

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 - Guelma
Faculté des Mathématiques, d'informatique et des Science de la matière
Département d'informatique



Mémoire de Projet de Fin d'Etudes

2^{ème} Année Master

Filière : Informatique

Optlon : Ingénierle des Médias

Thème :

Modélisation des états émotionnels d'un agent virtuel

Présenté par:

Beghour Fatima
Khelaifia Wahiba

Sous la direction de :

M^f. Soussi Hakim

Juin2016

Résumé

Le réalisme d'une application traitant de l'animation comportementale est fondé d'une part sur le réalisme du comportement lui-même et d'une autre part sur le rendu graphique des scènes produites par l'application.

Dans le cas particulier de l'animation comportementale, le réalisme doit être considéré au niveau macroscopique autant qu'au niveau microscopique.

- Le comportement macroscopique lié à une animation donnée, ce qui lui permet de ressembler à une foule réelle. 82
- Le réalisme microscopique concerne le réalisme du comportement de chaque individu virtuel dans un jeu de contexte, ou le comportement doit ressembler à celui l'un humain appartenant à une foule dans les mêmes contextes. 37

Dans ce travail, on a modélisé l'animation comportementale d'un individu virtuel dans un contexte sportif « saut en longueur » en essayant de satisfaire le réalisme microscopique.

Les mots clés : la réalité virtuelle, l'animation comportementale, l'infographie, l'attribut de caractère.

Table des matières

0.1	Introduction générale	6
1	L'animation comportementale	7
1.1	Introduction	8
1.1.1	Les exemples d'application de l'animation	8
1.1.2	Les premiers travaux d'animation comportementale	9
1.2	Quelques modèles comportementaux	10
1.2.1	Modèle « steering »	10
1.2.2	Modèle ViCrowd	11
1.2.3	Le système de classeurs (LCS)	12
1.2.4	Modèle de comportements scalables	12
1.2.5	Le modèle HiDAC	14
1.2.6	Le système MAGS	15
1.2.7	Le système MACES	16
1.3	Représentation de l'environnement	19
1.3.1	Représentation des environnements simples	19
1.3.2	Représentation des environnements complexes	20
1.4	Conclusion	23
2	L'infographie	24
2.1	Introduction	25
2.1.1	Autodesk 3dsMax (3d)	25
2.1.2	Autodesk Maya (3d)	26
2.1.3	Blender	27
2.2	Les étapes de création d'un objet 3D	28
2.2.1	La modélisation dans l'espace virtuel 3D	28

2.3	Conclusion	40
3	La conception	41
3.1	Introduction	42
3.2	Environnement de développement	42
3.2.1	Les Matériels utilisé	42
3.2.2	Le logiciel utilisé	42
3.3	La Modélisation 3D	43
3.3.1	Prise en main des fonctions de modélisation	43
3.3.2	La modélisation de notre agent et l'environnement	44
3.4	L'animation de l'agent	52
3.5	Le comportement du personnage	54
3.6	Conclusion	56
3.7	Conclusion générale	57

Table des figures

1.1	<i>Quelques exemples d'applications de l'animation comportementale[3].</i>	8
1.2	<i>Subdivision de l'espace de simulation en sous-espaces [7].</i>	11
1.3	<i>Classeur de holland[29].</i>	12
1.4	<i>Architecture d'agent à deux niveaux[18].</i>	13
1.5	<i>Les fonctions de comportement calculer la probabilité d'entrée indique indépendamment. Les distributions de probabilité qui sont calculées à partir des fonctions de comportement sont alors composées. Enfin, l'état suivant est sélectionné par échantillonnage sur la distribution de probabilité composée[29].</i>	14
1.6	<i>Les deux niveaux du système HiDAC[24].</i>	15
1.7	<i>Architecture de MAGS [19].</i>	16
1.8	<i>Exemple séquence d'images de l'évacuation de la foule avec haute ou. Faible pourcentage de leaders[25].</i>	17
1.9	<i>Architecture de PMFserv[29].</i>	18
1.10	<i>La boucle d'interaction principale[23].</i>	19
1.11	<i>Un champ potentiel attractif[11].</i>	20
1.12	<i>Modèle d'environnement Hiérarchie[12].</i>	22
1.13	<i>Capture d'écran d'une application interactive où un virtuel humain est capable de marcher à un emplacement sélectionné sans entrer en collision avec des boîtes à l'intérieur la salle. Les boîtes sont également soumis à changer de position. En bas à droite sa représentation correspondante par triangulation de delaunay[14].</i>	23
2.1	<i>Interface générale de 3ds Max Design[2]</i>	26
2.2	<i>Interface générale de maya [17]</i>	27
2.3	<i>Interface générale de blender [5]</i>	28
2.4	<i>Les phases de création d'un objet [6].</i>	29

2.5	<i>Exemple de modélisation polygonale[20].</i>	30
2.6	<i>Exemple de modélisation NURBS[4].</i>	31
2.7	<i>Les trois composants principaux du maillage.</i>	32
2.8	<i>Un exemple de translation.</i>	33
2.9	<i>Homothétie de centre O, avec $Sx = Sy = 2$.</i>	33
2.10	<i>Rotation du point P (autour de l'axe des cotes).</i>	34
2.11	<i>Rotation autour de l'axe des abscisses.</i>	35
2.12	<i>Rotation autour de l'axe des ordonnées.</i>	35
2.13	<i>L'application d'une texture sur un cube[27].</i>	36
2.14	<i>Le processus du UV unwrapping.[27]</i>	37
2.15	<i>Application de la texture.[17]</i>	37
2.16	<i>Erojection du rayon primaire [1].</i>	38
2.17	<i>Détermination du premier point d'intersection du rayon primaire avec le maillage[1].</i>	39
2.18	<i>Le rayon d'ombre (marron) ne coupe aucun objet[1].</i>	40
2.19	<i>Le rayon d'ombre (marron) coupe un objet (en vert)[1].</i>	40
3.1	<i>Exemple de modélisation des objets simples.</i>	43
3.2	<i>Exemple de modélisation d'un personnage.</i>	44
3.3	<i>Représentation du cube en mode Objet.</i>	45
3.4	<i>Représentation du cube en mode Edit.</i>	45
3.5	<i>Subdivision du cube de l'objet de base.</i>	46
3.6	<i>Le cube avant l'application de miroir Smooth et le Subsurf.</i>	46
3.7	<i>Le cube après l'application de miroir Smooth et le Subsurf.</i>	47
3.8	<i>Buste de notre personnage.</i>	47
3.9	<i>L'hauteur de corps de notre personnage.</i>	48
3.10	<i>La modélisation de cuisse.</i>	48
3.11	<i>la modélisation du bras et la main.</i>	49
3.12	<i>la modélisation du jambe et le pied.</i>	49
3.13	<i>L'ajout d'un sphère pour la création de tête.</i>	50
3.14	<i>Le personnage final.</i>	50
3.15	<i>Le résultat en mode texture.</i>	51
3.16	<i>Le résultat de texture après la coloration de l'image.</i>	51
3.17	<i>Représentation de l'environnement avant le texture.</i>	52

3.18	<i>Représentation de l'environnement après le texture</i>	52
3.19	<i>L'armature de notre personnage.</i>	53
3.20	<i>L'application de méthode Automatic Weights.</i>	53
3.21	<i>La création de l'action courir.</i>	54
3.22	<i>La trajectoire de l'animation.</i>	54
3.23	<i>Saut de premier personnage avec une force faible.</i>	55
3.24	<i>Saut de deuxième personnage avec une force moyenne.</i>	55
3.25	<i>Saut de troisième personnage avec une force fort.</i>	56

0.1 Introduction générale

La réalité virtuelle transcende ses origines et s'affirme aujourd'hui comme une nouvelle discipline de synthèse au sein des sciences de l'ingénieur, prises ici au sens large des sciences de la conception selon une épistémologie constructiviste. Elle manipule des images interactives, multimodales (3D, sonores, tactiles, proprioceptives), réalistes, animées en temps réel, et partagées sur les réseaux informatiques. En ce sens, elle constitue une évolution naturelle de l'informatique graphique. La réalité virtuelle repose alors sur l'interaction en temps réel avec des objets virtuels et sur la sensation d'immersion dans les mondes virtuels, elle s'apparente à une interface multi sensorielle, instrumentale et comportementale.

La réalité virtuelle contient deux composés l'animation comportementale et l'infographie.

- L'animation comportementale fait allusion aux différentes méthodes dont l'informatique se sert afin de modéliser les comportements des entités qui peuplent un monde virtuel et qui ont comme caractéristiques communes la capacité de percevoir ce monde ainsi qu'un certain niveau d'autonomie.
- L'infographie est un vaste domaine qui englobe toutes les applications orientées graphiques, c'est-à-dire permettant au final la production d'un objet. Le réalisme visuel des objets ne se contente pas de la précision des formes : il lui manque entre autres la couleur, les textures et les ombres. Aussi, l'informatique graphique se posa le problème de l'éclairage des objets par des sources lumineuses.

L'objectif de ce travail est de modéliser plusieurs personnages sportifs (des athlètes), chacun de ces athlètes effectue un saut en longueur selon la valeur d'un attribut de caractère force introduit par l'utilisateur.

Dans le premier et le deuxième chapitre nous aborderons consécutivement l'animation comportementale et l'infographie. Le troisième chapitre sera consacré à la conception et à la modélisation d'une application basée sur l'infographie et l'animation comportementale. Enfin, on finira par une conclusion générale.

Chapitre 1
L'animation comportementale

1.1 Introduction

L'animation (ou simulation) comportementale est une branche de l'animation ayant pour objectif la production de comportements pour les acteurs de l'animation.

L'animation par ordinateur est elle-même issue de l'informatique graphique, une discipline qui a pour but de produire des images grâce à des moyens informatiques.

Aujourd'hui, elle occupe une place prépondérante dans la création de films, de jeux vidéo, d'applications de réalité virtuelle (visites virtuelles de villes ou de lieux historiques, communautés virtuelles, simulations d'entraînement, etc...).

1.1.1 Les exemples d'application de l'animation

La Figure 1.1 présente quelque exemple de de l'animation , l'image (a) représente une scène du film « Mulan » de disney, les images (b, c) représentent respectivement des scènes de jeux vidéo « les sims 2 » et « Second Life », l'image (d) représente une simulation d'entraînement, finalement les figures (e et f) représentent des visites virtuelles dans des lieux touristiques.



FIGURE 1.1 – *Quelques exemples d'applications de l'animation comportementale*[3].

1.1.2 Les premiers travaux d'animation comportementale

Un premier ensemble d'approches a été étudié en parallèle dans la littérature, jusque dans le milieu des années 90, pour la définition du modèle décisionnel : ??

– stimulus/réponse :

se base sur un modèle du comportement dans lequel les actions sont des réponses directe à une simulation de l'environnement. Il utilise la plupart des cas des réseaux neurones formels qui représentent en fait une fonction mathématique dont les paramètres sont les valeurs issues de la perception de l'environnement[9], le principe d'avantage de cette approche est que les réseaux de neurones peuvent subir un processus d'apprentissage à partir d'une base d'exemple de couples (perception, action) qui permet par la suite au système de réagir de façon similaire. L'inconvénient majeur de ce système est qu'il n'est pas interprétable. En effet, un réseau de neurones est constitué de neurones liés entre eux par des transitions ayant chacune une valeur représentant le poids de cette transition. D'autre part, ce sont des modèles de type boîte noire dans lesquels il est impossible de modifier un paramètre sans reprendre le processus complet de configuration. Contrôler un mouvement complexe avec cette approche est de plus très complexe.

– règles de comportement :

Cette approche définit le comportement des objets par un ensemble de règles qui prend en entrée des informations restituant une certaine perception de l'environnement et qui produit en sortie un certain contrôle sur la motricité des objets.

Le comportement des objets est défini par un ensemble de règles et peuvent être représentés par un arbre de décision, chaque branche représentant un comportement différent. Une action satisfaisant les conditions de l'environnement courant sera choisie par l'application d'un algorithme de parcours d'arbre. Le problème de cette approche réside dans la pondération des différents comportements[30]. La solution la plus simple consiste à faire un choix implicite sur l'ordre des règles (suite de si ... alors ... sinon), mais cette solution ne permet pas la spécification de comportements complexes. Le parcours de l'arbre des possibilités avec pondération des branches permet de ne pas privilégier toujours la même règle, tandis que la prise en compte d'une hiérarchie d'experts permet de confronter plusieurs comportements concurrents, le choix final restant à l'expert de plus haut niveau.

– automates d'états finis :

97

L'automate définit les différents enchaînements conditionnels d'action possibles entre comportements[30]. En effet, chaque état de l'automate correspond à une tâche considérée comme unitaire et le passage d'un état à l'autre par les transitions de l'automate permet de modifier l'état et donc le comportement de l'agent. On a donc bien un système pour lequel l'enchaînement des actions dépend du contexte perçu mais aussi de la dernière action effectuée. Différentes approches ont permis d'augmenter le niveau de complexité du comportement d'écrit par des systèmes à base d'automates et aussi d'introduire plusieurs niveaux d'abstraction. Les piles d'automate ont apporté la notion d'appel de fonction. Lorsqu'un automate en appelle un autre. Le premier est empilé alors que le second est lancé. Une fois que le second est terminé. Le premier est dépilé et reprend[9].

Ces trois approches demeurent relativement simples, avec des champs de perception et d'action limités, et en outre ils ne prennent pas en compte l'aspect temporel. Afin de rendre compte de la complexité décisionnelle il est nécessaire de traiter conjointement les aspects continus et discrets, de coordonner les comportements concurrents et de gérer leur structure organisationnelle, c'est pourquoi les deux premières approches ont été assez vite abandonnées au profit de l'approche à base d'automates d'états finis dans leurs versions parallèles et hiérarchiques qui est maintenant couramment utilisée.

1.2 Quelques modèles comportementaux

1.2.1 Modèle « steering »

Ce modèle « steering » de [Reynolds ,1997] [7] est un modèle appliqué à des véhicules. Les comportements de base des véhicules sont : aller vers la cible, fuir la cible, suivre une cible dynamique, fuir une cible dynamique, arriver sur la cible, éviter les obstacles et suivre un chemin. Un véhicule est défini comme une particule qui possède une position, un poids et une vitesse. Cette vitesse est modifiée par l'interaction entre les forces. La direction de mouvement est incluse dans ce modèle pour calculer la nouvelle position.

A chaque pas de simulation, la force « steering » est appliquée à chaque véhicule pour changer sa vitesse. Une nouvelle position est calculée selon la manière suivante :

– $Steering_force = truncate(steering_direction, max_force)$

- Acceleration = steering_force / poids
- Vitesse = truncate (vitesse+ acceleration, max_vitesse)
- Position = position + vitesse

Ce modèle veut maintenir l'alignement de la vitesse dans un système de coordonnées local. Ce système est défini par 4 termes : un vecteur de position spécifiant l'origine locale et 3 termes représentant le repère local. Pour garder l'alignement de la vitesse à chaque pas de simulation, les 3 vecteurs de bases doivent faire des rotations vers la nouvelle direction[7].

1.2.2 Modèle ViCrowd

Introduit par [Musse, 2000][7], l'idée de contrôler les collisions est basée sur la détection des collisions et l'évitement de ces collisions. Ils sont dépendants de la densité des personnages. Avant de tester les collisions, il subdivise l'espace de simulation en sous-espaces où la détection des collisions est prise en compte (Figure 1.2). Avec des personnages dans un sous-espace, la distance permettant entre les personnages est deux mètres (densité forte dans un sous-espace) et de six mètres (densité faible). Grâce à deux positions (courant position et cible) de chaque personnage, on peut calculer une nouvelle position prédictive en fonction de sa vitesse et la détection des collisions est réalisée.

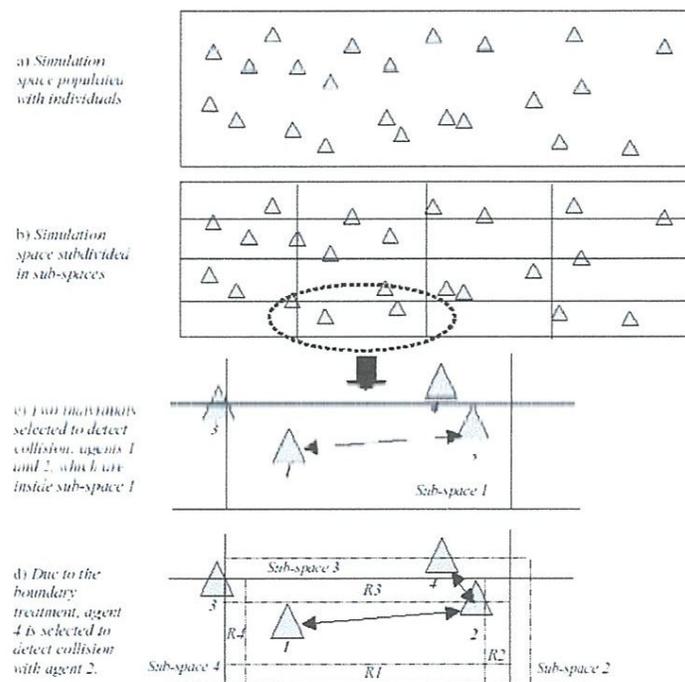


FIGURE 1.2 – Subdivision de l'espace de simulation en sous-espaces [7].

1.2.3 Le système de classeurs (LCS)

Introduits par John Holland en 1976, Les systèmes de classeurs sont des systèmes à base de règles de production qui construisent leur ensemble de règles de façon automatique.

Les classeurs, afin d'identifier et de renforcer un sous ensemble de règles qui sont capables de coopérer pour la réalisation d'une tâche spécifique. Ces systèmes utilisant les algorithmes génétiques[31].

Chaque classeur est formé de deux parties (Figure 1.3) : une règle de type « SI CONDITION ALORS ACTION », La base des travaux de Holland est fondée sur deux idées essentielles. Le premier est que la théorie Darwinienne de survie du meilleur individu peut être utilisée pour conditionner l'adaptation d'un système artificiel à un environnement inconnu. La seconde est qu'un agent peut apprendre à effectuer une tâche juste en interagissant avec un environnement partiellement connu et en essayant de maximiser les récompenses qu'il reçoit de cet environnement en retour de ses actions[29]. La partie condition représente une transcription de l'état



FIGURE 1.3 – Classeur de holland[29].

courant de l'environnement. Elle est codée à l'aide de l'alphabet ternaire (0,1, #) sous forme d'une chaîne de caractères de longueur fixe.

La partie action, ainsi que les messages en provenance de l'environnement, sont codés à l'aide de l'alphabet binaire (0,1) [31].

1.2.4 Modèle de comportements scalables

Introduit par Mankyu Sung et al[18], c'est une approche pour la simulation d'une foule, Dans cette approche, la foule est simulée sur deux niveaux (Figure 1.4).

Au haut niveau, nous adopter un mécanisme de contrôle distribué à base de situation donne à chaque agent dans une foule de détails spécifiques sur la façon de réagir à un moment donné en

qui??
fonction de son environnement local.

Le niveau bas, nous utilisons un système de probabilité qui calcule les probabilités sur les transitions d'état, puis des échantillons pour déplacer la simulation de l'avant[18].

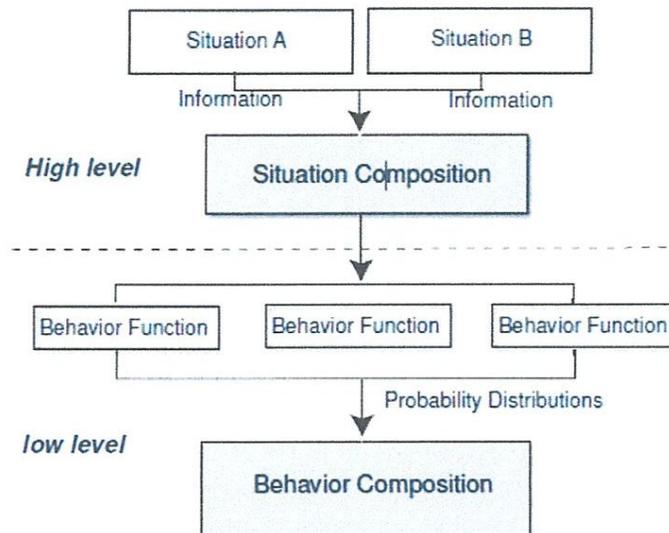


FIGURE 1.4 – Architecture d'agent à deux niveaux[18].

Les auteurs utilisent les probabilités dans les choix des actions des agents de la foule (Figure 1.5), ils supposent que les individus de la foule ne sont pas généralement connus, donc les comportements sélectionnés seront aléatoires selon la situation auxquelles ils sont soumis. La principale raison de l'utilisation des fonctions de comportement est que nous sommes capables de composer plusieurs comportements de composants pour synthétiser des comportements plus complexes qui incarnent toutes les influences des comportements composants. Composition de comportements est simplement la multiplication des probabilités des fonctions comportementales produisant.

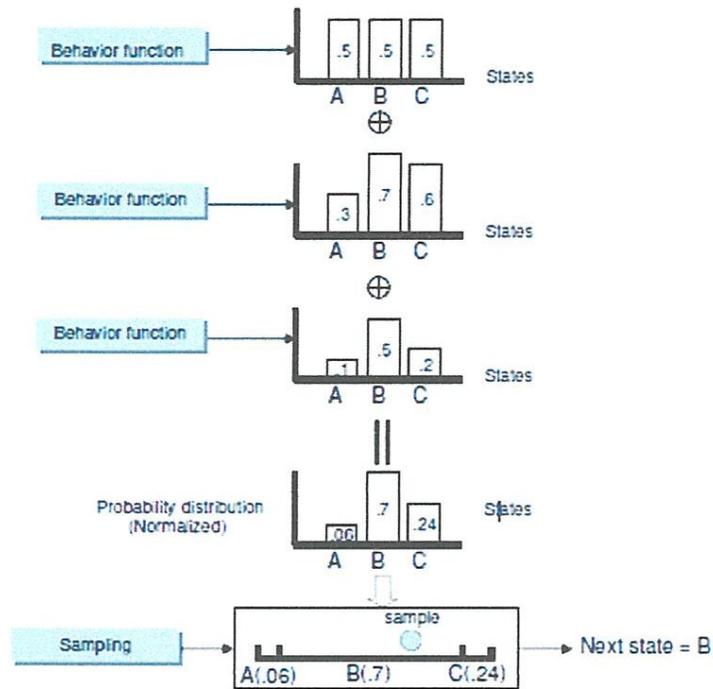


FIGURE 1.5 – Les fonctions de comportement calculer la probabilité d’entrée indiqe indépendamment. Les distributions de probabilité qui sont calculées à partir des fonctions de comportement sont alors composées. Enfin, l’état suivant est sélectionné par échantillonnage sur la distribution de probabilité composée[29].

1.2.5 Le modèle HiDAC

Le modèle HiDAC « High-Density Autonomous Crowds » [24], a pour objet de simuler les comportements correspondants au mouvement local dans les larges foules et peut simuler de façon réaliste un individu qui consiste à trouver son chemin, Ces comportements émergents sont entraînés par la combinaison de facteurs psychologiques (impatience, panique) afin de fournir une variété dans les comportements des agents, Ce qui leur permet de voir les conséquences de certains comportements sur le comportement émergent de la foule (par exemple, un agent sujet à la panique et qui a un degré d’impatience important, peut en se déplaçant faire tomber ou pousser d’autres agents et ainsi provoquer un blocage).

Dans HiDAC, chaque agent est doté d’une perception et il est capable de réagir aux objets statiques et dynamiques (par exemple, d’autres agents) qui se trouvent dans son espace.

Les comportements des agents sont calculés sur deux niveaux (Figure 1.6) :

Le haut niveau (la navigation, la communication, et la prise de décision) et le Bas niveau (La

perception, le mouvement de l'agent, la locomotion).

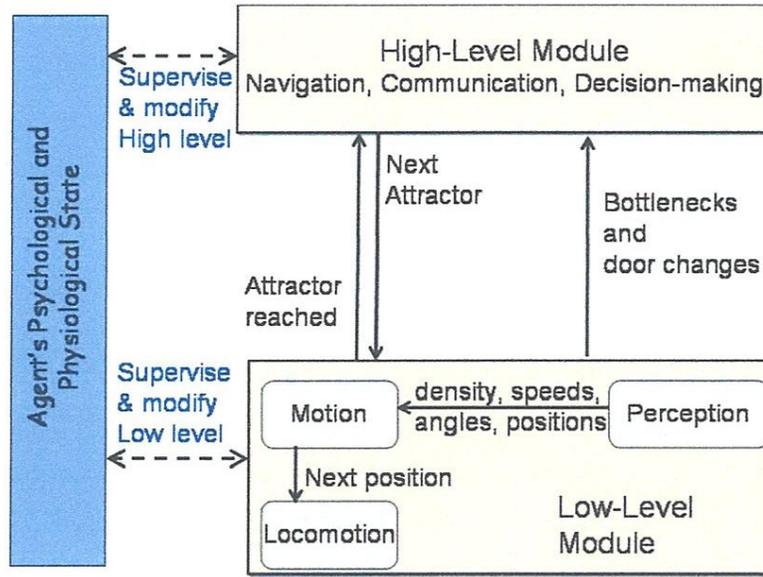


FIGURE 1.6 – Les deux niveaux du système HiDAC[24].

Le module du haut niveau permet de prendre des décisions concernant le prochain endroit à visiter en se basant sur les informations qu'il reçoit via la communication concernant l'état de l'environnement (un chemin à prendre qui mène à la destination), et les changements de porte qui ont été perçus par l'agent, Une fois la de haut niveau décide de la prochain endroit a été choisi, alors le haut niveau envoie cette information au bas niveau pour que ce dernier effectue les mouvements nécessaires pour l'atteindre [22].

Le sous-module de mouvement interroge le sous-module de Perception sur les positions et les angles des obstacles, foule densité avant de l'agent, et la vitesse de dynamiques obstacles. Sur la base de l'information perçue et interne état de l'agent (comportement actuel, la panique, l'impatience, etc.), le sous-module calcule la vitesse de mouvement et suivant la position de l'agent et envoie des informations à l'Locomotion sous module pour exécuter les pieds correcte mouvements[22].

1.2.0 Le système MAGS

Le système MAGS « Multi-Agent Geo-Simulation » est un simulateur multi-agents développé au sein du laboratoire informatique de l'université de Laval, permet de générer des géo-simulations impliquant des milliers d'agents (la perception, la navigation, ce qui rend les

capacités de décision) immergé dans des environnements géographiques virtuels[19].

Les agents dans MAGS (définis dans le module Agents, Figure 1.7) sont capables d'accomplir des activités spatiales et cognitives dans le système de géo-simulation. Pour ce faire, les agents sont capables de percevoir l'environnement spatial avec ce qu'il contient comme objets. L'environnement spatial doit être généré à partir des données stockées dans des systèmes d'information géographique SIG. Concrètement, l'environnement (défini dans le module Environnement, Figure 1.7) dans MAGS est représenté par une grille de cellules. Chaque cellule (correspondant à un bitmap) encapsule les informations pertinentes sur la portion du terrain qu'elle représente. L'accès à ces informations est à la base des capacités spatiales et cognitives dont bénéficient des agents MAGS.

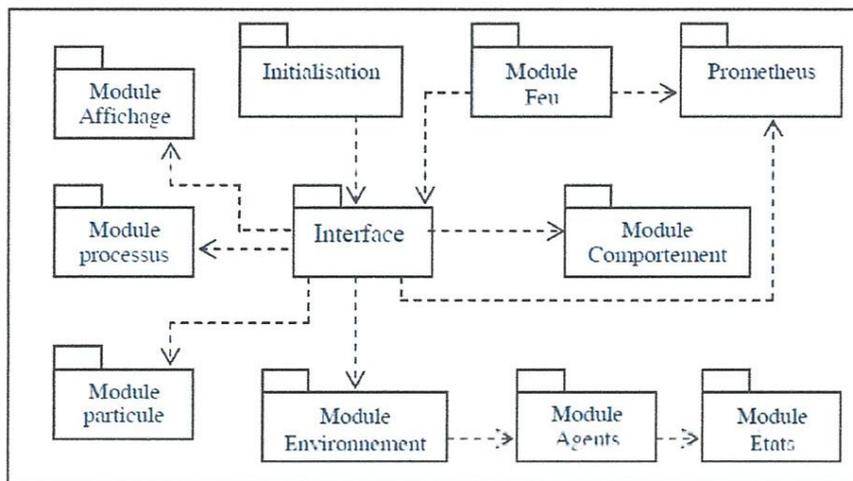


FIGURE 1.7 – Architecture de MAGS [19].

1.2.7 Le système MACES

Le système MACES « Multi-Agent Communication for Evacuation Simulation », est un système de simulation pour l'évacuation de foules. Dans MACES, on trouve trois grands genres d'individus qui peuvent apparaître dans une foule lors d'une évacuation :

Un leader non expérimenté est un agent qui peut aider les autres durant le processus d'évacuation en cherchant de nouveaux chemins, tout en faisant évoluer sa carte mentale de l'environnement à travers sa propre expérience ou via la communication. Les agents non leaders modélisent les personnes qui perdent leurs moyens lors des situations d'urgences où ils sont incapables de prendre la bonne décision et préfèrent suivre les autres. Contrairement aux agents leaders qui

représentent les personnes qui gardent leur calme (maîtrisent leur stress) lors des situations d'évacuation d'urgences[29].

La Figure 1.8 montre les différents comportements émergents obtenus en utilisant un pourcentage élevé de leaders vs faible pourcentage. L'image de gauche montre le pourcentage élevé de leaders qui ne connaissent pas l'environnement qui ont tendance à explorer la recherche de sorties. Par conséquent, nous pouvons observer de petits groupes de personnes à la recherche de l'espace et le partage d'informations qui donne un temps d'évacuation plus rapide. L'image de droite montre un petit pourcentage de leaders et donc un grand nombre d'adeptes. En conséquence, le comportement émergent se compose de quelques grands groupes de personnes errantes ensemble, après les autres et incapables de faire leur propres décisions, ce qui donne un plus temps global d'évacuation.

MACES se composent de deux niveaux : un haut niveau et un bas niveau.

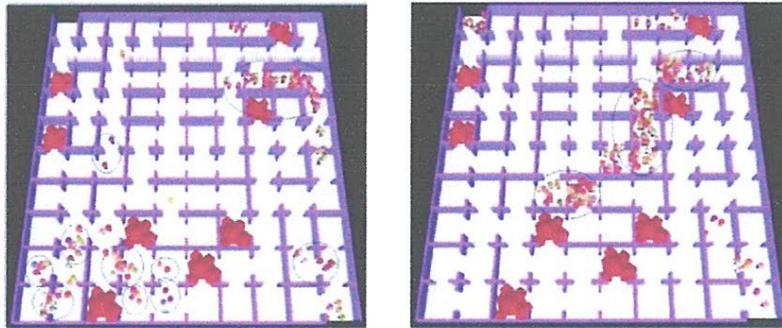


FIGURE 1.8 – Exemple séquence d'images de l'évacuation de la foule avec haute ou. Faible pourcentage de leaders[25].

Le haut niveau de MACES comporte trois étapes principales :

- choisir le chemin le plus court.
- mettre à jour la carte mentale.
- communication et partage des informations.

La première étape choisir le chemin le plus court à emprunter en prenant en compte l'état de sa carte mentale, L'agent réitère l'action toutes les fois qu'il tombe sur un chemin bloqué. Dans la deuxième étape, Les cartes mentales seront élargies comme agent explore les informations d'environnement et partage avec d'autres agents de la foule par la communication processus. Dans la troisième étape correspond à la communication à l'intérieur des pièces. Elle implique des informations sur la localisation des risques de blocage des chemins et des directions pos-

sibles dans une pièce qui ont été explorées[23].

Le bas niveau correspond au mouvement local à l'intérieur des pièces, lequel, est basé sur le modèle de Helbing. Il correspond au processus de déplacement d'un point attracteur (points d'intérêt) à un autre. Chaque point est placé à l'extérieur des portes correspondantes aux pièces (chambres), et chaque pièce comporte plusieurs portes (donc un point d'intérêt à chaque porte). A chaque fois qu'un agent accède à un point, alors il récupère de ce point des informations concernant la vitesse et la direction à prendre pour arriver au prochain point[29].

Pour obtenir un comportement humain plus réaliste, les auteurs ont défini un modèle nommé PMFserv, dont l'objet est de tenir compte d'éléments psychologiques. PMFserv est un système qui permet d'implémenter des modèles de comportements humain à partir les émotions et sources de motivation[23].

Les informations sont posées sur la mémoire de travail de telle sorte que chaque couche dépend des couches en dessous d'un cycle de décision de l'agent donné (voir Figure 1.9).

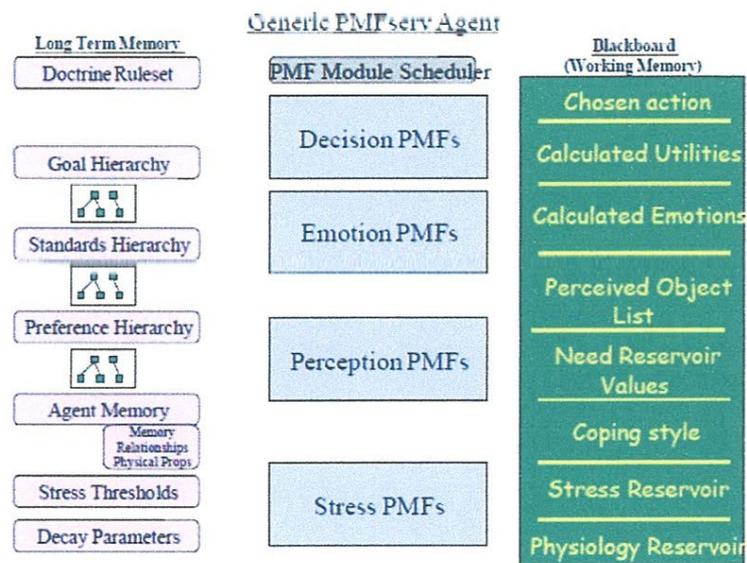


FIGURE 1.9 – Architecture de PMFserv[29].

L'observation de la mémoire de travail de bas en haut révèle le cycle de décision d'un seul agent. Les données physiologiques à travers une série de mesures (y compris l'excitation, la fatigue, la faim, soif, blessure, etc) sont combinés pour définir les niveaux d'une série de réservoirs de stress. Qui vont déterminer des éléments de style d'adaptation, chaque objet dans le système exécute ses règles de perception pour déterminer quels objets sont actuellement perceptibles par l'agent. Pour déterminer l'utilité émotionnelle d'une action en utilisant le modèle émotionnel

OCC qui génère 11 paires d'émotions. En additionnant ces émotions ils obtiennent une valeur d'utilité de l'action considérée. Ce processus est terminé pour chaque action offerte à la disposition de l'agent. L'action de la valeur d'utilité la plus élevée est sélectionnée et exécuté[23].

Dans la figure définit la combinant entre le système de simulation de foule (MACES) et le modèle psychologique (PMFserv). PMFserv fournit les motivations de l'agent, le stress, le style d'adaptation, les émotions, la personnalité et les décisions.

PMFserv met en uvre le processus de prise de décision tout en MACES gère les processus de planification situé à quelques (l'orientation et de mouvement local). Ainsi exécuter sa cognition pour déterminer les décisions d'action collective et individuelle et de transmettre des instructions de retour à MACES pour mener à bien les actions qui en résultent. La Figure 1.10 met en évidence la boucle d'interaction principale.

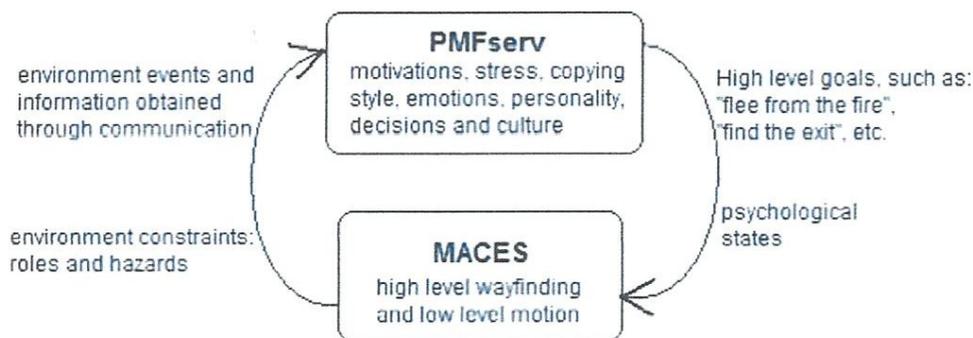


FIGURE 1.10 – La boucle d'interaction principale[23].

1.3 Représentation de l'environnement

Il ya plusieurs types pour représenter l'environnement Simple (basé sur les grilles de cellules, les champs de potentiels) et complexes (basée sur les Triangulations de Delaunay, modèles multi-couches).

1.3.1 Représentation des environnements simples

– Représentation basée sur les grilles

Les méthodes basées sur les grilles sont des méthodes simples (Figure 1.2), ils permettent de traiter la détection de collision[32], La décomposition à base de grille est utilisée par les

modèles d'automates cellulaires, elles sont idéalisation mathématique des systèmes physiques dans lesquels l'espace et le temps sont discrets, et les quantités physiques prennent un ensemble fini de valeurs discrètes. Un automate cellulaire est constitué d'un réseau régulier uniforme. L'état d'un automate cellulaire est complètement défini par les valeurs des variables à chaque cellule, Leur valeur est fonction des valeurs des variables des cellules voisines par l'application de règles locales situées aussi dans les cellules[33].

– Représentation basée sur les champs de potentiels

Introduite par Oussama Khatib[15], les méthodes basées sur les champs de potentiels sont des méthodes simples, l'avantage de cette méthode la capacité à générer en ligne anti-collision, chaque point de l'environnement est affecté d'un gradient de force, l'objet essaie de se déplacer robot pour être attiré par l'objectif tout en étant repoussé par les obstacles (Figure 1.11). L'environnement est représenté par un champ de potentiel (localement)

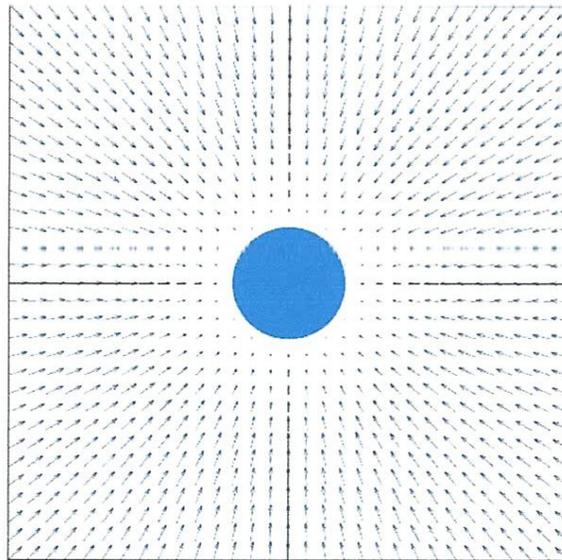


FIGURE 1.11 – *Un champ potentiel attractif[11].*

1.3.2 Représentation des environnements complexes

– Représentation multi-couches

Les modèles multi-couches ont pour objet de représenter les environnements ayant des structures topologiques complexes, Elle est illustrée sur la Figure 1.12, qui comprend des infor-

mations complètes et suffisantes environnement pour simuler la foule dans un environnement complexe. Modèle multicouche subdivisé en trois couches.

1. Niveau géométrique :

La partie principale de cette couche est le modèle géométrique 3D de l'environnement qui est utilisé pour l'affichage et l'extraction de l'information sémantique pour la prochaine couche de représentation sémantique.

2. Niveau sémantique :

La couche sémantique est représentée par :

- Des cartes de structures, il existe de nombreux objets, comme les mains courantes, les murs, les fenêtres...etc. Tous les objets placés dans la même région sont inclus dans le même bloc. Tous les blocs ainsi que leurs objets sont organisés en structure de la carte.
- Carte topologique est principalement utilisé pour stocker et récupérer les relations conjonctives entre les régions séparées (blocs).
- Des cartes de hauteurs, Les coordonnées de la hauteur de l'environnement sont indispensables pour l'animation des foules en particulier dans un environnement multicouche. Ils sont utilisés pour stocker l'information concernant les élévations numériques de surface.

3. Niveau application :

Aux fins de fournir une interaction efficace entre les piétons et l'environnement. Les informations de niveau sémantique pour générer des cartes de haut niveau telles que :

- La carte de perception des objets dans laquelle les blocs sont divisés en sous régions en utilisant une grille de cellules uniformes ; le but est de limiter la recherche des objets à une liste de cellules au lieu de tout le bloc.
- La carte de perception individuelle, Comme la carte de la perception des objets, une grille uniforme à deux dimensions dans chaque bloc est utilisée pour stocker tous les individus se trouvant au niveau du bloc.
- Le plan du parcours individuel, La plupart des systèmes de simulation de foule sont orientées vers un but. Chaque individu sera assigné un but lorsque l'individu est initia-

lisée. La carte topologique à base de blocs est utilisée pour générer la carte individuelle de chemin.

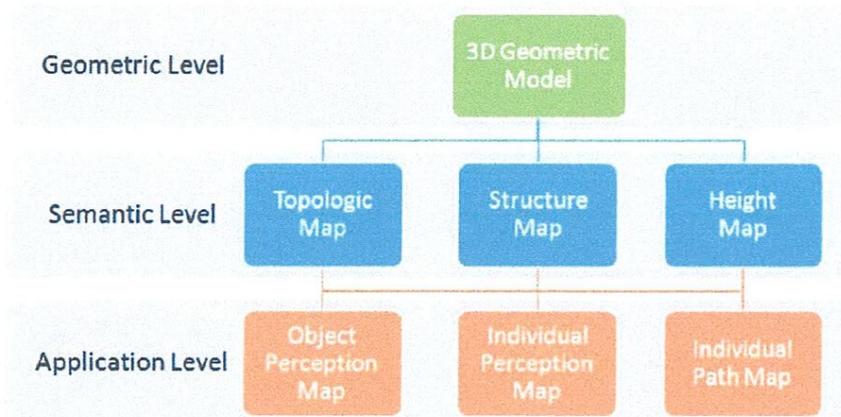


FIGURE 1.12 – *Modèle d'environnement Hiérarchie*[12].

– Représentation basée sur les Triangulations de Delaunay

Les triangulations de Delaunay offrent une représentation de l'environnement plus précise que les représentations à base de grille pour les environnements complexes. La triangulation de Delaunay crée un ensemble de triangles en fonction d'un degré de précision qui dépend de la courbure de l'espace[29].

Le fait que chaque point est relié à son plus proche voisin, cela permet de l'utiliser pour représenter l'espace de navigation (Figure 1.13).

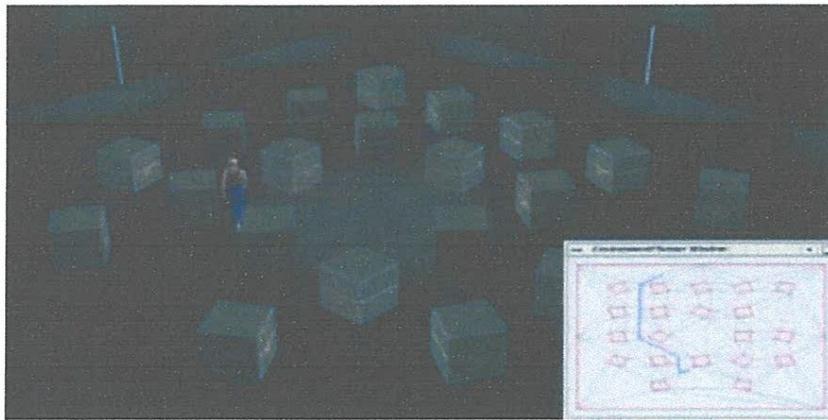


FIGURE 1.13 – Capture d'écran d'une application interactive où un virtuel humain est capable de marcher à un emplacement sélectionné sans entrer en collision avec des boîtes à l'intérieur la salle. Les boîtes sont également soumis à changer de position. En bas à droite sa représentation correspondante par triangulation de delaunay[14].

1.4 Conclusion

L'animation comportementale vise à aborder une dimension nouvelle de l'animation par ordinateur.

Elle est constituée d'un ensemble d'objets dynamiques doués d'une certaine autonomie et dont les évolutions dépendent des interactions qui peuvent se produire sous des formes très variées. Nous avons vue dans ce chapitre quelques modèles de l'animation comportementale ainsi que différents types de représentation des environnements.

Chapitre 2

L'infographie

2.1 Introduction

L'infographie (aussi appelée image de synthèse) est la science de l'image numérique. Elle définit les graphismes créés et gérés par ordinateur, dont la gestion se base sur une station graphique composée d'éléments matériels comme le processeur et la carte graphique (leurs puissances déterminent la vitesse de réponse et de calcul des unités graphiques).

Le terme infographie formé à partir de l'informatique et de graphisme. A l'origine il s'agit d'une appellation déposée en France par la société Benson en 1974. Mais on le mentionne depuis 1973 (IBM Systems Journal) à intervalles irréguliers, depuis la création expérimentale d'hologrammes par ordinateur[10].

Au début des années 1990, les graphistes disposent de plusieurs outils (logiciels). Les nouveaux artistes numériques se sont approprié le terme d'infographiste alors réservé aux ingénieurs concepteurs de ces outils [10].

L'infographie s'empare de la gestion, la médecine, la télévision, l'industrie du spectacle (jeux vidéo) et des films (Batman Begins et The Dark Night) ainsi que toutes les disciplines scientifiques, mathématiques, aéronautiques, mécaniques et tous les domaines de conception en général.

Parmi les logiciels les plus utilisés en infographie on trouve Autodesk 3dsMax (3d), Autodesk Maya (3d), blender.

2.1.1 Autodesk 3dsMax (3d)

3D Studio (l'appellation d'origine) c'est un logiciel de création et d'animation 3D, dont la première version date de 1990. Au début, la société s'appelée Autodesk, est ensuite elle est devenue en 1996 Kinetix, puis a fusionné en 1999 avec la société DiscreetLogic pour devenir Discreet. En 2005, la version du logiciel est 3DS Max 7.5 et en avril 2008 la version sortie 8.04 [6].

3DS Max est un des logiciels de création 3D les plus répandus, en raison de sa puissance. L'un de ses atouts est qu'il peut être étendu par le biais de plugins et de scripts développés par d'autres sociétés, qui permettent de lui ajouter de nouvelles fonctionnalités [6].

3dsMax est surtout utilisé par les développeurs de jeux et de films d'animations. Une autre version de ce logiciel est ; 3dsMaxDesign est plus adapté aux architectes, concepteurs, ingénieurs et spécialistes de la visualisation[2].

– Présentation l'interface de 3dsMax

La Figure 2.1 présente les commandes et les onglets les plus souvent utilisés dans 3ds Max Design.

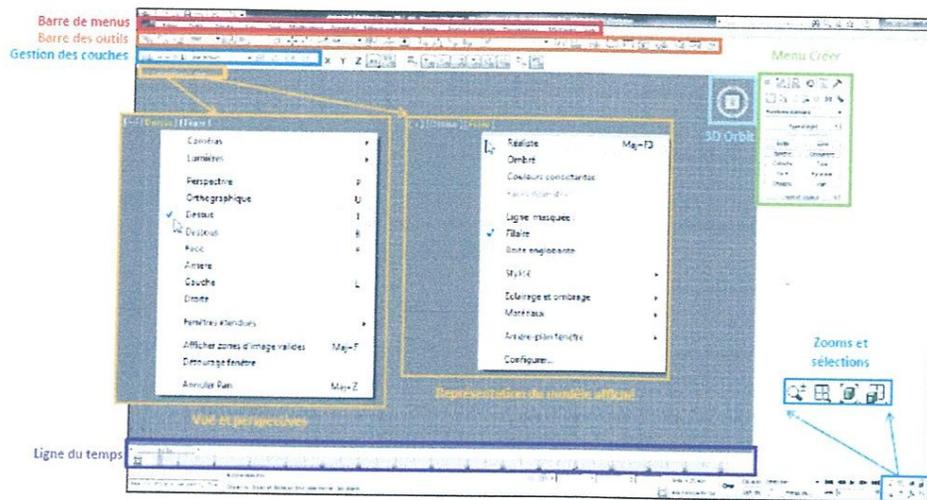


FIGURE 2.1 – Interface générale de 3ds Max Design[2]

2.1.2 Autodesk Maya (3D)

C'est un logiciel commercial (payant) Anciennement développé par Alias[8], il a hérité des technologies des premiers logiciels d'animation (Alias Power Animator, puis Alias Motion Builder qui est aujourd'hui leader en capture de mouvements) et de modélisation surfacique. Maya utilisé dans le domaine de l'animation où il est une référence. Les maisons d'effets visuels les plus connues telles que Industrial Light and Magic, et les jeux vidéo tels que Nintendo, Sony Computer Entertainment... [16]

– Présentation l'interface de maya

La Figure 2.2 suivante présente l'interface de maya.

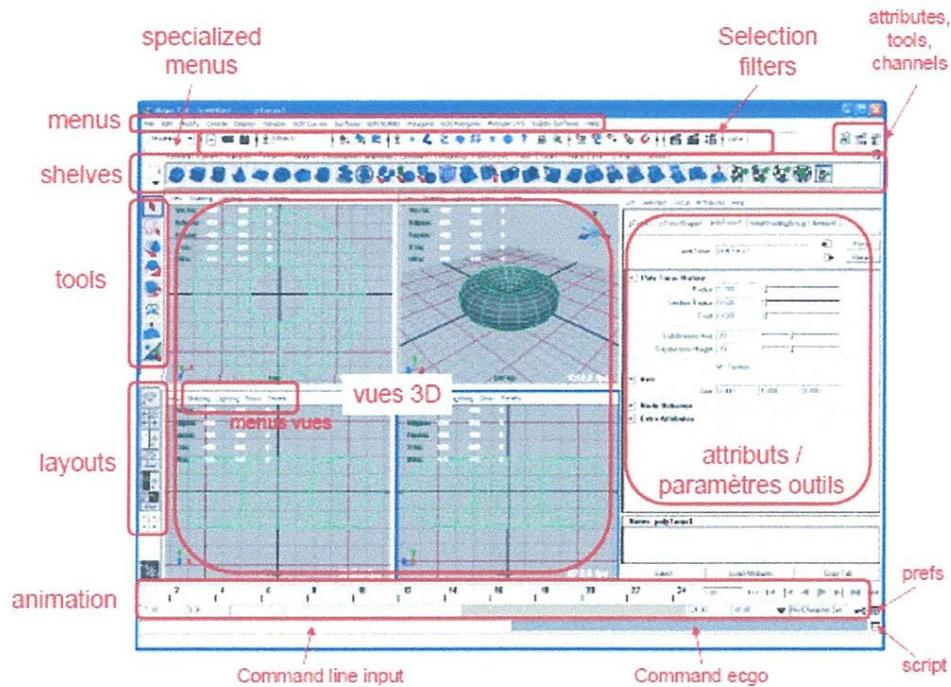


FIGURE 2.2 – Interface générale de maya [17]

2.1.3 Blender

Blender est un formidable outil graphique, permettant de créer des infographics à partir de scènes 3D, utilisé dans de nombreux films d'animation (Titanic, Avengers,...), l'art, et les jeux vidéo (Running Guys, Dead Cyborg)... [28]

– Présentation l'interface de blender

La Figure 2.3 suivante présente l'interface de blender

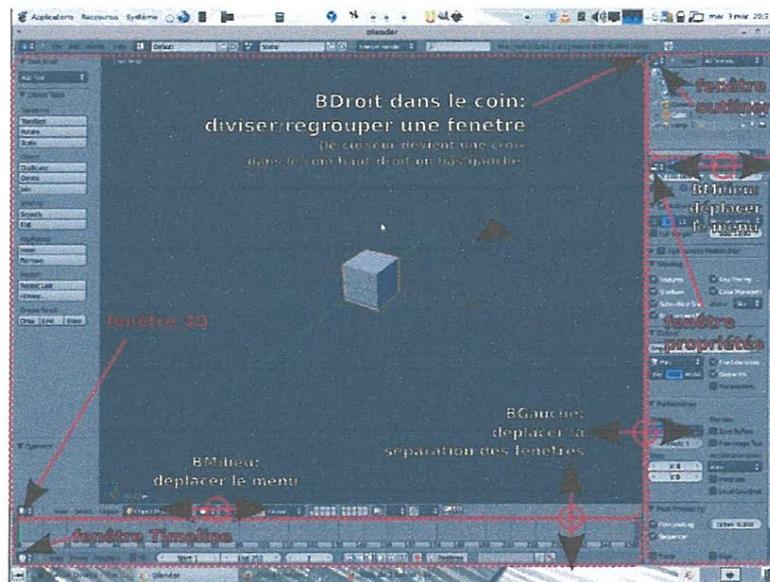


FIGURE 2.3 – Interface générale de blender [5]

2.2 Les étapes de création d'un objet 3D

2.2.1 La modélisation dans l'espace virtuel 3D

Tous les logiciels de 3D fonctionnent de la même manière. La modélisation consiste en la création d'objets. Qui est l'étape de base pour la réalisation d'une scène.

Afin de concevoir des objets dans votre scène en créant des objets standard, tels que des géométries 3D et des formes 2D, puis en appliquant des modificateurs à ces objets.

– La Modélisation avec 3ds Max

Pour créer des objets on sélectionne une catégorie d'objets dans le menu Créer, puis on clique dans la fenêtre pour définir les paramètres de l'objet. Dans La Figure 2.4 présente Les phases de création d'un objet (dans la première phase définit le Diamètre de pincer et la deuxième phase définit la hauteur de cylindre, la troisième phase augmentés les Côtés et la quatrième phase Segmenté la hauteur de cylindre). Le menu Créer comporte les fonctions de base suivantes : Géométrie, Formes, Lumières, Caméras, Assistants. Chaque catégorie contient de multiples sous- catégories [6].

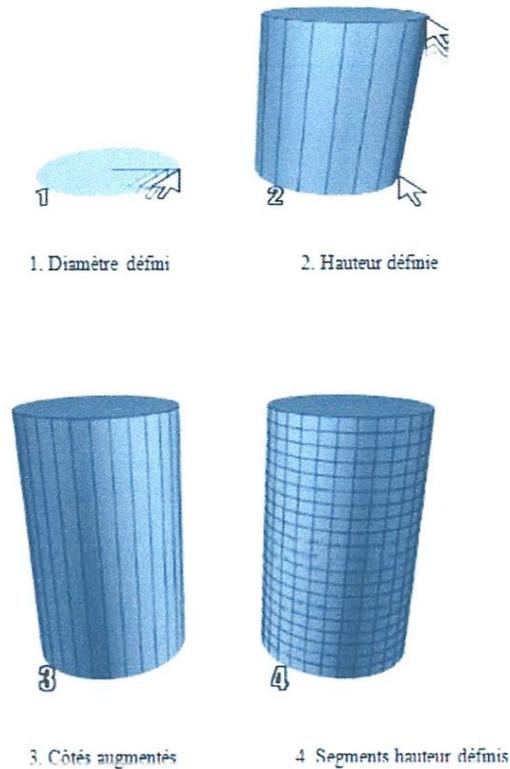


FIGURE 2.4 – Les phases de création d'un objet [6].

– La modélisation avec maya

Dans maya il y a deux façons de modéliser un objet 3D, la modélisation polygonale et la modélisation NURBS.

1. La modélisation polygonale :

C'est la modélisation la plus utilisée dans maya. Les surfaces polygonales peuvent être employées pour créer pratiquement n'importe quel type de forme et sont particulièrement efficaces pour créer des formes simples[20]. La Figure 2.5 présente un exemple d'une forme simple définie par des plusieurs faces et des arêtes.

Les polygones comprennent des surfaces planes appelées faces. La modélisation polygonale se fait à partir de primitif 3D comme un cube, une sphère, un cylindre et se traduit par la manipulation des composants qui les constituent : on parle alors de sommets (vertex), d'arêtes (edge), et de faces (face).

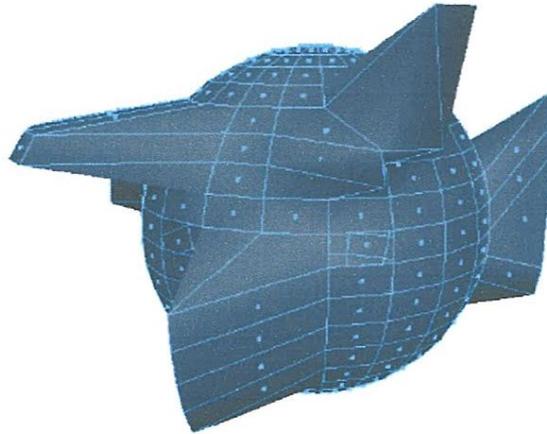


FIGURE 2.5 – *Exemple de modélisation polygonale[20].*

2. La modélisation NURBS

NURBS signifie Non Uniform Rational Bspline [26]. Ce sont des courbes mathématiques (comme la Figure 2.6) qui sont représentées par un ensemble d'équations. Ces courbes possèdent un point de contrôle qui permet de modifier sa forme. Elles sont faciles à l'emploi et peuvent approcher facilement le modèle demandé mais elles possèdent une flexibilité très limitée, ce qui rendra la modélisation difficile.

Les NURBS trouvent aujourd'hui leur utilisation dans la productique de matériels de très haute précision par exemple la microélectronique et le tracé des routages au sein des composants, ou la fabrication d'instruments de mesures comme les miroirs et lentilles utilisés en astronomie.

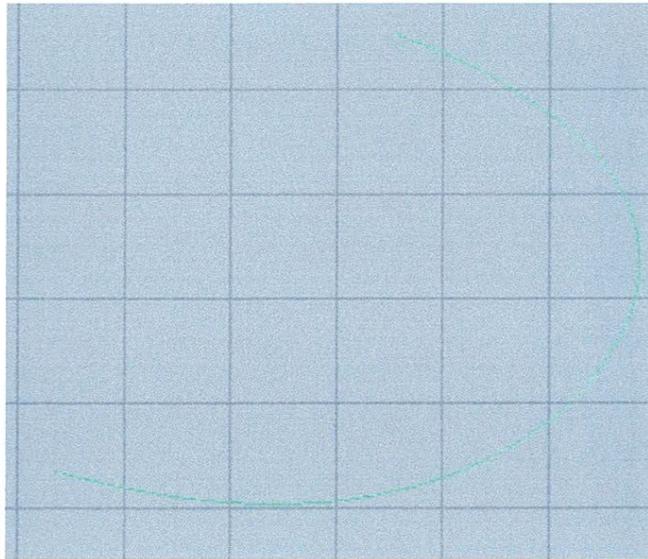


FIGURE 2.6 – Exemple de modélisation NURBS[4].

– Le maillage

C'est le nom donné à tout objet modélisé dans un espace tridimensionnel. Un maillage est un ensemble de sommets reliés entre eux par des arêtes[13], Il est tout d'abord composé de points, appelés vertices dans le langage des infographistes. Il comprend aussi des arêtes et des faces, Dans la (Figure 2.7) contient Les trois composants du maillage (la première phase contient les vertices et la deuxième définit les arêtes de polygone, la troisième phase contient les faces de polygone).

De façon générale, un logiciel d'infographie 3D enregistre les données concernant ces trois composants du maillage.

- Un vertice : C'est non seulement un point dans l'espace, mais c'est une unité à laquelle l'ordinateur associe des informations sur la position (du type $p(x ; y ; z)$), la couleur, etc.
- Une arête : C'est un segment qui relie deux vertices.
- Une face : Pour un logiciel d'infographie 3D, une face est un triangle composé de trois arêtes refermées et connectées par leurs sommets.

Plusieurs caractéristiques les distinguent :

- Si les sommets sont tous situés dans un même plan, le maillage sera 2D. Dans le cas contraire, ce sera un maillage 3D.
- Les éléments du maillage peuvent être de différents degrés. Si tous les éléments sont des triangles, on parle alors de maillage triangulaire. Si tous les éléments sont des tétraèdres, le

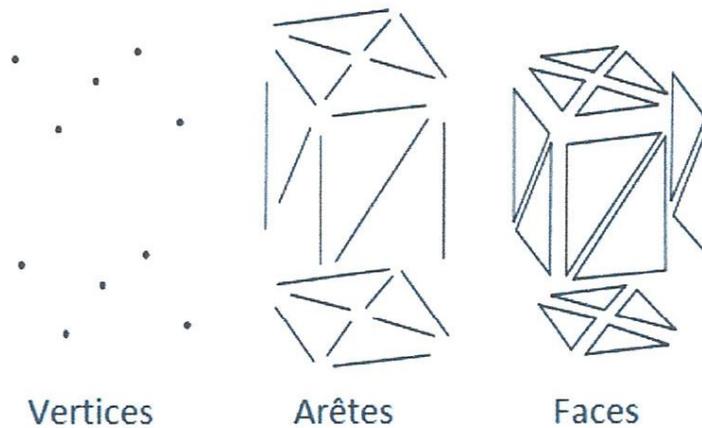


FIGURE 2.7 – *Les trois composants principaux du maillage .*

maillage est dit tétraédrique.

– La connectivité d’un maillage définit le type de connexions entre ses éléments. Un maillage est structuré si sa connectivité est fixe et identique pour chaque élément du maillage, par exemple, une grille. Un maillage est non-structuré si sa connectivité est quelconque, par exemple, la plupart des maillages triangulaires[13].

Différents types de manipulation peuvent être appliqués lors de la création de l’objet. Parmi ces manipulations on trouve : La translation, l’homothétie et la rotation.

– **La translation :**

Une translation (vectorielle) est simplement un déplacement d’objet.

Une translation de vecteur $\vec{u}(a; b; c)$, consiste à associer à un point $P(x; y; z)$ un point $P'(x'; y'; z')$ tel que : $x' = x + a$; $y' = y + b$; $z' = z + c$ Dans la Figure 2.8, nous voyons un exemple de translation d’objet. Elle a été faite dans le plan et non pas dans l’espace uniquement pour simplifier la démonstration. Le carré rouge est l’image du carré noir par la translation vectorielle $\vec{u}(2; 1)$.

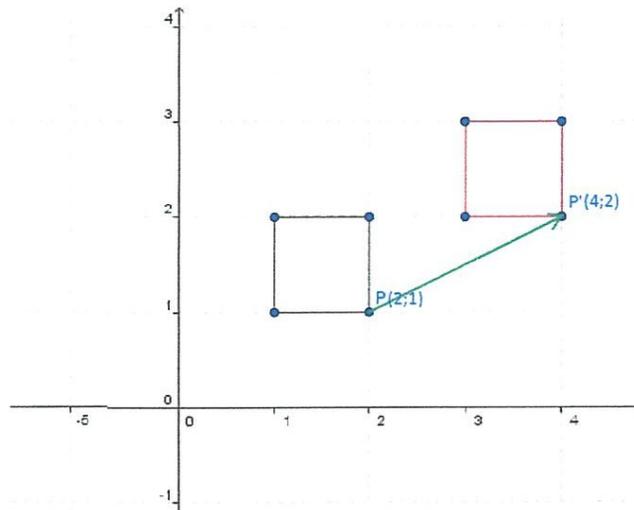


FIGURE 2.8 – *Un exemple de translation.*

– **L’homothétie :**

Parmi les manipulations les plus utilisées en infographie 3D, figurent les changements d’échelle (agrandissements ou rétrécissements) ; c’est pour cela que l’homothétie est utilisée. Dans la Figure 2.9 ci-dessous un exemple l’homothétie d’objet.

L’homothétie de centre O , centre du repère (tridimensionnel) consiste à associer à un point $P(x; y; z)$, un point $P'(x'; y'; z')$ tel que : $x' = x \times Sx$; $y' = y \times Sy$; $z' = z \times Sz$
 Sx , Sy , et Sz sont appelés des scalaires. Ce sont des réels qui « contrôlent » l’agrandissement (ou le rétrécissement) que l’on veut faire subir à un maillage sur les différents axes du repère (respectivement l’axe des abscisses, l’axe des ordonnées et l’axe des cotes).

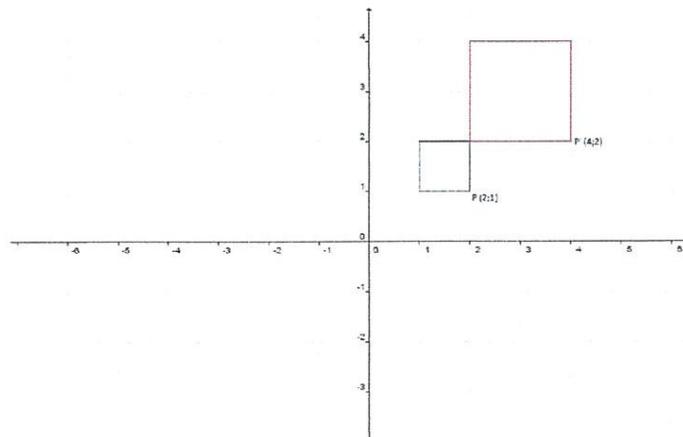


FIGURE 2.9 – *Homothétie de centre O , avec $Sx = Sy = 2$.*

– La rotation :

La rotation est une des manipulations de base en infographie 3D. Elle consiste à faire pivoter un objet (maillage) dans l'espace, autour d'un axe donné. En effet, les infographistes décomposent une rotation dans l'espace en trois mouvements : la rotation autour de l'axe des abscisses, des ordonnées, puis des côtes.

1. La rotation autour de l'axe des cotes :

La Figure 2.10 présente la Rotation du point P , Elle associe à un point $P(x; y; z)$, un point $P'(x'; y'; z')$, tel que : $x' = \cos(\alpha + \beta)xOP$

(Car $x'/OP = \cos(\alpha + \beta)$, avec OP : distance de P par rapport à l'origine O ; α : angle de rotation initial de P ; β : angle de la rotation à effectuer)

$$y' = \sin(\alpha + \beta)xOP$$

$$z' = z$$

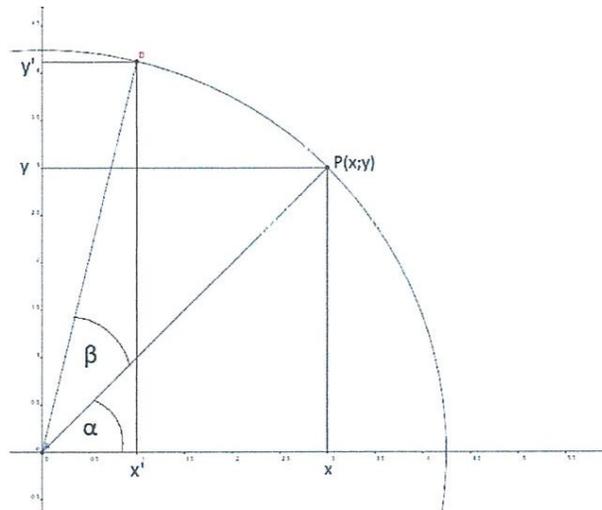


FIGURE 2.10 Rotation du point P (autour de l'axe des cotes).

2. La rotation autour de l'axe des abscisses :

La Figure 2.11 présente la Rotation autour de l'axe des abscisses, Elle associe à un point $P(x; y; z)$, un point $P'(x'; y'; z')$, tel que : $y' = \cos(\alpha + \beta)xOP$

(Avec α : angle de rotation initial de P autour de l'axe des abscisses ; β : angle de la rotation à effectuer autour de l'axe des abscisses)

$$z' = \sin(\alpha + \beta)xOP \quad x' = x$$

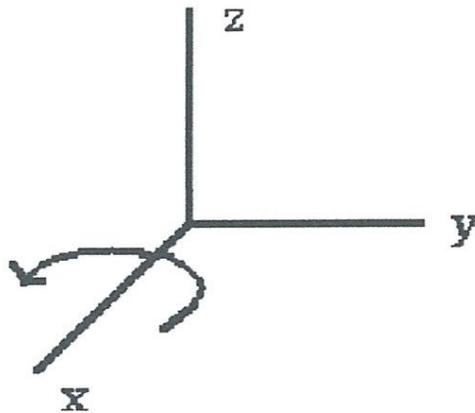


FIGURE 2.11 – Rotation autour de l'axe des abscisses.

3. La rotation autour de l'axe des ordonnées :

La Figure 2.12 présente la Rotation autour de l'axe des ordonnées, Elle associe à un point $P(x; y; z)$, un point $P'(x'; y'; z')$, tel que : $z' = \cos(\alpha + \beta)xOP$
 (Avec α : angle de rotation initial de P autour de l'axe des ordonnées ; β : angle de la rotation à effectuer autour de l'axe des ordonnées)

$$x' = \sin(\alpha + \beta)xOP$$

$$y' = y$$

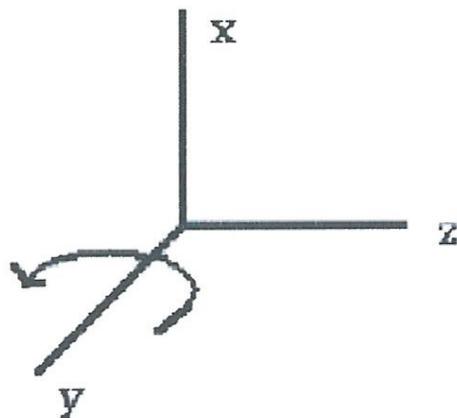


FIGURE 2.12 – Rotation autour de l'axe des ordonnées.

– La visualisation d’une scène virtuelle

– Les textures :

Les textures sont des images servant de « vêtements » pour un maillage. Effectivement, dans la réalité, les objets ont des caractéristiques spécifiques à leurs surfaces. Les traces dans le bois, la rugosité des pierres, la transparence des feuilles font partie de ces caractéristiques dites de surface d’un corps quelconque. Mais la modélisation en 3D des objets n’est qu’une représentation de leur volume et de leur aspect géométrique.

En 1974, Edwin Catmull (de Pixar) crée une méthode appelée le texture-mapping. Il s’agit d’appliquer une image « normale » (à deux dimensions) sur la surface d’un maillage tridimensionnel. On peut imaginer par exemple une boîte (qui représenterait le maillage) et du papier cadeau (qui représenterait la texture). Le texture-mapping consisterait à emballer la boîte avec le papier cadeau (comme dans la Figure 2.13)[1]. Plus exactement, le Texture-mapping se décompose en

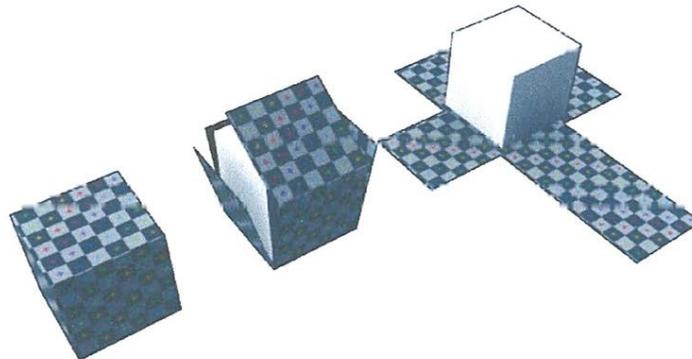


FIGURE 2.13 – *L’application d’une texture sur un cube*[27].

deux parties : l’UV Unwrapping, et l’application de la texture.

1. UV Unwrapping :

C’est un processus par lequel l’ordinateur détermine le patron d’un maillage (comme la Figure 2.14 détermine le patron en 2 dimensions d’un objet 3D). Effectivement, en Français, cela signifierait « déemballage en UV ». Mais que sont exactement les UV ? Nous savons que le patron du maillage est obtenu en 2 dimensions ; les lettres U et

V correspondent respectivement au nom des axes des abscisses et des ordonnées. Les lettres U et V remplacent X et Y, car ces deux lettres sont déjà utilisées par l'ordinateur dans les calculs des coordonnées des vertices.[27]

Il s'agit donc de donner à chaque vertice des coordonnées (du type $P(u, v)$) dans le plan plus. Ensuite, l'ordinateur peut tracer les faces, en fonction de la position des vertices qui la constituent, dans le plan. Le patron est donc obtenu.

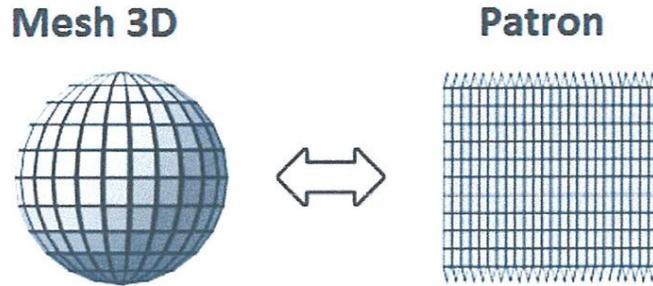


FIGURE 2.14 – Le processus du UV unwrapping.[27]

2. Application de la texture :

Il s'agit par la suite d'appliquer la texture et donc de placer une image sur le patron obtenu (comme la Figure 2.15) : c'est le texture-mapping. Effectivement, l'ordinateur enregistre pour chaque face, les pixels de la texture, appelés texels, qui lui appartiennent ainsi que leur position. Enfin, chaque face du maillage en 3D diffuse les texels qui lui ont été appropriés. Ainsi, la texture est appliquée sur la surface de l'objet modélisé.[17]

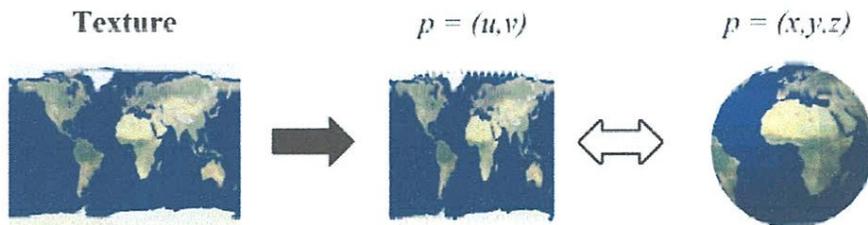


FIGURE 2.15 – Application de la texture.[17]

– Le rendu :

Le rendu C'est une étape pendant laquelle un logiciel appelé moteur de rendu 3D transforme une scène virtuelle tridimensionnelle en une image normale (2D). Dans la vie réelle, lorsque nous observons quelque chose, nos yeux perçoivent une image en «interprétant » les rayons lumineux qui parviennent à nos yeux. Il en est presque de même dans un logiciel de modélisation 3D ; il s'agit d'attribuer une couleur donnée à chaque pixel de l'image obtenue, en fonction des rayons de lumière qui proviennent d'une scène virtuelle 3D. Effectivement, afin d'illuminer une scène, plusieurs lampes virtuelles sont placées. Elles permettent de définir la couleur, l'intensité et la direction de la lumière éclairant la scène. Mais on sait que la lumière est émise en une infinité de rayons et dans toutes les directions, alors il ne serait pas efficace pour un ordinateur d'effectuer des calculs pour un très grand nombre de rayons, puisque la majorité de ceux-ci ne figureront pas sur l'image finale. Il s'agit alors de suivre le trajet inverse de chaque rayon lumineux qui parvient à l'image qu'on souhaite obtenir. On parle alors de Ray-Tracing, soit « lancer de rayons »[21].

Etape 1 : Le Ray-tracing est un algorithme de rendu d'image. Pour chaque pixel de l'image que l'on souhaite obtenir, il projette un rayon ; c'est un rayon primaire (Figure 2.16).

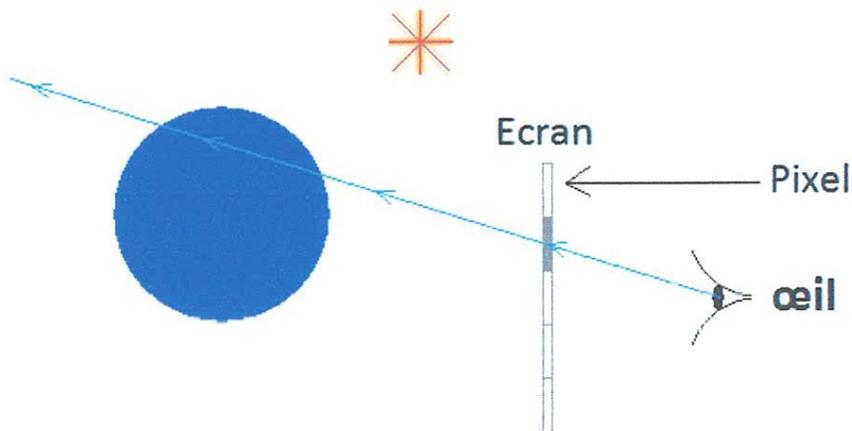


FIGURE 2.16 – Erojction du rayon primaire [1].

L'algorithme trouve le premier point (P) d'intersection de chaque rayon primaire avec un maillage de la scène 3D. (Figure 2.17). Il détermine ensuite à quel maillage, et à quelles faces

du maillage ce point appartient. Ceci permet donc de ne représenter que les faces visibles, car en effet, les points d'intersection des rayons primaires avec le maillage ne se trouvent que sur les faces visibles du maillage.

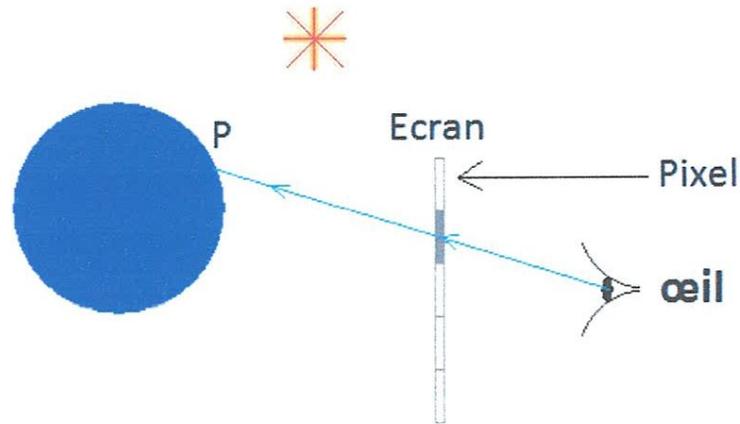


FIGURE 2.17 – Détermination du premier point d'intersection du rayon primaire avec le maillage[1].

Etape 2 : Il faut ensuite déterminer la couleur de la face à laquelle appartient ce point. Une couleur est attribuée à chaque face dans un maillage. à partir du point P (déterminé précédemment), un autre rayon est émis vers la «lampe ». Il est nommé paradoxalement shadow ray, ou « rayon d'ombre ».

Si aucun objet ne coupe ce rayon, alors l'algorithme saisit la valeur de l'intensité de la lumière émise par la lampe et l'affecte à la face du maillage. Ensuite, d'autres données entrent en jeu, comme par exemple l'angle entre le vecteur normal de la face et la lampe ; ceci permettra donc d'obtenir la couleur finale du pixel. (Figure 2.18).

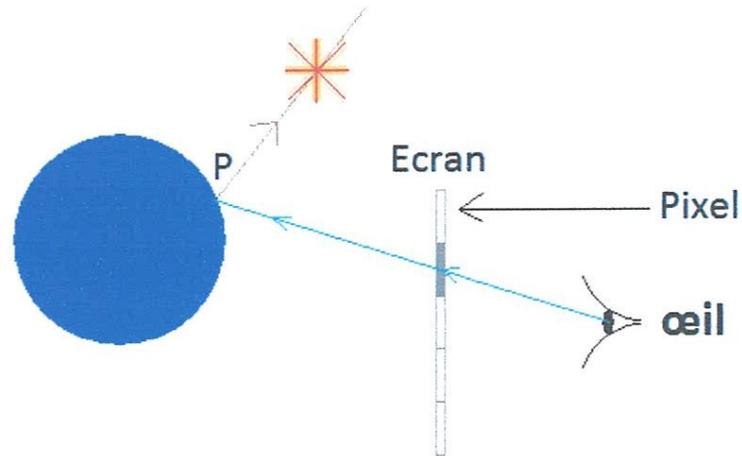


FIGURE 2.18 – *Le rayon d'ombre (marron) ne coupe aucun objet[1].*

Si le rayon d'ombre est coupé par un autre objet, la lumière n'atteint pas le point P ; il se trouve donc dans l'obscurité ; le pixel est par conséquent de couleur noire. (Figure 2.19)

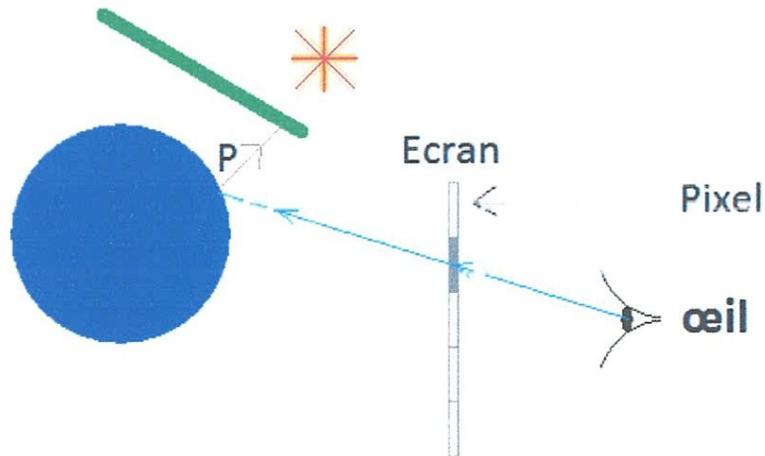


FIGURE 2.19 – *Le rayon d'ombre (marron) coupe un objet (en vert)[1].*

2.3 Conclusion

La conception d'images 3D, nécessite le passage par plusieurs étapes avant l'obtention d'une image finale. Le tout se fait via des logiciels de modélisations. Dans ce chapitre on a cités quelques logiciels de création et d'animation 3D généralement utilisés dans l'infographie, on vues aussi les différentes étapes de fonctionnement de l'infographie.

Chapitre 3

La conception

3.1 Introduction

Le but de ce sujet a été de développer une scène dans laquelle nous avons modélisé une application « saut en longueur » qui consiste à faire exécuter des agents virtuels avec chacun d’eux une force différentes. Pour ce faire nous nous sommes basés sur les techniques de l’infographie et de l’animation comportementale. Dans ce chapitre, nous vous présenterons les différentes étapes qui nous permettent de réaliser ce projet en commençant tout d’abord par la modélisation de l’environnement ainsi que des agents, puis par l’animation comportementale et enfin nous terminerons par des tests.

3.2 Environnement de développement

3.2.1 Les Matériels utilisé

Pour le développement de notre application nous avons utilisé un ordinateur ayant les caractéristiques suivantes :

Processeur : Intel (R) Pentium(R) 4 CPU 3.20 GHz.0

RAM : 1,00Go.

Disque dur : 150Go.

3.2.2 Le logiciel utilisé

L’environnement de développement qu’on a choisi pour la modélisation de notre application est « Blender » version « 2.75 ».

– Pourquoi Blender ?

La principale motivation qui nous a poussées à choisir Blender est qu’il est gratuit. En plus de ça, il dispose de la pluparts des fonctionnalités qu’on a besoins pour une modélisation (modélisation, animation et rendu en 3D). Il dispose aussi des fonctions avancées de modélisation, de sculpture 3D, de dépliage UV, de texturage. Il gère également le montage vidéo non linéaire, la création d’applications en 3D interactives, ainsi que diverses simulations physiques telles que les particules, les corps rigides, les corps souples et les fluides.

Blender est donc un logiciel complet pour réaliser des images et des animations de synthèse. Il

regroupe, au sein d'une interface stable et moderne, de puissants outils de modélisation, d'animation et de rendu.

3.3 La Modélisation 3D

La modélisation 3D est une technique de l'infographie faisant partie du domaine de l'informatique, qui vise à créer (dans un logiciel adapté) des objets en 3 dimensions. Elle est utilisée notamment dans l'élaboration de jeux vidéo, des films d'animation, ou encore de plans ou projets d'ordre professionnel (comme la reconstitution 3D de bâtiments afin de les construire ou les rénover).

3.3.1 Prise en main des fonctions de modélisation

Au niveau du logiciel de modélisation Blender, dans un premier temps il a donc fallu apprendre à utiliser ce logiciel qui est basé sur le langage de programmation Python. En effet bien qu'ayant de grosses capacités cet outil est très compliqué à prendre en main.

Pour avoir un apprentissage efficace nous avons donc décidé de regarder un grand nombre de tutoriaux, premièrement à propos des connaissances nécessaires pour une utilisation basique afin de se familiariser avec l'environnement complexe de Blender, puis à propos des fonctionnalités avancées dont nous aurions besoin tout au long de ce projet.

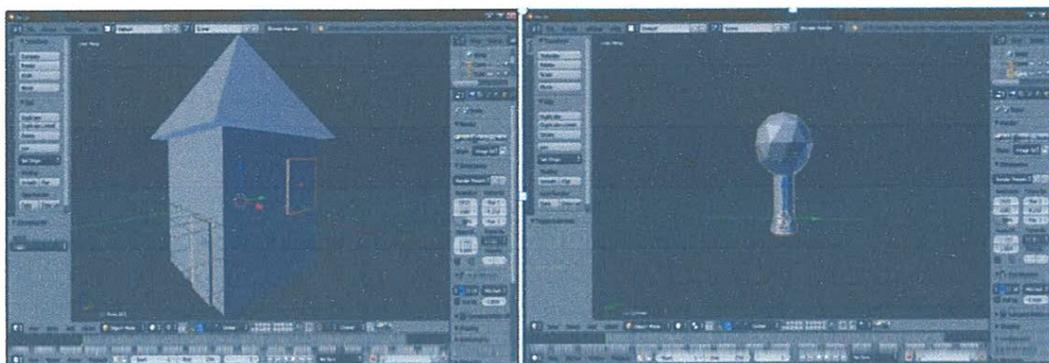


FIGURE 3.1 – Exemple de modélisation des objets simples.

Après avoir réalisé quelques modélisations assez simplistes et complexes à la fois comme dans la Figures 3.1, nous avons ensuite commencé à créer des objets beaucoup plus complexes (voire

Figure 3.2) qui représente un personnage virtuel et qui a été assez long à modéliser, mais qui nous a permis de bien assimiler les connaissances nécessaires pour la suite du projet. Ce personnage virtuel a été modélisé entièrement en se basant sur une sphère pour la tête et un cube pour le corps. Ces derniers ont été beaucoup modifiés en se basant sur les techniques de l'infographie pour arriver à un résultat plutôt satisfaisant.

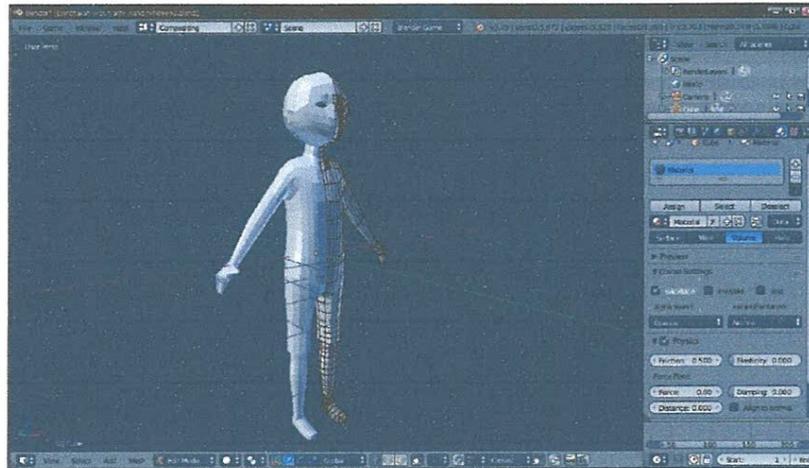


FIGURE 3.2 – Exemple de modélisation d'un personnage .

3.3.2 La modélisation de notre agent et l'environnement

– L'agent

Dans cette partie en va expliquer la création de notre agent :

Nous allons ajouter un cube (un Mesh qui est la base pour la plupart des objets 3D). Puisque nous allons modéliser un personnage, il sera intéressant de concevoir tout d'abord la moitié, et de créer l'autre moitié en utilisant un miroir.

Quand on a ajouté le cube, on passe en mode *Edit* pour pouvoir accéder et modifier les sommets de ce dernier. En d'autres termes, on accède au maillage de l'objet en question (Figure 3.3 et Figure 3.4).

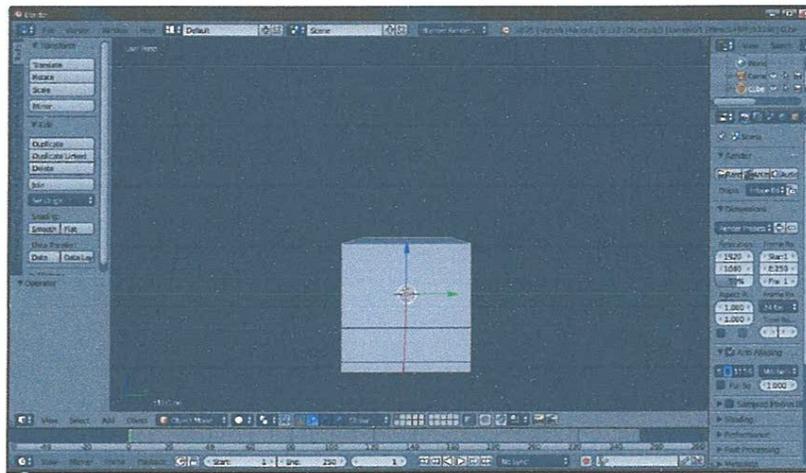


FIGURE 3.3 – Représentation du cube en mode Objet.

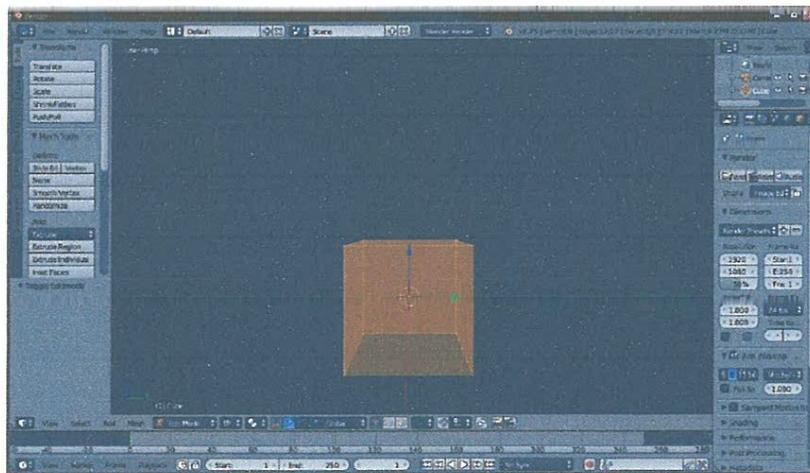


FIGURE 3.4 – Représentation du cube en mode Edit .

Premièrement on va subdiviser le cube, ce qui nous permet multiplier les sommets dans le but de supprimer la moitié gauche du cube (voire Figure 3.5).

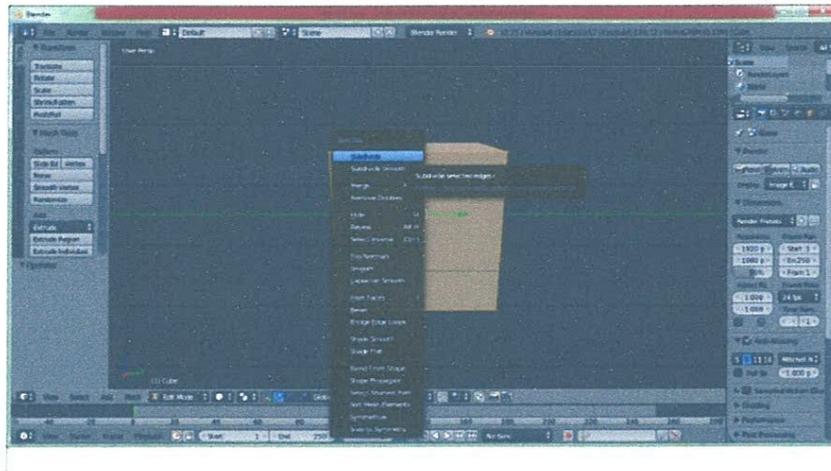


FIGURE 3.5 – *Subdivision du cube de l'objet de base.*

– L'application du *Miroir*

On applique le *miroir* selon l'axe des x. Sachant que le buste d'un humain n'est pas carré, on va lisser notre cube pour qu'il ressemble le plus possible à la réalité. En appliquant le *Smooth* et le *Suburf* sur notre cube, ce dernier devient lisse (Figure 3.6 et Figure 3.7).

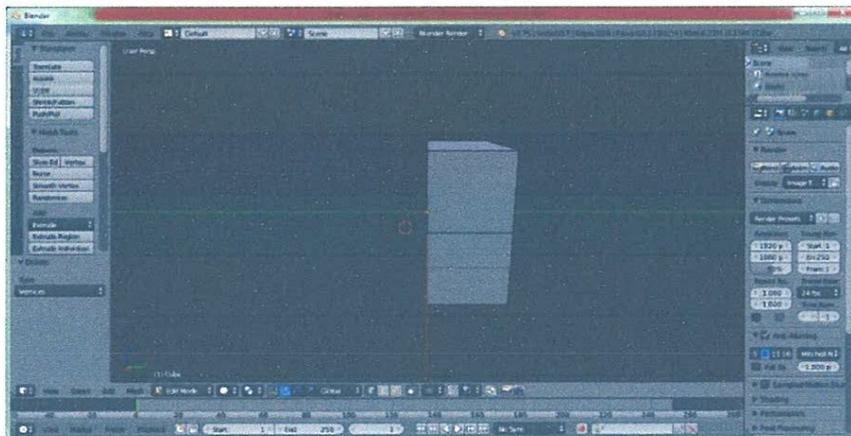


FIGURE 3.6 – *Le cube avant l'application de miroir Smooth et le Subsurf.*

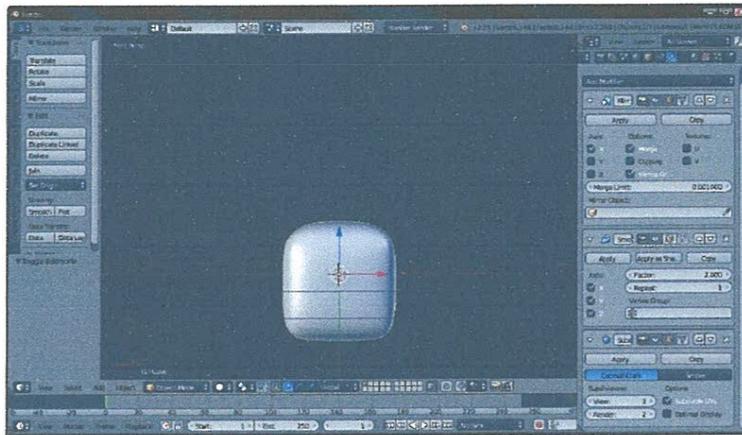


FIGURE 3.7 – *Le cube après l'application de miroir Smooth et le Subsurf.*

Après avoir multiplié le nombre des sommets afin de mieux contrôler notre cube lisse, on applique plusieurs étapes d'extrudassions, de translation, de rotation et de mise à l'échelle des différents côtés. A la fin, ce cube va devenir le buste de notre agent (Voir Figure 3.8). Après

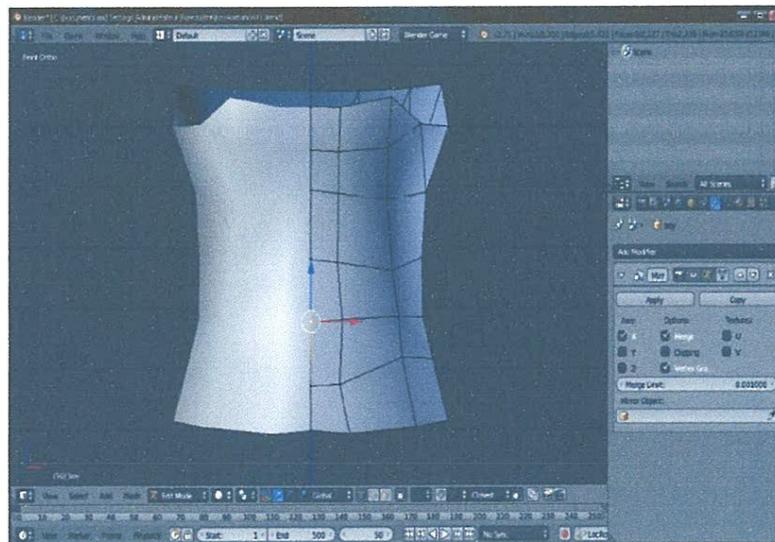


FIGURE 3.8 – *Buste de notre personnage.*

avoir construit le buste de notre personnage, on s'attaque maintenant au haut du corps. Pour ce faire, on va extruder vers haut le buste déjà obtenu. Après, on applique des translations afin d'obtenir le haut du corps (Figure 3.9). On extrude cette fois le buste vers le bas et on applique les mêmes étapes pour construire les cuisses (Figure 3.10). La construction des bras se fait à partir d'un cylindre sur lequel on va effectuer une découpe des deux extrémités. On applique

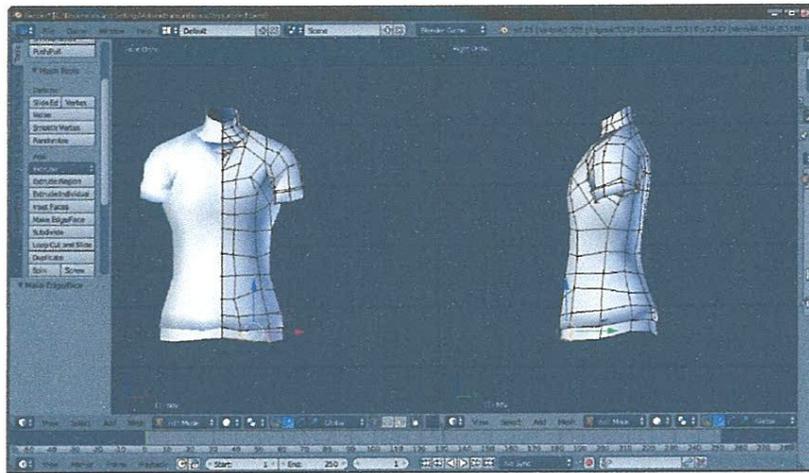


FIGURE 3.9 – *L'haut de corps de notre personnage.*

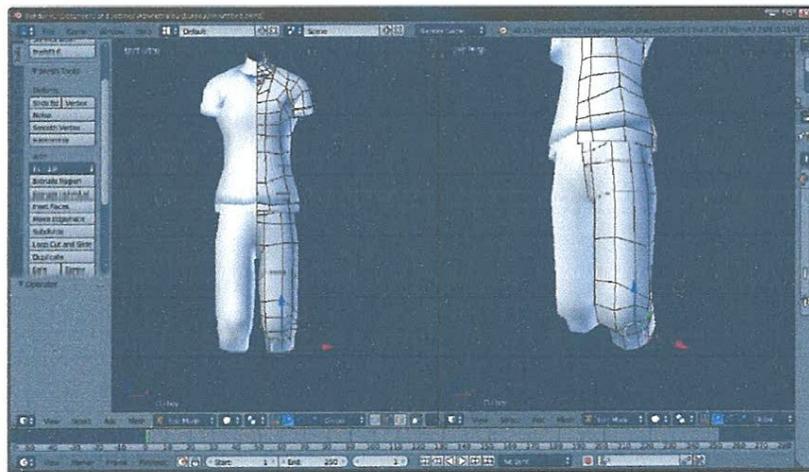


FIGURE 3.10 – *La modélisation de cuisse.*

les différentes étapes vues précédemment (extrudassions, translation, rotation) on obtient les bras ainsi que les mains de notre personnage (voir Figure 3.11).

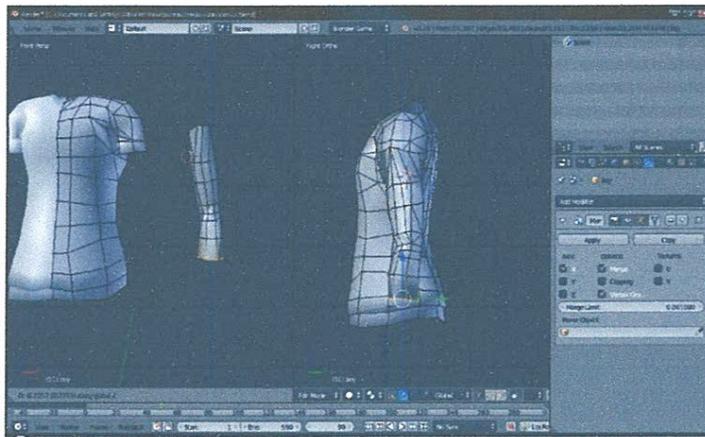


FIGURE 3.11 – *la modélisation du bras et la main.*

On fait la même chose pour modéliser les jambes et les pieds (Figures 3.12). Pour la tête,

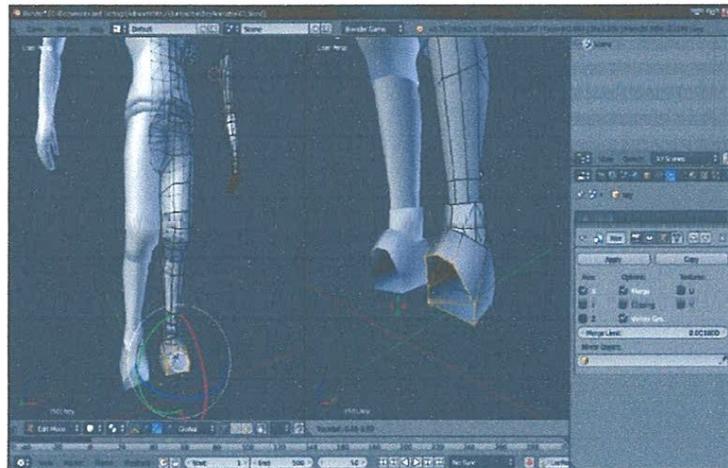


FIGURE 3.12 – *la modélisation du jambe et le pied.*

on va prendre comme forme de base une sphère (Figure 3.13). Après, l'application des étapes de modélisations on obtient la tête (Figure 3.14).

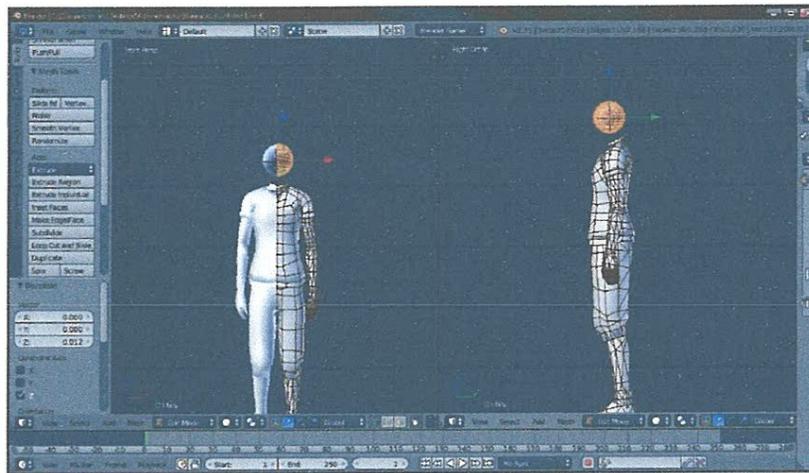


FIGURE 3.13 – L'ajout d'une sphère pour la création de tête.

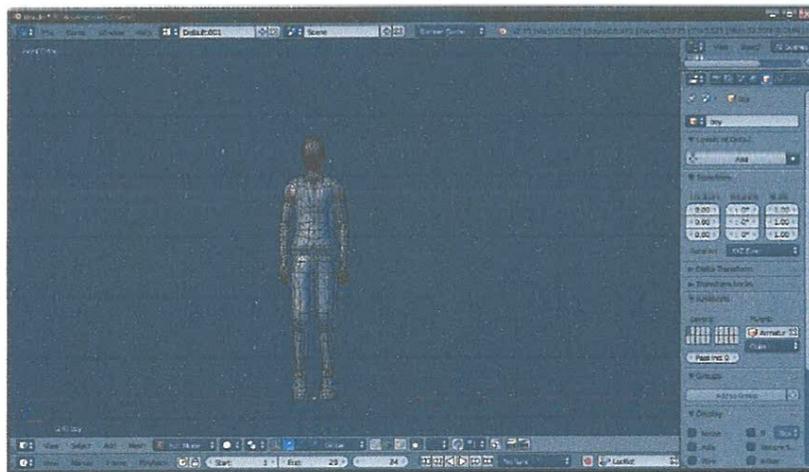


FIGURE 3.14 – Le personnage final.

– Texture et rendu :

Pour la texture de l'agent on a utilisé UV mapping qui va permettre à une image 2D de venir se positionner très précisément sur un volume (notre personnage). Comment passer d'une image 2D vers une image en 3D ? L'UV mapping tente de répondre à cette question en proposant d'attribuer à toutes les faces d'un objet un système de coordonnées à deux dimensions qui seront U et V et qui vont créer ce qu'on appelle une carte UV

Le mode d'emploi UV mapping Tout d'abord on va marquer les arêtes de dépliages, puis on va ajouter une image on mettant l'alpha à 0.

Après avoir collé l'image sur le personnage. On applique sur notre personnage l'Unwrap pour

obtenir la carte UV.. C'est sur cette carte qu'on va appliquer notre texture. La (Figure 3.15) montre le résultat en mode texture. Après avoir obtenu la carte UV on a colorié cette image

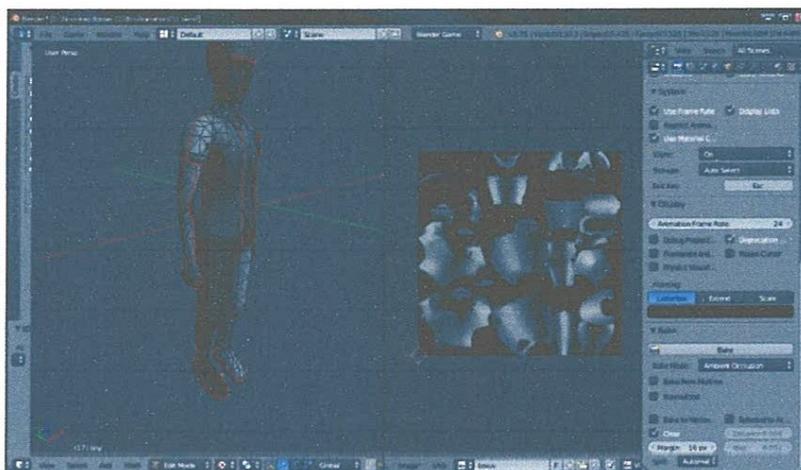


FIGURE 3.15 – *Le résultat en mode texture.*

dans un logiciel de peinture qui est Paint (Figure 3.16).

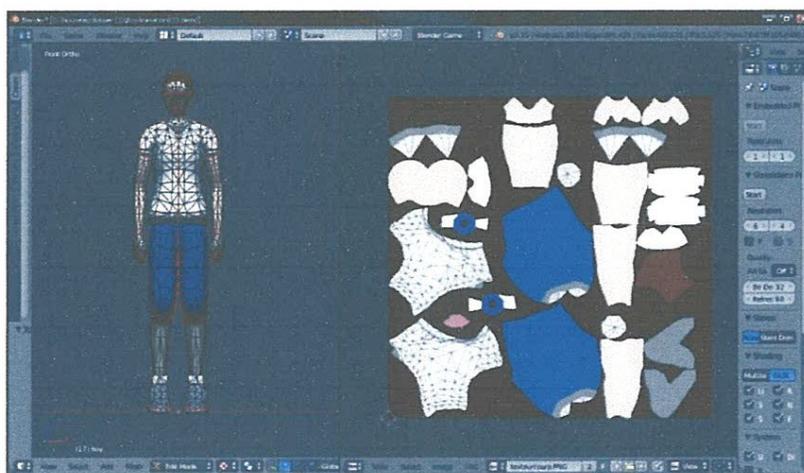


FIGURE 3.16 – *Le résultat de texture après la coloration de l'image.*

– L'environnement

Notre environnement est composé d'un terrain crée à partir d'un cube. Ensuite on a ajouté deux surfaces, la première représente la piste sur laquelle notre personnage va courir, la deuxième représente le bac à sable dans lequel notre personnage effectue son saut (voir Figure 3.17). Concernant les différentes textures de l'environnement on les a chargés de manière adéquate

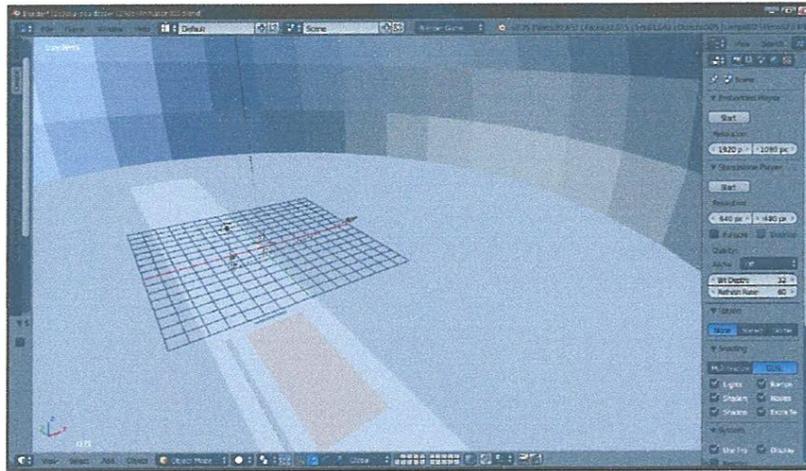


FIGURE 3.17 – Représentation de l'environnement avant la texture .

suivant le type de chaque élément de notre environnement ; le ciel, la piste ou encore le bac à sable (Figure 3.18).

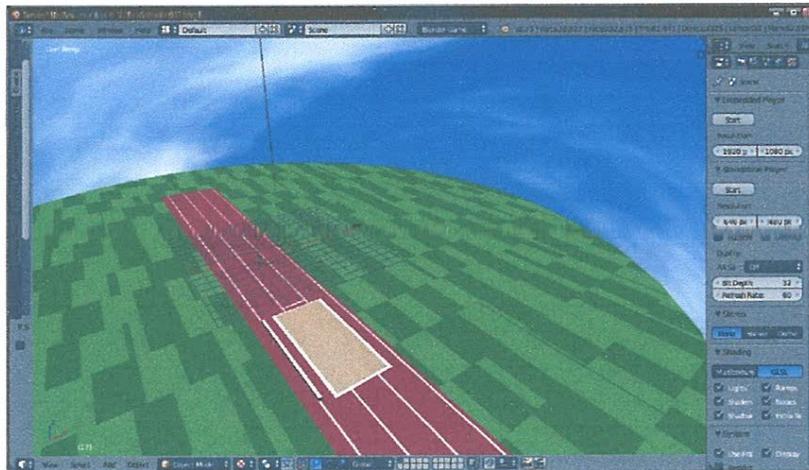


FIGURE 3.18 – Représentation de l'environnement après la texture .

3.4 L'animation de l'agent

Les armatures fonctionnent très simplement, comme les squelettes sur les êtres vivants. On va créer une armature constituée de plusieurs os qui vont déformer un maillage.

Dans l'étape suivante on a donné à l'agent un squelette (ou armature) qui le fera bouger. C'est l'art délicat du *rigging*. L'agent a un *rigging* composé de :

- Dix-huit os de type déformables sont : ventre, buste, col, tête, deux bras, deux avants bras, deux mains, deux cuisses, deux mollets, deux pieds, deux avants pieds(voir Figure 3.19).
- Douze os de type non déformables qui permettent de gérée le mouvement des pieds pour donner une souplesse à l’animation (voir Figure 3.19).

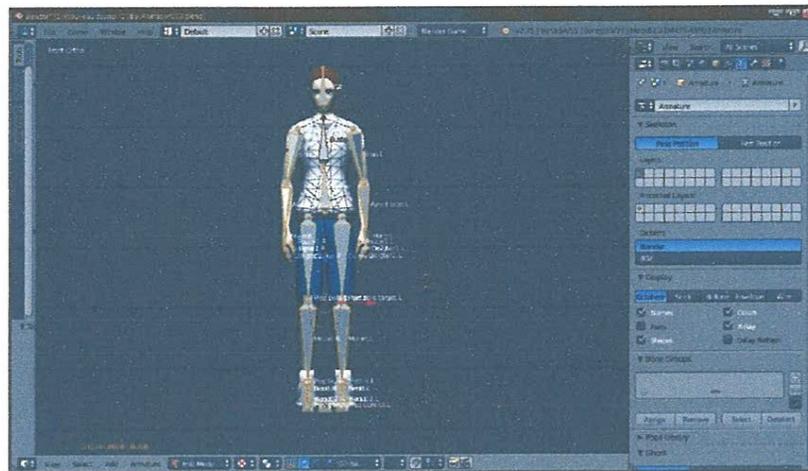


FIGURE 3.19 – L’armature de notre personnage.

Après avoir définis les armatures. On va les associées à notre personnage en utilisant La méthode *Automatic Weights* pour que ce dernier bouge si son squelette bouge (Figure 3.20). Après la

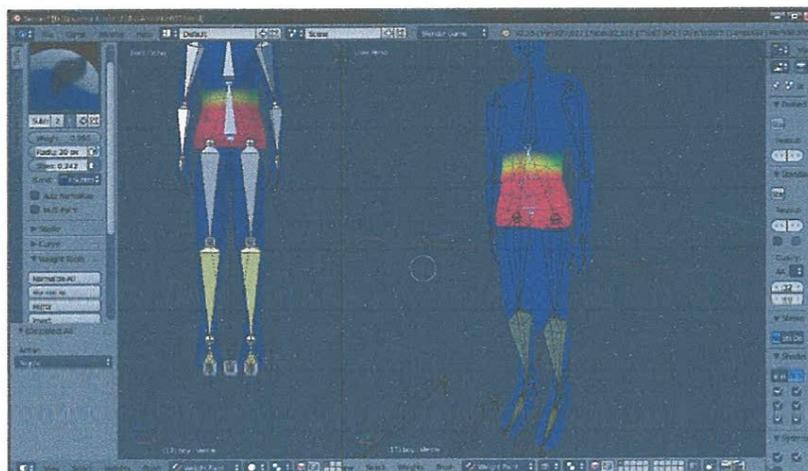


FIGURE 3.20 – L’application de méthode *Automatic Weights*.

création des os on a créé la action courir dans le pose mode à l’aide d’une image 2D (Figure 3.21).

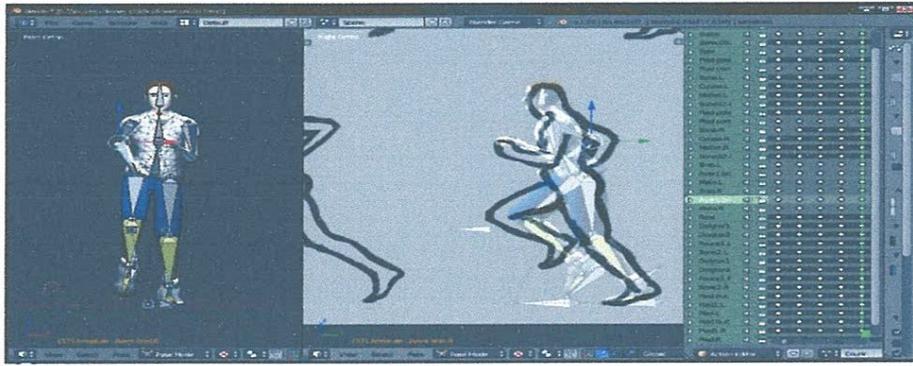


FIGURE 3.21 – *La création de l'action courir.*

On a ajouté à la fin de l'action courir une autre qui représente un saut dans le bac à sable. Pour ce faire, on a ajouté une trajectoire que l'agent va suivre (Figure 3.22).

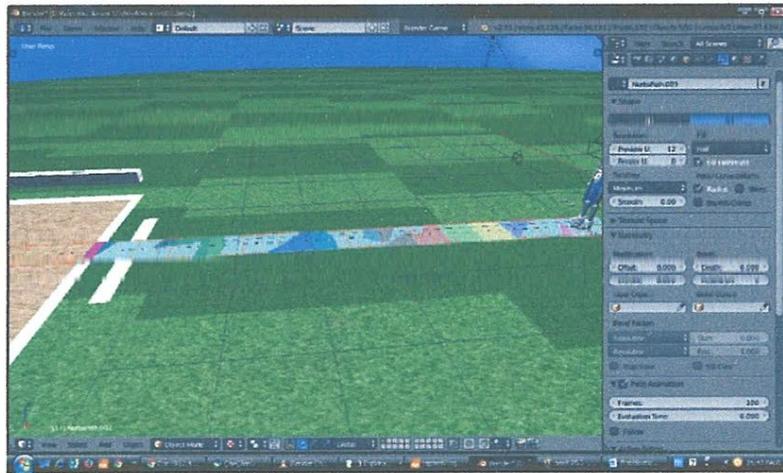


FIGURE 3.22 – *La trajectoire de l'animation.*

3.5 Le comportement du personnage

Après avoir créés trois personnages en utilisant les mêmes techniques de modélisation qu'on a vues précédemment, on s'attaque maintenant au comportement de chaque personnage. Dans cette étude on a ajouté un attribut de caractère qui est la *force*. Ce dernier possède une valeur pour chaque personnage.

— Cas 1 : Premier personnage

Lorsque on applique une force assez faible à notre premier personnage, ce dernier va effectuer un faible saut qui est inférieure à 3 mètres (voir Figure 3.23)

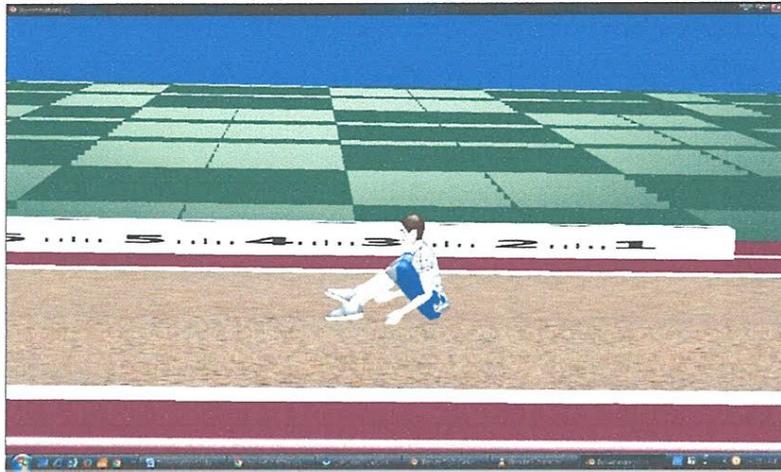


FIGURE 3.23 – *Saut de premier personnage avec une force faible.*

— Cas 2 : Deuxième personnage

Lorsque on applique une force moyenne à notre deuxième personnage, ce dernier va effectuer un saut un peu plus grand que celui du premier personnage (voir Figure 3.24)

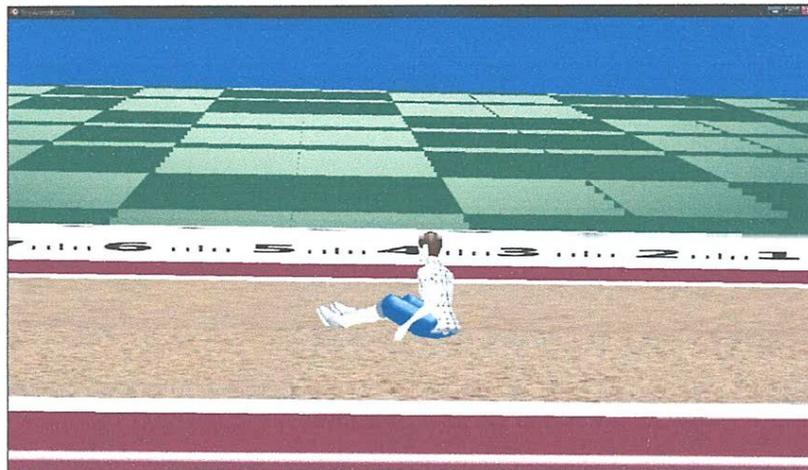


FIGURE 3.24 – *Saut de deuxième personnage avec une force moyenne.*

— Cas 3 : troisième personnage

Lorsque on applique une force élevée à notre troisième personnage, ce dernier va effectuer un saut plus grand que celui des deux premiers personnages (voir Figure 3.25)

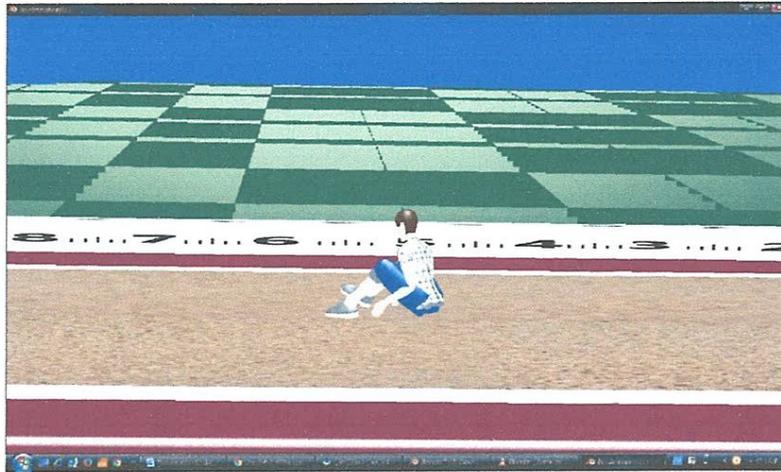


FIGURE 3.25 – *Saut de troisième personnage avec une force fort.*

3.6 Conclusion

Dans ce chapitre nous avons mis en place certaines techniques que nous avons vues dans le premier et deuxième chapitre.

Nous avons créé une application qui simule le saut en longueur de plusieurs sportifs (athlète). Ces derniers possèdent des comportements différents qui sont caractérisés par les valeurs de l'attribut de caractère force. Chacune de ces valeurs représente le niveau d'un athlète.

3.7 Conclusion générale

La réalité virtuelle ne se limite pas uniquement à la modélisation 3D, mais comme nous l'avons vue au cours de ce travail elle nécessite aussi l'incorporation du comportement aux entités qui constituent la scène. En d'autres termes, l'animation comportementale joue un rôle primordial dans la réalité virtuelle. Dans ce cadre, nous avons créé une application qui simule le saut en longueur de trois personnages, où nous avons associés à chaque personne une force. L'ajout d'un attribut de caractère comme la force rend le contrôle des entités virtuelles meilleur en influençant sur leurs comportements, comme on l'a vue avec les trois athlètes de notre application.

En ce qui concerne les perspectives, nous pouvons ajouter d'autres attributs de caractère comme l'âge, la vitesse et le poids de la personne pour obtenir de meilleurs comportements.

Bibliographie

- [1] Nicolas Moulin Andrawes Al Bahou, Eyasse Al Battah. *L'Infographie 3D en Médecine*. PhD thesis, Janvier 2012.
- [2] Corentin Haubruge Magali Bodart. *Modélisation et simulation de léclairage naturel au moyen de 3ds Max Design*. PhD thesis, Université catholique de Louvain Centre de Recherches en Architecture, 2012.
- [3] BOUSGUETITICHE. Modélisation topologique et sémantique de l'environnement. Master's thesis, 2011.
- [4] Lilian BUZER. *Modélisation NURBS, COURBES & SURFACES NURBS*. PhD thesis, Dept. Informatique.
- [5] Matthieu Dupont de Dinechin Viralata. *COURS BLENDER 3D : Aide mémoire support de cours*. PhD thesis, Mars 2010.
- [6] Sébastien Thon IUT de Provence. *3DS Max*. PhD thesis, Site d'Arles, 07 mars 2005.
- [7] Hiep Thuan Do. Simulation comportementale de la circulation à moto dans les grandes villes vietnamiennes. Rapport de stage Master Recherche 2IH, IRIT, Université Paul Sabatier, Toulouse, juin 2005.
- [8] James D. Foley, Richard L. Phillips, John F. Hughes, Andries van Dam, and Steven K. Feiner. *Introduction to Computer Graphics*. Addison-Wesley Longman Publishing Co., Inc., 1994.
- [9] gagui al. Navigation réactive d'humain virtuel par planification locale. Master's thesis, université de biskra.
- [10] Franck Alain GOMIS. *Cours d'Infographie !* PhD thesis, 2010.
- [11] Michael A. Goodrich. 1 introduction potential fields tutorial, 2000.
- [12] Hao Jiang, Wenbin Xu, Tianlu Mao, Chunpeng Li, Shihong Xia, and Zhaoqi Wang. A semantic environment model for crowd simulation in multilayered complex environment.

- In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology, VRST '09*, pages 191–198, New York, NY, USA, 2009. ACM.
- [13] Zhifan Jiang, Fabrice Jaillet, and Florence Zara. Reconstruction et complétion de maillages sous contraintes. Technical report, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/ École Centrale de Lyon, 2011.
- [14] Marcelo Kallmann, Hanspeter Bieri, and Daniel Thalmann. Fully dynamic constrained delaunay triangulations, 2003.
- [15] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1) :90–98, 1986.
- [16] Linko. *Tuto Maya refonte*. PhD thesis, 06/03/2011.
- [17] Michael Gleicher MankyuSung and Stephen Chenney. Scalable behaviors for crowd simulation.comput.graph. forum. 23(3) :519–528, 2004.
- [18] Bernard Moulin Nabil Sahli. La géo-simulation multi-agent : Un support pour la planification dans un environnement réel. *Revue d'Intelligence Artificielle*, 21(2) :151–182, 2007.
- [19] Michael O'Rourke. *Intro to Polygonal Modeling*. PhD thesis, maya 2012.
- [20] Romain Pacanowski. *Infographie, Rapport de stage sur les Différentiels de rayon*. PhD thesis, université de montréal, IFT-3051, Aout 2004.
- [21] N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '07*, pages 99–108, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [22] N. Pelechano, K. O'Brien, B.G Silverman, and N. Badler. Crowd simulation incorporating agent psychological models, roles and communication. In *International Workshop on Crowd Simulation*, 11 2005.
- [23] Nuria Pelechano. *Modeling Realistic High Density Autonomous Agent Crowd Movement : Social Forces, Communication, Roles and Psychological Influences*. PhD thesis, Philadelphia, PA, USA, 2006. AAT3225518

- [24] Nuria Pelechano and Norman I. Badler. Modeling crowd and trained leader behavior during building evacuation. *IEEE Computer Graphics and Applications*, 26(6) :80–86, 2006.
- [25] Les Piegl. On nurbs : a survey. *IEEE Computer Graphics and Applications*, 11 :55–71, 1991.
- [26] Francesca Delli Ponti. *Blender*. PhD thesis, Summer school on scientific visualisation.
- [27] Hakim Soussi. Modèle global et paramétrable, pour la gestion des foules d’agents en animation comportementale. Master’s thesis, Université de Bourgogne, 2011.
- [28] Jean-Marie Burkhardt Stanislas Couix. Réalité virtuelle, interface homme - machine et simulation. Master’s thesis, Université Paris, 2005.
- [29] stéphanesanchez. Mécanisme évolutionnistes pour la simulation comportementale d’acteurs virtuels. Master’s thesis, université toulouse, 2000.
- [30] Franco Tecchia, Céline Loscos, Ruth Dalton, and Yiorgos Chrysanthou. Agent behaviour simulator (abs) :a platform for urban behaviour development. In *The First International Game Technology Conference and Idea Expo (GTEC’01) in co-operation with ACM SIGGRAPH and EUROGRAPHICS*, Hong Kong, January 2001.
- [31] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55 :601–644, 1983.