



RAPPORT INFORMATIQUE GRAPHIQUE

Animation Avatar



1ère Année Spé Info


Sacha Lenartowicz

Manfred Madelaine

POLYTECH PARIS-SUD

Maison de l'ingénieur - bâtiment 620 rue Louis de Broglie
91405, Orsay

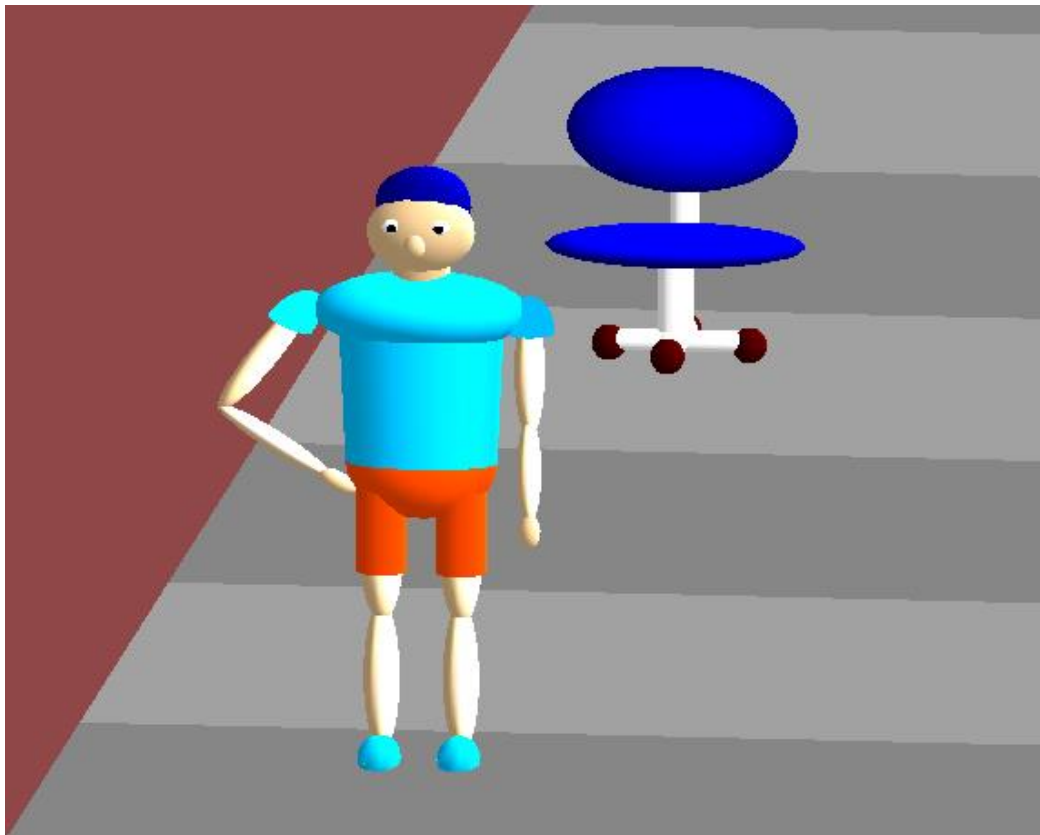
12/06/2016



Introduction

Le but de ce projet était de programmer l'animation d'un avatar en enrichissant l'ossature de ce dernier qui était fournie au préalable, lors du TD2, afin qu'on puisse se focaliser uniquement sur l'utilisation OpenGL. Pour cela, il nous a fallu comprendre comment ajouter des formes simples et complexes au modèle et également comment gérer leur animation.

Nous avons donc commencé par rendre l'avatar plus réaliste, avant de lui fournir les mouvements demandés. Nous avons enfin ajouté un environnement (décrit plus tard) et géré le déplacement du personnage dans celui-ci. Voici une capture d'écran de la version finale :



Sommaire

I- Récapitulatif du cahier des charges.....	3
a) Objectif principal.....	3
b) Graphe de scène.....	3
c) La modélisation de l'avatar	4
d) Les mouvements.....	6
e) Les transitions.....	7
f) Le déplacement et les collisions	8
g) L'environnement	9
h) Camera et autres.....	9
i) Récapitulatif des touches	10
 II- Perspectives d'évolutions.....	 11
 III-Accomplissements personnels.....	 12
 IV-Conclusion	 13

I- Récapitulatif du cahier des charges

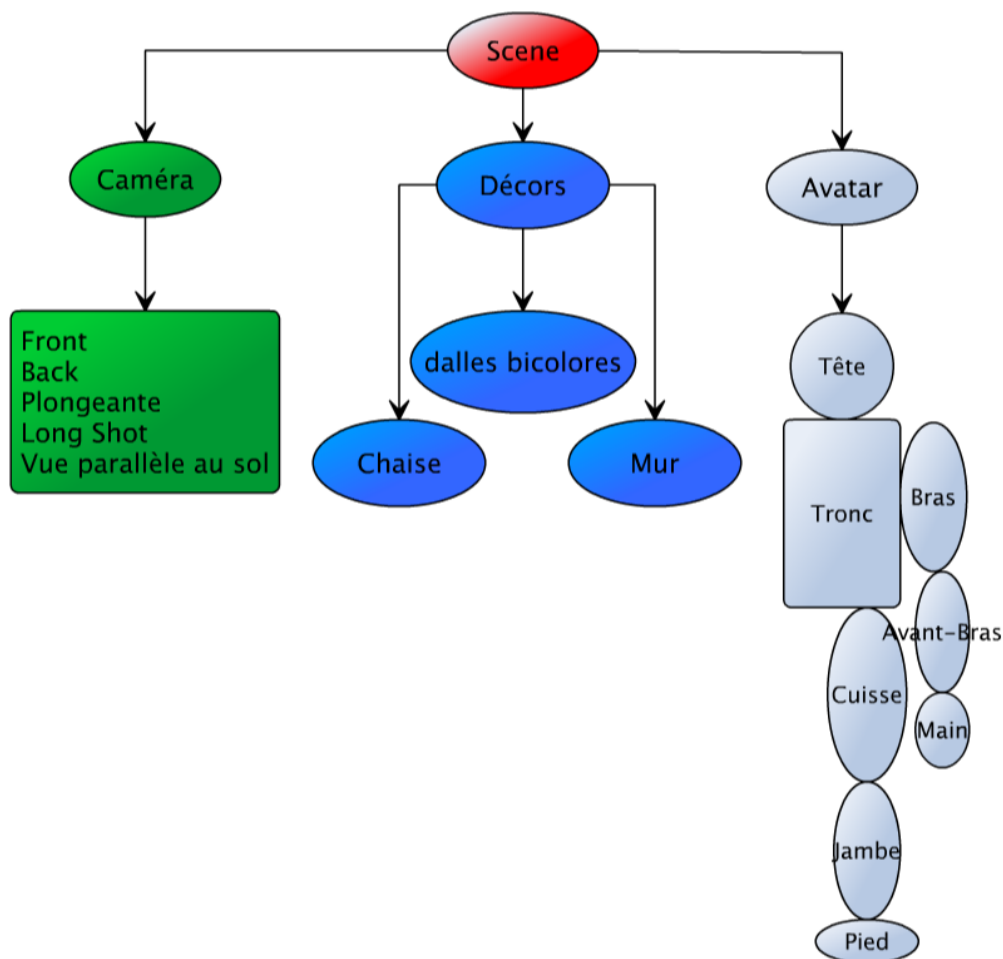
a) Objectif principal

Le programme doit permettre une gestion à la fois en mode manuel et aléatoire de l'avatar avec des animations fluides et cohérentes. Durant une utilisation manuelle, l'utilisateur est le seul maître des mouvements. Dans le cas inverse, les allures et déplacements se font de manière aléatoire.

b) Graphe de scène

Nous proposons un graphe de scène afin d'avoir une vision plus claire de la structure de la scène jouée dans notre programme :

Graphe de Scène



c) La modélisation de l'avatar

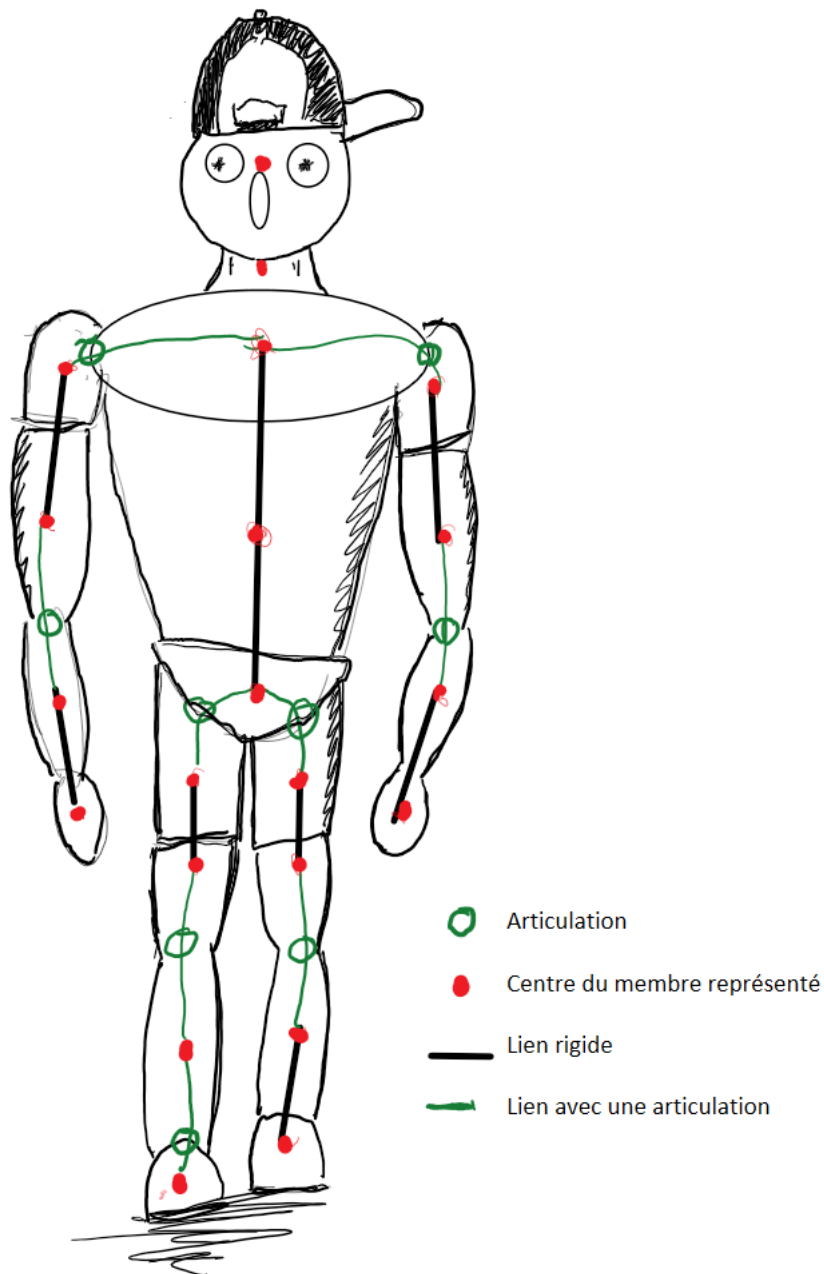
L'avatar que nous devions réaliser au cours du 2^{ème} TD n'était pas très réaliste. En effet, il ne possédait ni mains, ni pieds, ni même un visage. Les formes utilisées représentaient de manière très simpliste un être humain, et ses articulations restaient inexistantes.

Nous avons donc commencé par développer cette représentation, afin de pouvoir correctement implémenter les mouvements demandés dans la suite du sujet. Nous avons alors décidé de modéliser les membres (bras et jambes) par des ellipses, qui sont en fait des sphères que nous avons étirées grâce à la fonction `glScalef()`, permettant de modifier les échelles. Une fois les membres redessinés, nous avons ajouté les mains (sphères aplaties) et les pieds (demi-sphères fermées).

Ensuite, nous nous sommes attaqués au tronc de l'avatar. Nous avons remplacé le simple cylindre de base par une sphère écrasée servant de bassin, un cylindre représentant le tronc et une ellipse désignant la partie haute du tronc (le torse). Il était alors nécessaire de représenter les épaules par des demi-sphères creuses, symbolisant ainsi les manches d'un t-shirt, pour améliorer les transitions entre les divers éléments. Pour la même raison, nous avons utilisé deux cylindres sortant du bassin et épousant les cuisses afin de représenter un short.

Une fois les quatre membres terminés, il ne nous restait plus qu'à améliorer la tête du personnage. Dans l'optique d'indiquer à l'utilisateur le sens dans lequel est tourné le personnage, nous avons dessiné un nez en partant d'une sphère écrasée et des yeux, chacun formé de deux sphères, l'une pour le globe oculaire et l'autre pour symboliser la pupille. Nous avons terminé par la représentation d'une casquette, simplement formée d'une sphère non-transformées et d'une autre sphère extrêmement écrasée pour la visière.

Afin de mieux comprendre la construction de notre avatar et les liens entre chaque partie, quoi de mieux qu'un beau schéma ?



d) Les mouvements

L'avatar devait être capable de réaliser différents mouvements, à savoir marcher, courir, danser et s'asseoir. De plus, il devait également indiquer à l'utilisateur s'il est dans un état d'attente. Nous avons alors commencé par reprendre la fonction marcher (), réalisées lors de la seconde séance de TP en y apportant quelques modifications. Il était alors simple d'implémenter une fonction courir (), reprenant le principe de la fonction marcher (). Nous avons cependant jugé nécessaire d'effectuer des translations pour modifier la hauteur à laquelle l'avatar devait être affiché lors de la course afin de simuler un effet de sautaillement lors du déplacement de l'avatar.

Pour pouvoir s'asseoir, l'avatar doit se situer dos ou face à une chaise (situation expliquée plus tard). La fonction sassoir () est alors utilisée. Dans le cas où l'avatar est face à la chaise, il doit tout d'abord effectuer un demi-tour, en faisant appel à la fonction demiTour (). Une fois le personnage orienté dans le bon sens, il commence à s'asseoir : ses cuisses et ses mollets voient leurs angles modifiés dans la fonction adéquate, tandis qu'une translation du personnage vers le bas et vers l'arrière est effectuée dans la fonction translation_totale (). Une fois assis, le seul mouvement accessible est de se lever, implémenté par la fonction seLever (), dans laquelle l'évolution des angles est l'inverse de ce qui a été fait juste auparavant.

Il fallait également permettre à l'avatar de danser. Dans la fonction danser (), nous avons commencé par placer les bras du personnage à l'horizontale, et nous avons initialisé les angles des jambes à 0. Ensuite, les bras de l'avatar se plient et se déplient, tandis que les jambes font des va-et-vient au même rythme. Les mollets sont repliés lorsque la jambe avance, de telle sorte que l'avatar ne touche le sol qu'avec un seul de ses deux pieds. Pour terminer avec la danse, nous faisons reculer le personnage pour donner l'impression d'une sorte de « moon-walk ».

Enfin, lors de l'attente du personnage, celui-ci devait indiquer à l'utilisateur son état. Nous avons donc choisi de le faire tapoter du pied gauche, en modifiant la valeur de l'angle associé avec une fréquence plutôt élevée. De plus, lorsque le personnage commence à patienter, nous lui plaçons la main sur la hanche.

e) Les transitions

Nous avons pour objectif de réaliser des transitions fluides et le plus proche possible de la réalité. Ainsi, nous avons décidé que le personnage devait, à l'issue de chaque changement de mouvement, revenir à une position d'équilibre (tous les angles à 0). Cela nous a permis de coder un mouvement sans se préoccuper des autres mouvements. Nous avons également déclaré deux variables, `move` et `next_move`, qui nous ont respectivement permis de stocker le mouvement en cours et le prochain mouvement à exécuter. Ainsi, à chaque itération de la boucle principale, on vérifie, en fonction de ces deux variables et des angles de l'avatar, si ce dernier est dans la possibilité d'adopter ou non le nouveau mouvement demandé par l'utilisateur. Si c'est le cas, la variable temporelle `t` est remise à 0, et on débute le mouvement suivant. Nous nous assurons ainsi que l'avatar commence le mouvement dans la bonne posture.

La modélisation des bras par des ellipses nous a posé un léger problème. Etant donné leurs formes trop pointues, lors de certains mouvements, ces dernières se chevauchaient un peu trop et laissaient apparaître deux « coudes » au lieu d'un seul. L'une des solutions possibles aurait été d'ajouter une sphère pour symboliser le coude, mais cela désagrégeait malheureusement le rendu final des bras, rendu dont nous étions déjà très satisfaits. Nous avons donc opté pour un ajustement de l'avant-bras via des translations lors d'animations nécessitant des angles faibles pour les bras comme l'animation de la danse par exemple. En contrepartie, les bras paraissent un peu plus grands lors de ces mouvements.

Nous avons également décidé d'arrêter la marche, la course ou la danse de l'avatar une fois un certain temps dépassé. Ainsi, l'avatar repasse en mode « attente » si aucune touche n'a été pressée par l'utilisateur au cours de ce laps de temps.

Deux modes sont proposés dans ce programme, comme explicité dans le sujet de ce projet. Le premier est le mode manuel, celui que nous avons commencé à coder. Le deuxième est un mode aléatoire, qu'il était facile d'implémenter puisqu'il suffisait de modifier aléatoirement la variable `next_move`. Cet ajout n'a donc pas nécessité d'adaptation ou modification du code. La fonction `automatic_key()` permet de changer cette valeur. Lorsque le personnage est assez loin d'une chaise, le prochain mouvement pourra être

n'importe lequel, en excluant s'asseoir et se lever. Dans le cas contraire, l'avatar pourra soit s'asseoir, soit faire un demi-tour. S'il est assis, il ne peut que se lever. Le temps entre deux changements de mouvement est lui aussi aléatoire mais borné par une valeur maximale.

f) Le déplacement et les collisions

Pour pouvoir garder le personnage au centre de l'écran, nous avons choisi de ne pas effectuer la translation directement sur celui-ci mais sur l'ensemble du décor. On a l'impression que le personnage se déplace dans le décor, même si dans le code, c'est l'inverse.

Par commodité, nous avons déterminé qu'il était plus simple de faire avancer l'avatar sur une ligne droite plutôt qu'en 3 dimensions. L'avatar est donc coincé dans un couloir, dans lequel se situe à chaque extrémité une chaise, sur lesquelles il pourra bien évidemment s'asseoir.

La position du personnage est incrémentée à chaque itération de la boucle principale et également dans les fonctions danser(), marcher() et courir(). Ainsi, la translation se fait à l'aide de cette variable globale. Nous avons aussi utilisé la variable way, qui indique dans quel sens l'avatar est tourné. Elle nous permet de savoir s'il faut diminuer la valeur de la position (si l'avatar se déplace vers les x négatifs) ou bien l'augmenter (si le déplacement est vers les x positifs), et de faire une rotation de 180° dans le cas où le personnage doit effectuer un demi-tour.

Pour les collisions, il était alors simple de borner la variable stockant la position. La marche et la course sont alors stoppées si le personnage est sur le point de percuter une chaise. Pour pouvoir activer la danse, nous avons donné une marge supplémentaire, car l'amplitude des jambes faisait qu'elles rentraient en collision avec la chaise.

g) L'environnement

Comme décrit précédemment, le personnage évolue dans un couloir, et est bloqué par deux chaises. Nous avons donc représenté le couloir par un seul mur et un sol. Le sol est quant à lui composé de dalles bicolores, afin de bien se rendre compte que le personnage avance lors de la danse, la marche et la course.

Les chaises utilisées reprennent le modèle des chaises de bureau à roulettes. On retrouve des sphères aplaties pour le dossier et l'assise, des sphères pour les roulettes et des cylindres pour la structure. Les deux chaises sont orientées l'une face à l'autre.

h) Camera et autres

Nous avons vu en TP comment utiliser la souris pour pouvoir modifier l'angle de vue. Nous avons décidé de garder cette option, mais son utilisation restait laborieuse fortement contraignante. Nous avons donc décidé d'implémenter des vues prédéfinies permettant de passer de passer rapidement d'une vue à une autre. Nous vous proposons donc les angles de caméra suivants : face, dos, dessus, coté et décalé.

De plus, nous avons exploité lors du développement la possibilité de dilater la variable temporelle (ralentir ou accélérer le temps). Cela nous a permis de mieux observer l'exécution des divers mouvements en ralentissant le temps. Nous avons laissé cette option dans le programme rendu lors de la soutenance.

i) Récapitulatif des touches

Les touches suivantes sont également actives lors de l'exécution du code dans la console :

Touches	Actions
m	marcher
r	courir
d	danser
a	s'asseoir
l	se lever
s	faire demi-tour
0	mode manuel
1	mode aléatoire
4	vue de dessus
5	vue de derrière
6	vue de face
7	vue décalée
8	vue de profil
+	accélérer le temps
-	ralentir le temps

II- Perspectives d'évolutions

Le sujet que vous nous avez proposé peut facilement être soumis à diverses évolutions, permettant d'ajouter des fonctionnalités et d'accroître le réalisme. Nous aurions pu autoriser l'action de plusieurs mouvements à la fois, comme par exemple marcher et saluer. Cependant, il aurait pour cela fallu revoir l'organisation de la partie chargée de représentée les mouvements. La structure globale du programme est d'ailleurs la difficulté principale à laquelle nous avons été confrontés.

Nous nous sommes concentrés majoritairement sur la fluidité des mouvements et le rendu purement graphique de l'avatar. Il aurait été intéressant de développer la partie liée au déplacement et à la gestion des collisions. Nous avons vu d'autres groupes réussir ceci lors de la soutenance, et le rendu était intéressant. Enfin, après avoir réussi l'affichage d'un premier personnage, nous aurions aussi pu en afficher un second, qui aurait interagit avec ce dernier.

Nous aurions également pu pousser un peu plus le réalisme des personnages et des décors, créer un visage et y associer des expressions, ajouter d'autres éléments de décors en permettant à l'avatar d'interagir avec ceux-ci, ajouter des animations supplémentaires comme le saut d'obstacles, simuler un tir à l'arc, tourner la tête, s'accroupir et encore bien d'autres.

Nous n'avons pas pensé à assigner un objet au personnage, mais il aurait été intéressant de le faire tenir une épée, un bâton, un ballon de basket ou un bouclier et d'effectuer une animation en fonction de l'objet. En poussant un peu plus la réflexion en ce sens, on aurait pu choisir son objet au début et en changer à tout moment selon l'humeur.

La dernière perspective envisageable selon nous serait d'allier formes complexes et ajout de texture pour pousser encore plus le réalisme de la scène. La gestion de la lumière et des ombres serait également un point important à aborder afin de créer l'illusion du réel.

III- Accomplissements personnels

M. Lenartowicz :

Ce projet a été très intéressant à développer, car il m'a permis d'accéder à de nouvelles connaissances. Il était très important et difficile de, par exemple, bien organiser le programme, et de savoir où appeler telle ou telle fonction.

Le fait de pouvoir constater directement le résultat de notre travail fut quelque chose de très motivant, et cela m'a aussi permis de mieux comprendre comment une application 3D, comme un jeu vidéo, peut être structuré.

M. Madelaine :

L'animation de l'avatar représentait pour moi un véritable challenge et, une fois celle-ci faite, elle ouvra un large champ de possibilités jusque-là insoupçonnées : je peux maintenant envisager la création de diverses animations simples comme sauter, pointer du doigt, frapper, shooter, brandir une épée etc...

Mais ce n'est pas le seul point intéressant que j'ai appris au cours de ce projet. En effet, la gestion de la lumière et des reflets est plus complexe qu'elle n'en a l'air puis vient la gestion des textures (même si nous n'avons pas pu assimiler assez d'information pour ajouter ce procédé à notre projet).

IV- Conclusion

L'accomplissement de ce projet s'est déroulé dans une ambiance amicale. Nous avons travaillé de manière régulière afin de ne pas se retrouver bloqué par le temps. Nous avons tous les deux apporté nos idées à ce programme, ainsi que les différentes solutions pour les ajouter. Dans l'ensemble, nous sommes fiers de la quantité et de la qualité de ce que nous vous proposons.

J'espère que vous êtes également satisfait de notre programme. Nous remercions le corps enseignant pour la qualité de son enseignement, et pour les différentes réponses que vous avez pu nous apporter.