnement. Ces objets sont dans la plupart des cas des spécialisations de l'effecteur, représentant des équipements accessibles aux inter-acteurs. Trois sous catégories des objets statiques que sont :

- Les objets obstacles : les objets obstacles statique doivent être évités par des entités se déplaçant dans l'environnement. Par exemple ; un guichet de vente, un mur... .
- Les objets traversables : les objets traversables joignent plusieurs zones navigables de l'environnement, tout en étant eux-mêmes navigables lors de leur utilisation, ou obstacles dans le cas contraire. Par exemple ; un ascenseur, un escalier, un portique automatique.
- Les objets libres : les objets libres sont disposés de telle façon qu'ils ne gênent en rien la navigation, mais seraient considérés comme des obstacles s'ils étaient situés ailleurs. Par exemple ; un panneau d'affichage accroché en hauteur sur un mure ou sur un pilier.

1.5.2 Les objets dynamiques

Les objets dynamiques sont mobiles dans l'environnement. Ces objets peuvent indifféremment être des spécialisations de l'inter-acteur, comme l'humain virtuel, ou de l'effecteur, comme une valise, ou encore l'unification des deux, comme un agent de service dans un lieu public.

1.6 Conclusion

Nous nous sommes penchés tout au long de ce chapitre sur la représentation de l'environnement virtuel. Nous avons ainsi exposé les techniques permettant la représentation

de l'environnement, et son exploitation lors de la recherche de chemin. Ensuite, nous avons présenté les environnements informés. Enfin nous avons exposé l'intégration des objets interactifs.

Notamment concernant les représentations de l'environnement, on a pu voir que les approches approximatives offrent un accès rapide à la description topologique, alors que les approches exactes permettent la conservation maximale de l'information géométrique. Ces différentes techniques se basent, la plupart du temps, sur une projection en 2D de l'environnement pour rendre les calculs performants, mais cela introduit aussi une perte d'informations. Il est indéniable que dans le cadre de simulations devant produire des résultats comparables au réel, le réalisme prend la place la plus importante.

L'autre point concerne les paramètres devant être pris en compte pour évaluer un chemin. Certains de ces paramètres sont mis en avant par différentes études, mais aucune comparaison de leur importance relative n'est disponible.

Chapitre II

Navigation d'humains virtuels

2 Navigation d'humains virtuels

2.1 Introduction

L'animation est la deuxième étape importante de la synthèse d'images, on s'intéresse à un système continu qui évolue au cours du temps. Une animation est une séquence d'images, où chaque image est une observation du système à un instant donné [HA98]. Les images d'une animation sur l'écran d'un ordinateur constituent une discrétisation à la fois temporelle (fréquence d'affichage) et spatiale (résolution de l'écran) des états du système. On considère donc une animation comme étant une suite d'états du système pris à des intervalles temporels connus.

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

Dans ce chapitre, on s'intéresse à l'animation comportementale. Nous présentons les propriétés des humains virtuels et les comportements des individus, ensuite nous présentons le groupe des individus et son comportement, et la simulation de foule, et les relations entre les individus. Dans la dernière partie, nous présentons les processus mentaux de navigation.

2.2 Les individus

2.2.1 Modélisation des propriétés des humains virtuels

L'objectif majeur de la modélisation des comportements des acteurs est de construire des acteurs virtuels plus réalistes, ces acteurs sont intelligents et autonomes et avec une adaptation, perception et mémoire, capable d'agir librement et avec émotion, d'être conscient et imprévisible (Figure II.1).

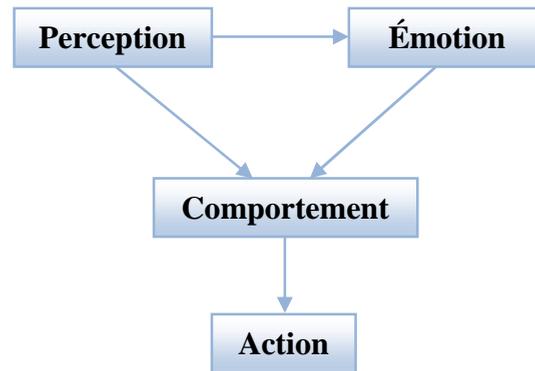


Fig.II.1. Structure du Modèle comportemental.

2.2.1.1 La perception

La perception est définie comme la conscience des éléments dans l'environnement à travers des sensations physiques. Il est réalisé en équipant les agents avec des détecteurs visuels, tactiles et auditifs ainsi ils simulent le comportement quotidien humain (aspect visuel, mouvement, réaction, ...). Le sous-système perceptuel le plus important est le système visuel. Une approche basée sur la vision [RMT90] est idéale pour le modelage d'une animation comportementale. Elle offre une approche universelle pour le passage d'information de l'environnement à l'acteur dans le contexte de recherche de chemin.

A un niveau plus haut, nous pouvons décomposer la perception comme suggérée par [BT98]. La perception d'un acteur peut être limitée aux objets et à d'autres acteurs dans le voisinage.

Mais cela limite le nombre de comportements possibles, parce que seules la présence et les caractéristiques d'un objet ou d'un acteur sont impliquées dans la sélection d'un comportement. Les actions des autres acteurs ne sont pas prises en considération.

Le module de perception produit trois types de perception :

1. La perception de la présence d'objets et d'acteurs ;
2. La perception des actions d'acteurs ;
3. La perception d'acteurs exécutant des actions sur des objets.

2.2.1.2 L'émotion

L'émotion peut être définie comme l'aspect affectif de la conscience : Un état de sentiment, une réaction psychique et physique (comme la colère ou la crainte),

subjectivement expérimenté comme un sentiment fort et physiologiquement qui augmente les changements préparant le corps pour une action vigoureuse immédiate.

Les acteurs doivent être capables de répondre, avec émotion à leur situation et agissant physiquement à cela. Les émotions visibles fournissent des designers avec un moyen direct pour affecter à l'utilisateur un état émotionnel propre à lui. Les acteurs sont donc équipés d'un modèle informatique simple de comportement émotionnel, qui est lié au comportement comme les expressions de visage qui peuvent être employées pour influencer leurs actions.

Une émotion est la réaction d'une personne à une perception. Celle-ci est amenée à répondre par une expression du visage, un geste, ou à choisir un comportement spécifique. Une émotion arrive entre une perception et la réaction suivante. Deux personnes différentes peuvent avoir des réactions différentes à la même perception, selon la façon dont ils sont affectés par cette perception [OT90].

Les émotions sont causées par la réaction aux objets, les actions d'agents et les événements. Les émotions causées par des événements peuvent être classées selon trois types d'émotions :

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

Chaque classe est caractérisée par des conditions d'apparition pour chacune de ses émotions et variables affectant son intensité. Les émotions auxquelles est soumis un acteur sont dues à sa perception.

2.2.1.3 Le comportement

Le comportement est souvent défini comme la voie dans lequel des animaux et les humains agissent. Il est usuellement décrit en termes de langage naturel qui a la signification sociale, psychologique ou physiologique, mais qui n'est pas nécessairement facilement réductible au mouvement d'un ou deux muscles.

Le comportement est aussi la réponse d'un individu, groupe, ou espèce à son environnement. Le comportement ne réagit pas seulement à l'environnement, mais inclut aussi le flux d'information par lequel l'environnement agit sur la créature vivante aussi bien sur la façon dont la créature code et emploie cette information.

Le comportement peut être décrit d'une façon hiérarchique. Le modèle comportemental décompose un comportement en comportements plus simple qui peuvent être décomposés plus loin. Chaque niveau de cette décomposition hiérarchique contient un ou plusieurs comportements exécutés séquentiellement, ou bien concurremment. Un niveau de la hiérarchie contenant plusieurs comportements pour être exécuté séquentiellement, appelé comportement.

Chaque comportement d'un ordre de comportement est appelé, cellule comportementale. La cellule comportementale contient les comportements qui sont exécutés chacun concurremment, ou exclusivement quand les règles d'inhibition sont spécifiées. Les comportements contenus dans une cellule comportementale sont de nature composée ou élémentaire.

Un comportement permet une décomposition récursive dans la hiérarchie. Un comportement élémentaire est placé au fond de la décomposition hiérarchique et encapsule un comportement spécialisé qui contrôle directement une ou plusieurs actions. Un comportement est exécuté récursivement : en fait la hiérarchie du comportement permet d'exécuter les comportements élémentaires des entités à chaque niveau de la structure de la hiérarchie.

Pour contrôler le comportement global d'un acteur, on exploite une pile de comportements. Au début de l'animation, l'utilisateur pousse un ordre de comportements dans la pile de l'acteur. À la fin du comportement actuel le système d'animation passe le comportement suivant de la pile et l'exécute. Ce processus est répété jusqu'à ce que la pile de comportement de l'acteur se vide.

Avec Ce contrôle de comportement employant une pile, un acteur devient plus autonome et crée ses propres sous buts en exécutant le scénario original.

2.2.1.4 L'action

Basé sur l'information perceptuelle, le mécanisme comportemental d'un acteur détermine les actions à exécuter. Les actions peuvent avoir plusieurs degrés de complexité. Un acteur peut se développer dans son environnement ou bien agir réciproquement avec l'environnement ou encore communiquer avec d'autres acteurs.

Les actions sont exécutées en employant une architecture de mouvement commune. Le module d'action gère l'exécution des actions employées par un comportement en animant un

modèle d'homme générique basé sur une hiérarchie de nœud. Il permet l'exécution simultanée ou séquentielle d'actions en gérant des transitions lisses entre des actions finales et des actions amorçantes [BMT95]. Une boucle comportementale conduit l'animation, son rôle est de mettre à jour l'état du monde virtuel.

A chaque itération, le temps est incrémenté, le monde virtuel est mis à jour avec en particulier une mise à jour de l'état de chaque objet et acteur. Dans le cas d'un acteur, la perception est d'abord exécutée, après ses émotions sont produites avant que son comportement et ses actions ne soient exécutés.

2.2.1.5 La Mémoire

La mémoire est d'habitude définie comme le pouvoir ou le processus de reproduction ou de rappel de ce qui a été appris et conservé, particulièrement par les mécanismes associatifs. La mémoire est aussi le dépôt pour des informations apprises et à conserver, générées à partir de l'activité d'un organisme ou d'une expérience.

2.2.2 Le comportement d'individu

Bien que la capacité à se mouvoir soit un comportement primordial de l'être humain, ce qui le caractérise vraiment est sa capacité à raisonner. En effet, un individu est capable de symboliser mentalement les actions qu'il veut exécuter, pour ensuite les organiser. Il est ainsi capable d'élaborer un plan comportemental lui permettant d'ordonner ses actions, et de gérer leurs liens de dépendance. Ce raisonnement va ainsi avoir un impact direct sur son déplacement.

Les tâches mentales prépondérantes concernant l'action d'un individu sur son environnement sont sans doute les tâches d'interaction. Celles-ci définissent la manière dont l'individu va pouvoir affecter d'autres entités, et ainsi modifier l'état du monde qui l'entoure. Ces tâches d'interaction entrent elles même dans un processus mental complexe, pouvant être directement désirées par l'individu, ou symboliser une étape de son raisonnement [Seb07].

Nous entendons par comportements d'individus ceux qui permettent à l'individu de se déplacer dans l'environnement. Par mais ces comportements en suite les comportements suivants :

2.2.2.1 Comportement de recherche et fuite

La recherche et la fuite sont deux comportements très simples qui déplacent un individu vers ou loin d'une position de cible avec une vitesse constante.

La différence entre la recherche et la fuite est que la recherche est l'acte d'orienter le caractère vers une position indiquée dans l'espace global. Ce comportement ajuste le caractère de sorte que sa vitesse soit radicalement alignée vers la cible. D'une manière simple, la fuite peut être considérée comme étant l'inverse de la recherche. Elle agit en orientant le caractère de sorte que sa vitesse soit radicalement alignée loin de la cible. La vitesse désirée se dirige dans la direction opposée [Rey99].

2.2.2.2 Comportement d'arrivée

Le comportement d'arrivée est une extension du comportement de recherche. De même que pour le comportement de recherche, il est utilisé pour orienter l'individu vers une cible indiquée. La différence importante se résume dans la manière selon laquelle l'individu atteint la destination.

Le comportement de recherche fait que notre individu atteint la cible à pleine vitesse. Il se déplace en fait davantage dans la direction actuelle et va ainsi générer plutôt un mouvement de danse semblable au comportement d'une mite autour d'une source lumineuse. Cependant, le comportement d'arrivée doit être tel qu'il y ait un ralentissement commandé du véhicule conformément au cahier des charges édicté par l'utilisateur, le mouvement doit s'arrêter à la position désirée (Figure II.2). Ceci fait provoquer le ralentissement du véhicule selon sa distance actuelle à la destination et le résultat est un véhicule qui s'arrête au niveau de la cible indiquée [Rey99].

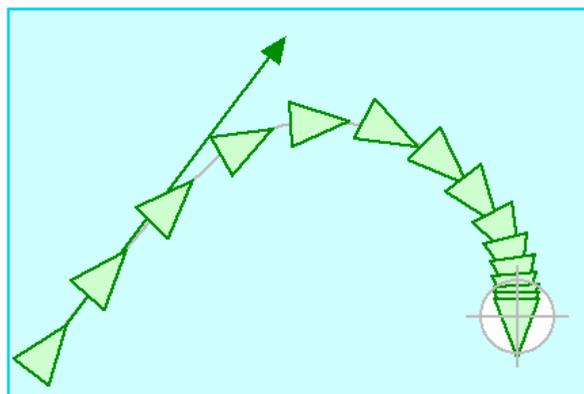


Fig.II.2. Le comportement d'arrivée.

2.2.2.3 Comportement d'évitement d'obstacles

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

La fuite a pour conséquence d'orienter l'acteur loin d'un emplacement donné, tandis que l'action d'évitement d'obstacles agit seulement quand un obstacle se trouve directement devant cet acteur. Par exemple, si un individu se déplace sur une trajectoire parallèle à un mur, l'évitement d'obstacles ne prendrait aucune mesure corrective de direction, mais la fuite essaierait de s'éloigner du mur, en suivant la perpendiculaire à ce mur.

Le but du comportement est de garder un cylindre imaginaire de l'espace libre devant l'acteur. Le cylindre se trouve le long de l'axe vers l'avant l'acteur, à un diamètre égal à la sphère de bondissement de l'acteur, et s'étend du centre de l'acteur pour une distance basée sur la vitesse et l'agilité de l'acteur.

Le comportement d'évitement d'obstacles considère chaque obstacle alternativement et détermine si ces obstacles possèdent des intersections avec le cylindre. En localisant le centre de chaque obstacle sphérique, le test d'intersection avec le cylindre est ainsi effectué très rapidement [Rey99].

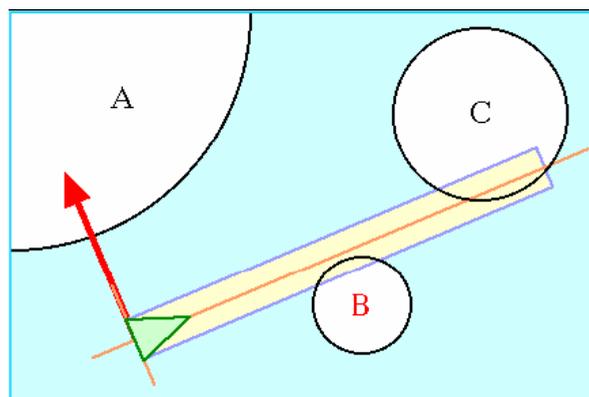


Fig.II.3. L'évitement d'obstacle.

Une note finale concernant l'interaction d'évitement d'obstacles et de recherche de but. D'une façon générale nous nous intéressons seulement aux obstacles qui sont localisés entre l'acteur et le but à atteindre.

2.2.2.4 Comportement de suivi de chemin

Ce type de comportement permet à un acteur de s'orienter le long d'une voie d'accès prédéterminée, telle qu'une chaussée, un couloir ou un tunnel. C'est en fait très distinct de l'action de contraindre un individu à suivre d'une manière rigide une voie d'accès tel que le roulement d'un train le long d'un rail, par exemple. Le comportement de suivi d'une voie d'accès est plutôt destiné à la production de mouvements tels que des personnes empruntant un couloir : les différentes voies d'accès restent proches et sont souvent parallèles à l'axe du couloir, elles sont néanmoins libres [Rey99].

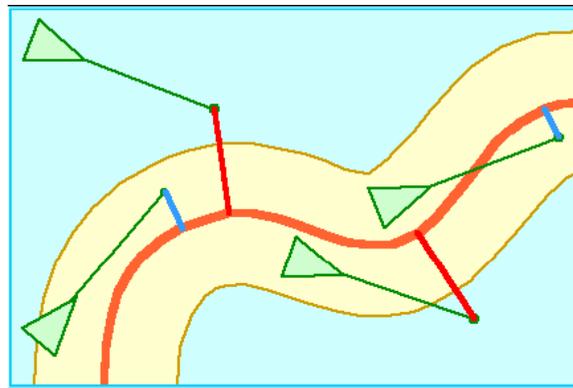


Fig.II.4. Le comportement de suivre de chemin.

Autrement dit, l'acteur vire loin de la voie d'accès, ou en est trop loin de celle-ci. Pour l'orienter en arrière vers la voie d'accès, le comportement de recherche est exploité pour une orientation en arrière vers la projection de la voie d'accès de la future position prévue, de la même manière que pour l'action d'évitement [Bee90].

2.2.2.5 Evitement de collision non-alignée

Le comportement d'évitement de collision non alignée se doit de prévenir les collisions entre des acteurs se déplaçant dans une direction arbitraire. Considérez votre propre expérience de marche à pied à travers une place ou une entrée pleine d'autres marchants : L'évitement de collisions implique la prévision de collisions potentielles et le changement de votre direction et de votre vitesse pour les prévenir [Rey99].

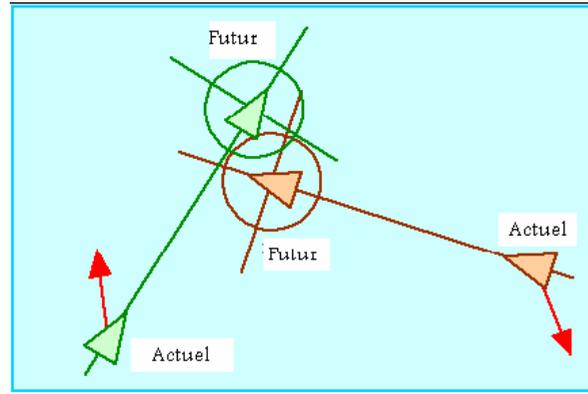


Fig.II.5. Le comportement d'évitement de collisions non-alignées.

2.3 Le groupe des individus

Un groupe peut être défini comme étant un nombre de personnes qui, pour une certaine période de temps, ont des interdépendances psychologiques ou de tâches, qui incluent l'interaction entre les personnes dans une situation. Le groupe formel existe par l'approbation d'une hiérarchie et a des fonctions spécifiques à remplir. Les groupes informels sont ceux qui se développent spontanément à cause de la proximité ou de la similarité du travail, de l'attraction mutuelle, du besoin mutuel d'accomplissement ou d'une combinaison de ces raisons.

La taille du groupe est un aspect important. Par exemple, il y a plus de chance qu'un individu en confronte un autre, dans un groupe de trois personnes (triade) que dans un groupe de deux (dyade), car l'individu compte sur le support de l'autre. Le troisième individu joue, dans ce cas, le rôle d'intermédiaire, dont la position renforce le poids de la position d'un des deux autres membres. Le nombre de relations affectant possiblement les individus d'un groupe est de $n(n - 1)/2$, où n est le nombre de personnes dans le groupe. Les groupes de taille paire ont plus tendance à avoir des désaccords que les groupes de taille impaire, probablement par l'émergence d'un médiateur lorsque la taille est impaire.

À partir d'environ sept jusqu'à environ quinze membres, les demandes faites sur le leader s'amplifient et il est de plus en plus différencié des autres. Plus le groupe est grand plus la tolérance à être dirigé augmente, plus les actions sont centralisées et les règles et procédures formalisées, plus la tendance à vouloir dominer des membres plus actifs est forte. Avec cette augmentation, l'atmosphère devient moins intime et la satisfaction des individus plus faible. Cela entraîne une tendance à former des sous-groupes.

La cohésion est le degré d'attraction entre les membres et leur groupe. Si la cohésion est forte, un membre a plus de chance de prendre des responsabilités pour le groupe, de participer aux rencontres, d'influencer les autres, de les écouter, et d'accepter leurs opinions. Aussi, les buts du groupe seront mieux estimés et les standards plus facilement adoptés. La cohésion est donc liée à l'efficacité du groupe à travailler dans le sens des buts du groupe, ainsi qu'à la satisfaction des membres [Pel03].

2.3.1 Le comportement de groupes

Dans son approche de simulation de comportements de groupes, Reynolds simule les nuées d'oiseaux, les troupeaux d'animaux et les bancs de poissons [Rey87]. Un groupe est l'élaboration d'un système de particules avec acteurs (oiseaux ou poissons) en tant que particules. Un groupe est supposé le résultat de l'interaction entre les comportements individuels des acteurs. De façon individuelle, les acteurs tentent de rester ensemble et d'éviter les collisions avec les autres acteurs, d'une part et les objets aux alentours, d'autre part. Les positions, les vitesses et les différentes orientations des acteurs sont connues dans le système à tout instant. L'animateur peut contrôler plusieurs paramètres globaux: [Fou06]

- Le poids du composant évitement d'obstacles.
- Le poids de la convergence vers le but.
- Le point du centrage du groupe.
- Le poids de l'égalité des vitesses.
- La vitesse maximale.
- L'accélération maximale.
- La distance minimale entre les oiseaux.

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

Le voisinage est spécifié par une distance qui est défini quand deux acteurs sont "voisin" et un angle qui définit la perception de l'acteur "champ de voisin" (Figure II.6).

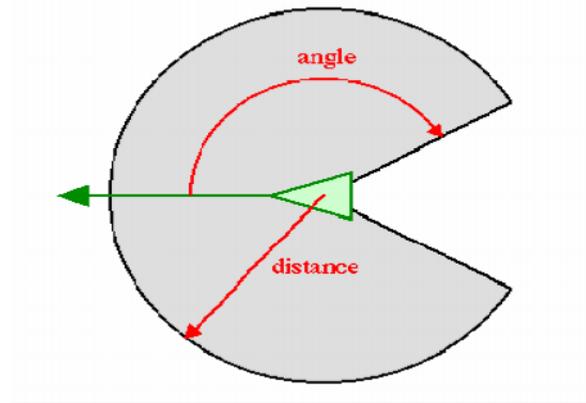


Fig.II.6. Notion de voisinage.

2.3.1.1 La séparation

Le comportement de séparation donne à l'acteur la capacité de maintenir une certaine distance de séparation des autres caractères qui lui sont proches. Cela peut être employé pour empêcher les acteurs de s'entasser ensemble. Pour calculer la direction de séparation, d'abord une recherche est faite pour trouver d'autres acteurs dans le voisinage indiqué. Pour chaque acteur voisin, la force répulsive est calculée en soustrayant les positions de notre acteur et l'acteur voisin, ensuite la normalisation de l'application à $1/r$ indemnité. (C'est-à-dire la position compense le vecteur est mesuré par $1/r^2$). Notons que $1/r$ est juste un arrangement et non une valeur fondamentale [Rey99].

Les forces répulsives pour chaque acteur voisin sont additionnées ensemble pour produire la force de direction (Figure II.7).

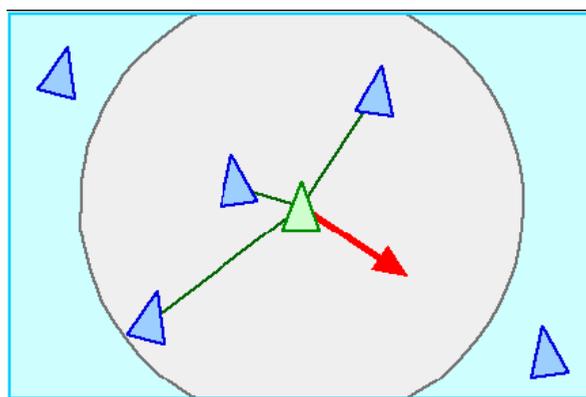


Fig.II.7. Le comportement de séparation.

2.3.1.2 La cohésion

Le comportement de cohésion donne aux acteurs la capacité de générer une situation de cohérence dans le groupe qu'ils forment (se rapprocher et former un groupe) avec d'autres acteurs voisins (Figure II.8).

La direction de la cohésion peut être calculée en déterminant tous les acteurs dans le voisinage local (comme décrit plus haut pour la séparation) et en calculant "la position moyenne" (ou "centre de gravité") des acteurs voisins.

La direction de la force peut ainsi être appliquée dans la direction de "la position moyenne" [Rey99].

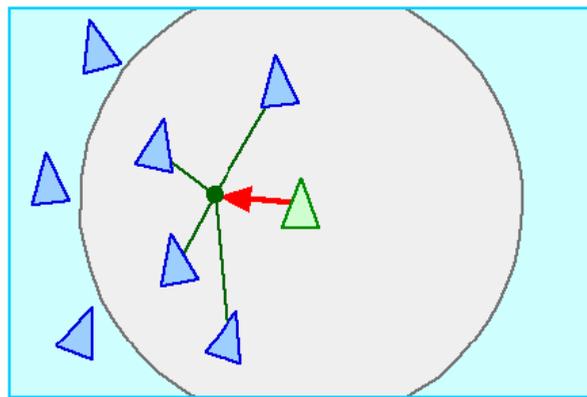


Fig.II.8. Le comportement de cohésion.

2.3.1.3 L'alignement

Le comportement d'alignement donne à l'acteur la capacité de s'aligner sur d'autres acteurs voisins (Figure II.9).

La direction d'alignement peut être calculée en déterminant tous les acteurs dans le voisinage local (comme décrit précédemment pour la séparation), en faisant la moyenne des ensembles des vitesses des acteurs voisins.

Cette moyenne est "la vitesse désirable", ainsi le vecteur de direction est la différence entre la moyenne et la vitesse courante de notre acteur. Cette direction a tendance à faire tourner notre acteur, il est ainsi aligné sur ses voisins [Rey99].

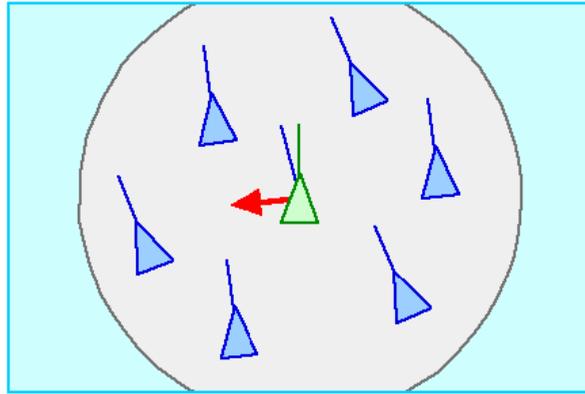


Fig.II.9. Le comportement de d'alignement.

Additivement aux comportements de séparation, de cohésion et le comportement d'alignement, ces comportements peuvent être combinés pour produire le modèle de comportement de masse, de troupeau et ou de colonie. En général, et pour la majorité des applications, il suffit simplement de résumer les trois directions des vecteurs de force pour produire une direction simple combinée pour les masses. Cependant, et pour le meilleur contrôle, il est utile d'abord de normaliser les trois composantes de direction pour les mettre ensuite à l'échelle par trois facteurs d'indemnité avant des les récapituler.

2.3.1.4 Le suivi de leader

Le comportement de suivi de leader, est provoqué par un ou plusieurs acteurs à l'effet de suivre un autre acteur en mouvement et considéré comme le leader. Généralement, ces acteurs tentent des rester près du leader, sans le cohue et faisant attention à rester loin de la voie du leader (dans le cas où ils arrivent à trouver des disciples devant le leader). De plus, s'il y a plus d'un disciple, ils doivent éviter de le heurter (Figure II.10) [Rey99].

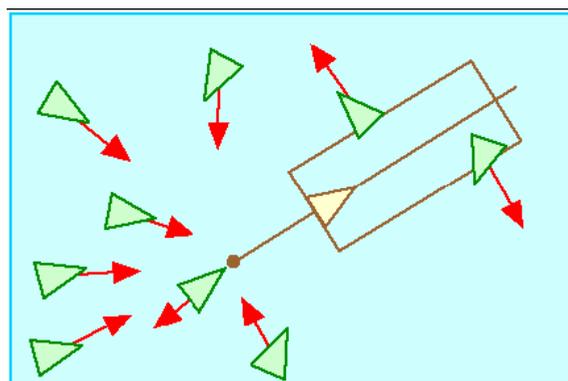


Fig.II.10. Comportement de suivi de leader.

2.4 La foule

2.4.1 Simulation de foule

Quelques travaux ont été discutés dans la bibliographie afin de simuler les foules virtuelles [MT01]. Le premier travail dans ce secteur a été effectué par Reynolds [Rey99] qui a décrit un modèle distribué de comportement pour simuler des bandes d'oiseaux constitués par des acteurs dotés de qualifications de perception. En fait, ces oiseaux maintiennent la position et l'orientation appropriées de la bande en équilibrant leur faculté d'éviter des collisions avec leurs voisins, pour assortir leur vitesse avec celle de leurs voisins ainsi que leur mouvement vers le centre de la bande. Le travail de Reynolds met en évidence une animation de groupe très réaliste, créée en appliquant des règles locales simples dans une structure de bande.

Mataric [Mat94] a décrit des comportements émergents de 20 robots mobiles, ce qui démontre le comportement d'errance de groupe, de poursuite, d'agrégation, de dispersion ainsi que d'assemblage. Quelques comportements complexes tels que l'assemblage ont été produits en combinant des comportements simples comme l'agrégation et le suivi par exemple. Cependant, des comportements hétérogènes sont étudiés pour fournir une hiérarchie de groupe où les robots peuvent agir différemment.

Brogan et Hodgins [BH97, BMH98] ont employé la dynamique pour modéliser le mouvement des groupes en utilisant les concepts de la physique. Ils reproduisirent des mouvements de robots, de cavaliers à jambes de bicyclette et des systèmes de masse-point basés sur la dynamique, un algorithme pour déterminer la position désirée pour chaque individu donné, les positions et les vitesses des créatures ainsi que les obstacles évidents. En effet, un processus de perception a été associé aux acteurs dans le but de déterminer les obstacles évidents pour chaque individu dans le groupe. Ce processus précède l'entrée en action de l'algorithme de déplacement.

Un autre travail intéressant a été présenté pour simuler des foules des fourmis d'une manière automatique [Ant98]. Le système de simulateur de foule est employé pour reproduire le mouvement de milliers de créatures, prenant en compte une combinaison des forces physiques et des règles du comportement.

SIMULEX [TLO03], développé par Thompson et Marchant, décrit un logiciel bidimensionnel modélisant l'évacuation d'une grande population dans un bâtiment. Le but de

SIMULEX était de modéliser le mouvement d'évasion de chaque occupant individuel dans un bâtiment, où la vitesse de marche de chaque personne est évaluée indépendamment de la densité moyenne du groupe dans un secteur défini d'un bâtiment.

Le groupe de Thalmann [MGT99] a développé un système de foule appelé ViCrowd. ViCrowd a présenté un modèle pour produire automatiquement des foules humaines basées sur des groupes, à la place d'individus. Les groupes sont des structures plus intelligentes, alors que les individus suivent les spécifications de groupes. ViCrowd est basé sur des règles locales appliquées aux groupes, où les groupes ne représentent pas une structure rigide. En outre, une définition des règles comportementales en utilisant des événements et des réactions conditionnelles est incluse pour fournir des foules plus autonomes.

La simulation des foules humaines dans le but de peupler des mondes virtuels fournit un sens plus réaliste de la présence virtuelle de groupe. Il existe plusieurs approches pour modéliser des foules autonomes [BCN97, BH97], [Rey99], [TT94]. Dans quelques environnements virtuels, il est souvent utile de simuler des populations d'une manière autonome, ainsi les agents ont un type de connaissance et peuvent se déplacer et agir l'un sur l'autre dans cet environnement. Cependant, selon l'application, différentes manières d'interaction peuvent être exigées afin de fournir une communication en temps réel entre les participants et les agents virtuels. Le groupe de Thalmann [MGT99] a défini trois niveaux d'autonomie : (Foule autonome, foule guidée et foule programmée).

Ces trois niveaux d'autonomie sont représentés en utilisant deux types d'interface : interface pré-établie ou guidée. L'interface pré-établie emploie un langage de scripts où l'action, le mouvement et les règles comportementales sont définis afin d'indiquer les comportements de foule. Tandis que l'action et le mouvement décrivent des comportements explicites de foule, appelés foule programmée, les règles comportementales sont employées pour définir la foule autonome. Toutes ces informations peuvent également être envoyées par un processus externe afin de guider des foules explicitement, pendant la simulation. Ils ont appelé ce type de foule «foule guidée».

Pour la gestion des foules, plusieurs niveaux de contrôle [MT01] peuvent être distingués, chacun d'entre eux ayant une influence sur l'autonomie des personnages la constituant mais aussi sur le temps de calcul global de l'application (tableau II.1 ce tableau est extrait des travaux de Raup Musse). La foule est ensuite décrite de manière hiérarchique (foule, groupes, individus) au travers de ViGrowd, l'architecture de simulation (Figure II.11).

Elle prédéfinit un certain nombre de comportements liés à des algorithmes de gestion de nuée [Rey 00] pour la gestion d'un groupe d'entités, le groupe est alors considéré comme une entité à part entière avec ses propres buts. Les entités ensuite ont le choix de suivre le groupe ou bien de changer de groupe.

Contrôle du comportement	Foules guidées	Foules Programmées	Foules autonomes
Niveau d'autonomie	Base	Moyen	Haut
Niveau d'intelligence	Base	Moyen	Haut
Rapidité d'exécution	Haute	Moyenne	Basse
Complexité des Comportements	Basse	Variable	Haute
Niveau d'interaction	Haut	Variable	Variable

Tab II.1. Comparaisons des différentes caractéristiques comportementales des foules en fonction du type de contrôle qui leur est imposé.

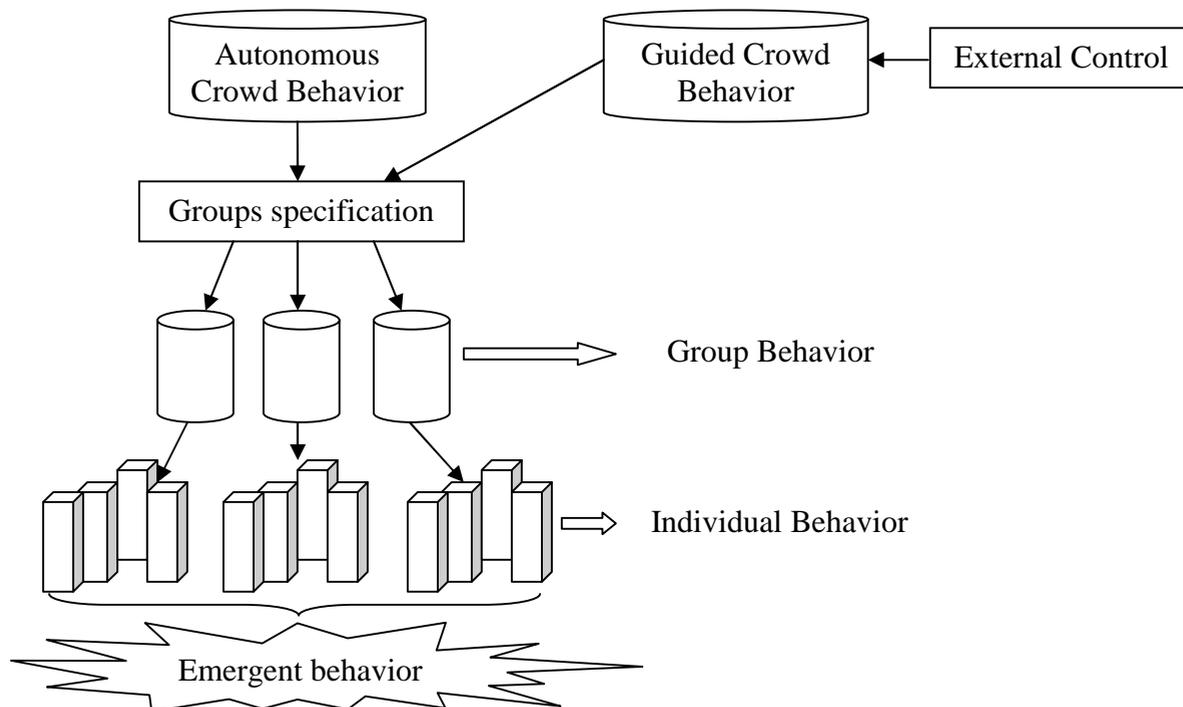


Fig.II.11. Architecture de contrôle des foules issue des travaux de Raupp Musse

2.4.2 Relations entre les individus

Au niveau individuel, les agents possèdent trois possibilités d'action interindividuelle. La première est l'évaluation des positions et la décision de qui doit changer de direction pour éviter une collision. Si l'agent est dans une région de densité forte en terme de nombre d'agents, la détection de collision survient seulement quand les individus sont près les uns des autres. Si l'agent est dans une région à faible densité, la détection est effectuée selon une distance plus grande entre les agents. La deuxième possibilité est d'attendre un agent du groupe qui est en retard, en vérifiant si tous les agents du groupe ont atteint le but. La dernière est de rencontrer un autre agent et d'évaluer les paramètres sociaux selon les règles définies dans le modèle social. Cette dernière possibilité peut changer le comportement du groupe et de la foule, comme nous l'avons vu précédemment. L'attroupement des agents est défini par les quatre règles suivantes [Pel03] :

- Les agents d'un même groupe partagent la même liste de buts ;
- Ils marchent à la même vitesse ;
- Ils suivent les chemins générés entre les buts ;
- Un agent peut attendre un autre membre du groupe quand il arrive à un but.

Ces quatre règles ont pour conséquence que les agents d'un même groupe marchent ensemble.

2.5 Processus mentaux de navigation

Une fois que l'être humain dispose d'une représentation mentale de son environnement, il peut l'exploiter afin d'organiser son déplacement. Ces processus mentaux sont scindés en trois groupes principaux [Seb07] :

1. Raisonement spatial, où l'individu va associer les tâches qu'il doit effectuer avec leur localisation, puis procéder à une sélection.
2. Planification d'un chemin, permettant à l'individu d'évaluer son déplacement au sein de l'environnement jusqu'à la tâche sélectionnée, et donc de se projeter dans le temps et l'espace.
3. Navigation et locomotion, qui vont gérer le déplacement local d'un individu tout au long du chemin planifié.

2.5.1 Planification d'un chemin

La planification de chemin (aussi dénommée navigation dans la littérature anglaise) consiste à lister les lieux par lesquels passer pour relier la position courante d'un humain à sa destination. Ce processus mental est bien entendu très fortement lié à la représentation mentale de l'espace, étant une exploitation directe des informations contenues dans la carte cognitive spatiale de l'individu.

Concernant le fonctionnement de la tâche de planification de chemin, plusieurs études montrent qu'il s'agit d'un procédé hiérarchique. Ainsi, S. Steck et H. Mallot [SM00] indiquent que des lieux marquants locaux et globaux interviennent dans le processus, mais de façon différente suivant les personnes : certaines n'utilisent que l'un ou l'autre, ou encore les deux à la fois. De plus, J. Wiener et H. Mallot [WM03] introduisent dans leur carte mentale hiérarchique les concepts de place et région définissant deux niveaux d'abstraction topologique. Ils expliquent que la planification de chemin se fait d'abord localement au niveau des places, puis globalement au niveau des régions dès que le processus s'éloigne du lieu marquant courant de l'individu (Figure II.12 En haut, l'environnement de test, et son équivalent mental en **a**. En **b** les conditions de départ (position et destinations), et en **c** les chemins hiérarchiques.).

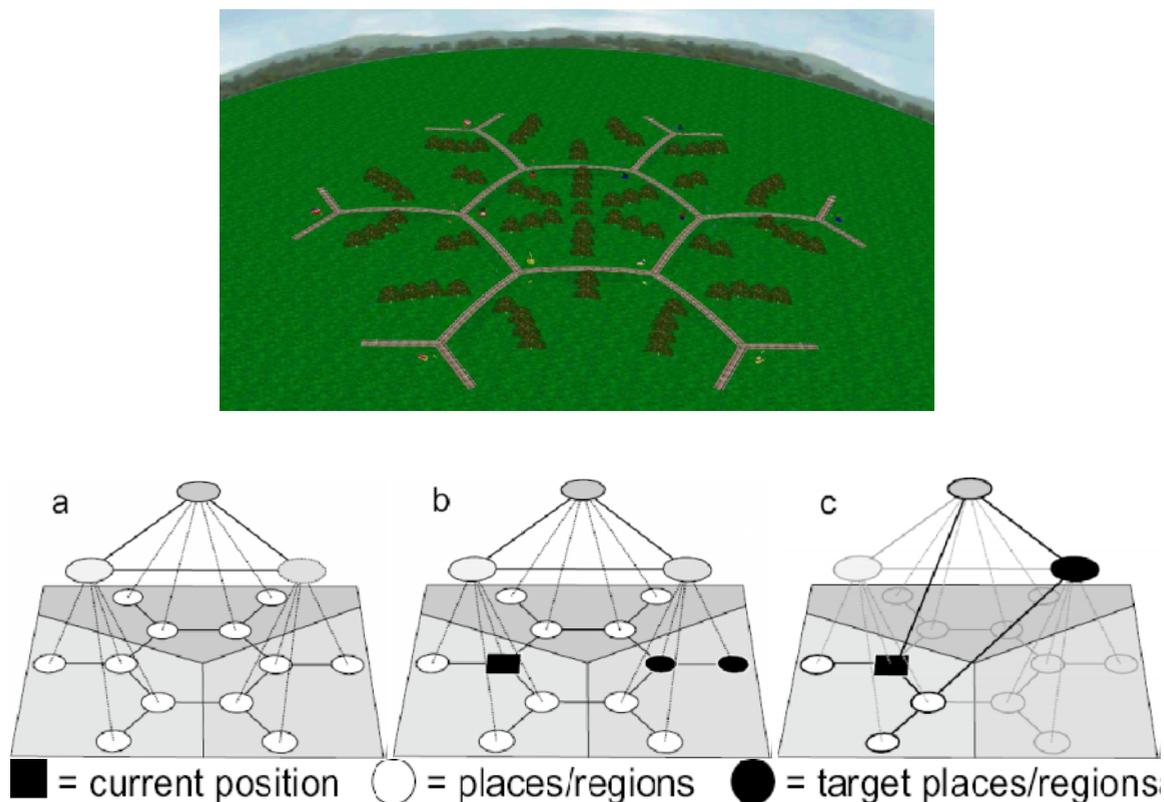


Fig.II.12. Représentation mentale hiérarchique pour la planification de chemin selon J.Wiener et H. Mallot.

2.5.2 Locomotion

La locomotion est le comportement humain permettant le déplacement de l'individu au sein de l'environnement, jouant de ce fait un rôle central dans de nombreuses activités. Celui-ci se doit d'être grandement adaptatif, notamment du fait de sa gestion de l'évitement de collision avec d'autres entités présentes dans l'environnement. Ainsi, W.H. Warren [**War95**] indique que la vision exerce un contrôle fondamental sur la locomotion humaine.

M. Relieu [**Rel96**] indique que le lien social entre piétons est caractérisé par le silence et l'indifférence. Afin de maintenir cet état de fait, E. Goffman [**Gof71**] énonce que trois comportements sont mis en œuvre lors de la locomotion :

L'externalisation : est le processus par lequel les gens fournissent constamment des informations sur leurs intentions de déplacement, que ce soit la direction, la vitesse, ou encore la longueur de pas. Cette transmission se fait par des successions de mouvements corporels, devant être perceptibles par l'entourage du piéton.

La scrutation : est le processus complémentaire à l'externalisation. Celui-ci permet à un piéton de regrouper sélectivement les informations émises par son entourage.

La minimisation : des ajustements signale que le piéton réduit l'amplitude et le nombre de changements d'orientation au cours de son trajet. Cela implique une anticipation dans la modification des trajectoires, afin de la rendre perceptible assez tôt par l'entourage, dans le but d'éviter une coordination ainsi qu'une interaction proche.

De plus, J.R.E. Lee et R. Watson [**LW92**] remarquent que les conditions socio-culturelles influent sur la minimisation des interactions, notamment par la prise en compte implicite du code de la route dans l'organisation globale des flux (comme l'évitement à droite).

Concernant l'interaction locale lors de la locomotion, E. Goffman introduit la région ovale de sécurité (Figure II.13). Sa profondeur est dépendante de la vitesse du piéton, définissant ainsi une zone d'anticipation, tandis que sa largeur détermine des paramètres d'aisance personnelle, désignant la distance acceptable pour passer à côté d'un obstacle, dont notamment une personne. De manière plus générale, M. Relieu [**PQ98**] énonce le principe de

discrimination urbaine, signalant que le piéton focalise son attention à l'intérieur de sa région courante pour en extraire une information pertinente.

A : Distance d'anticipation.

B : Distance latérale de sécurité.

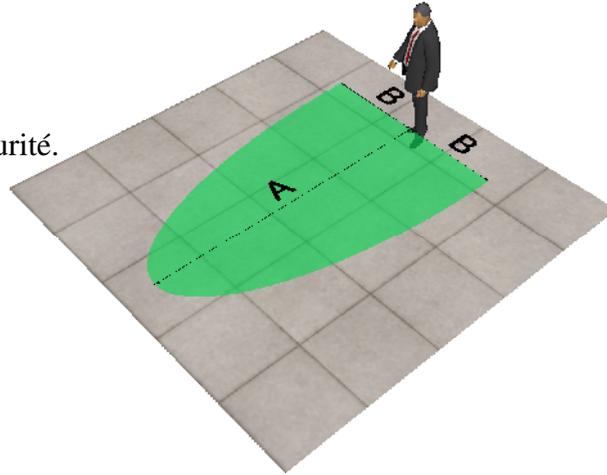


Fig.II.13. Représentation de l'aire de Goffman (en vert).

Concernant l'organisation dans l'espace de la locomotion, B. Hillier et al. [HPH+93] indiquent que la plupart des mouvements piétonniers suivent les lignes de vue pour le choix du chemin local, tentant ainsi de maximiser la distance couverte par le regard. M. Azimazadeh et B. Klarqvist [AK03] signalent que cette propriété a un impact direct sur le choix d'un chemin, et permet d'expliquer le fait que des chemins aller et retour soient généralement différents.

2.6 Conclusion

La simulation du comportement de navigation d'humanoïdes virtuels pose plusieurs problèmes et s'avère être un domaine pluridisciplinaire, ceci en plusieurs sens : d'une part en terme de domaines de recherche où les informations issues de la socio-psychologie sont importantes à prendre en compte pour la modélisation informatique du comportement, d'autre part en terme informatique où elle touche à plusieurs problèmes tels que la représentation de l'environnement, la planification de chemin, la détection de collision et la définition de comportements de plus ou moins haut niveau. Pour obtenir une architecture efficace alliant à la fois autonomie et rapidité, les compromis à faire sont nombreux.

Nous avons essayé de définir et de décrire dans ce chapitre «l'aspect d'autonomie» des acteurs. Nous avons présenté une décomposition des comportements de mouvement pour les acteurs autonomes selon une hiérarchie à trois niveaux : le choix, la direction et la locomotion.

Nous avons, ainsi, présenté une collection de comportements simples et communs de déplacement. (La recherche, l'arrivée, l'évitement d'obstacles, l'évitement de collisions,...). Nous avons terminé notre chapitre par une présentation des travaux sur la simulation des foules d'humains virtuels.

Chapitre III

Capture de mouvements

3 Capture de mouvements

3.1 Introduction

L'animation d'humains virtuels est un problème récurrent en synthèse d'images. Deux approches ont été développées au cours des dernières années. La première vise à améliorer les techniques de cinématique inverse et de manipulation d'images clefs pour augmenter le contrôle de l'infographiste sur l'animation. La seconde approche se focalisait sur la génération automatique de mouvements à partir de la simulation des forces animant un corps humain. Bien que donnant des résultats acceptables visuellement, elles atteignaient vite leurs limites, lorsque la complexité de l'humanoïde géré devenait trop importante. En effet un modèle complexe est très difficile à manipuler par des techniques d'images clefs ou de cinématique inverse, même pour un infographiste expérimenté. Et la seconde approche offre dans ce cas des temps de calculs beaucoup trop importants et une maîtrise aléatoire des paramètres.

Ici, au lieu de faire appel au talent artistique d'un infographiste, nous pouvons utiliser des systèmes qui vont enregistrer le mouvement d'un acteur, pour ensuite l'exploiter, et l'utiliser lors de la production d'image de synthèse. Ainsi on sera sûr, d'obtenir une animation réaliste. Divers types de méthodes sont employées pour arriver à capturer le mouvement basées sur différentes technologies de repérage d'un point physique dans un espace.

Les systèmes de capture de mouvement qui fournissent des animations de grandes qualités. Ils ont offert un moyen confortable de produire des animations réalistes. On les a utilisés au cours de nombreuses occasions comme, par exemple, dans la production d'effets spéciaux au cinéma, dans l'animation de shows télévisés, ou encore dans les jeux vidéo.

Nous allons présenter dans ce chapitre les systèmes de capture de mouvements. Trois grandes classes de systèmes sont utilisées : système opto-électronique, système mécanique, système magnétique, système vidéo et système sonore, et ses avantages et ses inconvénients de chaque système. Puis nous allons parler sur la méthode de mélange et adaptation de mouvements capturés. Nous finirons par une représentation de positionnement des marqueurs sur le corps du sujet.

3.2 Les systèmes de capture du mouvement

Les systèmes de capture du mouvement sont des outils “software ” et “ hardware ” qui permettent l’acquisition en temps réel ou en temps différé du mouvement d’un objet dans l’espace. Différents systèmes de capture existent sur le marché. Ils diffèrent essentiellement par leur technologie. Ils sont de nos jours fréquemment utilisés en production pour l’animation de personnages. Ces systèmes mesurent la position ou l’orientation des membres d’un acteur réel selon une fréquence d’échantillonnage qui leur est propre.

Les informations recueillies sont appliquées en temps réel ou en temps différé sur le personnage virtuel. Dans ce dernier cas, elles sont enregistrées et traitées par des logiciels spécifiques au système.

L’idée première d’associer des capteurs à un acteur pour en acquérir le mouvement a lentement évolué vers l’utilisation de systèmes d’acquisition plus simples à mettre en œuvre. La Souris, les tables de mixage... sont devenues de véritables systèmes de capture du mouvement [Web 1].

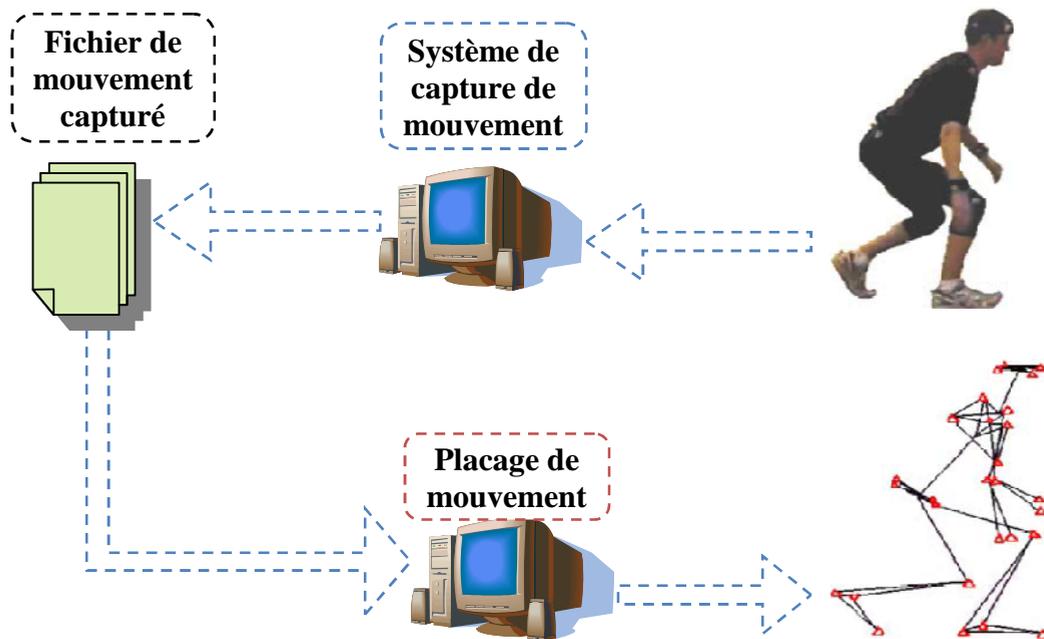


Fig.III.1. Principe de système de capture de mouvement.

Les systèmes d’acquisition de mouvement humain sont classés en tant que systèmes extérieur-entrant, intérieur-sortant et intérieur-entrant [Men03].

Ces appellations sont représentatives de l'emplacement des sources capturées et des récepteurs.

- Les systèmes extérieur-entrant où les informations sont collectées par des récepteurs externes (non rattachés au corps humain) et les données acquises sont émises par des sources placées sur le sujet en mouvement. Les systèmes d'acquisition utilisant des caméras comme récepteurs font partie de cette catégorie (les marqueurs positionnés sur le corps étant les sources),
- les systèmes intérieur-sortant où des récepteurs intérieurs placés sur le corps du sujet collectent des informations émises par des sources externes, comme par exemple les systèmes d'acquisition magnétique où les récepteurs calculent les informations à partir de champs magnétiques ambiants,
- les systèmes intérieur-entrant où les sources et les récepteurs sont rattachés au corps du sujet. Par exemple, les systèmes d'acquisition mécaniques où les récepteurs sont des potentiomètres ou goniomètres et l'émetteur est le squelette humain font partie de cette catégorie.

Tous ces systèmes permettent donc d'acquérir le mouvement humain de manière réaliste. Chacun possède ses avantages et inconvénients qui influent directement sur la qualité du mouvement et le temps de production. Parmi tous les systèmes existants on distingue néanmoins 3 grandes familles : Les systèmes opto-électroniques, magnétiques, mécaniques. Cependant, il existe d'autres systèmes d'acquisition moins répandus pour l'animation de mannequin tels que l'acquisition vidéo (qui se base sur une ou plusieurs images pour retrouver les positions et orientations des différentes parties du corps), les systèmes d'acquisition sonore (qui utilisent la vitesse de propagation du son dans l'air) ou la capture par centrales inertielles [**Men03**].

Quel que soit le système d'acquisition utilisé, les données récupérées doivent souvent être soumises à une phase de traitement permettant par exemple d'estimer les positions des centres articulaires ou de gérer des problèmes d'occultations de marqueurs dans le cas d'une acquisition optique.

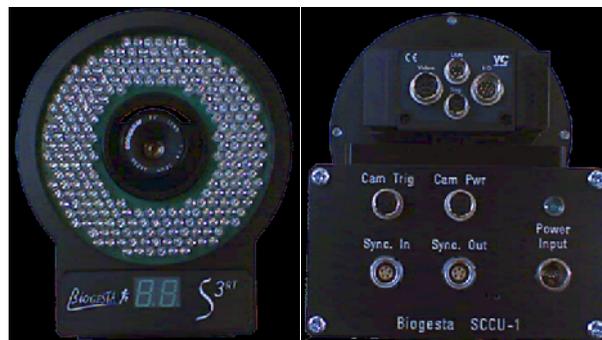
3.2.1 Les systèmes d'acquisition opto-électroniques

3.2.1.1 Marqueurs passifs

Les systèmes opto-électroniques à marqueurs passifs sont sûrement, à l'heure actuelle, le type de système d'acquisition le plus répandu. Les deux principales sociétés concurrentes qui commercialisent ces produits sont **Vicon** (Figure III.2.a) [Web 2] et **Motion Analysis**. La société française **BIOGESTA** (Figure III.2.b) [Web 3] commercialise aussi ce type de produit, notamment le système **Saga-3RT**.



(a) Plateau de capture de Motion Vicon.



(b) Caméra BIOGESTA.

Fig.III.2. Exemple de système d'acquisition opto-électronique.

Le principe de fonctionnement est analogue à celui des radars : les caméras émettent un rayonnement (Chaque caméra est équipée d'un projecteur (LED) en général rouge et/ou infrarouge), réfléchi par les marqueurs (dont la surface est composée d'une matière ultra

réfléchissante) puis renvoyé à ces mêmes caméras. Celles-ci ne sont sensibles qu'à un type de longueur d'onde et visualisent les marqueurs sous forme des taches vidéos blanches (ou en niveau de gris pour les caméras les plus récentes) [Bou09].

Les marqueurs sont suivis en deux dimensions par chacune des caméras. Une reconstruction en trois dimensions est possible lorsque chaque marqueur est repéré par au moins deux caméras. Le principe de reconstruction volumique se base sur les méthodes de triangulation spatiale. A noter qu'une calibration du volume d'étude est toujours nécessaire afin, notamment, de déterminer les positions et les orientations relatives des caméras ainsi que de définir le repère de référence (Figure III.3).

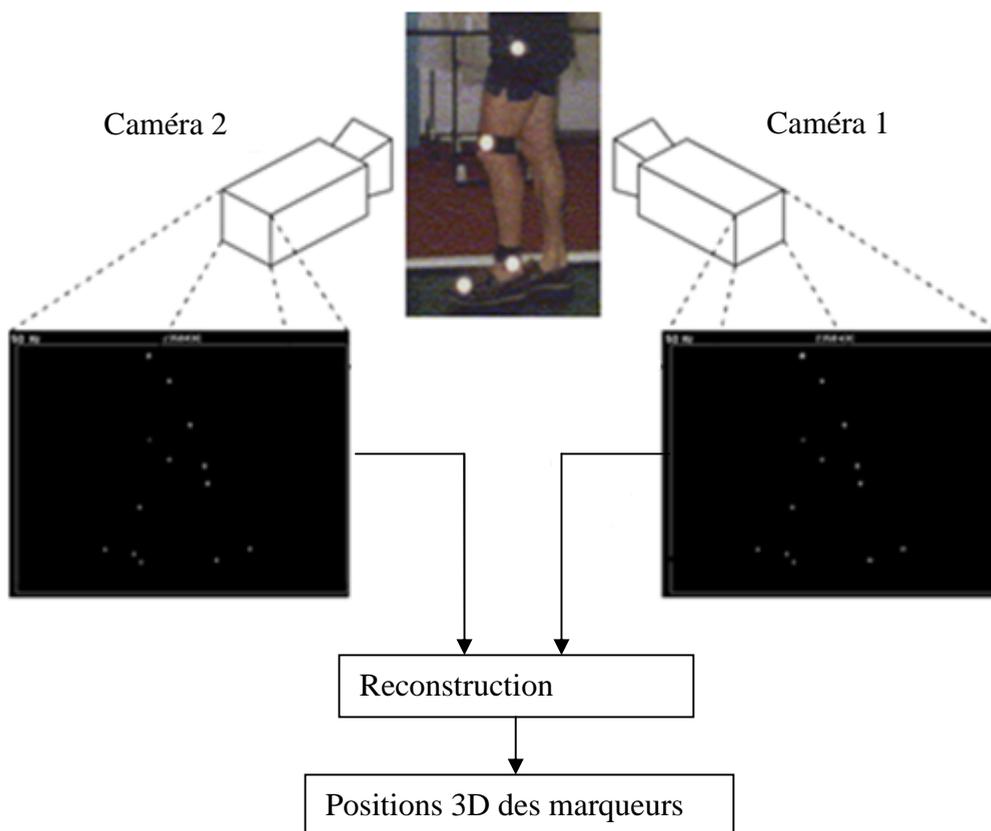


Fig.III.3. Principe du système d'acquisition optique.

L'avantage principal est la liberté de mouvement est très peu altérée par des marqueurs petits et légers. Cependant, un des inconvénients majeurs est le non différenciation des marqueurs.

3.2.1.2 Marqueurs actifs

Ce type de systèmes est composé de marqueurs qui émettent directement un signal infrarouge (LED), chaque marqueur identifié par les paramètres des ondes émises (longueur

d'onde et fréquence), se différencier les uns des autres. Ces marqueurs sont donc identifiés automatiquement et immédiatement en temps réel comme étant des points dynamiques avec des coordonnées 3D dans l'espace virtuel [Bou09].



Fig.III.4. Système Optotrak Smart Markers.

Les principaux avantages sont d'une part la différenciation automatique et robuste des marqueurs, et d'autre part la distance importante qu'il peut y avoir entre le sujet en mouvement et les caméras. Aucun projecteur n'est nécessaire donc l'enregistrement peut se réaliser en lumière ambiante. L'inconvénient majeur est le dispositif, le plus souvent filaire (Figure III.4), des marqueurs qui ont besoin d'une alimentation et d'une synchronisation. Ce dispositif gêne le sujet ou, tout au moins limite les gestes à analyser (mouvement du matériel embarqué).

(a) Les avantages des systèmes opto-électroniques : [Men03]

- Grande précision des données acquises dans la plupart des cas ;
- Un grand nombre de marqueurs peuvent être utilisés ;
- Il est facile de changer la configuration des marqueurs ;
- Estimation possible du squelette interne grâce à des groupes de marqueurs ;
- Le sujet est faiblement contraint dans ses mouvements par le système ;
- Le champ d'acquisition est relativement grand (dépendant du nombre de caméras) ;

- Très grande fréquence d'acquisition possible.

(b) Les inconvénients des systèmes opto-électroniques :

- Traitements coûteux liés aux nommages et occultations de marqueurs ;
- Traitements coûteux liés à la reconstruction des centres articulaires ;
- Mesures en extérieur impossibles dans le cas de systèmes à base de lumière infra-rouge.

3.2.2 Les systèmes d'acquisition magnétiques

Les systèmes d'acquisition magnétiques se composent d'un ensemble de récepteurs qui mesurent leurs positions et orientations spatiales par rapport à un émetteur par l'intermédiaire d'une unité électronique embarquée. Ces récepteurs (ou capteurs) sont placés sur le corps du sujet et reliés à une unité électronique de contrôle par des câbles. Cette unité collecte les positions et orientations des différents capteurs et les transmet à un ordinateur. La figure III.5 montre quelques exemples de systèmes magnétiques.



(a) MotionStar Wireless



(b) StarTrak de Polhemus



(c) StarTrak de Polhemus

Fig.III.5. Exemple de système d'acquisition magnétique.

L'émetteur génère un champ électromagnétique (de 3 à 15 mètres) modulé à basses fréquences qui est détecté par les récepteurs. Les informations captées sont filtrées puis amplifiées. Une fois les informations transmises à l'ordinateur, un logiciel retrouve la position cartésienne et l'orientation de chaque capteur. En sortie, le système renvoie la position et l'orientation de chaque capteur au cours du temps. Une dernière phase de traitement est

donc nécessaire (mais généralement non traitée par le système) pour retrouver les positions approchantes des centres articulaires et les repères locaux associés aux segments du squelette [Men03].

(a) Les avantages des systèmes magnétiques :

- Capture presque temps réel ;
- Positions et orientations des capteurs disponibles sans traitement ;
- Aucune occultation de capteurs.

(b) Les inconvénients des systèmes magnétiques :

- Contraintes sur l'environnement (absence de métaux, faible zone d'acquisition) ;
- Contraintes sur le sujet (câbles, capteurs fragiles) ;
- Fréquence d'acquisition insuffisante pour certains mouvements (au maximum 150Hz) ;
- Difficile de changer le positionnement des capteurs.

3.2.3 Les systèmes d'acquisition mécaniques

Les systèmes d'acquisition mécaniques se présentent sous la forme d'un ensemble de structures rigides liées par des potentiomètres ou autres périphériques similaires de mesures angulaires. Ces capteurs sont placés près des principaux centres articulaires et adoptent la posture du corps humain (Figure III.6). Ici, la mesure est proprioceptive et c'est le mouvement des articulations de l'exosquelette qui est enregistré avec une grande précision [Men03].

(a) Les avantages des systèmes mécaniques :

- Environnement peu contraint (grand champ, insensible au métal ou au soleil) ;
- Phase de traitement simplifiée (aucune occultation et bruit minime sur les mesures).

(b) Les inconvénients des systèmes mécaniques :

- Faible fréquence d'acquisition ;
- Système portatif contraignant pour le sujet (limitation des mouvements possibles) ;
- Le positionnement des capteurs est fixée ;

- Problème du calcul de la translation globale.



BodyTracker de Puppet Works ET



Gipsy2 de Analogus



La réalisation du film Avatar a également utilisé cette technologie [Web 4].

Fig.III.6. Exemple de système d'acquisition mécanique.

3.2.4 Les systèmes d'acquisition vidéos

Les systèmes d'acquisition vidéo se basent sur une ou plusieurs images pour retrouver les positions et orientations des différentes parties du corps humain au cours du temps. Ces systèmes rencontrent de nombreuses difficultés du fait de la complexité des images 2D à traiter. En effet, de multiples facteurs comme une grande quantité de textures et lumières, les occultations, les informations de profondeur non fournies, la calibration délicate des caméras influencent la phase d'identification de l'image. Un modèle géométrique simplifié (généralement composé de cylindres articulés) est le plus souvent employé pour faciliter l'identification du corps humain en mouvement à partir des images 2D. Dès lors, les algorithmes de reconstruction tentent de calquer le modèle géométrique à l'image pour

retrouver sa posture au cours du temps. Trois grandes familles de méthodes se distinguent actuellement [Men03] :

- (1) Les algorithmes basés modèles de mouvements qui utilisent les connaissances sur le mouvement pour prédire les images futures. L'image et le modèle sont traités dans deux espaces différents. Le modèle fournit un ensemble d'hypothèses concernant le mouvement futur. La validité de ces hypothèses est ensuite testée dans l'espace image. Les hypothèses sont raffinées et renvoyées dans le domaine du modèle afin d'être corrigées (Figure III.7). Etant donné la complexité des mouvements humains, cette approche reste délicate car le processus de prédiction peut amener la phase de test à considérer une recherche sur un trop grand espace de possibilités.

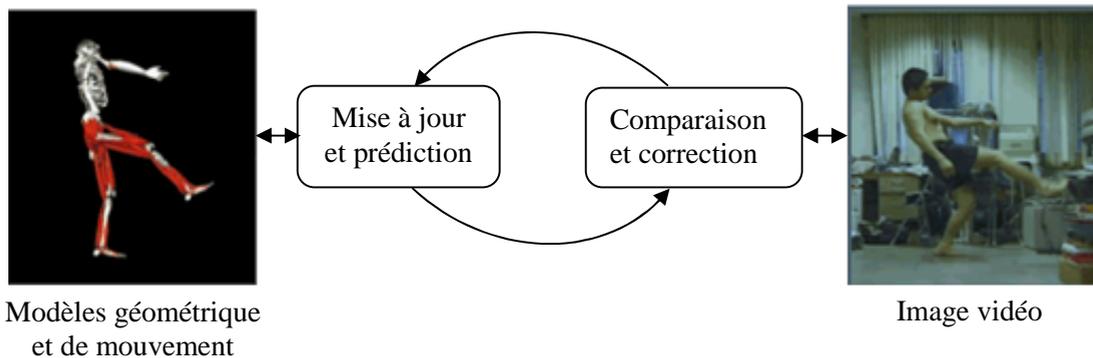


Fig.III.7. Système d'acquisition vidéo par cycles de prédiction/correction.

- (2) Les algorithmes utilisant la vidéo inverse proposent une alternative à l'estimation basée modèle. Cette technique, souvent utilisée en robotique, ne pose aucun a priori sur la nature du mouvement. Une projection non linéaire de l'espace du modèle géométrique (positions et orientations des articulations) dans l'espace image est effectuée. Cela implique plusieurs transformations cartésiennes aux articulations ainsi que des projections en perspective. La vidéo inverse est alors utilisée pour inverser la projection et retrouver une posture approchante du squelette à partir de l'image vidéo. Ces algorithmes sont toutefois très sensibles aux minimums locaux du fait de la nature hautement non linéaire de la projection.
- (3) La dernière famille d'algorithmes utilisent la stéréoscopie et plusieurs points de vue d'une même scène. En analysant les images et en mettant en correspondance les informations, ces techniques essaient de limiter les problèmes d'occultations

rencontrés dans les deux familles précédentes. Toutefois, d'autres problèmes interviennent ici tels que la précision de calibration des cameras ou la mise en correspondance des informations.

Les travaux actuels se focalisent sur la mise en œuvre de modèles géométriques de plus en plus complexes permettant de restreindre quantitativement l'ensemble des postures possibles. Une autre approche tend à proposer des modèles de mouvement offrant de meilleures prédictions [Men03].

(a) Les avantages des systèmes d'acquisition vidéos :

- Environnement peu contraint (prises de mesure en extérieur possibles),
- Aucune contrainte physique sur le sujet,
- Coût du matériel minime.

(b) Les inconvénients des systèmes d'acquisition vidéos :

- Analyse complexe de l'image (souvent assistée par l'utilisateur),
- Aucune information de profondeur (pour la reconstruction mono caméra),
- Occultations.

3.2.5 Les systèmes d'acquisition sonores

Les systèmes d'acquisition sonores travaillent généralement dans le domaine des ultrasons. Ils se composent d'un ensemble d'émetteurs triangulaires, de récepteurs et d'une unité de contrôle électronique. L'émetteur se compose de trois microphones, pour saisir les informations nécessaires à la reconstruction du positionnement en 3D. La distance d'un récepteur à l'émetteur se calcule en fonction de la vitesse de propagation du son dans l'air. La position du récepteur dans l'espace 3D est déduite par triangulation. L'orientation se calcule par rapport au repère constitué par les trois microphones [Men03].

(a) Les avantages des systèmes sonores :

- Grande précision ;
- Bon marché.

(b) Les inconvénients des systèmes sonores :

- Petit volume de travail ;
- Pas d'obstacle entre l'émetteur et le récepteur sinon perte de signal ;

- Perturbations dues à la température de l'air ;
- Sensibilité aux bruits de fond et autres sources ultrasonores.

3.3 Mélange de mouvements capturés

Lorsque l'on utilise une base de données de mouvements capturés, il peut parfois être intéressant de déformer ces mouvements capturés, soit en adaptant un mouvement capturé soit en mélangeant différents mouvements capturés, pour répondre à de nouvelles contraintes.

Les méthodes de mélange de mouvements cherchent à obtenir un mouvement par composition de plusieurs mouvements capturés. Si on a un mouvement de marche et un mouvement de saut, n'ayant pas été capturé ensemble mais séparément, on peut vouloir les enchaîner (Figure III.8). Pour cela il faut faire une transition entre la fin du mouvement de marche et le début du mouvement de saut. On utilise les techniques de mélange de mouvements pour réaliser cette transition.

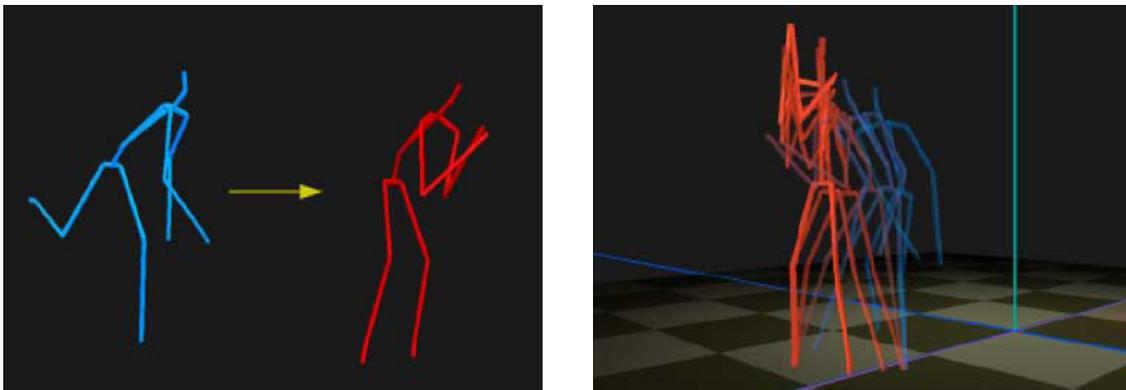


Fig.III.8. A gauche, les deux positions à mélanger. A droite, l'animation de transition entre les ces deux positions.

Les premiers travaux dans ce domaine proposèrent des interpolations linéaires dans une base de données de trajectoires capturées [AW01] [AW00] [Pet03]. Une autre approche consiste à raisonner posture par posture plutôt que de considérer un mouvement dans sa globalité. Ainsi, des graphes de mouvements sont proposés [KGP02] [LCR+02] : chaque nœud correspond à une posture et chaque arc est une transition possible entre 2 postures [WB03] [LL04] [GSK+03].

3.4 Adaptation de mouvements capturés

Le principe de l'adaptation d'un mouvement est de modifier un mouvement capturé afin de résoudre des contraintes cinématiques ou dynamiques différentes, comme par exemple modifier un mouvement de préhension pour saisir un objet dont la position est différent par rapport à celle de la trajectoire capturée. Pour résoudre ce type de problème, une approche consiste à imposer des contraintes (position à un instant donné) et à utiliser des méthodes d'optimisation non-linéaire pour trouver une solution [WK88] [WP95] [Coh92] [NM93] [LGC94] [FP03]. Ces méthodes se limitent généralement à des contraintes cinématiques, même si quelques travaux tentent de tenir compte partiellement de la dynamique [ZMC+05] [PB00] [ZH99] [AFO05] [KHL05].

Ainsi, Ménardais et Multon proposent une architecture logicielle permettant à la fois de combiner et d'adapter des mouvements capturés [Men03]. A partir de cette architecture, Kulpa [Kul05] s'intéresse à l'adaptation de mouvements pour des contraintes cinématiques, dynamiques et cinétiques pour proposer une méthode de simulation de mouvements appelée MKM (Manageable Kinematic Motion) [MKB08].

Enfin, il peut également être nécessaire d'adapter les trajectoires articulaires (obtenues par capture de mouvement) pour un nouveau squelette de morphologie différente. Plusieurs travaux s'intéressent à cette phase d'adaptation : Monzani et al. utilisent un squelette intermédiaire pour adapter des trajectoires articulaires à une nouvelle morphologie [MBB+00], Choi et Ko proposent une adaptation temps réel des mouvements d'un personnage à un autre [CK00] ou encore Ménardais et al. proposent une adaptation de trajectoires capturées pour un nouveau mannequin de morphologie différente ou pour un nouvel environnement.

Enfin, Pronost et al. proposent une interpolation, à partir d'une base de données de mouvements capturés, en fonction des paramètres anatomiques d'une personne à simuler [PDB+06]. Yamane et al. proposent une approche d'animation de personnage manipulant des objets combinant un planificateur de trajectoires et une base de données de mouvement capturés [YKH04].

3.5 Positionnement des marqueurs

Pour enregistrer le mouvement, c'est-à-dire la position et l'orientation dans le repère terrestre de chacun des segments précédemment cités, plusieurs marqueurs sont positionnés sur la peau du sujet.

Le choix de la position des marqueurs est essentiel pour le traitement ultérieur des données. Les marqueurs sont positionnés en fonction de repères anatomiques de référence et contribuent à la localisation des centres articulaires. Plus le nombre de marqueurs utilisés est grand plus la détermination de la position et de l'orientation des segments est précise. Nous avons tout de même opté pour un nombre de marqueurs faible afin de réaliser plus rapidement les enregistrements (Figure III.9). Au moins trois marqueurs sont nécessaires pour calculer l'orientation et la position de chaque segment. Le positionnement des marqueurs ainsi que le nombre de marqueurs à disposer est une problématique d'actualité dans le domaine de l'analyse du mouvement humain. Les systèmes d'acquisition évoluent de façon très rapide et pour réaliser aujourd'hui des mesures avec de très nombreux marqueurs (jusqu'à une centaine avec une dizaine de caméras). Des stratégies de définition des repères segmentaires sont ainsi à définir en fonction des marqueurs vus ou perdus par le système d'acquisition. Le Tableau III.1 donne la localisation anatomique des marqueurs [Bou09].

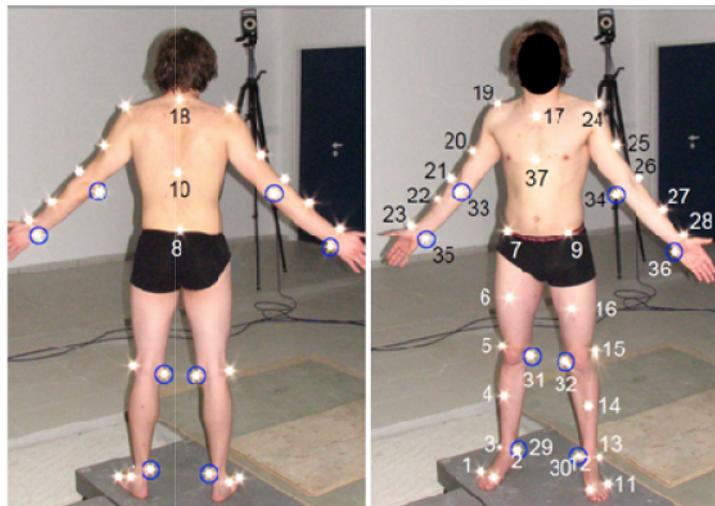


Fig.III.9. Set de 37 marqueurs disposés sur le corps du sujet.

Numéro marqueur	Dénomination	Côté	Segment
1	métatarse pha 5	droit	ped
2	métatarse pha 1	droit	ped
3	maléole externe	droit	tibia-pied
4	face médiale du tibia	droit	tibia
5	épicondyle latéral	droit	tibia-fémur
6	face médiale de la cuisse épine iliaque	droit	fémur
7	antérosupérieure	droit	hanche
8	Sacrum épine iliaque	----	hanche
9	antérosupérieure	gauche	hanche
10	vertèbre T10	---	hanche-tronc
11	métatarse pha 5	gauche	ped
12	métatarse pha 1	gauche	ped
13	maléole externe	gauche	tibia-pied
14	face médiale du tibia	gauche	tibia
15	épicondyle latéral	gauche	tibia-fémur
16	face médiale de la cuisse	gauche	fémur
17	entre art. sternoclaviculaires	---	tronc-clavicule
18	Cervicale C7	---	tronc-tête
19	acromion	droit	clavicule-bras
20	face latérale du bras	droit	bras
21	épicondyle	droit	bras-avant bras
22	face latérale avant bras	droit	avant bras
23	poignet interne	droit	avant bras-main
24	acromion	gauche	clavicule-bras
25	face latérale du bras	gauche	bras
26	épicondyle	gauche	bras-avant bras
27	face latérale avant bras	gauche	avant bras
28	poignet interne	gauche	avant bras-main
29	malléole interne	droit	tibia-pied
30	malléole interne	gauche	tibia-pied
31	épicondyle interne	droit	tibia-fémur
32	épicondyle interne	gauche	tibia-fémur
33	épitrochlée	droit	bras-avant bras
34	épitrochlée	gauche	bras-avant bras
35	poignet externe	droit	avant bras-main
36	poignet externe	gauche	avant bras-main
37	sternum	---	tronc

Tab III.1. Liste des marqueurs disposée sur des points anatomiques superficiels de référence

3.6 Conclusion

Pour conclure sur la capture de mouvement, nous pouvons dire, que c'est la seule technique qui reproduise le réalisme des mouvements d'une personne d'une façon sûre et fiable. Mais elle a un défaut majeur pour l'édition de mouvement, qui est son non adaptabilité aux physionomies différentes, entre le personnage virtuel, et le personnage réel capturé. En effet, la capture d'un mouvement effectuée par une personne de très grande taille, ne peut pas être directement utilisée sur un personnage virtuel plus petit. C'est pourquoi, on doit faire appel à des techniques qui vont adapter le mouvement pour faire en sorte que celui-ci soit utilisable sur n'importe quel acteur virtuel. D'autres approches font la démarche inverse, c'est à dire, au lieu d'enregistrer la réalité, on va essayer de simuler dans son ensemble, cette réalité.

Nous allons présenter dans la seconde partie du mémoire, notre modèle de simulation de comportement de navigation de piétons, résolvant quelques problèmes d'évitement de collisions entre individus.

Contribution

Chapitre IV

Modèle de croisement de flux de piétons

4 Modèle de croisement de flux de piétons

4.1 Introduction

L'étude des circulations de personnes est une problématique nécessitant d'appréhender le comportement humain. Ceci est d'autant plus vrai dans notre domaine d'application, qui est la caractérisation des flux de personnes dans les lieux d'échanges, où ce comportement n'est pas seulement un déplacement dans un environnement mais inclut aussi des interactions avec divers équipements.

L'humain virtuel est l'acteur exclusif de nos simulations. Notre objectif est de proposer un modèle d'humanoïde aussi réaliste que possible d'un point de vue comportemental. Celui-ci doit être doué de certaines aptitudes d'un humain réel, comme l'observation, la mémoire, ou la capacité à se déplacer. Cette dernière doit répondre aux logiques de déplacement caractéristiques des foules, tout en offrant un niveau de contrôle individuel.

L'objectif de notre étude est de simuler problème de croisement de flux de piéton, nous allons utiliser l'animation comportementale pour animer les humains virtuels. Nous assurons l'évitement de collisions entre les individus et les obstacles de l'environnement. Nous allons proposer deux modèles de simulation d'animation de foules. Dans le premier modèle, nous allons représenter l'humain virtuel par de formes géométriques simples, et l'animation est une translation entre les points. Dans le second modèle nous allons représenter l'humain virtuel par des modèles 3D, et nous utilisons de mouvements capturés pour animer l'humain virtuel.

4.2 L'objectif du système

Nous allons décrire l'architecture logicielle pour effectuer la simulation d'animation de foules de piétons. Cette architecture a pour but de simule de croisement du flux de piétons, et d'éviter la collision entre les piétons.

4.3 L'architecture de notre système

Le schéma suivant présente l'architecture globale de notre système :

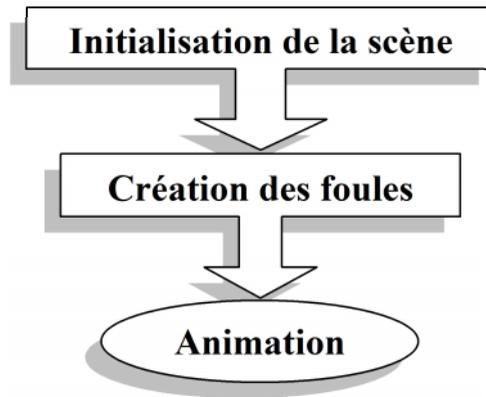


Fig.IV.1. L'architecture du système.

4.3.1 L'animation

Notre système utilise l'animation comportementale. Ce type d'animation base sur trois phases suivant (Figure IV.2) :

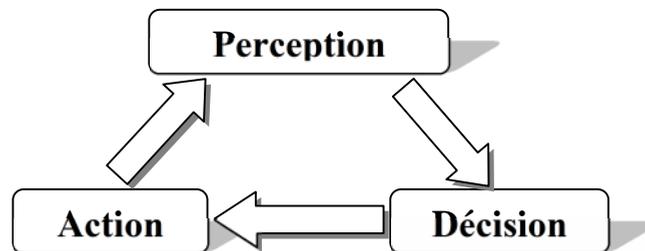


Fig.IV.2. Boucle d'animation comportementale.

- Perception des obstacles et autres individus dans l'environnement, ce qui fournit des informations sur la nature et la position des obstacles.
- En utilise ces informations par le modèle comportemental pour décider de l'action à exécuter, ce qui va produire des paramètres pour une procédure de mouvement.
- Réalisation du mouvement par l'individu, en action finale.

4.3.1.1 Perception

La perception est définie comme la conscience des éléments dans l'environnement qui exprimée à travers des sensations physiques. Elle est réalisée en équipant les acteurs de détecteurs visuels tactiles et auditifs. Ainsi, ils simulent le comportement quotidien humain

telle que la locomotion adressée visuellement, la manipulation d'objets et la réponse aux sons et aux énonciations.

Le sous-système perceptuel les plus importants est le système visuel. Une approche basée sur la vision est idéale pour la modélisation d'une animation comportementale et offre surcroît une approche universelle pour le passage d'information de l'environnement à l'acteur dans le contexte de recherche de chemin, d'évitement d'obstacles.

4.3.1.2 Décision

Module de décision qui se manifeste essentiellement dans les techniques d'évitement d'obstacle et de collisions. Pour cela, nous entamons les étapes d'évitement et les différentes manières pour sa mise en œuvre.

Une étude plus complète sur l'évitement de collision sera détaillée dans l'étape suivant.

4.3.1.3 Action

C'est la dernière phase dans le cycle d'animation comportementale, l'action est présentée par le déplacement des acteurs et les groupes dans la scène.

Le fait de déplacer un objet vers son but est très simple, car on peut exprimer le mouvement de l'acteur par une droite vers son but,

4.4 Structure de notre système

Nous allons discuter dans cette partie les divers aspects de notre système.

Dans un premier temps, nous présentons la manière dont l'environnement virtuel est créé ainsi que la façon dont la scène est discrétisée sous la forme de cellules dans le but d'en former un graphe.

En second lieu, nous présentons notre algorithme modifié de l'algorithme A* qui est utilisé dans l'environnement dynamique. Cet algorithme est employé par chaque individu à l'intérieur de sa région visible. Nous concentrons, alors, notre discussion sur les types des situations d'évitement de collision. Les individus doivent éviter des collisions entre eux ainsi qu'avec l'environnement.

Partant de la position courante d'un piéton et d'une destination cible, l'algorithme exploite la carte topologique au niveau supérieur du modèle de l'environnement. En appliquant des algorithmes de recherche de chemin lié à chaque région, le piéton peut

projeter un chemin de l'endroit courant à la frontière entre la région courante et la prochaine. Le processus est répété dans la prochaine région, et ainsi de suite, jusqu'à ce qu'il se termine au niveau du but de destination.

La structure générale du système comporte plusieurs modules qui permettent de simuler notre système, le schéma suivant montre le fonctionnement de notre système :

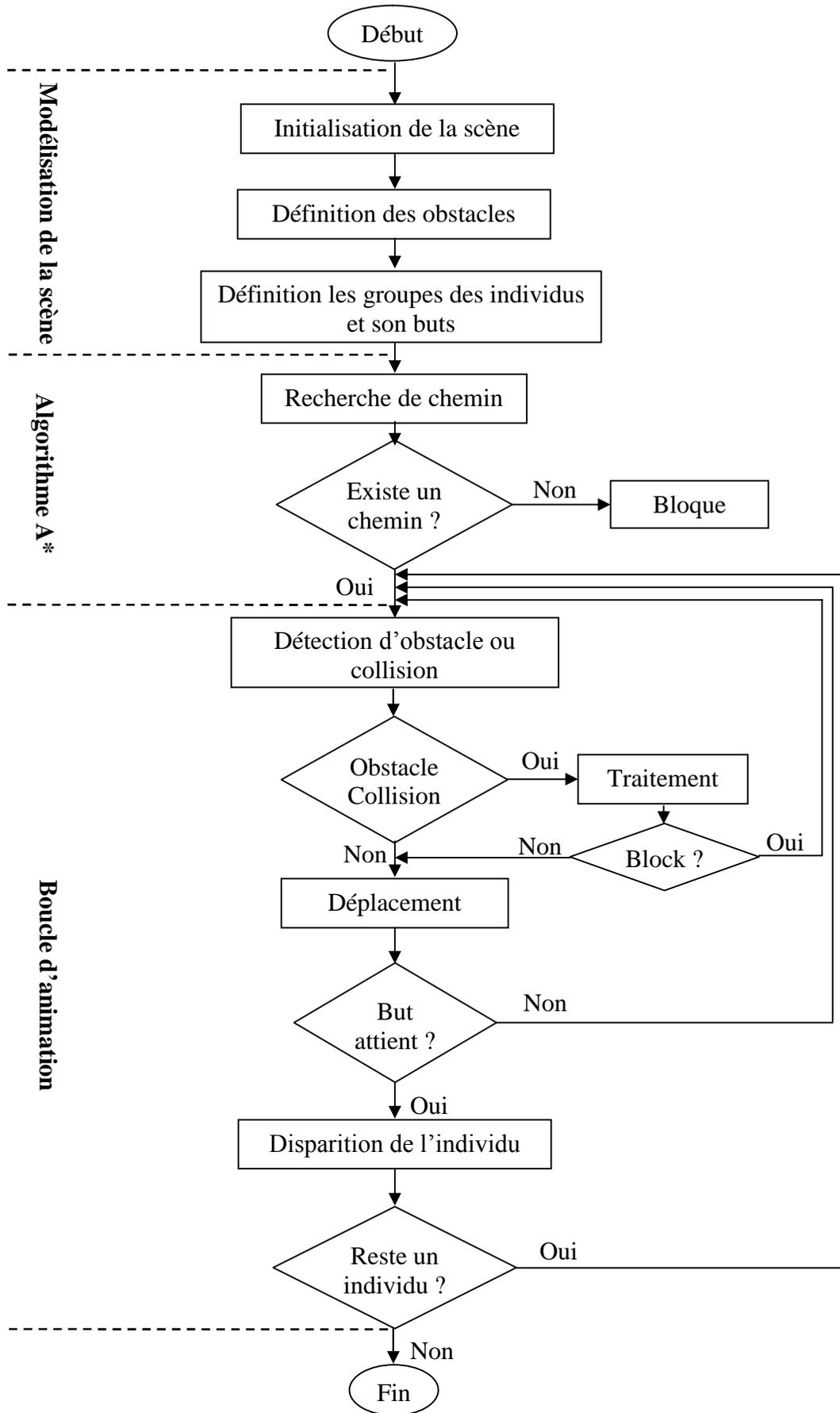


Fig.IV.3. Organigramme du système.

4.4.1 Modélisation de la scène

Le but de notre système est de simuler le mouvement des foules dans un environnement dynamique. La scène est ainsi formée par un ensemble d'obstacles, et un ensemble de groupes des individus et ses buts.

- Les obstacles : il y a deux type ;
 - ✓ Des objets simple : l'utilisateur pues choisir le nombre et la position, ce type devant être évite par les individus.
 - ✓ Les murs qui est défini le contour de la scène.
- La foule : est un ensemble de groupes, chaque groupe est constitué par un certain nombre d'individus. Nous allons présenter les individus au le première modèle par de forme géométrique simple et dans le deuxième modèle par des humains virtuel 3D.

L'utilisateur du système peut modifier la foule par initialisation :

- ✓ Nombre de groupes ;
- ✓ Nombre des individus dans un groupe ;
- ✓ La position de groupe.
- Les Buts : l'utilisateur peut choisir la position de but de chaque groupe, chacun à une position et un couleur différente aux autres buts.

4.4.2 Notre modèle de foule

Nous avons défini notre foule comme étant un ensemble de groupe dont chacun est composé par des individus. Notre modèle distribue les comportements de foule aux groupes et puis aux individus, chaque individu a un ensemble de comportements et chaque groupe des individus a un ensemble de comportements (Figure IV.4).

- Comportement d'individu :
 - ✓ L'évitement de collision entre les autres individus ;
 - ✓ L'évitement d'obstacle ;
 - ✓ La recherche de bus.
- Comportement de groupe :
 - ✓ Recherche le but ;

- ✓ La cohésion entre les individus d'un groupe.

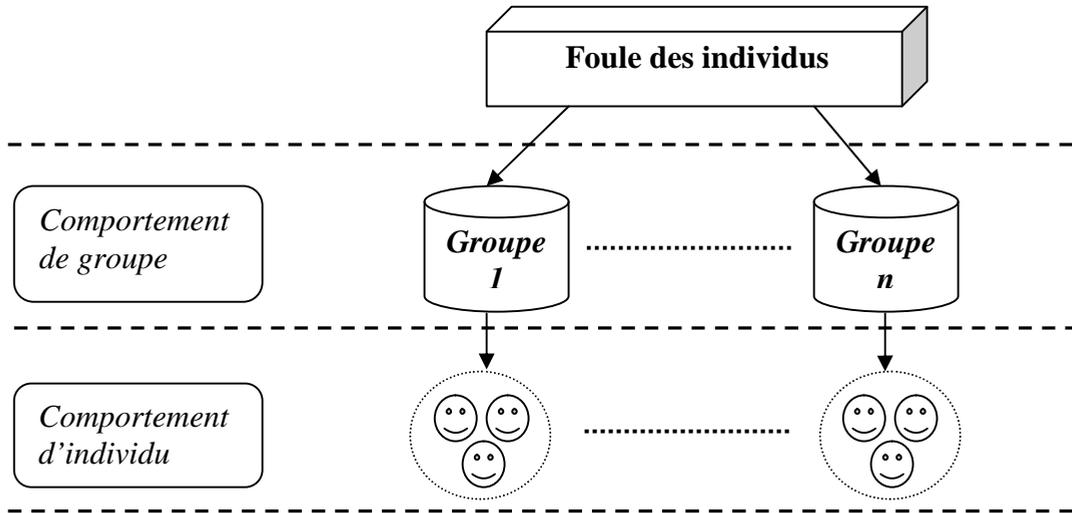


Fig.IV.4. Modèle de foule.

4.4.3 Discrétisation la scène

Nous allons présenter notre environnement par la représentation approximative. Méthode approximative est sans doute les plus utilisées en animation comportementale, du fait facilité de mise en œuvre. Ces représentations vont décrire l'espace libre de navigation avec des formes géométriques simples telles que carré ou rectangle en **2D** ou sous la forme de cubes ou parallélépipèdes en **3D**.

Notre approche suit une stratégie de décomposition de cellules (grilles uniformes). Elle a été employé dans le cadre de l'algorithme A* modifié pour déterminer des chemins pour un ensemble d'acteurs. La première tâche se cet algorithme est la discrétisation de la scène en cellules. Chaque acteur occupe une cellule alors qu'un obstacle peut occuper plusieurs cellules (Figure IV.5).

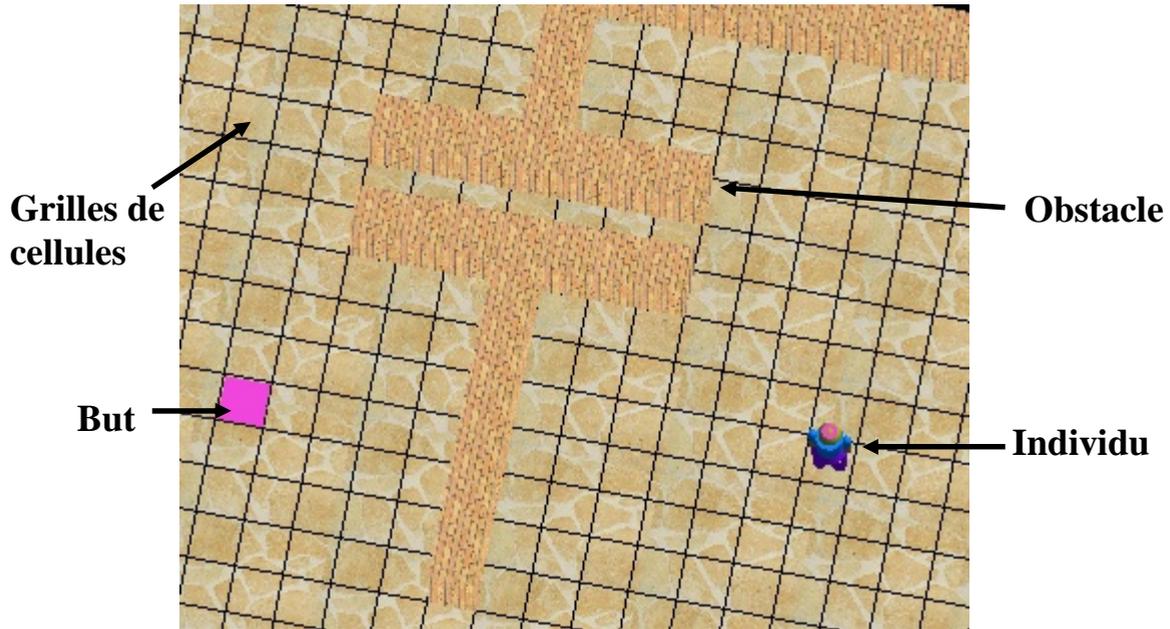


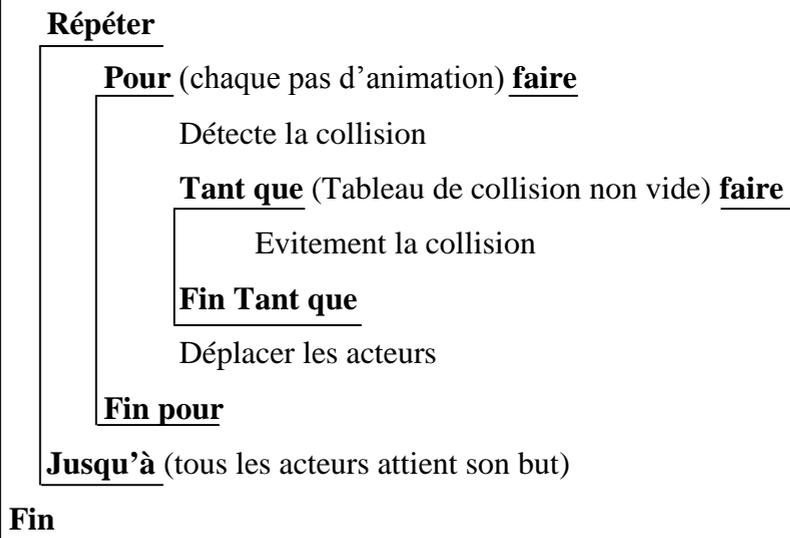
Fig.IV.5. Discretisation la scène.

4.4.4 Recherche de chemin

Rechercher de chemin consiste à trouver un chemin optimal (généralement le plus court) entre un point de départ et un point de destination dans un environnement virtuel tout en évitant les différents obstacles qui peuvent se présenter. Traditionnellement, la planification de chemin a été résolue en utilisant un algorithme heuristique de recherche tel que le très célèbre algorithme A^* [BPW93, BTD98], algorithme directement couplé à l'animation d'acteurs de bas niveau.

4.4.5 Boucle d'animation

Les acteurs autonomes sont capables de projeter automatiquement de chemins autour des obstacles statiques et dynamiques dans l'environnement. L'algorithme A^* tient compte seulement des obstacles statiques, le problème la plus importante est comment un acteur évite les autres acteurs, pour traiter ce problème on utilise un algorithme qui détecte la collision entre les acteurs. Ainsi, à chaque pas de l'animation d'un acteur et avant d'obtenir la prochaine position, chaque acteur doit traiter la prévision de collision pour éviter la collision avec les autres acteurs d'environnement.

Boucle d'animation**4.4.5.1 La prévision de collision**

La prévision de collision est la perception des autres acteurs en mouvement et qui sert par la suite à l'évitement de collisions. Le principe de cette méthode est de tester à chaque pas d'animation (un pas en avant) s'il existe un acteur qui occupe le même endroit. C'est-à-dire : est-ce que le point destination de l'acteur en mouvement est occupé par un autre acteur, dans le cas positif alors une collision est détectée.

La figure suivant illustre la mise en œuvre de prévision de collision :

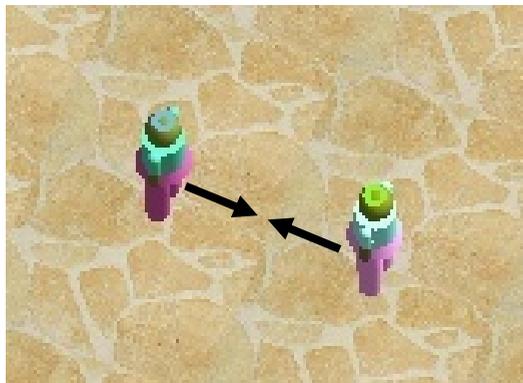
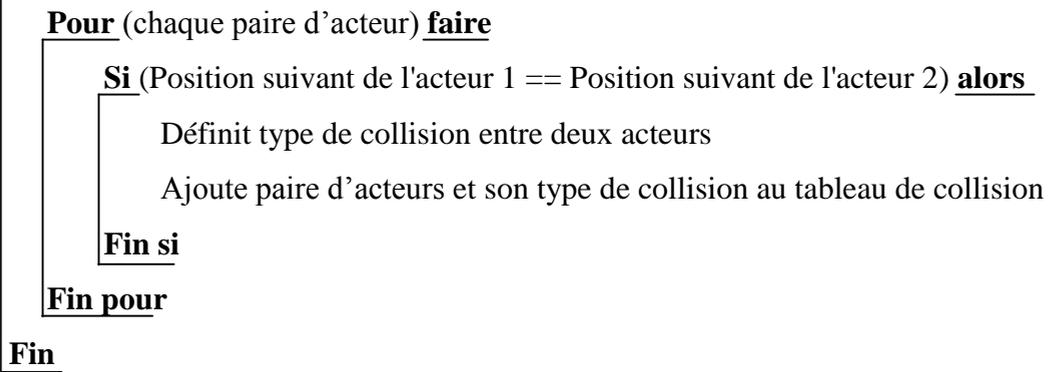


Fig.IV.6. Prévision de collision.

Détecte la collision



Dans la réalité, il y a trois types de collisions possibles (Figure IV.7) :

- **Collision face à face** : se produit si les agents se déplacent l'un vers l'autre ;
- **Collision en arrière** : se produit si deux acteurs marchent dans la même direction et l'un derrière à l'autre ;
- **Collision de côte** : se produit si les deux agents marchent au la même cellule au même temps.

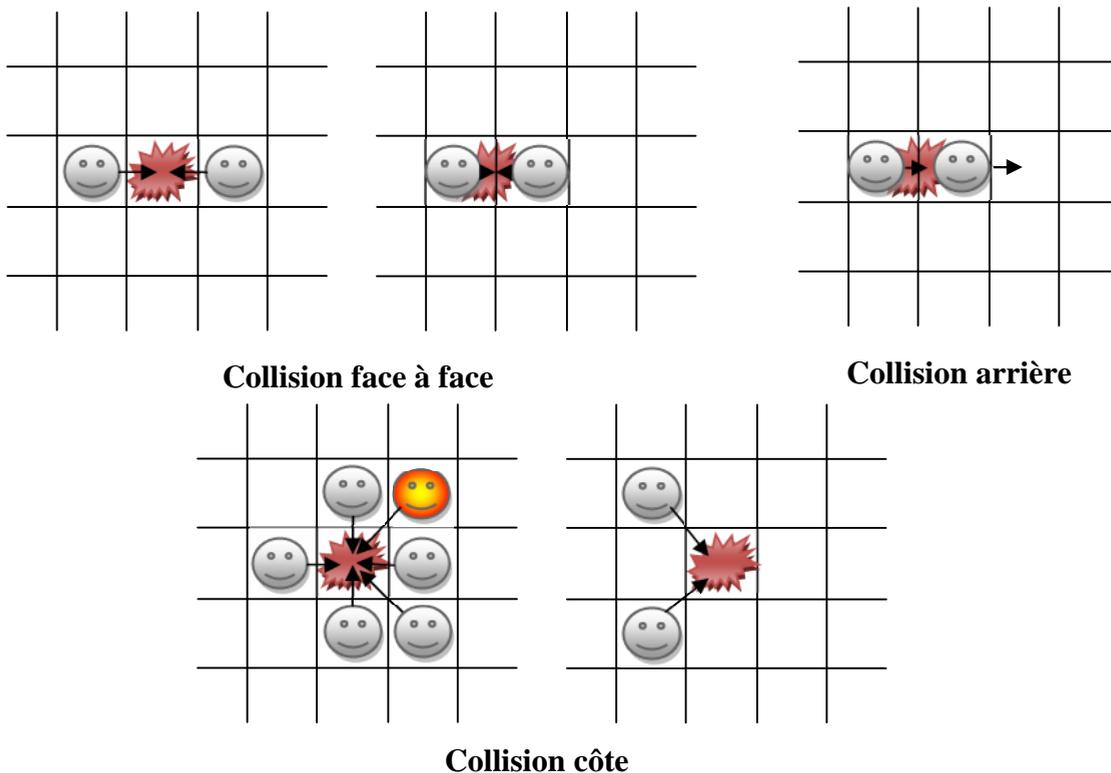


Fig.IV.7. Les trois types de collisions.

4.4.5.2 Evitement de collision

- *Evitement d'obstacle par un acteur :*

Le principe évitement d'obstacle est très simple : lorsqu'un obstacle est détecté par un acteur, ce dernier doit décider dans quelle direction il doit l'éviter en droite ou en gauche.

- *Evitement d'obstacle par un groupe d'acteurs :*

Etant donné que chaque groupe est guidé par un chef, et ce chef est un acteur alors le même principe appliqué à l'acteur est appliqué au groupe. Cependant un problème se pose à ce niveau :

Lorsqu'un obstacle est détecté par un groupe d'acteurs, le leader va choisir une direction, il va prendre son chemin, comment les autres acteurs vont éviter cet obstacle ?

Plusieurs méthodes existent, on suggère que l'intégralité du groupe va suivre le leader dans sa direction en évitant l'obstacle. Une autre voie est à envisager, elle est édictée par le réalisme exigé dans le comportement du groupe. Celui-ci, peut, par exemple se diviser en sous-groupes, un groupe suit le leader et l'autre prend l'autre chemin et se réunissent après l'évitement.

Dans notre modèle, nous avons décidé de suivre la seconde stratégie, car elle est simple à élaborer et elle donne à notre scène un réalisme plus tangible surtout dans le cas d'évitement d'un obstacle ou d'évitement d'une collision.



Fig.IV.8. Evitement d'obstacle par un groupe.

- *Evitement de collision entre acteurs*

L'évitement de collision entre des acteurs d'une scène dynamique peut impliquer quelques problèmes qui apparaissent seulement quand nous traitons une scène contenant

beaucoup d'acteurs. Une méthode pour éviter la collision entre acteurs peut être non efficace lorsque leur nombre est très élevé. Il existe plusieurs contraintes et variables dans un environnement complexe, un environnement complexe étant composé par des obstacles fixes et des acteurs, par de petites régions de marche ainsi qu'un nombre important d'acteurs.

Cependant, si la structure du groupe doit être préservée, ceci ajoute un autre paramètre dans la complexité de l'évitement de collision de la foule.

4.4.5.2.1 Collision face à face

Dans la vie réelle, les gens préfèrent passer à côté avec la moindre déviation de leur chemin. La plupart des observations prouvent que l'acteur a trois manières différentes d'éviter la collision :

- Changement de direction seulement ;
- Changement de vitesse seulement ;
- Changement de direction et de vitesse.

Si on n'a trouvé aucun comportement qui évitera la collision, l'acteur doit s'arrêter. Ceci permettra à l'autre acteur impliqué dans la collision pour éviter le sujet, qui peut alors reprendre la marche.

4.4.5.2.2 Collision en arrière

Une collision en arrière est une collision où l'acteur cible de la collision est devant l'acteur, mais l'acteur se déplace plus rapidement que l'acteur cible, ou l'acteur se déplace et l'acteur cible bloque, ainsi l'acteur provoquera une collision avec l'arrière de l'acteur cible. Pour traiter ce type de situation, l'acteur possède deux choix :

- Ralentir à la même vitesse que l'acteur cible et marcher derrière lui ;
- Dépasser l'agent cible en choisissant la direction appropriée.

4.4.5.2.3 Collision de côté

Ce type de collision est traité d'une manière semblable à la collision face à face.

- Changement de direction seulement ;
- Bloque l'acteur cible.

Chaque acteur possède un but et essaye de l'atteindre en suivant son chemin initial. Après avoir évité la collision, l'acteur devrait retourner à son chemin d'une manière fine afin de sembler normal ou changer complètement le chemin.

4.4.5.3 Comportements d'évitement

Comme nous l'avons vu plus haut, chaque action d'évitement de collision a besoin de différents comportements et de traitement différent. La liste de comportements qui peuvent être employés dans l'évitement de collisions est :

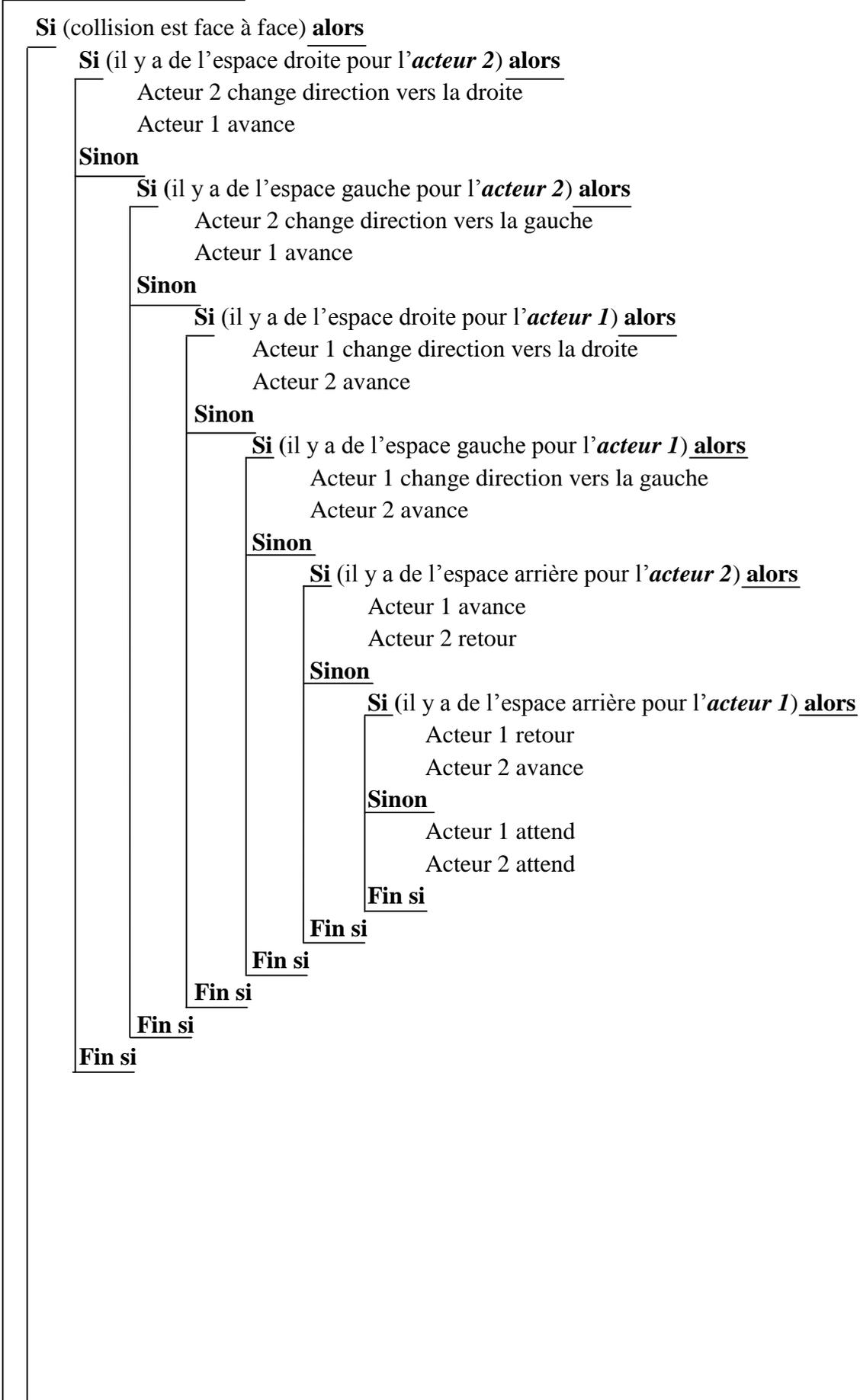
- Avancement ;
- Changement de directions (gauche ou droite) ;
- Attente ;
- Retour.

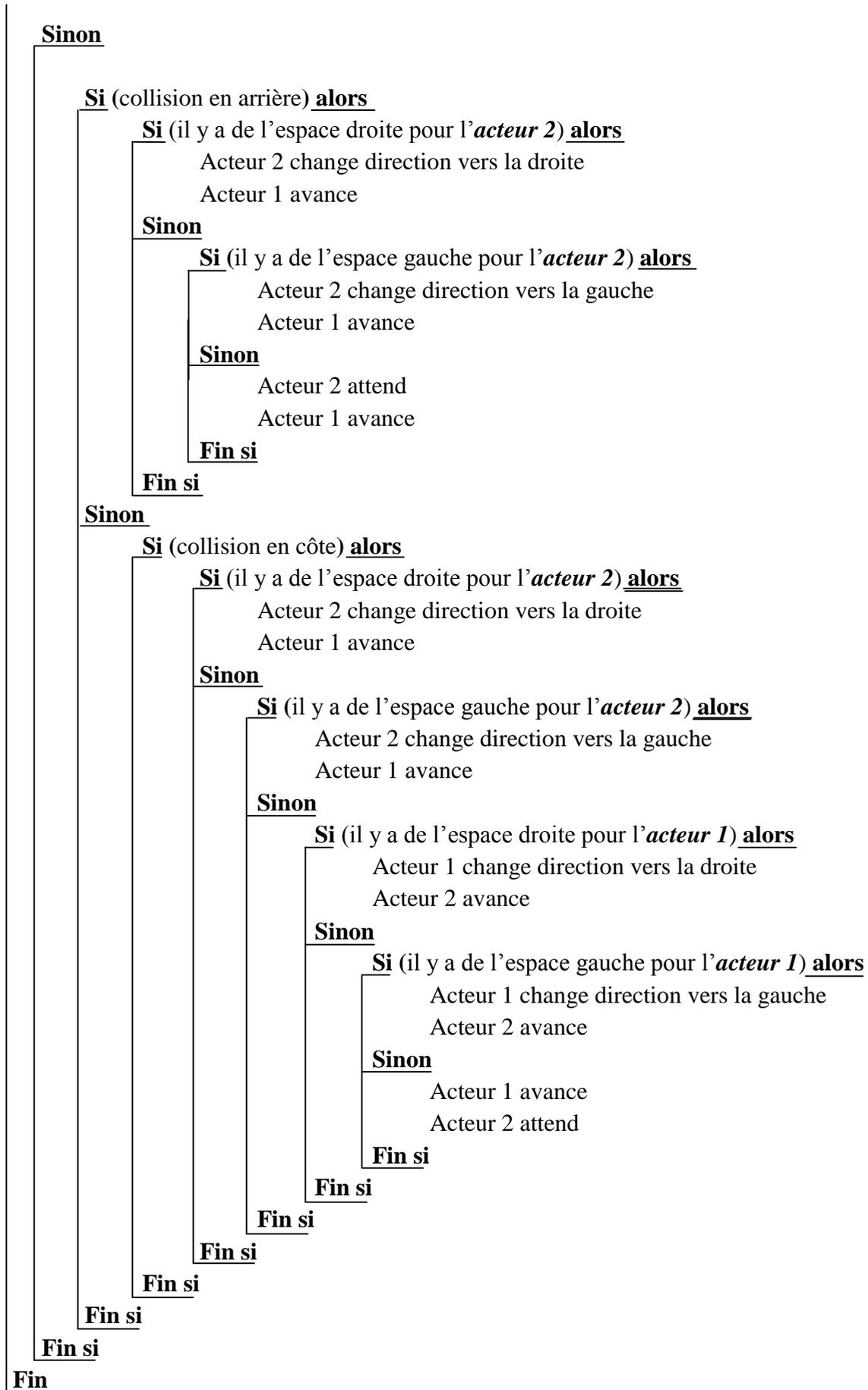
Le processus d'évitement de collision est semblable pour toutes ces situations en appliquant l'algorithme ci-dessus pour chaque paire d'agents dans les collisions. Le tableau suivant présente tous le cas de traitement de l'évitement de collision :

Type de collision	Acteur 1	Acteur 2
Face à face	Avancement	droite
	Avancement	gauche
	Droit	avancement
	Gauche	avancement
	Avancement	retour
	Retour	avancement
Arrière	Avancement	dépassement par la droite
	Avancement	dépassement par la gauche
	Avancement	attente
Côte	Avancement	droit
	Avancement	gauche
	Droit	avancement
	Gauche	avancement
	Avancement	attente
	Attente	avancement

Tab IV.1. Tableau de traitement de l'évitement de collision

Traitement la collision





4.4.5.3.1 *Croisement de groupes*

Cette situation est très souvent rencontrée dans un environnement peuplé, lorsque deux groupes d'acteurs se déplaçant dans des directions opposées et tentent de s'éviter mutuellement. Dans la réalité, ils forment des lignes de piétons se déplaçant dans la même direction (Figure IV.9).

En cas de collision, l'algorithme d'évitement de collision acteur par acteur est appliqué. Les comportements d'évitement de collision utilisés sont : l'avancement, le changement de directions, l'attente et le retour. Le dernier comportement est employé dans le cas de blocage, quand il n'y a plus aucun espace permettant à l'acteur d'avancer.



Fig.IV.9. Croisement de groupes.

4.4.5.3.2 *Goulots d'étranglement*

Les goulots d'étranglement des piétons peuvent être rencontrés dans des localisations telles que l'entrée de couloirs, les escaliers, les passages souterrains. Dans la réalité, la priorité est accordée au plus proche du centre du goulot d'étranglement (Figure IV.10). L'algorithme A^* résout directement cette situation et en cas de collision, l'algorithme d'évitement de collision acteur par acteur est appliqué.

Les comportements d'évitement de collision utilisés sont : l'avancement, le changement de directions et l'attente.



Fig.IV.10. Goulot d'étranglement.

4.4.5.3 *Alignement dans deux directions*

Les piétons forment des files d'attente devant les sorties, devant les portes d'un métro,... Quand un véhicule arrive, les piétons attendent que les passagers de l'intérieur descendent ou sortent du véhicule, c'est uniquement après cette étape que les autres passagers commencent à monter etc.... (Figure IV.11).



Fig.IV.11. Alignement dans deux.

Dans notre système, une priorité est accordée aux acteurs d'une direction, les autres attendent jusqu'à ce qu'il y'aurait un espace libre de déplacement. En cas de collision, l'algorithme d'évitement de collision acteur par acteur est appliqué. Les comportements d'évitement de collision utilisés sont : l'avancée, le changement de directions, l'attente et le retour. Le dernier comportement est employé dans le cas de blocage, quand il n'y a aucun espace permettant d'avancer.

Les acteurs qui sont moins prioritaires doivent choisir entre deux possibilités :

- Reculer pour laisser le passage aux autres acteurs.
- Appliquer l'algorithme **A*** pour trouver un nouveau chemin.

4.4.5.3.4 Cas des passages étroits

Cette situation est observée à l'intérieur des couloirs, sur les trottoirs, ou corridor de piétons. D'une façon générale les piétons forment des lignes pour chaque groupe. Le choix de la direction est pris à partir des comportements sociologiques des piétons (droite ou gauche). L'algorithme A* fournit le chemin à suivre, le changement de la direction permettant le dépassement est choisit en cas de collision arrière (Figure IV.12).



Fig.IV.12. Passage étroit.

4.4.5.4 Formation de files

La formation de files est l'exemple de phénomènes collectifs le plus étudié dans la littérature. La formation de files est une forme d'organisation collective qui apparaît lorsque deux flux de piétons se déplacent en sens inverses dans une rue ou un couloir. Deux mécanismes différents ont été proposés pour expliquer ce phénomène. Le premier met en jeu un comportement de suivi : lorsqu'un piéton se déplace dans une foule, celui-ci aurait tendance à suivre les personnes se déplaçant dans le même sens que lui. En répétant des manœuvres d'évitement, les premiers piétons du flux auraient donc tendance à entraîner les piétons suivants du même côté qu'eux, formant des groupes d'individus qui se suivent [KS02, NSK+06, RAB+08]. Le second mécanisme proposé ne fait appel qu'à un comportement d'évitement. Dans ce cas, un piéton ne chercherait pas directement à suivre les individus qui se déplacent dans le même sens que lui, mais il éviterait de manière plus marquée ceux qui se déplacent dans le sens opposé [Hel95, YHD+05]. Par évitements successifs, un piéton finirait par se retrouver, par hasard, derrière un autre se déplaçant dans le même sens, ce qui constitue une position de marche ne nécessitant plus d'évitement. Au fur et à mesure que des files de piétons se forment, il devient de plus en plus difficile de marcher ailleurs que

dans une file. Ici le comportement de suivi ne résulte pas de la volonté explicite du piéton, mais d'une conséquence de son comportement d'évitement (Figure IV.13).

Nous avons utilisé dans notre modèle le deuxième mécanisme pour résoudre le problème de croisement entre deux flux de piéton.



Illustration du phénomène de formation de files [Meh10].

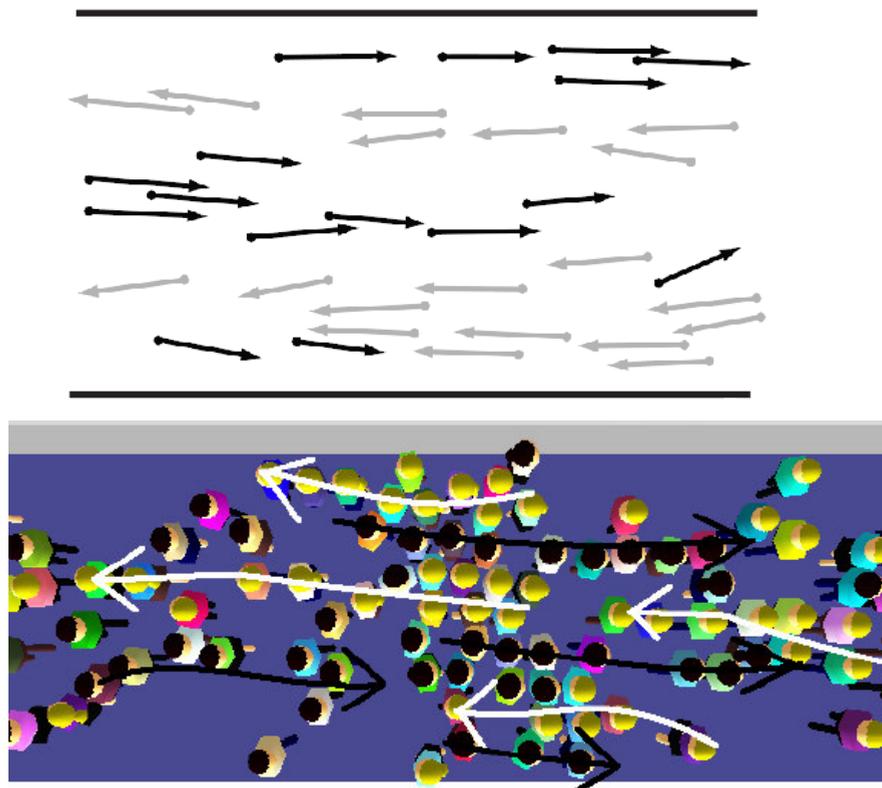


Fig.IV.13. Lane formation in pedestrians.

4.4.5.5 Recalcul du chemin

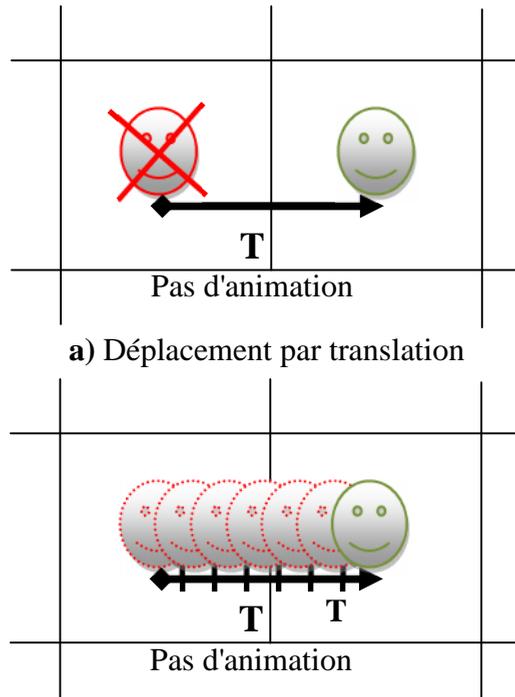
Le principe de recalcul du chemin est de recherche un nouveau chemin pour chaque individu qui est éloigné à son chemin initiale, et pour rechercher le nouvelle chemin nous allons utiliser l'algorithme A^* . Alors les individus qui sont utilisé cette procédure son les individus qui fait la procédure d'évitement de collision entre des autres individus plusieurs fois successive. Nous avons proposé une condition pour utiliser cette procédure, si l'individu éloigné à son chemin initiale des deux cellules (c-à-d. la distance entre le point actuelle et le point suivant du chemin et égale ou supérieur à deux cellules) donc cet individu exécute la procédure de recalcul du chemin.

4.4.5.6 La procédure de déplacement les acteurs

Après la procédure d'évitement de collision en va à la dernière procédure de boucle d'animation, la procédure de déplacement les acteurs. Le principe de cette procédure est pour chaque acteur si en peut déplace (n'est pas bloqué par un autre acteur) alors l'acteur déplace à la cellule actuelle vers la cellule suivant de se chemin.

Dans le première modèle d'application l'opération de déplacement d'une cellule vers cellule suivant modélise par une translation d'un point vers un autre point, c'est-à-dire, on efface l'acteur depuis le premier point et on recrée de deuxième point. La procédure de déplacement de première modèle donne une animation plus loin à l'animation réelle, par ce qu'est un processus discontinu (Figure IV.14.a).

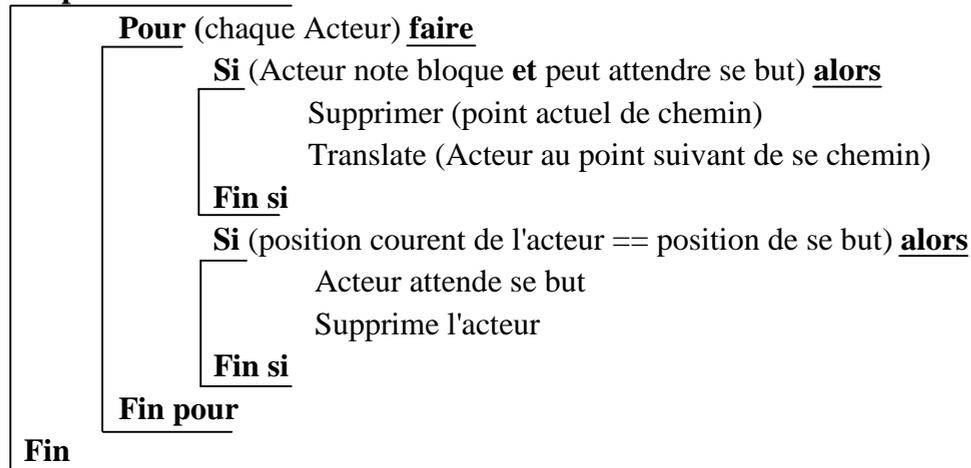
Pour obtenir une animation plus réaliste, nous allons produire un autre modèle de notre application. Dans ce modèle nous allons concentrer d'intégré de mouvement réelle (mouvement capture) à la procédure de déplacement. C'est-à-dire, on divise le pas d'animation au plusieurs sous pas, et pour chaque sous pas on associe par un intervalle de mouvement capture (Figure IV.14.b).



b) Déplacement par capture de mouvement

Fig.IV.14. Procédure de déplacement d'acteur

Déplacer les acteurs



4.5 Conclusion

Nous avons décrit dans ce chapitre un modèle pour simuler le croisement de flux de piétons. Nous avons concentré notre effort sur la recherche de chemin ainsi que sur les techniques d'évitement de collisions entre les piétons dans un environnement dynamique. Ces deux techniques rendent les simulations plus cohérentes et plus réalistes. Plus de plus de réalisme, nous allons remplacer la génération automatique de mouvement par des mouvements capturées.

Chapitre V

Implémentation et résultats

5 Implémentation et résultats

5.1 Introduction

Nous avons proposé dans la partie précédente deux modèles de simulation de croisement de flux de piéton, un modèle sans capture de mouvement et l'autre avec capture de mouvement. Néanmoins, une dernière étape est nécessaire afin de concrétiser ce travail. On effectue, ces modèles ne peuvent être utilisés sans être intégrés dans deux applications. Du nom de "**SimCroi V1**" et "**SimCroi V2**", ces applications concrétisent le travail présenté et démontrent sa faisabilité.

Nous commencerons, dans ce chapitre 5, par décrire l'application qui intègre le modèle proposé. Nous aborderons ensuite,

5.2 SimCroi

SimCroi est l'acronyme de **Simulation Croisement**. C'est un nouveau système qui permet de simuler le croisement de flux d'humains virtuels, on assure l'évitement de collision aux obstacles dans l'environnement et entre les autres humains virtuels.

Dans la première version de notre application, nous avons modélisé les individus par une forme géométrique simple, et l'animation modélise par une translation d'un point à un autre point (Figure V.1).

Dans la deuxième version, nous avons modélisé les individus par un modèle de l'humanoïde 3D, et l'animation modélise par une capture de mouvement pour obtenir une animation plus réaliste (Figure V.2).

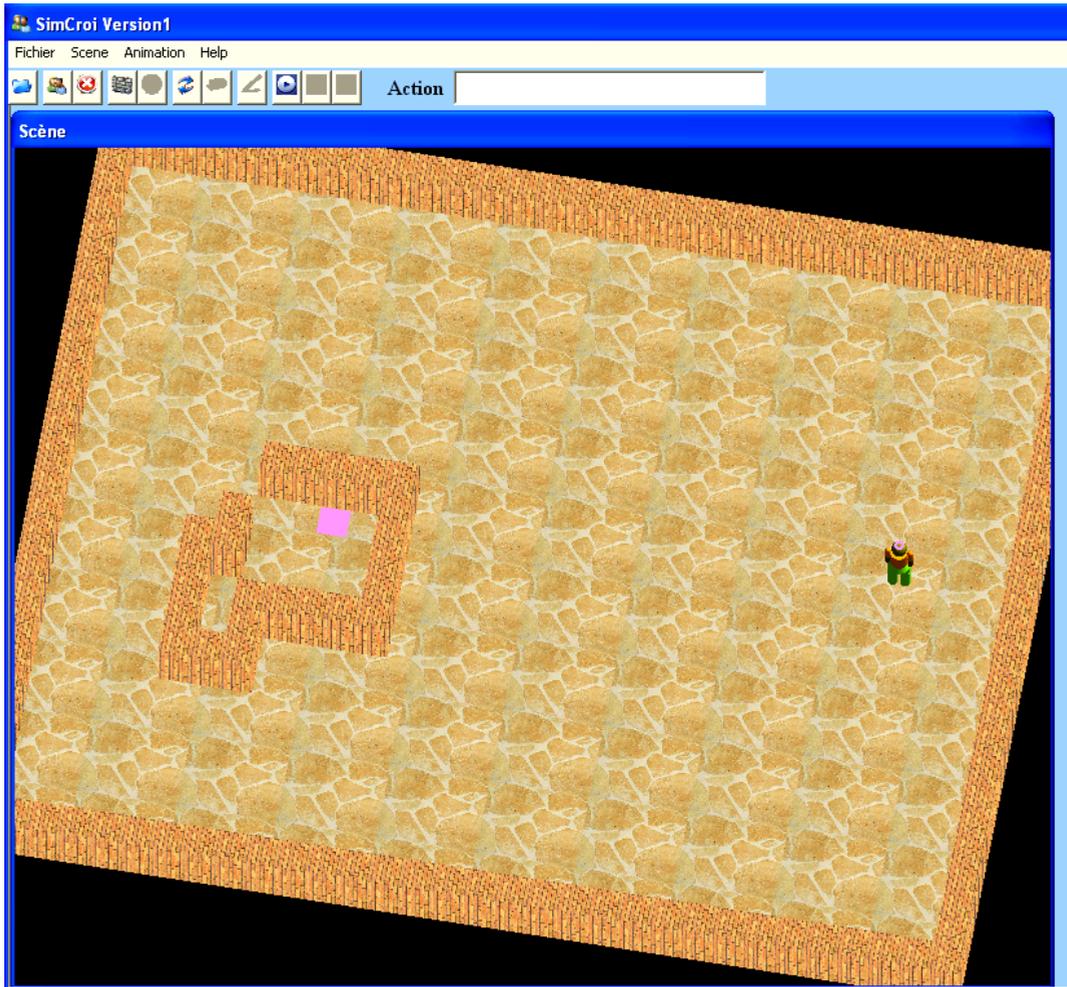


Fig.V.1. L'interface de première version d'application.

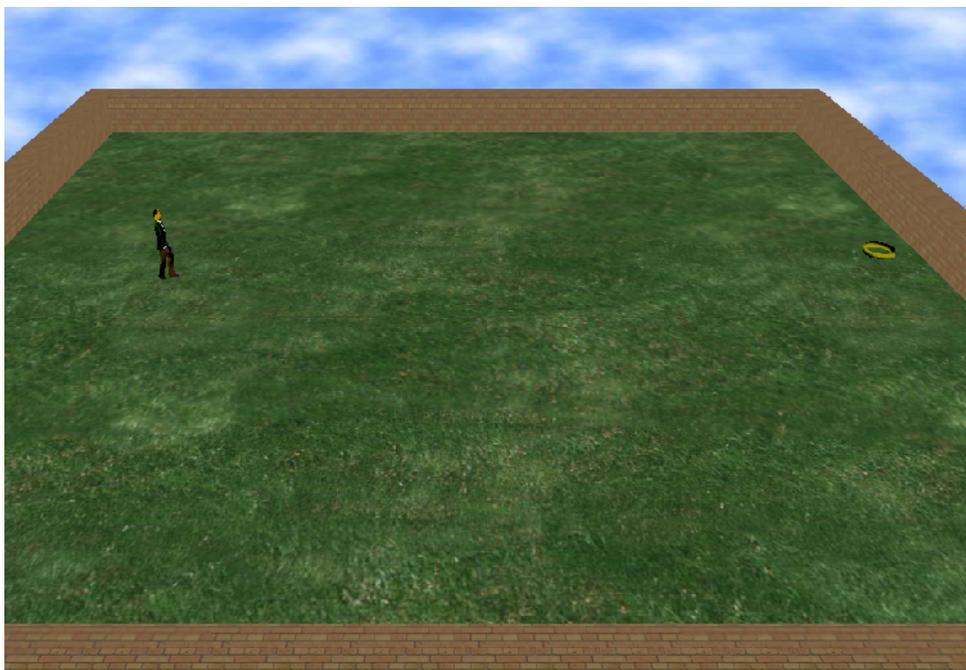


Fig.V.2. L'interface de deuxième version d'application.

5.3 L'environnement de développement

C'est un logiciel tridimensionnel, qui fonctionne sur une machine à base d'un processeur Intel Pentium 4, vitesse 3.2 Ghz, RAM 1,12 Go et carte graphique intégrée de 128Mo, utilisant le système d'exploitation Windows XP.

La première version d'application est implémente en C++ Builder 6, utilisant la bibliothèque d'OpenGL pour la modélisation de scène.

La deuxième version d'application est implémente en Visual Studio 10, utilisant la plateforme OGRE 3D pour la modélisation de scène, et 3DS Max pour modélise les individus.

5.3.1 3DS MAX

Anciennement connu sous le nom "3D Studio," 3DS Max est un programme de modélisation 3D, d'animation et de rendu de la division Media and Entertainment d'Autodesk, Inc, San Rafael, CA¹. Largement utilisé dans les domaines des jeux interactifs, des effets visuels pour les films et des modèles de conception industrielle.

L'humain virtuel nécessite un mécanisme gérant sa locomotion. Nous utilisons pour cela des mouvements capturés, que nous avons choisis à partir d'une base de données de mouvements capturés fournie par Carnegie-Mellon Graphics Lab². Pour adapter ces mouvements capturés à la morphologie de l'humain virtuel ainsi que de les mélanger et les ajuster, nous avons utilisé 3DS Max. Notre choix s'est tourné vers ce modèle car il produit des mouvements valides vis-à-vis des lois biomécaniques, qu'il permet un contrôle fin des déplacements (Figure V.3).

¹ www.discreet.com

² <http://mocap.cs.cmu.edu>

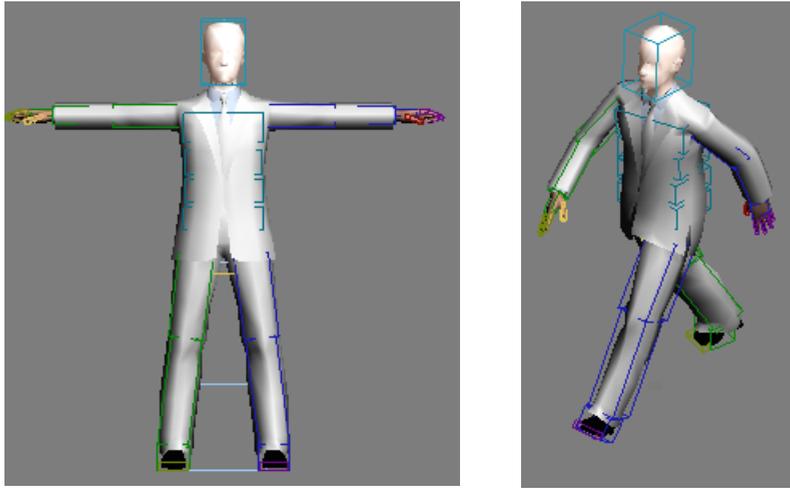


Fig.V.3. Modèle de l'humanoïde 3D.

5.3.2 Ogre 3D

OGRE signifie Object-Oriented Graphics Rendering Engine, soit Moteur de Rendu Graphique Orienté-objet en français.

OGRE est un moteur 3D open-source multi-plate forme (Linux, Win32, MacOSX, iPhone) orienté scène qui permet à partir d'objets à facettes de réaliser un environnement tridimensionnel qui sera perçu par un rendu bidimensionnel au travers d'une ou plusieurs caméra virtuelle.

OGRE est entièrement écrit en C++ et de part sa conception, il permet aussi bien d'utiliser DirectX qu'OpenGL ! Il intègre les dernières possibilités de ces deux bibliothèques et est en constante évolution. Par contre, OGRE est SEULEMENT un moteur 3D, il faut rajouter des modules pour gérer le son, la physique, le réseau...

Historique

Le projet Ogre est né en 1999 de l'esprit d'un certain *Steve Streeting*, programmeur connu sous le pseudonyme de *sinbad* et qui a eu l'idée de construire un moteur 3D indépendant de la plateforme et de l'API utilisée (Direct3D ou OpenGL). Le projet est rapidement créé sur *SourceForge* au début de l'année 2000 et grossira progressivement sous la supervision de *sinbad*. Celui-ci se retire en Avril 2010 pour des raisons de santé mais reste un contributeur actif.

La version 1.0.0 du moteur sort en février 2005 sous le nom de code *AzaThot*. Les versions 1.x se succèdent au fil des ans et de façon régulière, toujours avec un nom de code désignant une créature mystique.

En avril 2010 est sortie la version 1.7.1 dénommée *Cthugha*, qui en profite pour changer de licence et passer en MIT en lieu et place de la LGPL utilisée jusqu'alors.

Avantages

- Multi plateforme : OGRE fonctionne aussi bien sous Windows, Linux ou MAC.
- Compatible OpenGL et DirectX.
- Très performant : il gère les effets graphiques les plus récents et affiche un Frame Rate très satisfaisant.
- Offre une architecture "professionnelle" : son design est tel qu'il peut parfaitement s'intégrer à des applications graphiques professionnelles. Le travail en équipe ne pose pas de problème grâce à la POO et sa modularité offre de grandes possibilités de personnalisations.
- Très bien documenté : OGRE offre une API de très bonne qualité, un manuel complet et des tutoriaux tous très bien fait. Le forum est aussi très actif et peu beaucoup aider.

Inconvénients :

- Difficile à prendre en main : en effet même si il est très bien documenté et bien pensé, OGRE reste un moteur très (trop ?) complet et par conséquent complexe, tant dans le mode de fonctionnement que dans l'utilisation. Il vous faudra de nombreuses heures avant de le prendre en main. Mais une fois cet effort effectué, il sera votre meilleur outil !
- Trop 'fouillis' : OGRE permet de faire tellement de choses qu'il comporte beaucoup de dossiers et de fichiers qui paraissent « ésotériques » lors des premières utilisations. Mais au fur et à mesure, on se rend compte à quel point cette organisation est logique et performante. Tout est question de temps...
- Optimisé pour VC Toolkit et Visual Studio : c'est pour moi un inconvénient qu'un moteur Open Source tel que OGRE soit optimisé pour être utilisé par des outils propriétaire de Microsoft. La plupart des plugins et outils sont tous développés pour VS. La configuration avec d'autre IDE ou sur d'autre plateformes que Windows est donc plus complexe et nécessite des connaissances supplémentaires. Mais ceci tend à changer avec le temps et j'espère que l'open source sera de plus en plus représentée dans la programmation.

5.3.2.1 L'entité

Une entité, c'est la représentation dans la scène d'un modèle 3D, aussi appelé *mesh* (mesh signifie maillage en anglais).

Le mesh est l'ensemble des polygones élémentaires (des quadrilatères et des triangles, le plus souvent) que l'on crée dans un logiciel de modélisation, comme 3DS Max ou Blender ou Maya, et qui constituent votre modèle 3D complet. Tous ces polygones sont reliés entre eux par leurs sommets. Plus il y a de sommets, et donc de polygones, plus le mesh est précis. Généralement, le mesh possède aussi une ou plusieurs textures (une image plaquée sur les polygones) qui lui donnent un aspect plus réaliste qu'une couleur unie lors du rendu. Enfin, pour certains modèles comme les personnages, le mesh possède un squelette qui permet de créer des animations pour son mouvement.

Les mesh sont gérés par le gestionnaire **MeshManager**. Ils sont généralement chargés à partir du format de fichier spécifique à OGRE : le format ".mesh". Ces fichiers sont créés grâce aux outils d'exportation (section 3.2.3) disponibles dans la plupart des logiciels 3D, comme ils peuvent être créés manuellement en appelant la méthode **MeshManager::createManual**.

5.3.2.2 Le SceneManager

Afin de pouvoir gérer l'ensemble des objets de notre scène tout au long du déroulement du programme, OGRE introduit la classe de gestion de la scène "*SceneManager*".

Concrètement, le SceneManager s'occupe de garder une trace de tous les modèles, lumières, caméra et autres objets que peut contenir la scène. C'est à lui que revient la tâche de créer tous ces objets et de nous permettre ensuite d'y accéder. Par conséquent, l'insertion d'objets dans la scène passe toujours par lui (ou par l'un des éléments déjà insérés) et par les méthodes qu'il propose pour cela.

Il en existe diverses variantes en fonction de la scène que l'on veut réaliser ; selon que celle-ci sera en intérieur ou en extérieur, par exemple, on pourra utiliser un SceneManager différent.

5.3.2.3 Ogre Exporter

Les exportateurs sont des plugins de modèleur 3D qui écrivent le maillage et le squelette d'animation et le texture dans un format de fichier compréhensible par OGRE. Ces fichiers sont respectivement au format **.mesh**, **.skeleton** et **.material**.

Nous allons utiliser *OgreMax Scene Exporter*. Il est le plugin du modèleur 3DS Max utilisé pour l'exportation des scènes de format compréhensible par 3DS Max vers des scènes de format compréhensible par OGRE. Nous avons utilisé la version 2.4 du 21 Septembre 2011.

5.4 L'architecture SimCroi V1

L'architecture de première version d'application est sous la forme suivant :

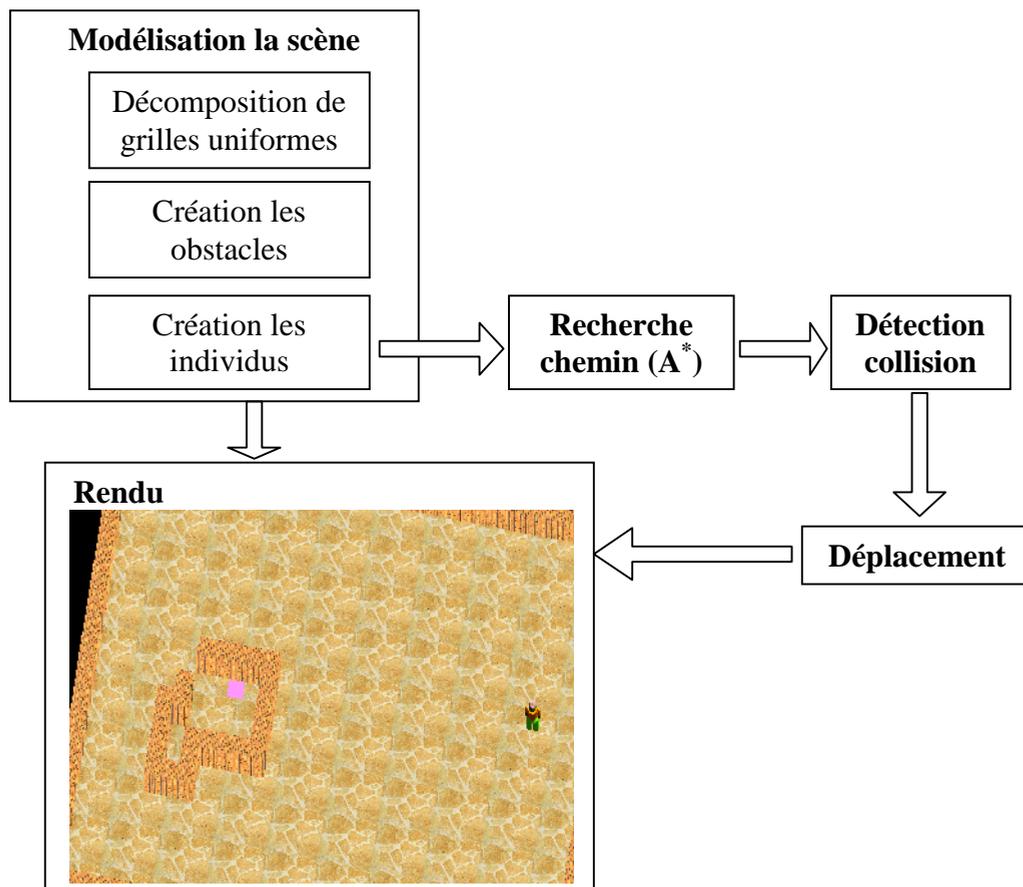


Fig.V.4. L'architecture de première version d'application.

5.4.1 Résultats

La combinaison de l'algorithme A^* et les comportements d'évitement de collision donné des bons résultats de simulate de croisement de flux de piéton.

L'interface utilisateur a été conçue pour permettre une utilisation facile du logiciel :

La simulation peut être interrompue à tout moment ;

Elle peut être exécutée pas à pas ;

Peut change l'angle de vue ;

L'utilisateur peut créer sa scène virtuelle de différentes manières.

Figure suivant présente résultat obtenue après d'applique l'algorithme A^* .

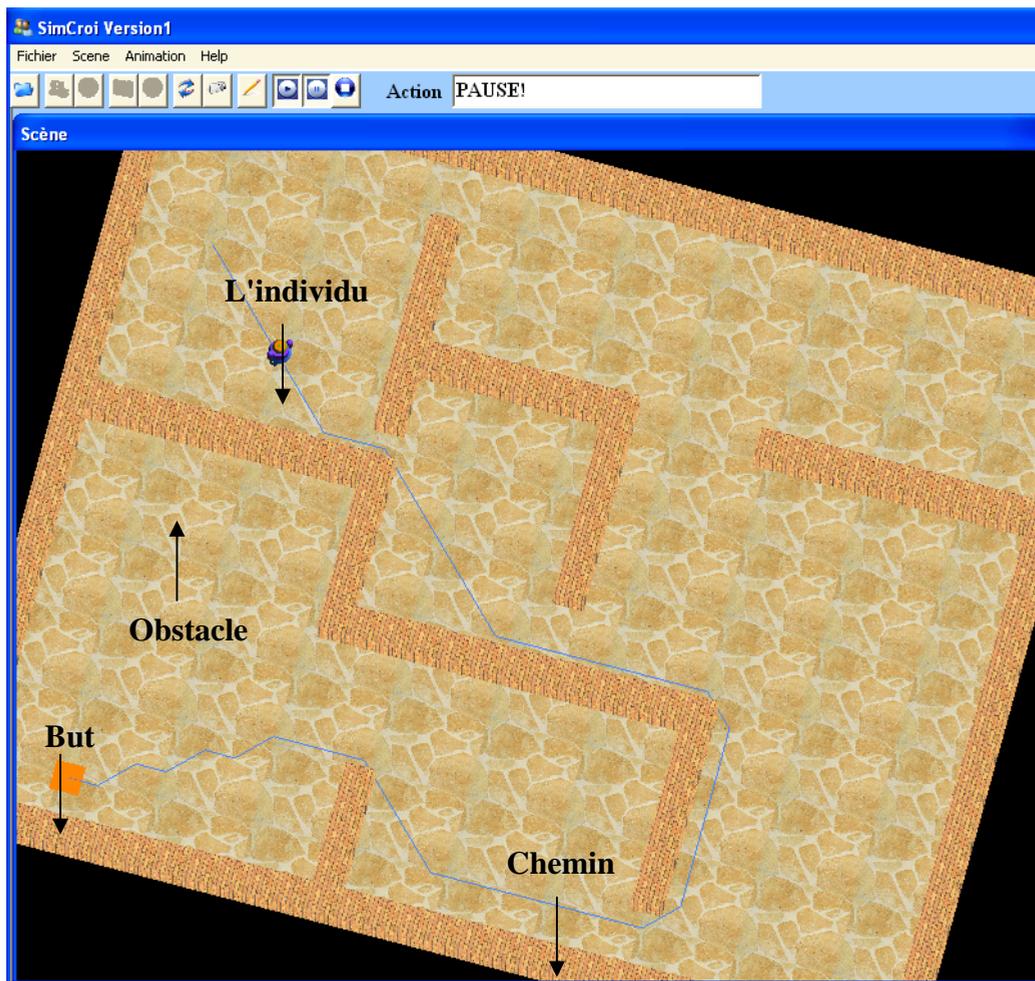


Fig.V.5. Résultat de l'algorithme A^* .

Les Figures suivants présentent trois situations de croisement de flux de piétons.

Figure V.6 présente le résultat obtenu de simulation croisement entre deux groupes de piétons, on applique l'algorithme d'évitement de collision acteur par acteur. Les comportements d'évitement de collision utilisés sont : l'avancement, le changement de directions, l'attente et le retour.

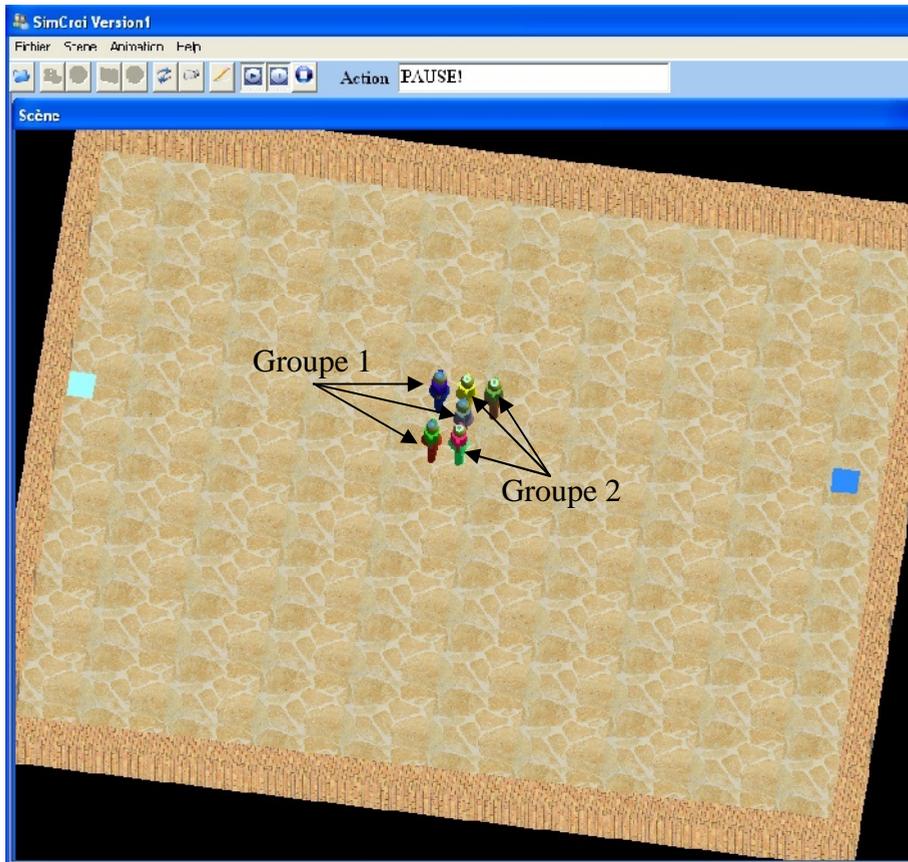


Fig.V.6. Croisement entre deux groupes.

Figure V.7 présente le résultat obtenu de simulation croisement entre deux groupes de piétons avant une porte (alignement dans deux directions). Les comportements d'évitement de collision utilisés sont : l'avancement, le changement de directions, l'attente et le retour.

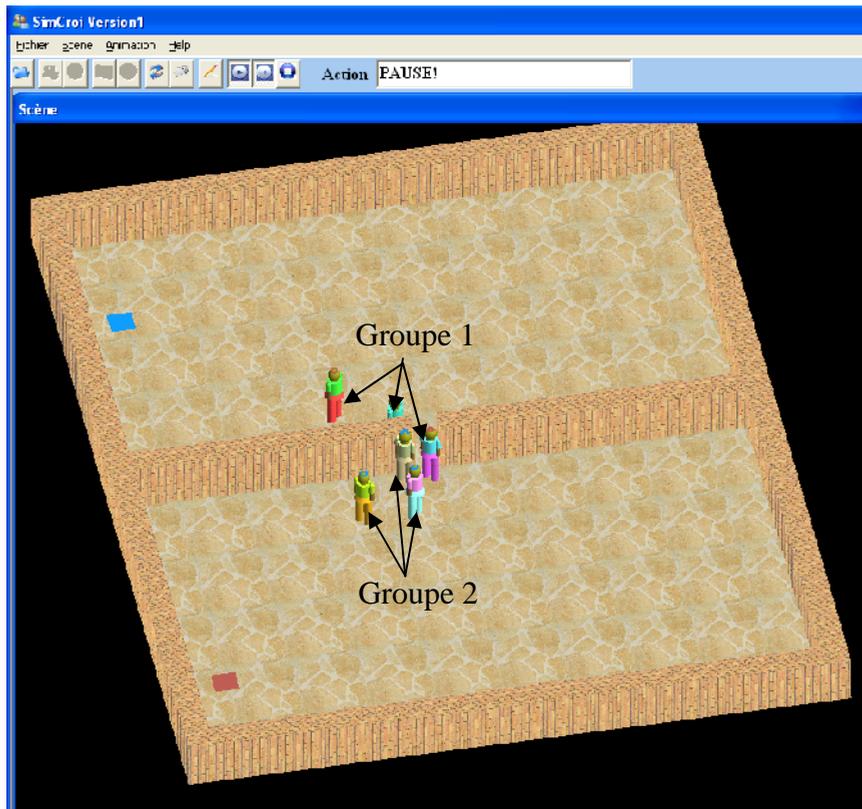


Fig.V.7. L'alignement dans deux directions.

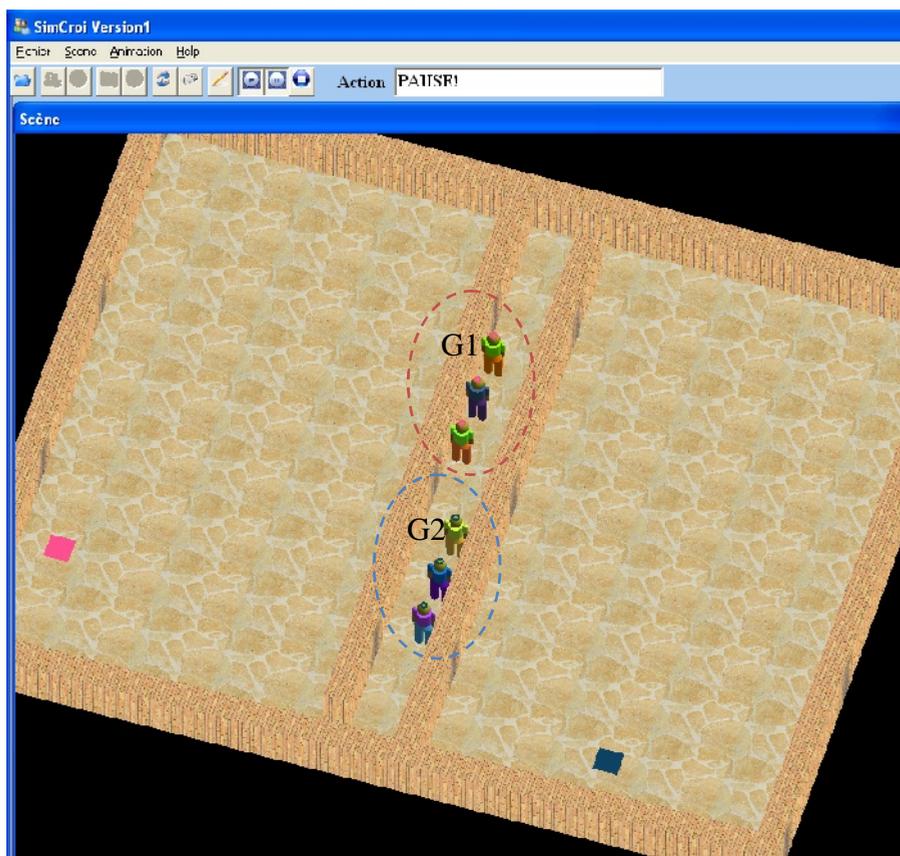


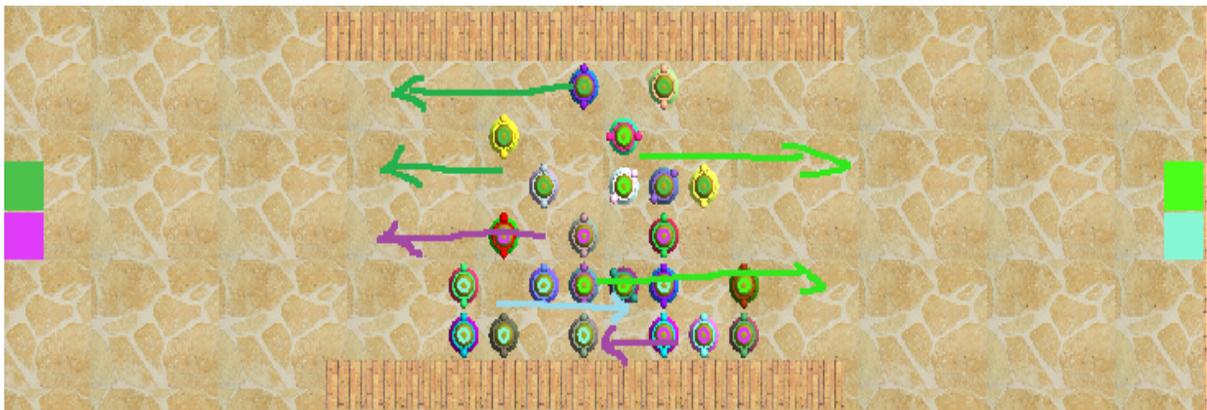
Fig.V.8. Passage étroit.

Figure V.8 présente le résultat obtenu de simulation croisement entre deux groupes de piétons dans couloire. Les comportements d'évitement de collision utilisés sont : l'avancement, le changement de directions et l'attente.

Les Figures suivants présentent le résultat qui nous allons obtenir de simulation croisement entre deux flux de piétons. Pour résoudre le problème de croisement nous allons utiliser les comportements d'évitement de collision entre les individus, nous allons voir après l'analyse les groupes forme de lignes.



a) Deux flux des individus opposent.



b) Résultat de croisement entre le deux flux.

Fig.V.9. Croisement entre de flux.

5.5 L'architecture SimCroi V2

L'architecture de deuxième version d'application est sous la forme suivant :

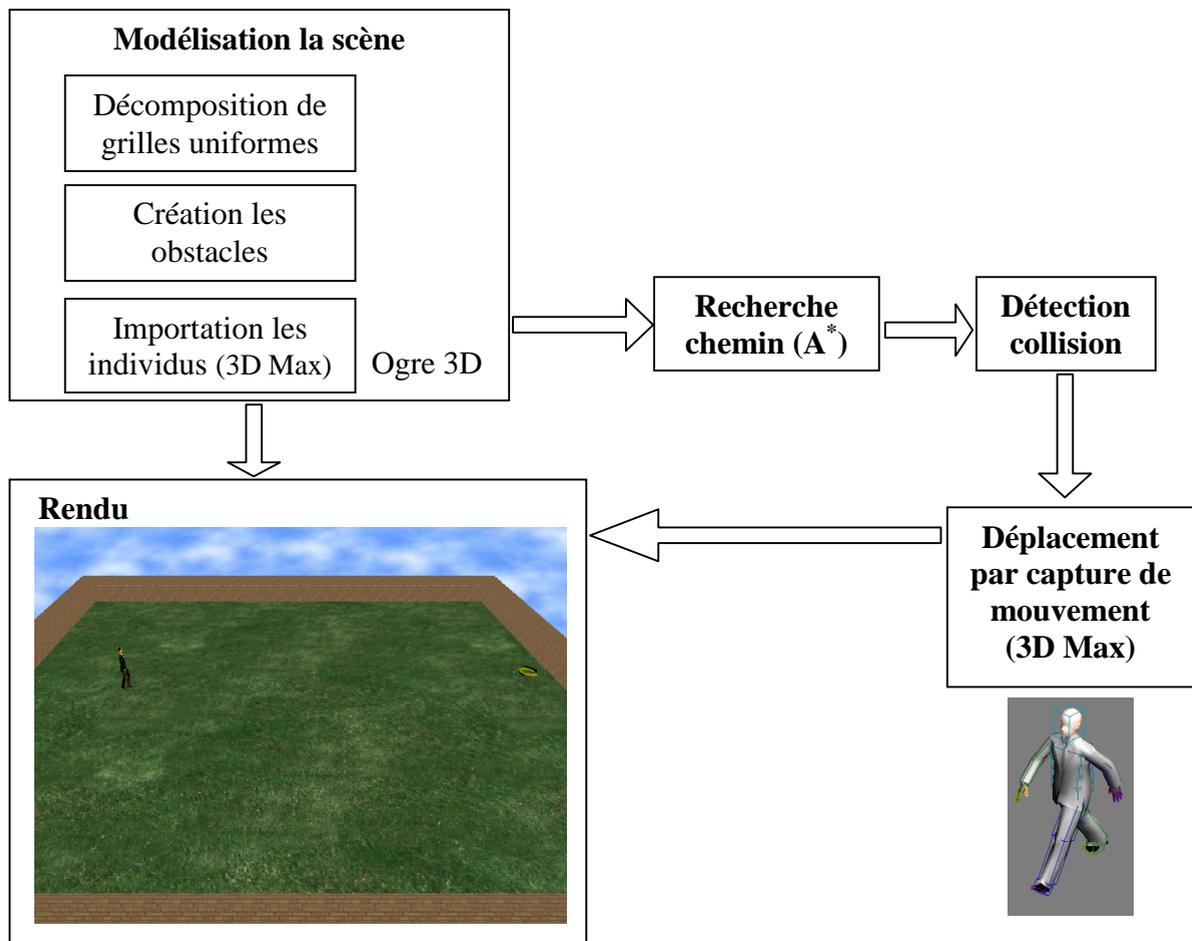


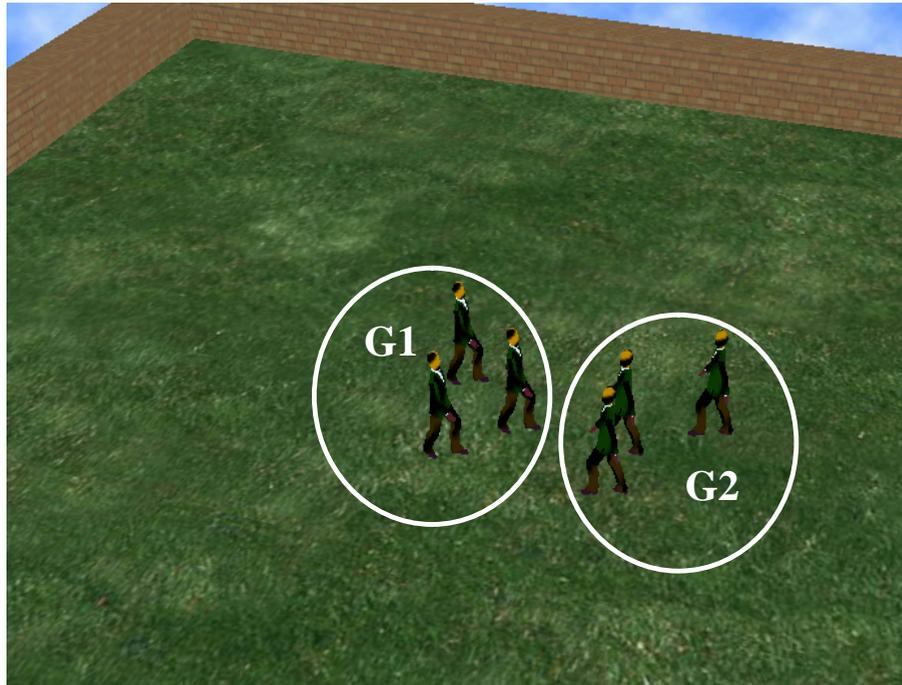
Fig.V.10. L'architecture de deuxième version d'application.

5.5.1 L'exportation les individus

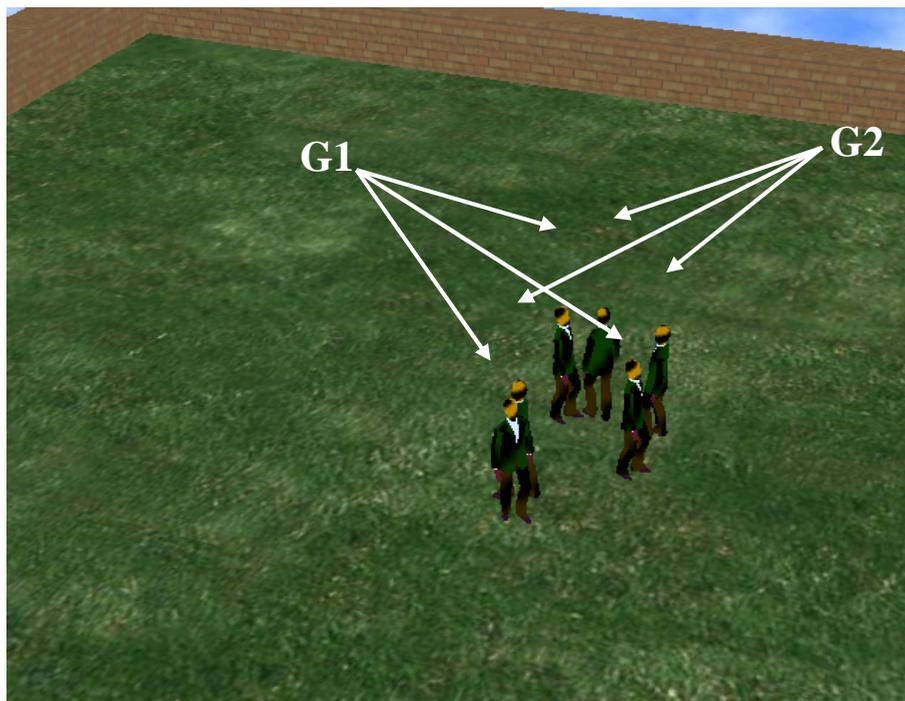
Nous allons utiliser 3D Max pour modélise les individus 3D et pour intègre l'animation capture sur le modèle, ensuite nous utilisons OgreMax exporter pour exporter le modèle vers Oger 3D. OgreMax exporter donne trois fichiers comme résultat, ".mesh", ".skeleton" et ".material".

5.5.2 Résultats

Après de change le modèle des individus de la première version d'application par de modèle 3D et d'intégré les mouvements capture, nous allons obtenir les résultats qui présente dans les figures suivants.

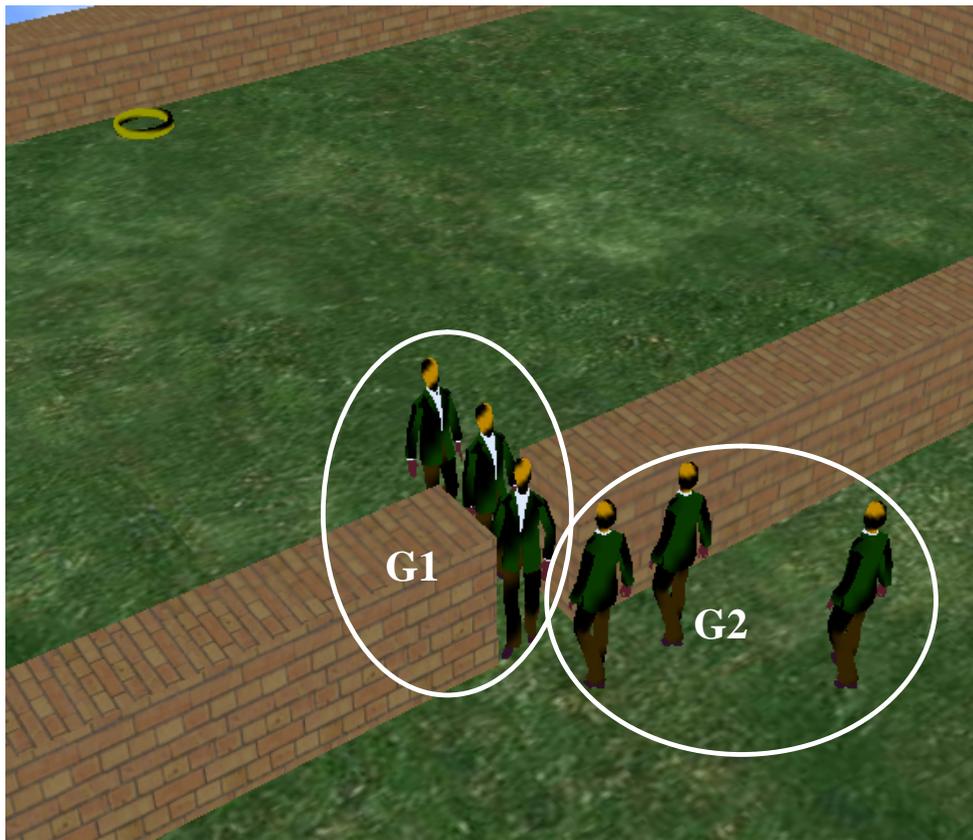


a) Les deux groupes de piéton avant croisement.



b) Les comportements d'évitement de collision utilisés sont acteur par acteur.

Fig.V.11. Croisement entre deux groupes.



Les deux groupes de piéton avant une sortie.

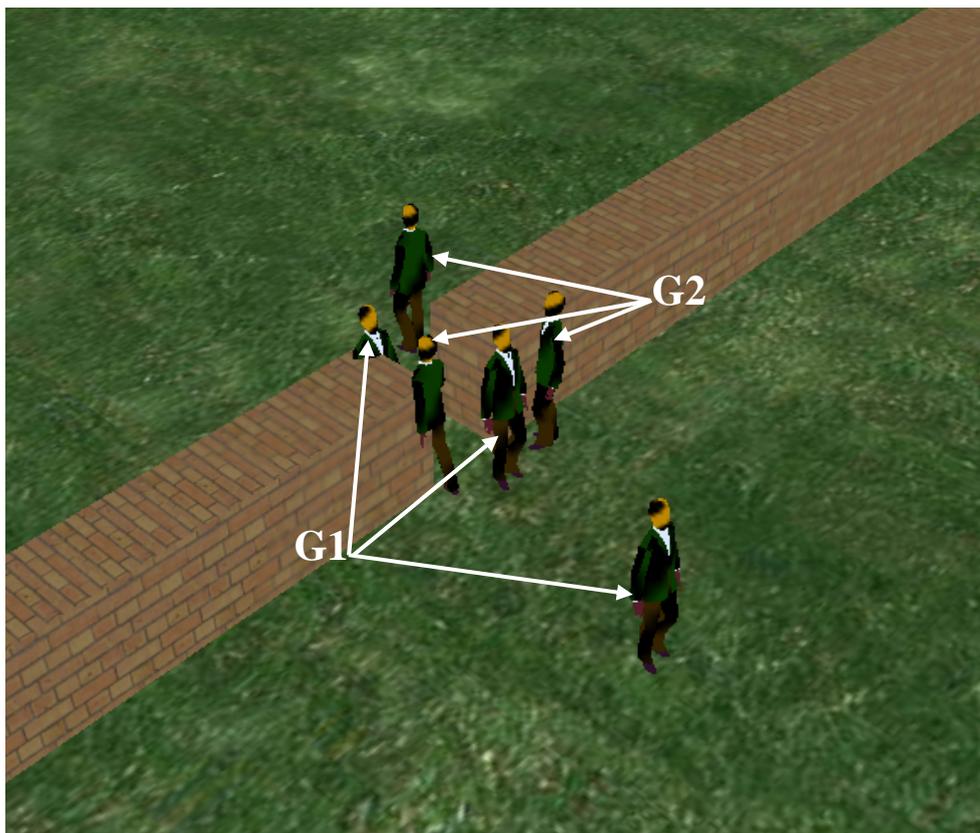
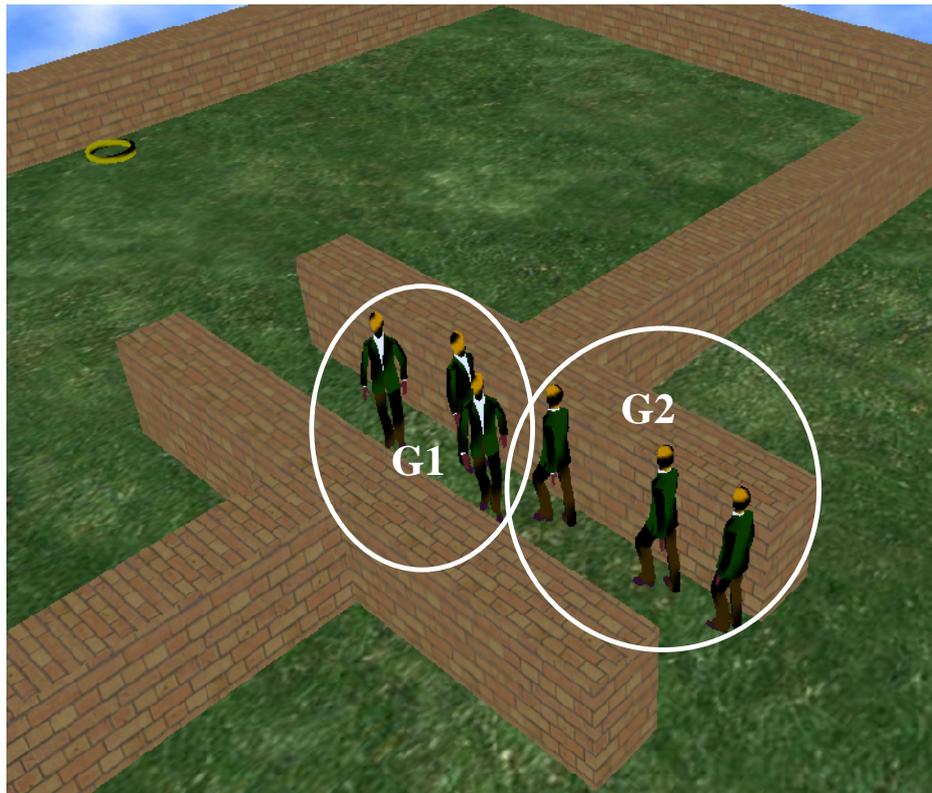


Fig.V.12. L'alignement dans deux directions.



Les deux groupes de piéton dans couloir.

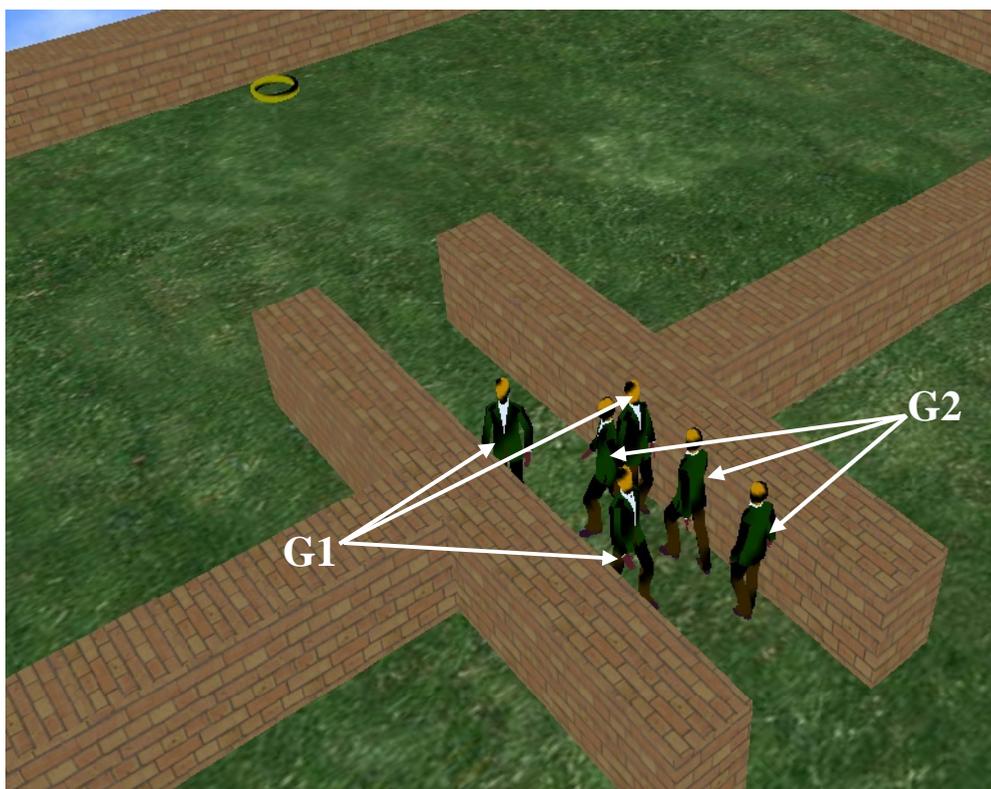


Fig.V.13. Passage étroit.

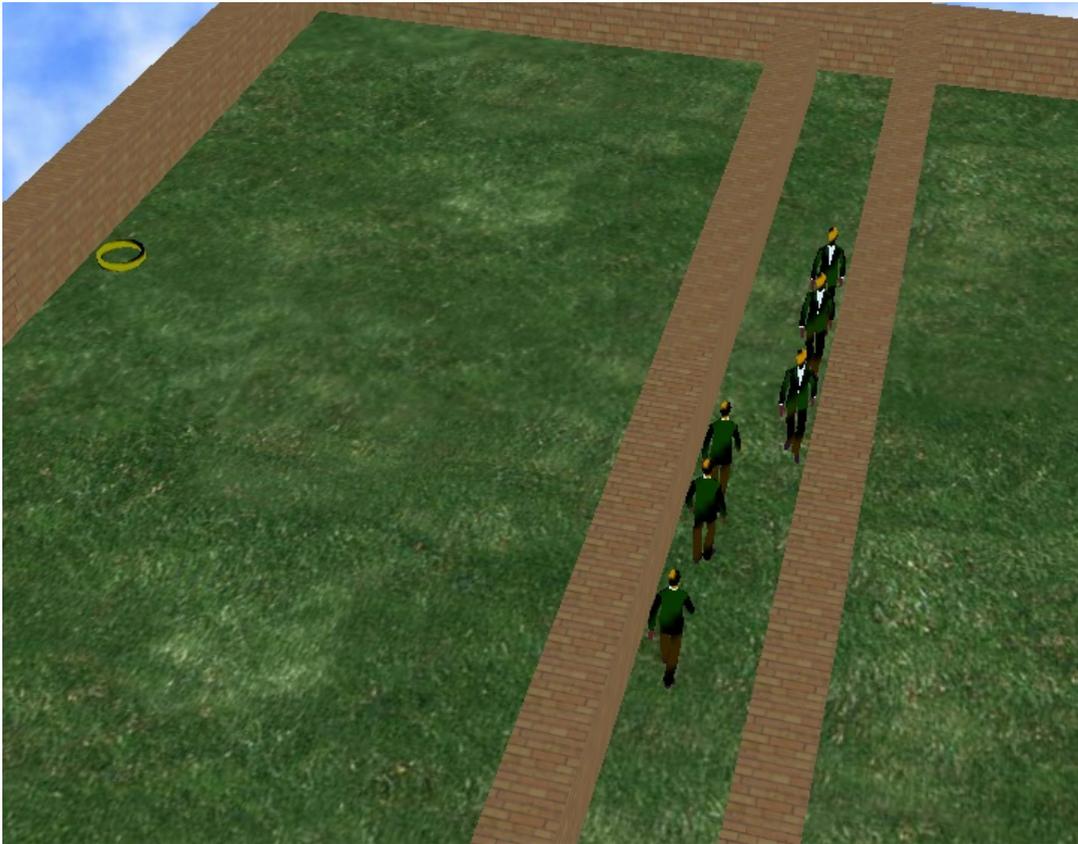


Fig.V.14. Passage dans un couloir.

La procédure de évitement de collision qui nous allons utiliser au la première version d'application donnée une simulation plus réaliste de croisement de flux de piéton, mais la procédure de déplacement donne une animation plus loin à l'animation réelle, par ce que donnée une animation discontinu.

Dans deuxième modèle nous allons concentrer d'intégré de mouvement réelle (mouvement capture) à la procédure de déplacement. Nous allons obtenu une animation plus réaliste.

5.6 Conclusions

Nous avons abordé tout au long de ce chapitre les résultats obtenus lors de ce mémoire. Ces résultats prennent tout d'abord la forme d'une architecture dédiée à la simule de croisement de flux de piéton.

Ce logiciel concrétise le modèle présenté comme contribution, tout en apportant un certain nombre de fonctionnalités permettant de les exploiter.

La première application concentre sur les techniques d'évitement de collision entre les acteurs, ainsi que sur la recherche de chemin dans un environnement dynamique.

La deuxième application concentre d'intègre de mouvement capture à la première application, pour faire une compare entre les deux résultats.

Notre modèle a pu être employé comme une plateforme de simulation l'animation de flux de piétons dans des situations réelles telles que : la montée et la descente dans un métro, la marche sur un trottoir ou sur corridor de piétons, etc.

Conclusion générale

Conclusion générale

Nous avons intéressé de simule l'animation des foules de piéton, on exploitant le principe de l'animation comportementale pour animer les individus. Nous avons concentré notre effort sur les techniques d'évitement de collision ainsi que sur les algorithmes de recherche de chemin dans un environnement dynamique.

Nous avons proposé deux modèles permettant de simuler le croisement de flux de piétons. Le première modèle utilise de formes géométriques simples pour modéliser les individus et l'animation de ces individus est faite par une translation. Dans le deuxième modèle nous avons remplacé les formes géométriques par de modèles 3D (Humain 3D) et nous avons intégré de mouvement capturés pour animer les individus. Ces modèles s'articulent autour de trois problématiques : recherche du chemin optimal, l'évitement de collision, et représentation 3D de l'humain virtuel avec l'intégration capture de mouvement.

Nous avons proposé pour chaque individu un ensemble de règles comportementales pour résoudre le problème l'évitement de collision entre les individus et un ensemble de règles de sociales pour le croisement de flux entre deux groupes de piétons. Nous avons choisi l'algorithme A^* pour trouver un chemin ou un itinéraire initial. Ces deux techniques rehaussent le réalisme de la simulation d'animation d'une foule des individus.

Perspectives

Afin d'accroître les performances de ce modèle, il peut être important de se pencher sur différents autres aspects. Nous avons relevés trois axes permettant d'obtenir un modèle plus réaliste :

Le premier point consiste à utiliser un modèle d'environnement urbain.

L'ajout des règles sociales et psychologiques aux comportements de navigation.

La prise en compte de la relation inter et intra groupe pour bien simuler le réalisme de mouvement. .

Bibliographie

Bibliographie

[ACF01]: Arikan (O.), Cheney (S.) et Forsyth (D. A.). *Efficient multi-agent path planning*. Dans : Computer Animation and Simulation 01, éd. par Magnenat-Thalmann (N.) et Thalmann (D.), pages 151-162, 2001.

[ACT06]: Abaci (T.), Ciger, (J.) et Thalmann, (D.). *Planning with smart objects*. In International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, pages 25–28, 2006.

[AFO05]: Arikan (O.), Forsyth (D. A.) et O'Brien (J. F.). *Pushing People Around*. Proceedings of SIGGRAPH Symposium on Computer Animation, pages 59-66, 2005.

[AK03]: Azimzadeh (M.) et Klarqvist (B.). *Phenomenal space : attitudes and methods*. Dans Space Syntax, 4th International Symposium, University College of London, 2003.

[Ant98]: AntZ, 1998. <http://www.siggraph.org/s98/conference/sketches/animation/an2.html>

[Aub07]: Aubry (S.). *Annotations et gestion des connaissances en environnement virtuel collaboratif*. PhD thèses, Université de Technologie de Compiègne, France, 2007.

[AW00]: Ashraf (G.) et Wong (K.C.). *Generating Consistent Motion Transition via Decoupled Framespace Interpolation*. EUROGRAPHICS, vol. 19, no. 3, 2000.

[AW01]: Ashraf (G.) et Wong (K. C.). *Constrained Framespace Interpolation*. Computer Animation, pages 61-72, 2001.

[BCN97]: Bouvier (E.), Cohen (E.) et Najman (L.). *From crowd to airbag deployment: particle systems, a new paradigm of simulation*. Dans: Journal of Electronic Imaging, vol. 6, no. 1, pages 94-107, 1997.

[Bee90]: Beer (R.D.). *Intelligence as Adaptive Behavior: Experiments in Computational Neuroethology*. New York: Academic Press. (1990). Voir aussi: <http://yugoth.ces.cwru.edu/johng/robopaper/robopaper.html>

[BH97]: Brogan (D.C.) et Hodgins, (J.). *Group Behaviours for Systems with Significant Dynamics*. Dans: Autonomous Robots, pages 137-153, 1997.

- [BL91]:** Barraquand (J.) et Latombe (J.-C.). *Robot motion planning : a distributed representation approach*. International Journal of Robotics Research, pages 628–649, 1991.
- [BMH98]:** Brogan (D.C.), Metoyer (R.A.) et Hodgins (J.K.). *Dynamically simulated characters in virtual environments*. Dans: IEEE C. G. A. vol.18, no. 5, pages 58-69. 1998.
- [BMS04]:** Botea(A.), Müller (M.), et Schaeffer (J.). *Near optimal hierarchical path-finding*. Journal of Game Development, pages 7–28, 2004.
- [BMT95]:** Boulic (R.), Mas (R.) et Thalmann (D.). *Position control of the center of mass for articulated figures in multiple support*, Dans: Computer Animation and Simulation'95, éd. Par Springer-Verlag (New-York), pages 130-144, 1995.
- [Bou09]:** Boutin (L.). *génération de trajectoires pour la robotique humanoïde a partir de mouvements humains*. PhD Thèses, Université de Poitiers, 2009.
- [BPW93]:** Badler (N.), Phillips (C.) et Webber (B.). *Simulating Humans*. Computer Graphics Animation and Control. Oxford University Press, New York, NY, 1993.
- [BTh98]:** Bandi (S.) et Thalmann (D.). *Space discretization for efficient human navigation*. Computer Graphics Forum, vol. 17, no. 3, pages 195-206, 1998.
- [BT98]:** Becheiraz (P.) et Thalmann, (D.). *A Behavioral Animation System for Autonomous Actors personified by Emotions*. Dans: Proceedings of the First Workshop on Embodied Conversational Characters (WECC '98), Lake Tahoe, California. 1998.
- [BY95]:** Boissonnat (J.-D.) et Yvinec (M.). *Géométrie algorithmique*. Collection Informatique. EDISCIENCE international, 1995.
- [BY98]:** Boissonnat (J.-D.) et Yvinec (M.). *Algorithmic Geometry*. Cambridge. University Press, 1998.
- [Che87]:** Chew (L. P.). *Constrained delaunay triangulations*. Dans Proceedings of the third annual symposium on Computational geometry, pages 215–222, 1987.
- [CK00]:** Choi (K.J.) et Ko (H.S.). *On-line motion retargetting*. The Journal Of Visualisation and Computer Animation, vol. 11, no. 5, pages 223-235, 2000.

[CL03]: Choi (M. G.), Lee (J.) et Shin (S. Y.). *Planning biped locomotion using motion capture data and probabilistic roadmaps*. ACM Transactions on Graphics, vol. 22, no. 2, pages 182-203, 2003.

[Coh92]: Cohen (M.F.). *Interactive Spacetime Control for Animation*. Proceedings of SIGGRAPH Computer Graphics, vol. 26, no. 2, pages 293-302, 1992.

[CRR01]: Caselli (S.), Reggiani (M.), et Rocchi (R.). *Heuristic methods for randomized path planning in potential fields*. University Parma. 2001.

[DK03]: Duckham (M.) et Kulik (L.). *"SimplestPaths": Automated route selection for navigation*. Dans Spatial Information Theory: Foundations of Geographic Information Science, édité par Kuhn (W.), F.Worboys (M.), et Timpf (S.), vol. 2825 de Lecture Notes in Computer Science, pages 182–199, 2003.

[Don4]: Donikian, (S.). *Modélisation, contrôle et animation d'agents virtuels autonomes évoluant dans des environnements informés et structurés*, pages 236, 2004.

[Doy02]: Doyle (P.). *Believability through context using "knowledge in the world" to create intelligent characters*. In Proceedings of the first international joint conference on Autonomous agents and multi-agent systems, pages 342–349. 2002.

[Fou06]: Foudil (C.). *Animation comportemental : Simulation de foules d'humains virtuels*. Thèse de Doctorat, Université de Biskra, 2006.

[FP03]: Fang (A C.) et Pollard (N.S.). *Efficient Synthesis of Physically Valid Human Motion*. Proceedings of SIGGRAPH, vol. 22, pages 417_426, 2003.

[Gof71]: Goffman (E.). *Relations in public: micro-studies of the public order*. New York: Basic books, 1971.

[GSK+03]: Gleicher (M.), Shin (H.J.), Kovar (L.) et Jepsen (A.). *Snap-Together Motion: Assembling Run-Time Animations*. Proceedings of Symposium on Interactive 3D Graphics, 2003.

[GSN04]: Gloor (C.), Stucki (P.), et Nagel(K.). *Hybrid techniques for pedestrian simulations*. Dans 4th Swiss Transport Research Conference, Monte Verità, Ascona, 2004.

[Gut05]: Gutiérrez (M. A). *Semantic Virtual Environments*. PhD thèses, Ecole polytechnique fédérale de Lausanne, 2005.

[HA98]: Hègron (G.) et Arnaldi (B.), *Computer Animation: Motion and Deformation Control*. Cambridge, Grande-Bretagne, Eurographics'92 Tutorial Notes, Eurographics Technical Series, 1998.

[Hel95] : Helbing (D.). *Quantitative Sociodynamics : Stochastic Methods and Models of Social Interaction Processes*. Kluwer Academic, Dordrecht, 1 edition, 1995.

[HHK05]: Hartwig(H.), Hochmair et Karlsson (V.). *Investigation of preference between the least-angle strategy and the initial segment strategy for route selection in unknown environments*. Lecture Notes in Computer Science, vol. 3343, pages 79 – 97, 2005.

[HKL+99]: Hoff III (K. E.), Keyser (J.), Lin (M.), Manocha (D.), et Culver (T.). *Fast computation of generalized Voronoi diagrams using graphics hardware*. Computer Graphics, vol. 33, pages 277–286, 1999.

[HPH+93]: Hillier (B.), Penn (A.), Hanson (J.), Grajewski (T.), et Xu (J.). *Natural movement: or, configuration and attraction in urban pedestrian movement*. Environment and Planning B: Planning and Design, vol. 20, pages 29–66, 1993.

[KAA+03]: Kallmann (M.), Aubel (A.), Abaci (T.), et Thalmann (D.). *Planning collision free reaching motions for interactive object manipulation and grasping*. Dans Computer Graphics Forum, vol. 22, pages 313–322, 2003.

[KBT03]: Kallmann (M.), Bieri (H.), et Thalmann (D.). *Fully dynamic constrained Delaunay triangulations*. Geometric Modelling for Scientific Visualization, 2003.

[KGP02]: Kovar (L.), Gleicher (M.) et Pighin (F.). *Motion Graphs*. Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pages 473-482, 2002.

[KHL05]: Komura (T.), Ho (E.S.L.) et Lau (R.W.H.). *Animating reactive motion using momentum-based inverse kinematics*. Computer Animation and Virtual Worlds, vol. 16, pages 213-223, 2005.

[KL98]: Kallmann (M.) et Thalmann (D.). *Modeling objects for interaction tasks*. Dans EGCAS, 1998.

- [KL02]:** Kallmann (M.) et Thalmann (D.). *Modeling behaviors of interactive objects for real time virtual environments*. Journal of Visual Languages and Computing, vol. 13, pages 177–195, 2002.
- [Kor85]:** Korf (R. E.). *Depth-first iterative-deepening: An optimal admissible tree search*. Artificial intelligence, vol. 27, pages 97–109, 1985.
- [KS02] :** Kirchner (A.) et Schadschneider (A.). *Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics*. Institut für Theoretische Physik, Universität zu Köln A, 2002.
- [Kuf98]:** Kuffner (J. J.). *Goal-directed navigation for animated characters using real-time path planning and control*. Lecture Notes in Computer Science, vol. 1537, pages 171-179, 1998.
- [Kuf99]:** Kuffner (J. J.). *Autonomous agents for real-time animation*. PhD thèse, Stanford University, 1999.
- [Kul05]:** Kulpa (R.). *Adaptation interactive et performante des mouvements d'humanoïdes synthétiques : aspects cinématiques, cinétiques et dynamiques*. PhD thèses, INSA Rennes, 2005.
- [LA98]:** Logan (B.) et Alechina (N.). *A* with bounded costs*. Dans Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), pages 444–449, 1998.
- [Lam03]:** Lamarche (F.). *Humanoïde virtuels, réaction et cognition : une architecture pour leur autonomie*. PhD thèse, I.F.S.I.C., Université de Rennes I, IRISA-INRIA, 2003.
- [Lau83]:** Laumond (J.P.). *odel structuring and concept recognition: Two aspects of learning for a mobile robot*. Dans: International Joint Conférence on Artificial Intelligence (IJCAI), pages 839-841, 1983.
- [Lau89]:** Laumond (J.P.). *A learning System for the understanding of a mobile robot environment*. Dans: Machine and Human Learning, éd. Par Kodratoff (Y.), pages 161-171, 1989.
- [Lat91]:** Latombe (J.-C.). *Robot Motion Planning*. Boston, Boston: Kluwer Académie Publishers, 1991.

[LCR+02]: Lee (J.), Chai (J.), Reitsma (P.S.), Hodgins (J.K.) et Pollard (N.S.). *Interactive Control of Avatars Animated with Human Motion Data*. Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pages 491-500, 2002.

[LD04]: Lamarche (F.) et Donikian (S.). *Crowds of virtual humans: a new approach for real time navigation in complex and structured environments*. Computer Graphics Forum, Eurographics'04, 2004.

[LGC94]: Liu (Z.), Gortler (S. J.) et Cohen (M. F.). *Hierarchical Spacetime Control*. Proceedings of SIGGRAPH, pages 35-42, 1994.

[LL04]: Lee (J.) et Lee (K. H.). *Precomputing Avatar Behavior From Human Motion Data*. Eurographics SIGGRAPH Symposium on Computer Animation, pages 79-87, 2004.

[LRD+90]: Lengyel (J.), Reichert (M.), Donald(B.R.), et Donald (P.G.). *Real time robot motion planning using rasterizing computer graphics hardware*. Dans SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques, pages 327–335, 1990.

[LW92]: Lee (J. R. E.) et Watson (R.). *Regards et habitudes des passants*, vol. 57-58, 1992.

[Mat94]: Mataric (M. J.). *Learning to Behave Socially*. Dans: Cliff (D.), Husbands (P.), Meyer (J.A.) et Wilson (S.), ed^s, *From Animals to Animats: International Conference on Simulation of Adaptive Behavior*, pages 453-462, 1994.

[MBB+00]: Monzani (J.-S.), Baerlocher (P.), Boulic (R.) et Thalmann (D.). *Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting*. Proceedings of Eurographics, 2000.

[Meh10] : Mehdi (M.). *Étude expérimentale et modélisation des déplacements collectifs de piétons*. PhD thèse. Université Toulouse III - Paul Sabatier. 2010.

[Men03]: Ménardais (S.). *Fusion et adaptation temps réel de mouvements acquis pour l'animation d'humanoïdes synthétiques*. PhD thèses, IRISA Rennes, 2003.

[MGT99]: Musse (S.R.), Garat (F.) et Thalmann (D.). *Guiding and Interacting with Virtual Crowds in Real Time*. Dans: Proceedings of Workshop Eurographics Computer Animation and Simulation, 1999.

- [MKB08]:** Multon (F.), Kulpa (R.) et Bideau (B.). *MKM: A Global Framework for Animating Humans in Virtual Reality Applications*. Presence, vol. 17, no. 1, pages 17-28, 2008.
- [MMK+04]:** Ménardais (S.), Multon (F.), Kulpa (R.) et Arnaldi (B.). *Motion blending for real-time animation while accounting for the environment*. Proceedings of the Computer Graphics International (CGI'04), 2004.
- [Mou09]:** Mouttapa (T. I.). *Interaction et connaissance: construction d'une expérience dans le monde virtuel*, 2009.
- [MT01]:** Musse (S.R.) et Thalmann (D.). *Hierarchical Model for Real Time Simulation of Virtual Human Crowds*. Dans: IEEE Transactions on Visualisation and Computer Graphics, vol. 7, n.2, pages 152-164, 2001.
- [Nil82]:** Nilsson (N.J.). *Principles of artificial intelligence*. Springer-Verlag Berlin Heidelberg New York, 1982.
- [NM93]:** Ngo (J.T.) et Marks (J.). *Spacetime Constraints Revisited*. Proceedings of SIGGRAPH, pages 343-350, 1993.
- [NSK+06] :** Nishinari (K.), Sugawara (K.), Kazama (T.), Schadschneider (A.), et Chowdhury (D.). *Modelling of self-driven particles : Foraging ants and pedestrians*. Physica Am vol. 372, pages 132-141, 2006.
- [Osa04]:** Osaragi (T.). *Modeling of pedestrian behavior and its applications to spatial evaluation*. Dans AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, pages 836–843, 2004.
- [OT90]:** Ortony (A.) et Turner (T.J.). *What's basic about basic emotions?* Psychological Review. pages 315-331, 1990.
- [Ott05]:** Otto (K.A.). *The semantics of multi-user virtual environments*. In SVE. pages 10, 2005.
- [Ove02]:** Overmars (M.H.). *Récent developments in motion planning*. Dans: International Conférence on Computational Science (S), pages 3-13, 2002.

[PB00]: Pollard (N.S.) et Behmaram-Mosavat (F.). *Force-Based Motion Editing for Locomotion Tasks*. Proceedings of the IEEE International Conference on Robotics and Automtion, 2000.

[PCM+06]: Pettré (J.), Ciechowski (P.H.), Maim (J.), Yersin (B.), Laumond (J.P.), et Thalmann (D.). *Real-time navigating crowds: scalable simulation and rendering*. Computer Animation and Virtual Worlds, vol. 17, pages 445–455, 2006.

[PDB+06]: Pronost (N.), Dumont (G.), Berillon (G.) & Nicolas (G.). *Morphological and stance interpolations in database for simulating bipedalism of virtual humans*. Visual Computer, vol. 22, pages 4-13, 2006.

[Pel03]: Pelletier (P.). *Simulation de comportements de foule*. M.Sc thèse, Université de Laval, 2003.

[Pet03]: Pettré (J.). *Planification de Mouvements de Marche pour Acteurs Digitaux*. PhD Thèses, Thèse de doctorat de l'Université Paul Sabatier, LAAS-CNR, Toulouse, 2003.

[PL03]: Pettré (J.), Laumond (J.-P.) et Siméon (T.). *A 2-stages locomotion planner for 273 digital actors*. Dans: Proc. of the 2003 ACM SIGGRAPH Eurographics Symposium on Computer Animation (SCA '03), pages 258-264, 2003.

[PQ98]: Relieu (M.) et Quéré (L.). *Les risques urbains: Acteurs, systèmes de prévention, chapter Mobilité, perception et sécurité dans les espaces urbains*. Anthropos : Collection ville. 1998.

[RAB+08] : Robin (T.), Antonini (G.), Bierlaire (M.), et Cruz (J.). *Specification, estimation and validation of a pedestrian walking behavior model*. Transportation Research Part B: Methodological, vol. 43, 36-56, 2009.

[Rel96]: Relieu (M.). *Voir et se mouvoir en marchant dans la ville*. Dans Le courrier du CNRS : Villes, pages 107–109, 1996.

[Rey87]: Reynolds (C.W.). *Flocks, herds and schools: a distributed behavioural model*. Dans: SIGGRAPH'87, vol. 21, no. 4 of Computer Graphics, pages 25-34, 1987.

[Rey99]: Reynolds (C. W.). *Steering Behaviors For Autonomous Characters*. GDC 99, pages 763-782, 1999.

- [RKB+94]:** Reich (B.), Ko (H.), Becket (W.), et Badler (N. I.). *Terrain reasoning for human locomotion*. Dans *Computer Animation*, pages 996–1005, 1994.
- [RMT90]:** Renault (O.), Magnenat-Thalmann (N.) et Thalmann (D.). *A vision based approach to behavioural animation*. *Journal of Visualization and Computer Animation*, vol. 1, no 1, pages 18-21, 1990.
- [Seb07]:** Sébastien (P.). *Caractérisation des niveaux de services et modélisation des circulations de personnes dans les lieux d'échanges*. PhD thèse. Université de Rennes I, IRISA, France, octobre 2007.
- [Sep07]:** Septseault (C.). *Représentation d'environnements virtuels informés et de leur dynamique par un personnage autonome en vue d'une crédibilité comportementale*. PhD thèse. Université de Bretagne Occidentale. 2007.
- [SM00]:** Steck(S.) et Mallot(H). *The role of global and local landmarks in virtual environment navigation*. *Presence: Teleoperators and Virtual Environments*, vol. 9, pages 69-83, 2000.
- [ST05]:** Shao (W.) et Terzopoulos (D.). *Autonomous pedestrians*. Dans *SCA 05: Proceedings of the 2005 ACM SIGGRAPH Eurographics symposium on Computer animation*, pages 19-28, 2005.
- [TB96]:** Thrun (S.) et Bücken (A.). *Integrating grid-based and topological maps for mobile robot navigation*. Dans: *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 944-951, 1996.
- [TC00]:** Tecchia (F.) et Chrysanthou (Y.). *Real time rendering of densely populated urban environments*. Dans: *Rendering Techniques '00 (10th Eurographics Workshop on Rendering)*. pages 45-56, 2000.
- [TCP06]:** Treuille (A.), Cooper (S.), et Popović (Z.). *Continuum crowds*. Dans *SIGGRAPH '06: ACM SIGGRAPH 2006*, pages 1160–1168, 2006.

- [TLO03]:** Thompson (P.), Lindstrom (H.), Ohlsson (P.) et Thompson (S.). *Simulex: Analysis and Changes for IMO Compliance*. Dans: Proceedings of 2nd International Conference: Pedestrian and Evacuation Dynamics, pages 173-184, 2003.
- [TT94]:** Tu (X.) et Terzopoulos (D.). *Artificial Fishes: physics, locomotion, perception, behaviour*. Dans: SIGGRAPH 94 Conference Proceedings, pages 43-50, 1994.
- [War95]:** Warren (W.H.). *Handbook of perception and cognition: Perception of Space and Motion*. Chapter Self-Motion: Visual Perception and Visual Control, pages 263–325. 1995.
- [WB03]:** Wang (J.) et Bodenheimer (B.). *An Evaluation of a Cost Metric for Selecting Transitions between Motion Segments*. Eurographics SIGGRAPH Symposium on Computer Animation, pages 232-238, 2003.
- [WK88]:** Witkin (A.) et Kass (M.). *Space Constraints*. Proceedings of ACM SIGGRAPH, pages 159-168, 1988.
- [WM03]:** Wiener (J.) et Mallot (H.). *Fine-to-coarse Route Planning and Navigation in Regionalized Environments*. Spatial Cognition & Computation, vol. 3, pages 331–358, 2003.
- [WP95]:** Witkin (A.) et Popovic (Z.). *Motion Warping*. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 105-108, 1995.
- [YCD+05] :** Yu (W. J.), Chen (R.), Dong (L. Y.), et Dai (S. Q.). *Centrifugal force model for pedestrian dynamics*. Physical Review E., 2005.
- [YKH04]:** Yamane (K.), Kuffner (J.J.) et Hodgins (J.K.). *Synthesizing animations of human manipulation tasks*. Proceedings in SIGGRAPH 2004, pages 532-539, 2004.
- [YSS+98]:** Yahja (A.), Stentz (A.), Singh (S.) et Brumitt (B. L.). *Framed-quad tree path planning for mobile robots operating in sparse environments*. Dans: IEEE Conférence on Robotics and Automation (ICRA). Leuven, Belgium, 1998.
- [ZH99]:** Zordan (V. B.) et Hodgins (J. K.). *Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation*. Proceedings of Computer Animation and Simulation, pages 13-22, 1999.

[ZMC+05]: Zordan (V. B.), Majkowska (A.), Chiu (B.) etFast (M.). *Dynamic Response for Motion Capture Animation*. Proceedings of SIGGRAPH, vol. 24, no. 3, pages 697-701, 2005.

[Web 1]: <http://www.metamotion.com/motion-capture/motion-capture.htm>.

[Web 2]: <http://www.vicon.com>.

[Web 3]: <http://www.biogesta.fr>.

[Web 4]: http://infografilles.fr/motion_capture_222.htm.