



ÉCOLE CENTRALE DE LYON

RAPPORT DE FIN DE PROJET

VISUALISATION DE MOUVEMENT 3D ACQUIS PAR MOTION CAPTURE

PAR 114

Pauline ROMAGON

Quentin LEMAIRE

Tuteurs : Stéphane DERRODE et Romain VUILLEMOT

Conseillère : Sandrine BEC

Promotion entrante 2015

11 avril 2017

Résumé

Le projet d'application recherche 114 a vu le jour en juillet 2016, à la suite de l'acquisition d'une plateforme de Motion Capture au sein de l'Ecole Centrale de Lyon. Son objectif principal est de réaliser des rendus graphiques visuels esthétiques plus complexes que ceux fournis par le logiciel associé au système d'acquisition 3D, puis d'automatiser la réalisation de ces rendus.

Dès septembre, nous nous sommes lancés dans ce projet, encadrés par deux tuteurs scientifiques Stéphane Derrode et Romain Vuillemot ainsi qu'une conseillère Sandrine Bec, également responsable de l'ensemble de l'activité PAR.

Une première partie du travail a été de nous approprier la plateforme Amigo et le logiciel Nexus associé, à travers des acquisitions simples. Nous étions le premier groupe à découvrir cette plateforme. L'élaboration de nouvelles acquisitions plus complexes a posé des problèmes de gestion du logiciel résolus notamment grâce à l'aide du support téléphonique Biometrics de la plateforme. Ce support nous permet aujourd'hui de maîtriser l'étape d'acquisition ce qui constitue une base solide et pérenne pour les projets futurs qui travailleront sur cette plateforme.

En ce qui concerne les modélisations, les premières ont été effectuées avec différents modules de Python. Rapidement limités, nous avons évolué vers une représentation sous forme d'animation 3D avec le logiciel Blender qui permettait davantage de liberté. Pour cela nous nous sommes concentrés sur des extraits précis d'acquisition afin d'obtenir des rendus visuels plus travaillés. La modélisation d'un personnage puis son animation à partir des données 3D extraites ont abouti à des rendus très satisfaisants bien que non automatisés.

Nous avons par ailleurs continué à programmer avec Python dans le traitement des données pour ce qui est de l'aspect audio de nos rendus. Les programmes audios ont été créés de sorte qu'ils s'adaptent à toute acquisition quelle que soit sa durée et le nombre de ses marqueurs. Ce travail, initié par le binôme, s'inscrit pleinement dans l'objectif d'automatisation des rendus.

De belles perspectives s'offrent aux prochains projets travaillant sur la plateforme puisque notre travail a permis une avancée considérable dans la découverte et l'appropriation de la plateforme et de ses utilisations diverses.

Table des matières

1	Remerciements	3
2	Introduction	4
3	Contexte scientifique et technique	5
3.1	Introduction à la Motion Capture	5
3.2	La Plateforme AMIGO	6
3.3	Problématique et enjeux	6
3.4	La démarche globale : étapes et outils techniques	6
4	L'acquisition des données	7
4.1	Le fonctionnement de la plateforme	7
4.2	Faire une acquisition	8
4.3	Les difficultés rencontrées et solutions adaptées	8
4.4	Automatisation et perspectives	9
5	Le traitement des données	10
5.1	Objectifs	10
5.2	Solutions envisagées	10
5.3	Mise en place et résultats	10
5.3.1	Vitesse	10
5.3.2	Accélération	11
5.3.3	Représentation graphique des données	11
5.3.4	Lissage des données	12
5.3.5	Le calcul du maximum et minimum des valeurs d'un paramètre donné pour l'ensemble des marqueurs	12
5.4	Perspectives futures	13
6	Rendus, modélisation et animation 3D	14
6.1	Objectifs	14
6.2	Solutions envisagées	15
6.2.1	Une première approche : le module Plotly de Python	15
6.2.2	Une autre approche : le module Matplotlib de Python	15
6.3	Mise en place et résultats : travail sur le logiciel Blender	17
6.3.1	Modélisation d'un personnage	17
6.3.2	Création d'une armature animée à partir des données exportées	17
6.3.3	Mise en mouvement du personnage modélisé	18
6.4	Perspectives futures	19

7 L'aspect audio	20
7.1 Objectifs	20
7.2 Solutions envisagées	20
7.2.1 Paramètres relatifs aux marqueurs	20
7.2.2 Paramètres relatifs au son	21
7.2.3 Choix	21
7.3 Mise en place et résultats	22
7.3.1 Principe général de création de la musique	22
7.3.2 Paramètres utilisés	22
7.3.3 Définition du problème d'un point de vue musical	23
7.3.4 Automatisation	23
7.3.5 Résultats	23
7.4 Perspectives futures	24
8 Partage et pérennité du projet	25
8.1 Mise en open source de notre travail sur le site web Github	25
8.2 Les tutoriels d'utilisation de la plateforme	25
9 Conclusion	26
A Les types de données	27
B Les modules de Python	28
C Détails du traitement audio	29

Chapitre 1

Remerciements

Nous tenons avant toute chose à remercier les différents acteurs qui, de près ou de loin, ont participé au bon déroulement de ce projet.

Merci à nos tuteurs Stéphane Derrode et Romain Vuillemot pour leur conseils judicieux et leur disponibilité tout au long de l'année.

Merci à La Région ainsi qu'à l'Ecole Centrale de Lyon sans qui nous n'aurions pu travailler sur cette plateforme.

Merci à notre conseillère Sandrine Bec pour son aide pertinente.

Merci à Emeline et Arthur pour la mise en commun de notre travail au sein de cette plateforme.

Merci à Thomas Martin du service support Biometrics pour son aide précieuse face aux difficultés rencontrées avec le logiciel Nexus.

Merci à Bastien Louvet et plus généralement à l'équipe du projet d'étude 41 pour leur coopération.

Merci à Marius Blanchard pour sa participation sur le traitement sonore.

Merci à l'association Centrale Television Network pour le prêt de matériel.

Enfin, merci à tous ceux qui nous ont encouragés.

Chapitre 2

Introduction

De nos jours, la Motion Capture, technique d'enregistrement des mouvements d'objets animés ou d'êtres vivants, a permis de nombreuses avancées dans le domaine médical et dans celui de l'animation (cinéma et jeux vidéos).

Il existe différents types de Motion Capture. La Motion Capture optique est la plus courante, il s'agit de placer des marqueurs sur l'objet ou le modèle ; ces marqueurs vont, tout en suivant le mouvement, recevoir des rayons infrarouges envoyés par les caméras qui vont à leur tour détecter les réflexions des rayons sur les marqueurs et donc les positions de chacun. Ces positions sont donc représentées virtuellement et en temps réel par ordinateur. Cependant, si des obstacles entravent le parcours des rayons entre la caméra et les marqueurs cela pose problème. C'est là qu'intervient la Motion Capture Gyroscopique : les marqueurs sont remplacés par des capteurs gyroscopiques et inertiels qui permettent de repérer en temps réel l'angle et la position des capteurs, avec l'aide d'un émetteur qui assure la transmission des données vers l'ordinateur.

Toutefois, la motion capture est également possible sans capteurs : par exemple, pour les expressions du visage, des animations peuvent être réalisées à partir de modélisations 3D d'humains filmés appliquées à d'autres modèles. Enfin, la motion capture par Phosphore consiste à recouvrir le modèle de phosphore, visible dans le noir, et à filmer ce modèle à l'aide de couples de caméras et d'éclairages stroboscopiques afin d'en obtenir une modélisation visible et virtuelle.

Dans le cadre de ce projet d'application, nous travaillerons avec la Motion Capture Optique. Dans ce rapport, nous présenterons dans un premier temps le contexte scientifique et technique du projet ainsi que ses objectifs et enjeux ; nous détaillerons ensuite, pour chacune des étapes de notre démarche, les objectifs, les solutions envisagées et mises en place, puis les résultats obtenus et les perspectives correspondantes. Enfin, nous conclurons sur la globalité du projet et sur l'importance du travail fourni pour les projets amenés à travailler sur cette plateforme. La figure 2.1 détaille l'articulation des différentes étapes de notre démarche et présente les éléments clés de leur fonctionnement.

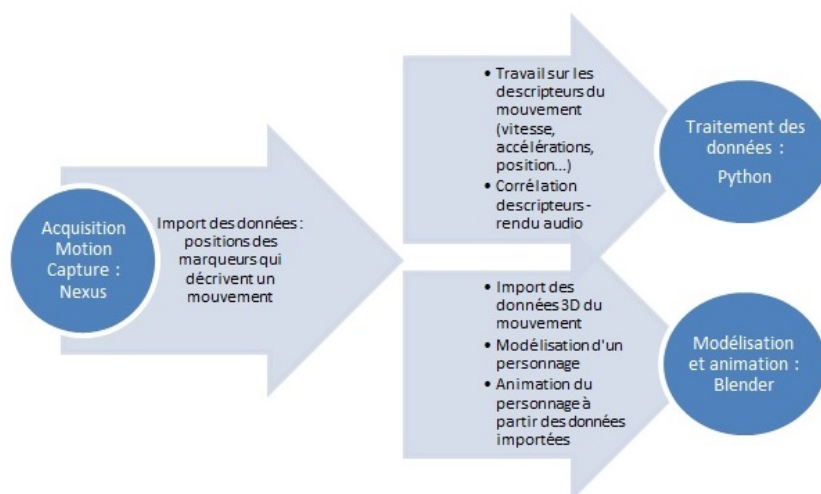


FIGURE 2.1 – Articulation des différentes étapes du projet et description de leur fonctionnement clé

Chapitre 3

Contexte scientifique et technique

3.1 Introduction à la Motion Capture

La Motion Capture, ou capture de mouvement, est une technique qui permet d'enregistrer les mouvements ou bien les positions d'objets ou d'êtres vivants afin d'en obtenir une modélisation virtuelle sur ordinateur. Lors d'une acquisition, une restitution visuelle est faite en temps réel grâce à un logiciel associé au matériel utilisé. Ces acquisitions peuvent être stockées et ensuite exploitées pour générer des animations.

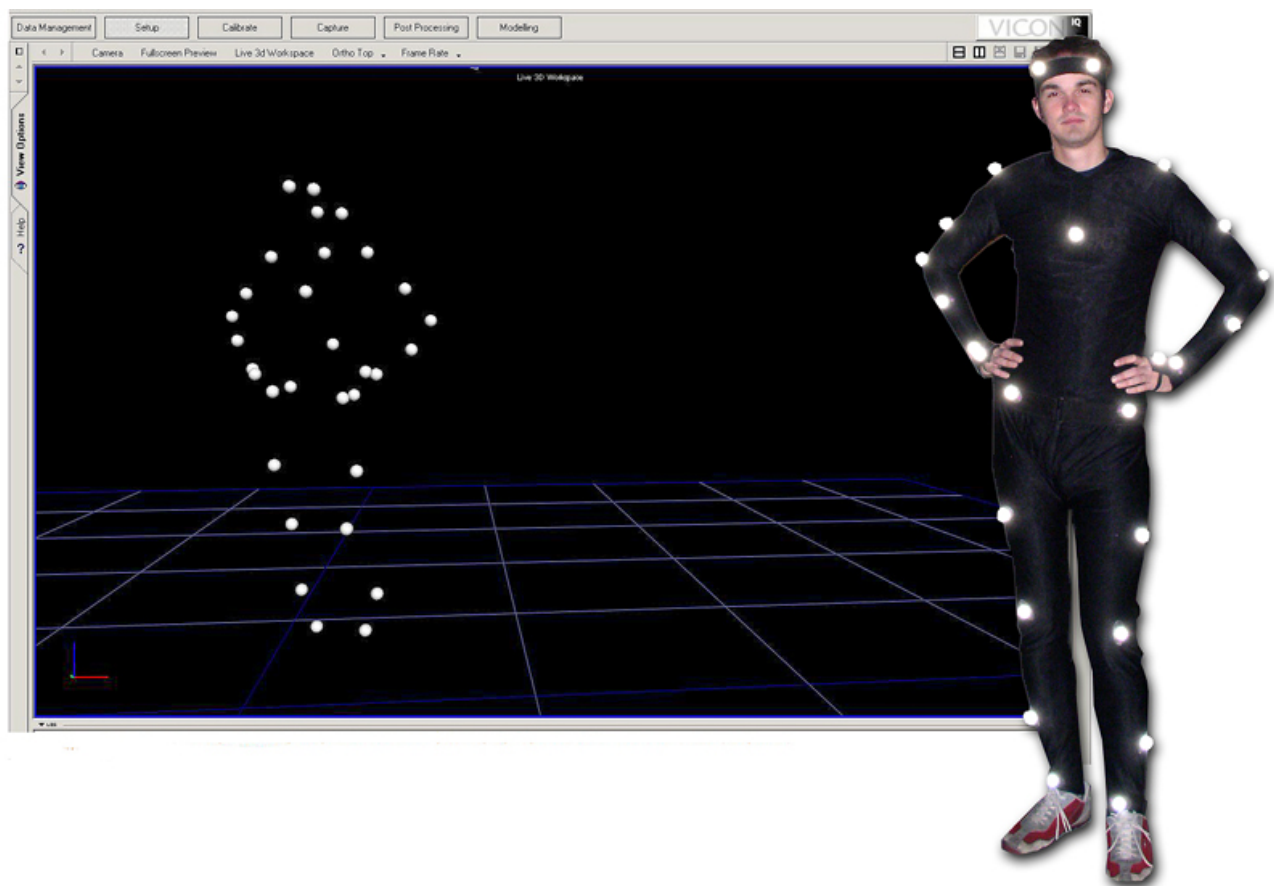


FIGURE 3.1 – Exemple d'un modèle humain avec des marqueurs ainsi que sa représentation visuelle
Source : <http://www.assignmentpoint.com/science/computer/motion-capture.html>

Depuis 1990, la Motion Capture (figure 3.1) est utilisée dans le cinéma afin de produire différents types d'animations à des fins cinématographiques : des images de synthèse, des effets spéciaux et des outils artistiques. Toutefois c'est dans le jeu vidéo que la Motion Capture s'est majoritairement développée. Les graphismes sont

passés de dessins animés exagérés à des mouvements beaucoup plus réalistes adaptés notamment à des jeux sportifs ou guerriers.

Des avancées dans le domaine médical ont également été permises : en autres la visualisation de radio à l'aide de la 3D.

3.2 La Plateforme AMIGO

La plateforme AMIGO a été récemment acquise par l'Ecole Centrale de Lyon notamment grâce à l'aide de la région, et installée au bâtiment W1, elle comprend un dispositif de Capture du Mouvement.

Six caméras nous permettent de suivre en temps réel le mouvement d'un objet ou d'un corps humain, grâce à la reconnaissance de marqueurs infrarouges réfléchissants judicieusement placés. Ces caméras ont une haute définition spatiale et temporelle et une fréquence d'acquisition de 250Hz.

Le logiciel Nexus, fourni avec le dispositif, permet non seulement de suivre les mouvements grâce à une restitution visuelle extrêmement précise mais également de les étudier, les exploiter, et d'en extraire des données. Il permet également de rejouer les mouvements, d'ajouter des segments liés entre les marqueurs qui permettent par exemple de reconstituer un squelette.

L'acquisition de cette plateforme a été source de nouveaux projets d'étude et de projets d'application en recherche.

3.3 Problématique et enjeux

L'objectif principal de ce projet d'application recherche est de **générer des rendus graphiques** esthétiques et plus complexes que ceux fournis par le logiciel Nexus ; l'enjeu étant donc d'enrichir les rendus des mouvements 3D en prenant en compte certains descripteurs de mouvement comme l'accélération, la vitesse, l'amplitude du mouvement (que l'on peut extraire grâce à Nexus). Cette première composante de la problématique concerne majoritairement l'aspect artistique des rendus graphiques.

La Motion Capture est toutefois un processus non automatisé qui a comme défaut de représenter un coût élevé et demander un travail manuel conséquent. Une composante non négligeable du projet concerne alors la réalisation automatique des traitements, l'objectif secondaire est donc de **développer des programmes capables de générer automatiquement ces rendus graphiques**.

3.4 La démarche globale : étapes et outils techniques

La démarche globale du projet est scindée en plusieurs étapes auxquelles des outils spécifiques sont associés.

- **L'acquisition des données et le logiciel Nexus** : la première étape consiste à faire une acquisition en Motion Capture, c'est-à-dire à extraire des données 3D. Pour ce faire, nous avons travaillé à la plateforme Amigo et le logiciel Nexus présentés précédemment. Les données ont été exportées sous deux types différents détaillés dans l'annexe A.
- **Le traitement algorithmique des données avec le logiciel de programmation Python et ses modules** : les données extraites ont été exploitées via le logiciel de programmation Python, notamment via les modules détaillés dans l'annexe B. D'une part, cela a permis l'étude de descripteurs des marqueurs (vitesse, accélération, position...), et d'autre part cela a permis un travail conséquent pour générer une partie audio des rendus corrélée avec les descripteurs.
- **Le rendu, la modélisation et l'animation 3D avec le logiciel Blender** : finalement, la dernière étape de la démarche aura été de modéliser un personnage, puis de l'animer grâce aux données de Motion Capture importées, afin de générer un rendu visuel, le tout grâce au logiciel Blender qui permet de créer et animer tous types de personnages ou d'objets.

4.2 Faire une acquisition

Avant de faire une acquisition, il est nécessaire de calibrer le matériel : cela revient à calibrer les caméras, calibrer l'origine de l'espace 3D, puis traiter et éliminer les différents détails parasites (points de l'espace détectés par les caméras et présents sur l'écran, là où aucun marqueur n'est).

Pour procéder à une acquisition, il faut ensuite calibrer un sujet : cela revient à faire un enregistrement court de l'objet ou du corps considéré, à relier graphiquement les marqueurs par des segments et lier les segments entre eux afin d'obtenir un squelette, puis appliquer une "pipeline" propre au logiciel Nexus (outil qui va valider la construction du squelette).

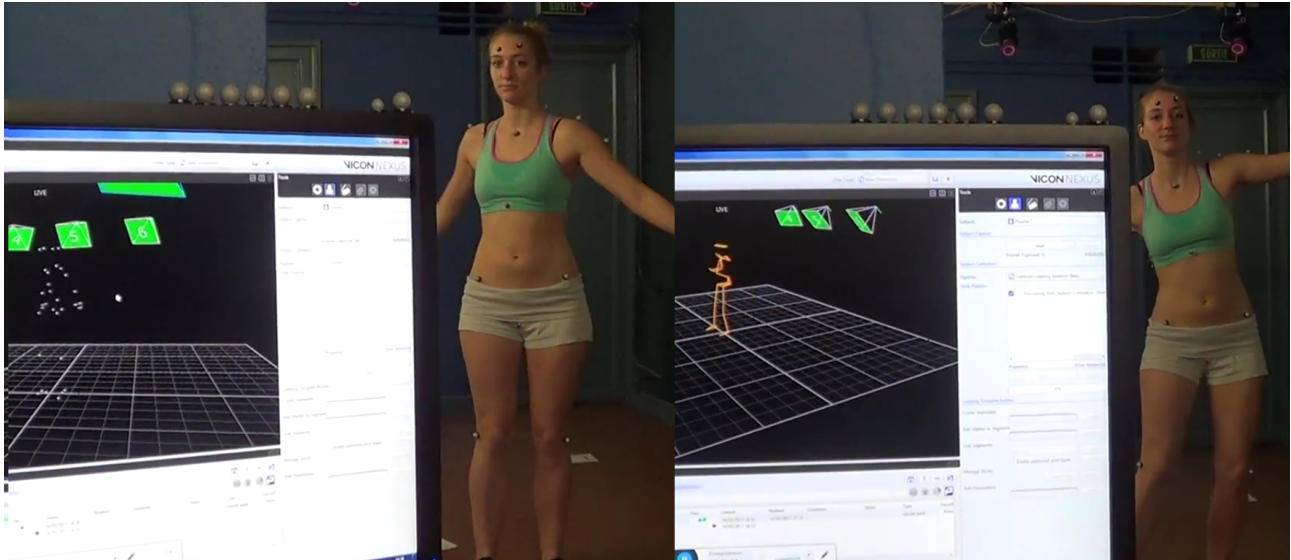


FIGURE 4.2 – Aperçu de la reconnaissance du sujet avant (gauche) et après (droite) l'étape de calibration

La figure 4.2 représente l'interface avant et après la calibration d'un sujet. Une fois le sujet calibré, tout enregistrement suivant reconnaîtra automatiquement le squelette. Cela sera notamment utile pour la reconstitution des trajectoires perdues : pour certains marqueurs "perdus" (non visible lors d'un mouvement - on parle alors de "gap"), il est possible de retrouver leur position à partir des positions de leurs marqueurs voisins.

Remarque : lors de la création des segments dans la calibration du sujet, il est important de labéliser tous les marqueurs en suivant la labélisation présentée sur la figure 4.1, cela s'avère utile à la fois lorsque l'on comble les gaps et à la fois dans l'export des données (noms des marqueurs exportés avec leurs positions).

4.3 Les difficultés rencontrées et solutions adaptées

La découverte du fonctionnement des acquisitions s'est faite progressivement :

- les premières acquisitions étaient de courte durée (environ 5 secondes) avec peu de marqueurs (deux, par exemple coude et poignet) : le mouvement était peu reconnaissable, la création d'un segment et l'ajout d'une pipeline fonctionnaient.
- les suivantes étaient plus complexes (10 secondes environ) et comptaient environ 10 marqueurs (par exemple 7 : cou, épaule, bras, coude, 2 au poignet, main) : la création des segments fonctionnait, certains marqueurs étaient perdus lors des acquisitions mais l'ajout correct d'une pipeline permettait de combler ces gaps.
- les dernières acquisitions représentaient un squelette complet : quelle que soit la durée de l'enregistrement, la calibration du sujet posait problème.

Lors du processus d'acquisition sur un squelette complet, nous avons dû faire face à de nombreuses difficultés.

L'ajout d'une pipeline qui valide le squelette construit requiert une méthodologie rigoureuse de labélisation, de création des segments et de liaison qu'il a été possible d'établir seulement après de multiples essais. Dans

la plupart des cas, l'ajout de pipeline n'aboutissait pas, et sans pipeline les enregistrements contenaient des gaps en permanence. C'est avec l'aide du support de l'entreprise Biometrics que cette méthodologie a pu être complétée (ajout d'un ordre de construction, principe de liaisons libres et d'origine "World").

Tous ces principes seront expliqués plus en détails dans les tutoriels vidéos dont nous parlons dans la partie 8.2.

De plus, toute modification de distance entre deux marqueurs entraînait la non-reconnaissance du segment qui les lie : un vêtement ample ou qui bouge nuisait au processus d'acquisition. Ne possédant pas de combinaison moulante spéciale pour la Motion Capture, nous avons effectué les acquisitions suivantes en plaçant les marqueurs sur la peau.

4.4 Automatisation et perspectives

Finalement, les nombreux essais d'acquisitions ont permis une meilleure appropriation du fonctionnement du logiciel qui nous permettront de transmettre aux prochains groupes la bonne méthodologie de calibration de sujet. Cette méthodologie s'inscrit pleinement dans l'objectif d'automatisation des rendus, puisque l'étape d'acquisition est alors automatique une fois le sujet calibré.

De plus, cela représentera un gain de temps pour ceux amenés à travailler sur la même plateforme : si les premières acquisitions sur squelette complet duraient initialement 4h, aujourd'hui elles sont possibles en moins de deux heures.

Chapitre 5

Le traitement des données

5.1 Objectifs

Avant toute utilisation des données à des fins d'animation, nous nous sommes intéressés au traitement des données pour faire ressortir des valeurs intéressantes et caractéristiques du mouvement, cela a notamment facilité l'utilisation des données.

Ce traitement aide à la représentation, permet de mieux visualiser les données (par exemple par l'intermédiaire de courbes) ou de pré-calculer certaines valeurs.

5.2 Solutions envisagées

Pour traiter les données, nous avons décidé d'un certain nombre de calculs et de fonctionnalités à mettre en place :

- Le calcul de la vitesse de chaque marqueur
- Le calcul de l'accélération de chaque marqueur
- La représentation graphique de la vitesse, de l'accélération ou de la position par rapport à l'origine
- Le lissage des valeurs pour réduire le bruit capté lors de l'acquisition
- Le calcul du maximum et minimum des valeurs d'un paramètre donné pour l'ensemble des marqueurs

De nombreux choix ont été nécessaires pour les méthodes de calcul : c'est le cas pour la vitesse, l'accélération ou surtout pour le lissage.

Ces choix seront détaillés dans la section suivante.

5.3 Mise en place et résultats

5.3.1 Vitesse

Pour calculer la vitesse, nous utilisons le principe de la dérivation numérique.

Notons $x^i(n)$ la position du i^{ieme} marqueur pour la mesure i . Grâce à la fonction *diff* de la bibliothèque *numpy* de Python, nous obtenons la vitesse du marqueur i selon x par :

$$v_x^i(n) = x^i(n+1) - x^i(n) \quad (5.1)$$

On remarque que le résultat obtenu est proportionnel à la vitesse à un facteur près (facteur qui correspond à la durée entre deux mesures, qui est constante) [1].

Comme nous n'allons utiliser la vitesse qu'à titre de comparaison entre les marqueurs dans la suite, nous garderons cette valeur telle quelle.

Pour obtenir ensuite la vitesse globale d'un marqueur, nous avons pris le module de la vitesse de chaque coordonnée selon la formule suivante :

$$v^i(n) = \sqrt{(v_x^i(n))^2 + (v_y^i(n))^2 + (v_z^i(n))^2} \quad (5.2)$$

5.3.2 Accélération

Comme pour le calcul de la vitesse, nous avons réitéré le même principe au le calcul de l'accélération à partir des valeurs de vitesses obtenues.

Une fois de plus, le résultat ne sera que proportionnel à l'accélération mais ce ne sera pas pénalisant pour l'utilisation que nous en ferons.

5.3.3 Représentation graphique des données

Toutes les figures qui seront affichées dans la suite de cette section proviennent de l'acquisition du mouvement simple d'un bras, le même mouvement sera utilisé pour la représentation simple à l'aide des bibliothèques *matplotlib* et *plotly* détaillées dans la partie 6 : Rendus, modélisation et animation 3D.

L'objectif de cette représentation n'est donc pas d'avoir une représentation graphique exacte des données mais de pouvoir comparer les valeurs pour plusieurs marqueurs.

L'algorithme réalisé doit alors mettre à la même échelle chaque graphique pour permettre une comparaison claire.

L'abscisse utilisée ici correspondra au numéro de la mesure qui est proportionnelle au temps, en effet, la fréquence d'acquisition est de $250Hz$. Une fois de plus, il ne s'agit pas d'avoir des valeurs exactes donc nous ne convertirons pas cette valeur.

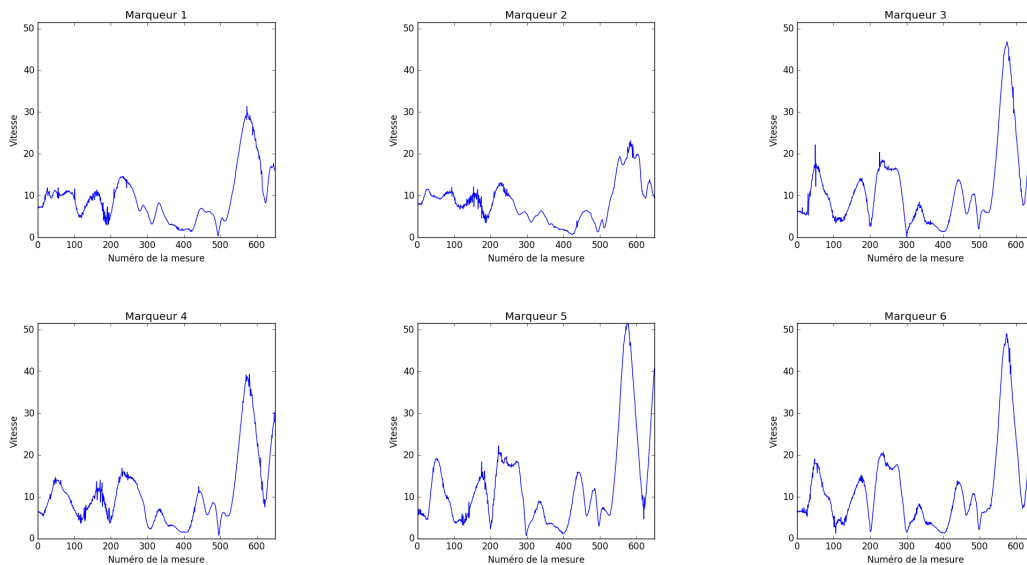


FIGURE 5.1 – Représentation de la vitesse de six marqueurs

Le résultat obtenu sur la figure 5.1 nous permet bien de pouvoir comparer les vitesses des différents marqueurs du bras et donne une bonne idée de l'allure des courbes. On remarque par exemple que la vitesse des marqueurs du poignet (6 et 7) est plus élevée que celle du coude (marqueur 2).

Cependant, il y a un bruit assez important dans les données que nous allons essayer de lisser.

5.3.4 Lissage des données

L'objectif de cette section est d'éviter les fluctuations brutales des données dans un espace de temps très réduit à cause du bruit. Ces fluctuations pourraient avoir un impact sur le résultat dans les parties 5.3.5 : Couleurs des marqueurs et 7 : Audio que nous développerons plus tard.

Un choix a de nouveau été nécessaire, en effet, il fallait que le lissage soit suffisant pour que les fluctuations n'aient pas d'impact dans les études futures et que les données soient déformées le moins possible. Plusieurs méthodes ont donc été testées.

Nous avons retenu la méthode qui nous semblait être le meilleur compromis : l'utilisation d'un lissage par moyenne mobile [2] d'ordre 5 pour la vitesse et d'ordre 7 pour l'accélération. Soit $x(n)$ la $n^{ième}$ mesure de la liste de données à lisser, on définit alors $x_l(n)$ la mesure lissée à l'ordre 5 par :

$$x_l(n) = (x(n-2) + x(n-1) + x(n) + x(n+1) + x(n+2))/5 \quad (5.3)$$

Voici les 6 courbes de vitesses précédentes lissées :

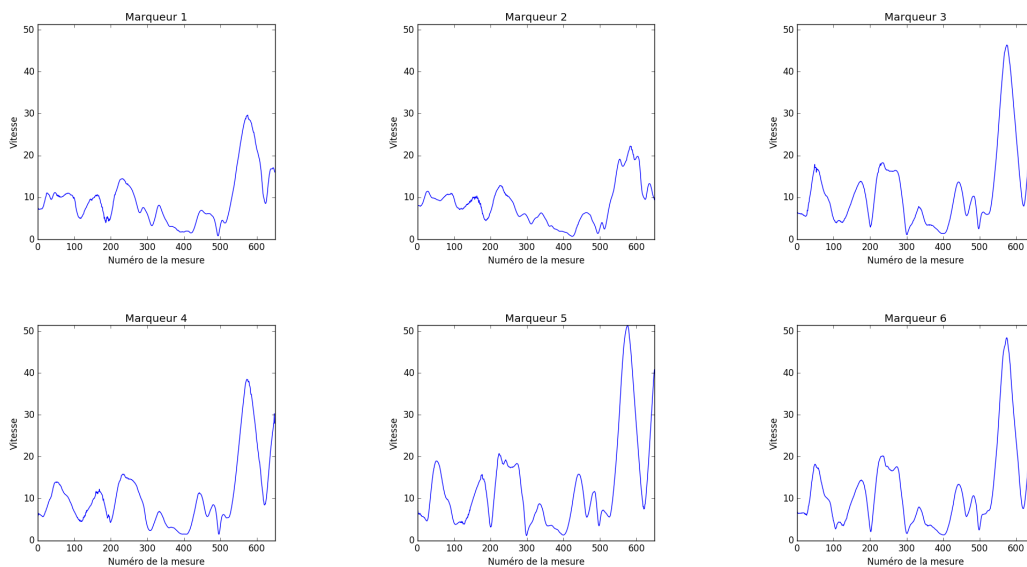


FIGURE 5.2 – Représentation de la vitesse lissée de six marqueurs

Le résultat, figure 5.2, convient à nos attentes puisque le bruit est atténué et les données sont restées fidèles au mouvement.

5.3.5 Le calcul du maximum et minimum des valeurs d'un paramètre donné pour l'ensemble des marqueurs

Comme expliqué précédemment, les valeurs de la vitesse et de l'accélération ne nous intéresseront pas mais seront seulement utilisées à titre de comparaison.

Nous avons donc implémenté un algorithme qui permettra de calculer le maximum et le minimum d'une liste de données pour chaque marqueur. Par exemple, appliqué à la vitesse, l'algorithme retourne le maximum et minimum de vitesse atteint parmi l'ensemble des vitesses atteintes par l'ensemble des marqueurs. Cela fournit donc une plage de données dans laquelle toutes les valeurs de vitesses de tous les marqueurs seront comprises.

Application : couleur par marqueur

Une des applications possibles, en dehors de la comparaison, est d'associer une couleur par marqueur à chaque instant en fonction de la donnée choisie (position, vitesse, accélération, etc ...).

Cela permettra de mettre en avant les marqueurs lorsque les valeurs sont plus élevées par des couleurs chaudes.

Pour cela, nous avons choisi d'utiliser une affectation linéaire des couleurs entre la valeur minimale et maximale. La minimale correspond au bleu et l'on passe par les couleurs de l'arc-en-ciel pour atteindre le rouge à la valeur maximale [3].

5.4 Perspectives futures

Dans une optique d'automatisation, les études effectuées sur le traitement des données sont très bonnes car aucune étape manuelle n'est nécessaire.

A l'avenir, d'autres traitements pourront s'avérer nécessaires pour des représentations et des études plus poussées et il sera sûrement utile d'essayer des méthodes de lissage et de calcul plus optimisées, cela n'a pas été nécessaire dans le cadre de notre projet.

Chapitre 6

Rendus, modélisation et animation 3D

6.1 Objectifs

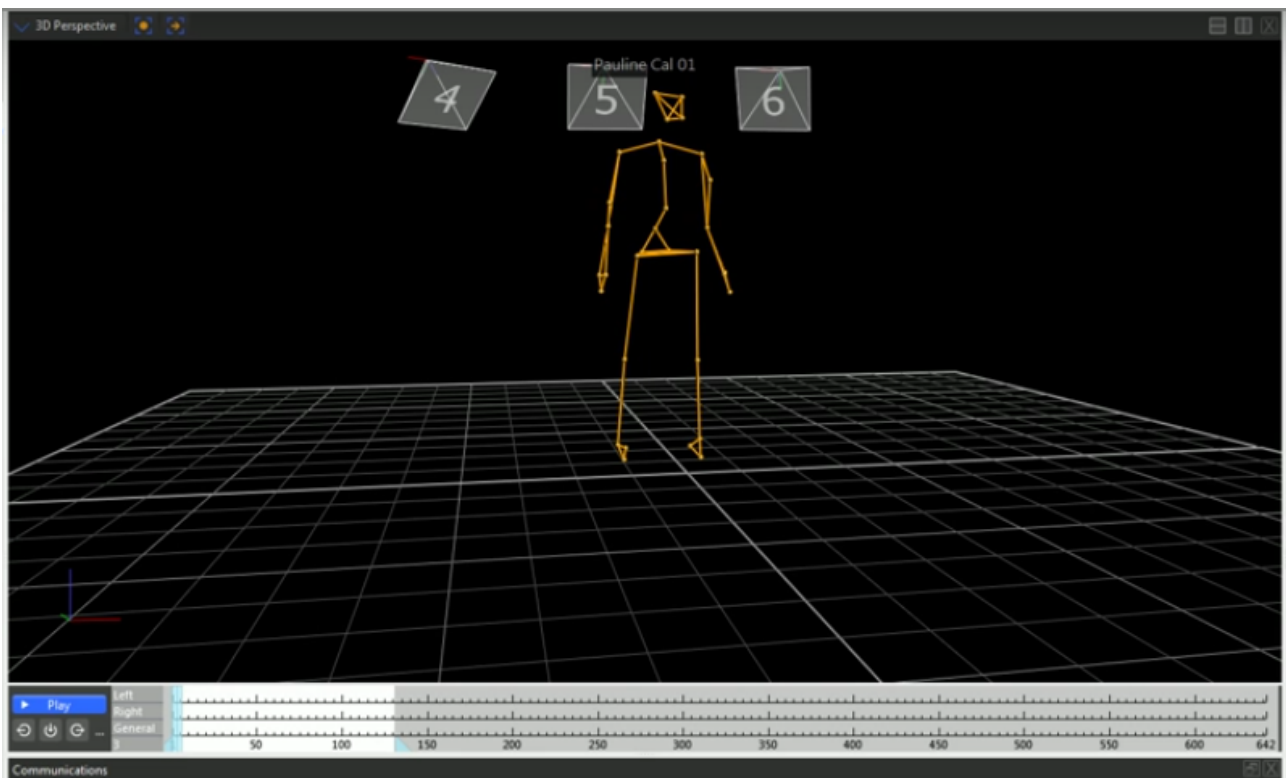


FIGURE 6.1 – Aperçu d'un squelette modelisé via le logiciel Nexus

La figure 6.1 représente le rendu visuel de l'interface Nexus avec laquelle nous avons travaillé. On devine un squelette humain reconstitué par des segments à l'aide des marqueurs présents. Si cette représentation est assez pratique pour perfectionner des détails de l'acquisition, il est cohérent de ne pas la juger esthétique pour autant.

L'objectif est donc de réaliser des rendus graphiques plus esthétiques, bien que la notion soit abstraite, en s'appuyant sur l'étude de paramètres propres au mouvement acquis (vitesses des marqueurs, positions, accélérations...).

6.2 Solutions envisagées

6.2.1 Une première approche : le module Plotly de Python

Une première étape a consisté à reproduire les positions des marqueurs. Pour ce faire, nous avons travaillé avec le module Plotly de Python : il nous a permis d'appréhender une première représentation 3D des données extraites.

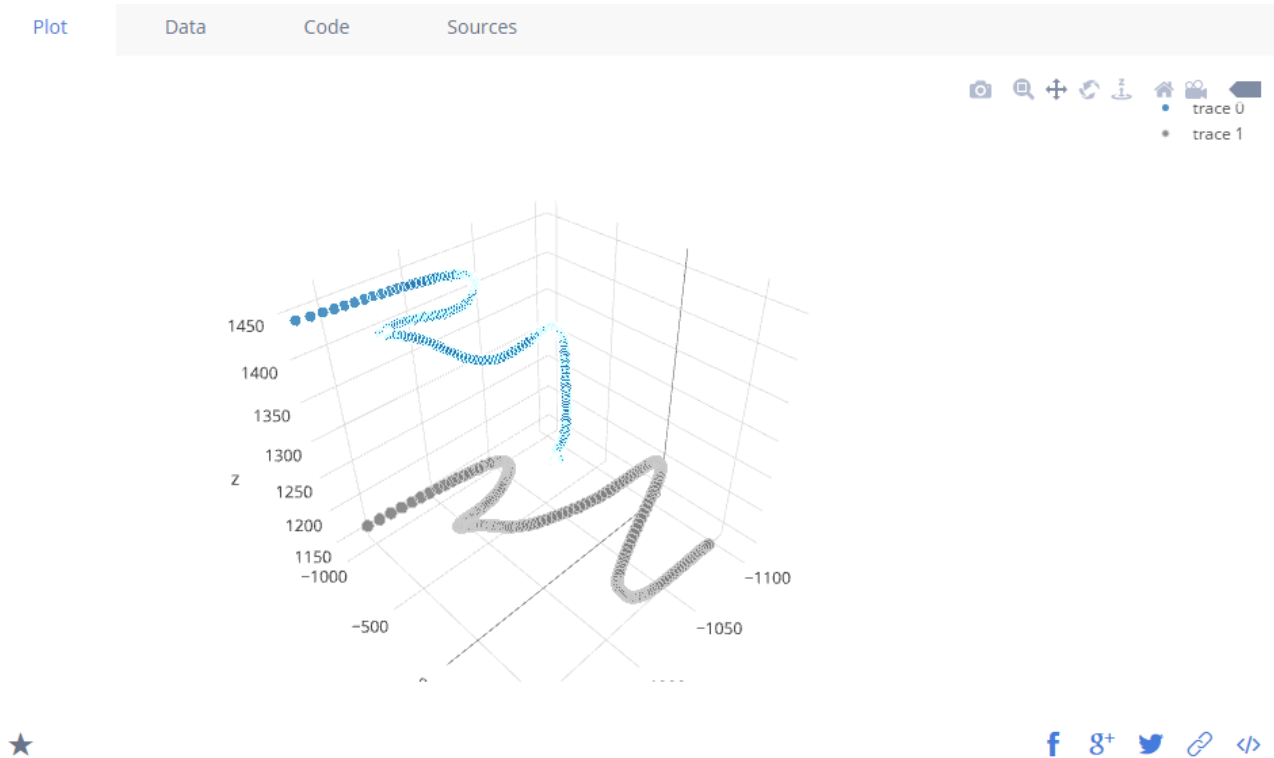


FIGURE 6.2 – Capture d'écran d'une représentation avec Plotly

La figure 6.2 représente les données de deux capteurs (coude et poignet). Le module a pour avantage de rendre toute représentation disponible sur internet grâce à l'envoi et au traitement de chaque donnée sur un serveur. L'animation manque toutefois de fluidité et conserve les trajectoires des marqueurs en affichant toutes les positions prises successivement. Elle est disponible sur : https://youtu.be/muG6M_u_07o

Face à ces premiers rendus peu pratiques nous avons décidé de travailler avec un autre module de Python.

6.2.2 Une autre approche : le module Matplotlib de Python

Ce deuxième module [4] a permis d'obtenir des rendus disponibles en local (non plus disponibles sur internet), ce qui s'est avéré suffisant pour l'exploitation de ces rendus ; les visuels étaient également plus complexes.

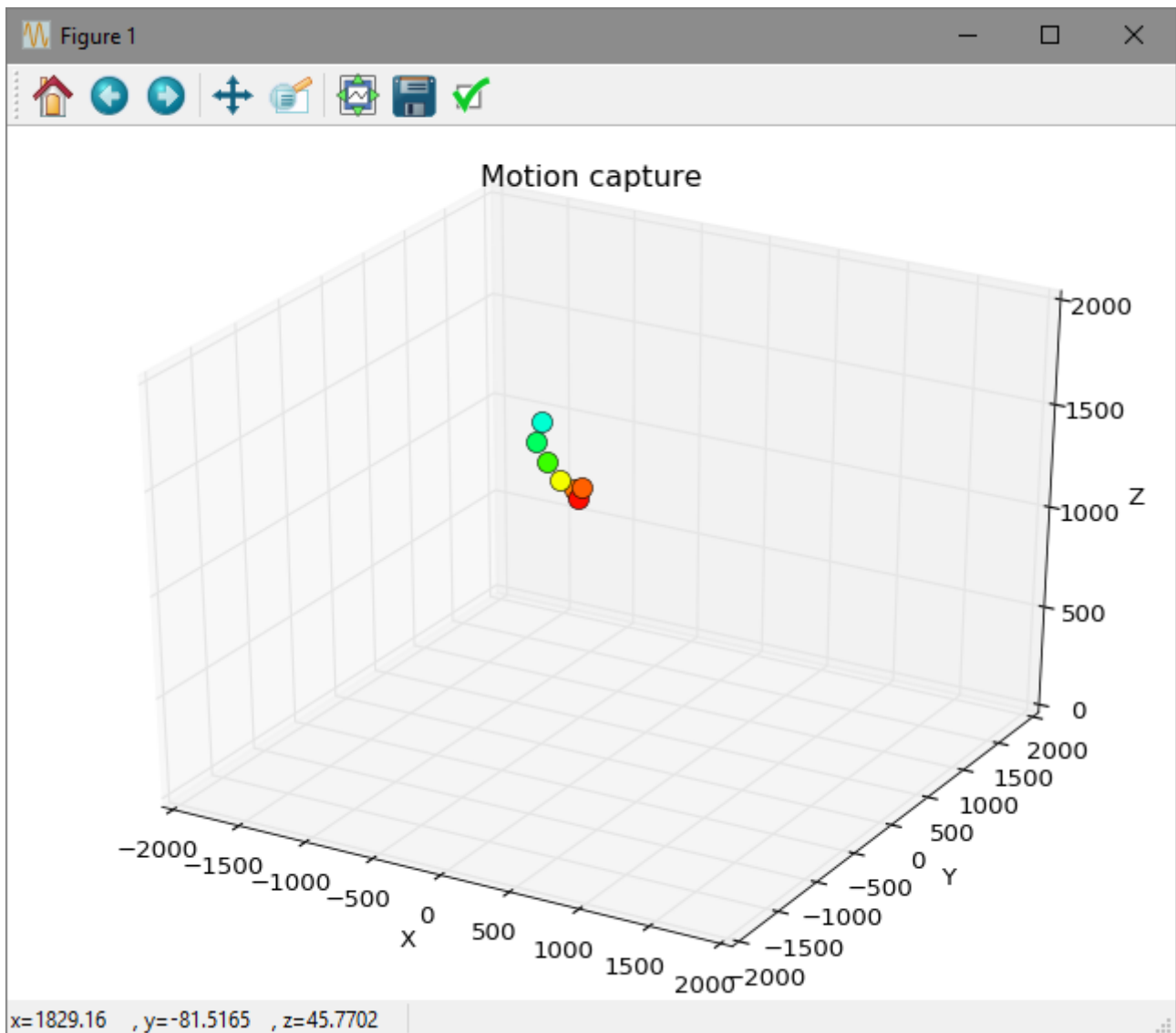


FIGURE 6.3 – Capture d'écran d'une représentation avec Matplotlib

La figure 6.3 représente le rendu 3D fourni en utilisant le module Matplotlib : l'acquisition correspondante contient 7 marqueurs répartis de l'épaule jusqu'à la main, le mouvement est un bras qui se balance. Une vidéo du rendu est disponible ici : https://youtu.be/8LnIc9n_TuQ

Cette deuxième approche est plus concluante. Elle a pour avantage de ne poser aucune limite de vitesse : les données n'ont pas à être téléchargées comme avec le module précédent, ce qui rend la vitesse du mouvement dans le rendu égale à la vitesse du mouvement acquis par Motion Capture.

Les tracés des trajectoires ont été supprimés et une méthode de programmation "orientée objet" permet de traiter les marqueurs individuellement et de leur affecter des couleurs selon des paramètres qui leur sont propres, grâce à la fonction détaillée dans le chapitre précédent.

Enfin, le travail avec le module Matplotlib a pour avantage de conserver l'automatisation des rendus : aucun travail manuel n'est effectué, le rendu est généré quels que soient le nombre de marqueurs et la durée de l'acquisition.

Si travailler avec le module Matplotlib a certains avantages par comparaison avec le module Plotly, les rendus restent toutefois simples (affichage des marqueurs seulement, pas de liberté de modélisation), c'est donc naturellement que nous avons choisi de travailler sur Blender, logiciel de modélisation et d'animation 3D. Il s'est avéré que les autres groupes de projet d'étude et d'application travaillaient aussi sur ce logiciel, ce qui a permis la mise en commun de nos travaux et une certaine entraide dans l'appropriation du logiciel.

6.3 Mise en place et résultats : travail sur le logiciel Blender

L'appropriation du logiciel Blender s'est déroulée en premier lieu grâce à un cours en ligne fourni sur le site Openclassroom [5] ; puis certaines étapes ont été appréhendées grâce aux différents tutoriels rédigés par l'équipe du projet d'étude 41.

6.3.1 Modélisation d'un personnage

Avant toute animation, il faut choisir puis modéliser un personnage ou objet à animer. Notre choix s'est arrêté sur un mannequin en bois, personnage créé lors des travaux pratiques de l'apprentissage avec Openclassroom.

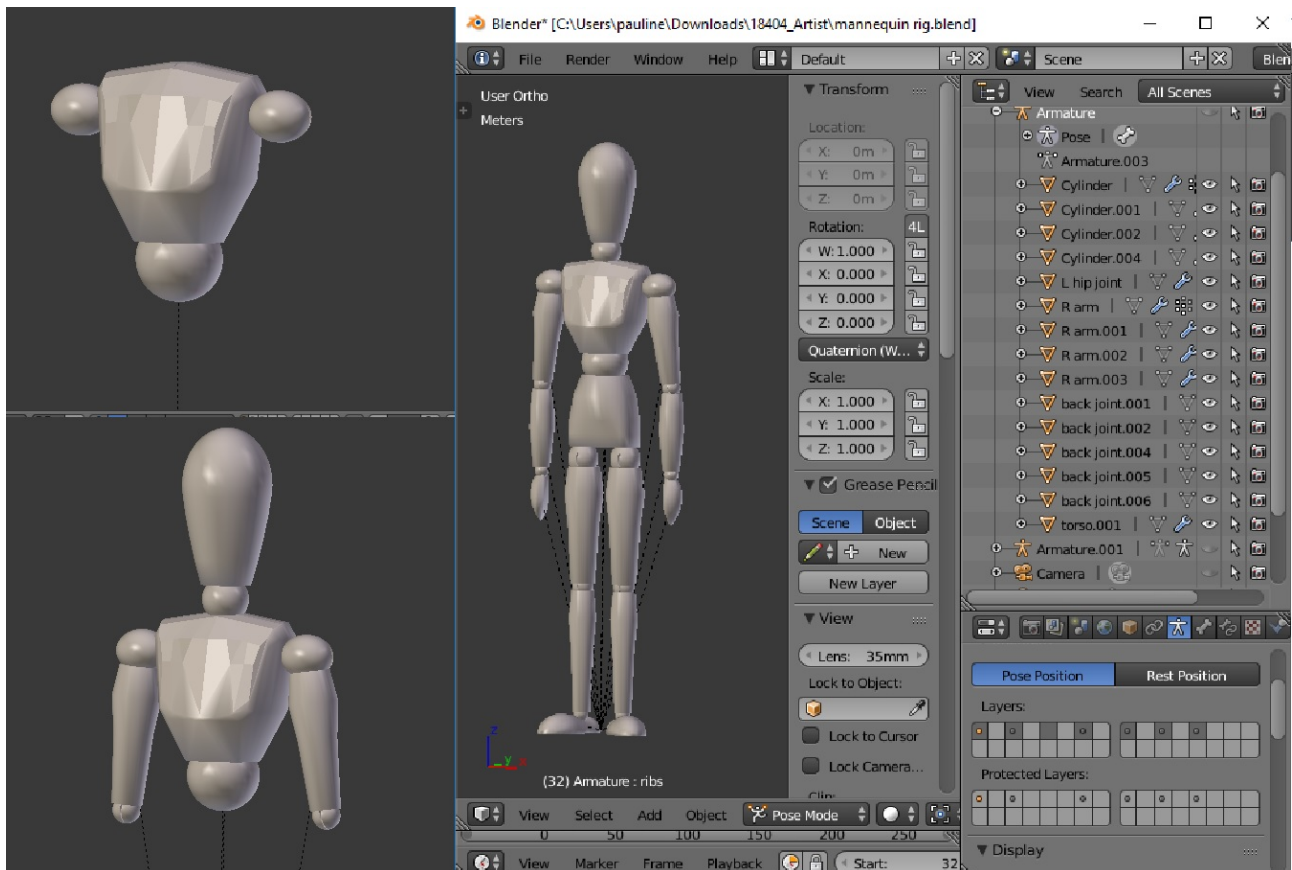


FIGURE 6.4 – Aperçu des différentes étapes de modélisation d'un personnage

Les multiples étapes de modélisation visibles sur la figure 6.4 ont été réalisées par :

- L'ajout d'un objet de forme simple et pré-existante dans le logiciel (par exemple un cylindre pour l'avant-bras, une sphère pour les articulations, un cube pour le torse...)
- Le maillage plus ou moins fin de cet objet (pour cette étape nous avons respecté les conseils des cours suivis sur Openclassroom et avons opté pour un compromis entre précision et lourdeur des calculs)
- Le déplacement manuel des points du maillage pour donner la forme voulue à l'objet initial,
- L'ajout de texture aux surfaces.

6.3.2 Création d'une armature animée à partir des données exportées

Il est possible d'importer dans Blender les données qui ont été exportées avec Nexus si le fichier est sous le format ".c3d" : le logiciel reconstitue alors les marqueurs sous formes de croix dans l'espace 3D, qui suivent l'animation acquise. La figure 6.5 ci-dessous donne un aperçu de l'espace de travail une fois les données importées.

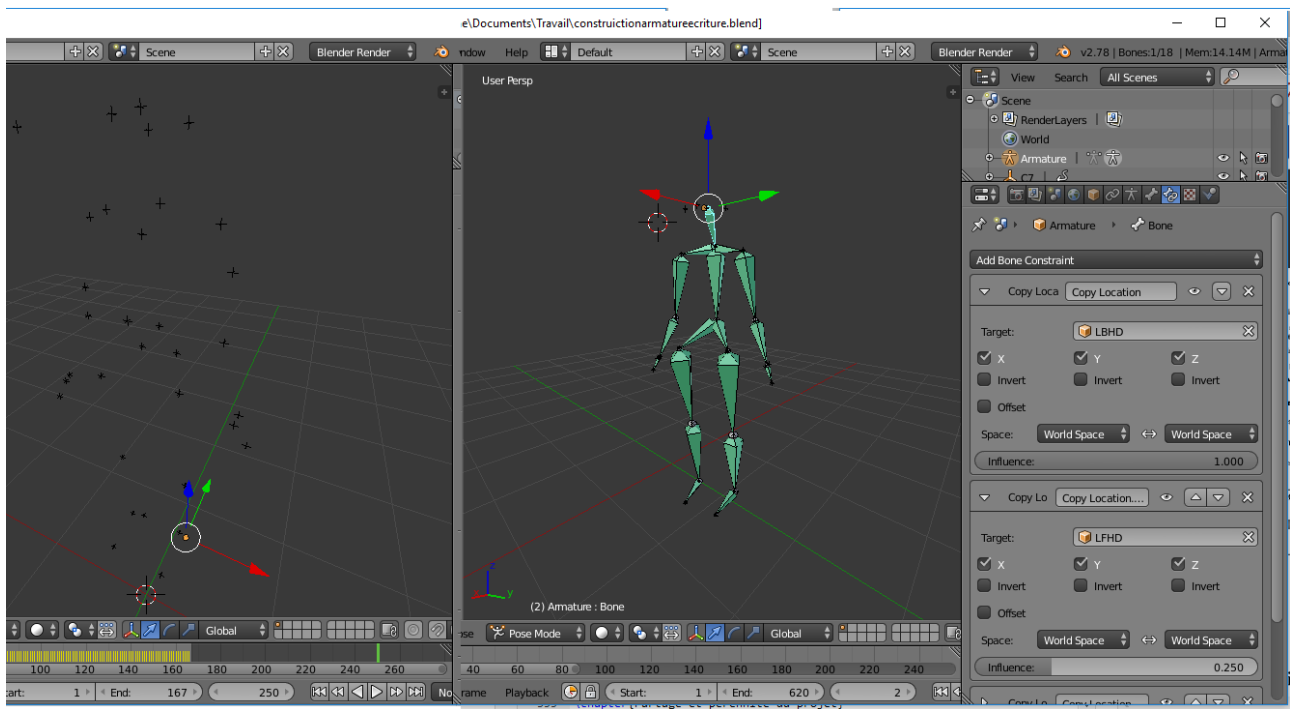


FIGURE 6.5 – Aperçu de différentes étapes de création de l'armature

On parle d'armature puisqu'il s'agit ici d'un squelette qui va imposer une animation (un mouvement) à un personnage déjà créé. La création de l'armature requiert ici un travail manuel dont les différentes étapes sont :

- L'import des données sous le format ".c3d" ;
- L'ajout successifs de "bones" qui correspondront aux os de l'armature ;
- Le positionnement des "bones" sur les marqueurs ;
- L'ajout de contraintes de localisation et de direction des "bones" par rapport aux différents marqueurs, qui permet de faire correspondre le mouvement des marqueurs et le mouvement des bones ;
- La liaison de tous les bones entre eux ;
- L'export de l'armature finie sous format ".bvh" .

La création de cette armature permet parfois de simplifier le nombre de marqueurs : il est possible d'en supprimer certains, ils étaient indispensables pour reconstituer des pertes de données lors de l'acquisition avec Nexus mais ne le sont plus pour la création de l'armature.

6.3.3 Mise en mouvement du personnage modélisé

La dernière étape de l'animation consiste à faire coïncider l'armature animée et le personnage. Une fois de plus, cela demande un travail manuel d'ajustement des dimensions et des proportions de l'armature au personnage, puis un travail manuel d'ajustement des poids d'influence des os de l'armature sur les objets qui constituent le personnage.

Par exemple, l'os "avant-bras" de l'armature devra contrôler totalement l'objet "avant-bras" de notre modèle, il aura donc un poids d'influence fort sur cet objet.

La figure 6.6 illustre le parallèle entre le mouvement réel, le squelette reconstitué en Nexus à partir des données acquises et le personnage animé à partir des données exportées.

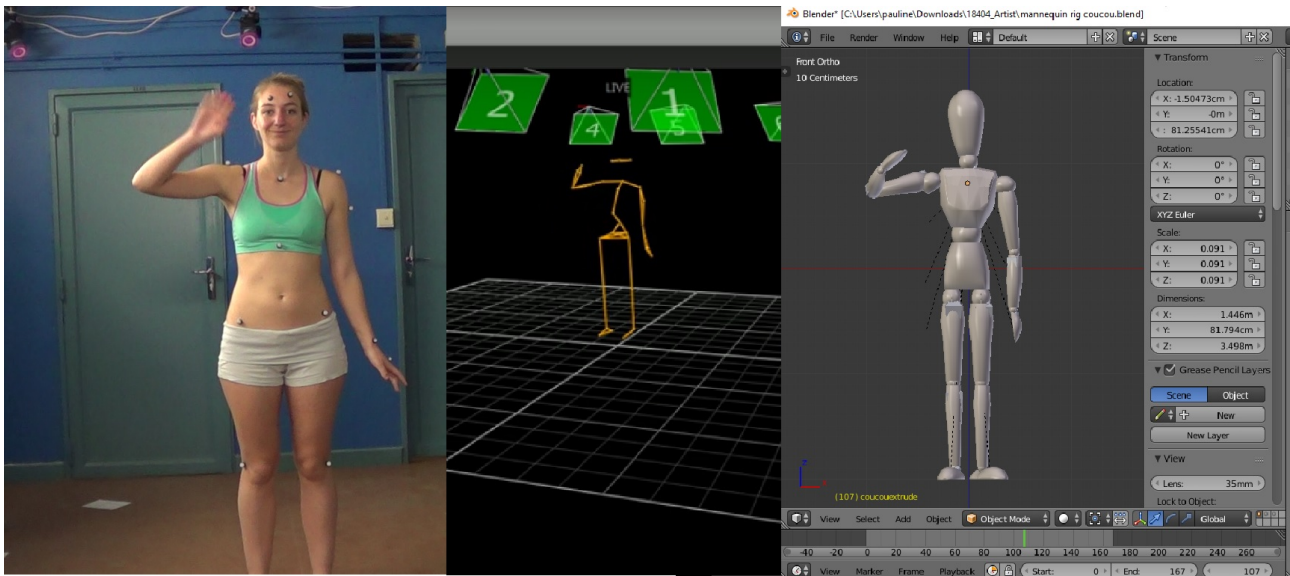


FIGURE 6.6 – Mise en parallèle du mouvement réel, de l'interface Nexus et du personnage animé

Si l'animation a été réussie sur des mouvements simples, l'étape de mise en mouvement du personnage, plus particulièrement l'ajustement de l'influence de l'armature sur l'objet s'est avérée plus compliquée pour des mouvements complexes tels que de la course, des sauts ou des mouvements de torsion du buste.

Finalement, nous avons deux rendus : le mannequin qui effectue un mouvement de "coucher", et un autre rendu avec un saut.

6.4 Perspectives futures

Toujours dans une optique d'automatisation du processus de génération des rendus, il y a plusieurs pistes pour simplifier les étapes manuelles.

La création d'un squelette en Nexus (pour automatiser les acquisitions) et la création de l'armature en Blender semblent être deux étapes similaires et redondantes.

D'une part, il serait intéressant de chercher à générer l'armature dans Blender de manière automatique puisqu'il est possible de programmer des scripts Python dans Blender. En effet, nous avons pu remarquer que lors de l'import des données dans Blender que le logiciel importait également les labels donnés à chaque marqueur lors de la construction du squelette sous Nexus. Il serait donc intéressant de créer un programme qui construirait l'armature automatiquement à partir des noms des différents marqueurs.

D'autre part, une seconde perspective serait de tenter d'exporter le squelette construit en Nexus, sous le même format que l'armature construite en Blender (format ".bvh") pour l'appliquer directement au personnage modélisé et éliminer ainsi l'étape de création de l'armature.

Chapitre 7

L'aspect audio

7.1 Objectifs

Les données acquises par Motion Capture sont en général exclusivement utilisées pour des représentations graphiques comme celles que nous avons abordées précédemment. Nous nous sommes posé la question d'une représentation différente des données qui permettrait de fortement élargir les perspectives du projet.

Une représentation auditive des données présente comme avantage de pouvoir être superposée à la représentation visuelle pour l'appuyer et la compléter ainsi de pouvoir être écouté individuellement. Ce choix a également été une initiative personnelle cohérente avec nos centres d'intérêt.

7.2 Solutions envisagées

Plusieurs possibilités étaient envisageables pour effectuer ce rendu audio. Nous nous sommes donné l'objectif de créer une corrélation entre des paramètres relatifs aux marqueurs et des paramètres relatifs au son.

7.2.1 Paramètres relatifs aux marqueurs

Plusieurs paramètres étaient exploitables, la réflexion s'est surtout portée autour des paramètres et des pistes suivants :

- Les coordonnées spatiales : on peut par exemple modifier l'un des paramètres sonores lorsque le marqueur est à une certaine distance de l'origine ou lorsqu'il rentre dans une zone définie de l'espace.
- La vitesse ou l'accélération : étudier les fluctuations des valeurs atteintes
- La position relative d'un marqueur par rapport à un autre
- Le marqueur en lui-même : à chaque marqueur peut être associé un paramètre sonore en tenant compte de l'un des paramètres cité précédemment de ce marqueur.

Nous nous sommes ensuite posé plusieurs questions. Faut-il moyenner certaines valeurs (par exemple la vitesse de chaque marqueur) ou faut-il faire correspondre chaque vitesse à un paramètre sonore ? Dans le second cas, la représentation serait plus complète et représenterait spécifiquement cette acquisition alors que dans le premier cas plusieurs acquisitions totalement différentes pourraient être représentées par une même piste sonore.

Deux voies correspondantes de représentation ont alors été envisagées :

- Soit on souhaite avoir une représentation sonore prenant l'ensemble des paramètres en compte sans effectuer de moyenne ou de simplification, le résultat ne sera pas forcément musical mais il représentera entièrement et spécifiquement l'acquisition traitée.
- Soit on souhaite obtenir une représentation sonore musicalement cohérente et qui pourra accentuer les paramètres choisis lors d'une représentation graphique mais qui ne pourra pas représenter à elle seule l'acquisition. Il faut alors faire des choix de simplification car il ne sera pas musicalement possible par exemple de prendre en compte la vitesse de 40 marqueurs. Dans cette représentation, l'acquisition fait office de "chef d'orchestre".

7.2.2 Paramètres relatifs au son

Les paramètres sonores que l'on peut faire varier peuvent être très différents selon le type de représentation choisi ; en voici les principaux :

- Le volume
- La fréquence
- Le tempo
- Le nombre de pistes sonores
- La présence ou non de certaines pistes sonores
- Le contenu de la piste sonore (une même piste sonore peut alterner entre différents contenus)

7.2.3 Choix

Le choix de la représentation sonore s'effectue en deux étapes :

- Le choix d'une des deux directions possibles évoquées précédemment
- L'association de paramètres audios et de paramètres relatifs aux marqueurs

Notre réflexion s'est alors portée sur ces différentes possibilités :

Première idée :

On peut associer un son (par exemple une sinusoïde) de fréquence différente et spécifique à chaque marqueur et on peut ensuite régler individuellement le volume de chaque son en fonction d'un paramètre du marqueur (vitesse ou distance par rapport à l'origine).

De cette façon, la personne qui écoute le son pourra directement avoir une idée claire du mouvement qui est à la base de l'acquisition.

Seconde idée :

Notre seconde idée fut de pouvoir générer une musique qui va s'adapter seule à l'acquisition, elle pourra accompagner la vidéo et sera totalement en adéquation avec elle. Pour cela la plus grande difficulté est de garder une cohérence musicale, il faut respecter la notion de tempo, de battement par minutes, de mesure, de tonalité ...

Il est également nécessaire que la musique finale soit construite, qu'il y ait une phase d'introduction, une cohérence le long du morceau avec par exemple un refrain, éventuellement des moments de solo, etc ...

Il faut donc choisir des paramètres intéressants et qui donneront un résultat musicalement intéressant entre les paramètres sonores et ceux de l'acquisition.

Troisième idée :

Enfin notre dernière idée partait d'une base ludique, il ne s'agit plus de représenter ou d'accompagner une acquisition mais de savoir, lors de la réalisation de l'acquisition, la musique que vont jouer nos mouvements .

L'idée était de reprendre exactement le modèle du chef d'orchestre, lors de l'acquisition avec une main on va contrôler le volume sonore (en fonction de sa position sur l'axe z) et avec l'autre main l'intensité musicale (de la même façon en fonction de sa position sur l'axe z).

L'intensité musicale peut être définie de plusieurs façon mais n'a pas vraiment de définition théorique car il s'agit essentiellement d'une perception qui peut être individuelle dans certains cas. On peut par contre jouer sur certains paramètres pour modifier l'intensité en termes de nombre d'instruments, d'ajout de solo, de nombre de notes jouées par mesure ...

On peut bien entendu rajouter des paramètres musicaux associés à des mouvements différents et non uniquement les coordonnées sur l'axe z des deux mains. Il pourrait même être intéressant de faire l'acquisition avec plusieurs personne, chaque personne contrôlant un des instrument à travers son volume et son intensité.

Notre choix

Notre choix s'est porté sur la seconde idée : créer une musique qui pour accompagner la vidéo et qui s'accorde avec elle.

Ce choix s'est effectué en accord avec nos affinités, il permet de donner un grand aspect musical à notre projet et le second choix paraissait être le plus intéressant.

7.3 Mise en place et résultats

7.3.1 Principe général de création de la musique

Pour construire la musique, nous allons tout d'abord constituer une liste de sample musicale (il s'agit d'une piste sonore d'un unique instrument très court et que l'on peut répéter sans coupure) cohérentes entre elles, c'est à dire qu'elles seront de la même tonalité, au même tempo et auront la même couleur musicale, le même style.

Ensuite, l'objectif sera de les assembler, de les lier et de les superposer pour avoir un résultat musical cohérent avec l'acquisition.

7.3.2 Paramètres utilisés

Dans un premier temps, il a fallu définir quel paramètre du mouvement va intervenir sur quel paramètre musical. Notre objectif était de faire en sorte que la musique puisse accentuer le mouvement, que "l'intensité" du mouvement influe donc sur "l'intensité" de la musique. Il faut donc pouvoir donner une définition à ces deux intensités.

Intensité musicale

Pour définir l'intensité musicale nous avons choisi plusieurs paramètres :

- Le nombre d'instrument : plus nous souhaitons une intensité forte, plus le nombre d'instruments sera élevé (arrivée des percussions, etc ...)
- Lors des phases les plus intenses du mouvement, certains des instruments pourront effectuer des solos, avec pour seule contrainte d'avoir un seul solo à la fois pour garder une cohérence musicale.
- Lorsque le début du mouvement est assez calme par rapport au reste, il peut être musicalement intéressant de mettre une partie introductive unique pour améliorer la cohérence et la construction musicale.
- Enfin, nous avons décidé d'attribuer plusieurs pistes musicales à chaque instrument qui vont intervenir à différents moments, en fonction de l'intensité du mouvement. Soit les pistes seront proches et de la même intensité, dans ce cas elles serviront seulement à rendre la musique moins répétitive ; soit elles seront d'une intensité (plus énergique, rythme plus important, etc ...) différente et interviendront donc à des moments différents. Ainsi lorsque chaque sample est importé dans l'algorithme, il faut associer une intensité à chaque sample qui sera comparée relativement à celle des autres (pour définir cette intensité il faut rentrer un nombre entre 0 et 100, avec 0 peu intense et 100 très intense).

Intensité du mouvement

Pour définir l'intensité du mouvement, on peut se baser sur plusieurs paramètres comme la vitesse, l'accélération, la distance entre les marqueurs, la vitesse de rotation ...

Dans une optique de simplification et pour avoir un premier résultat rapidement, nous avons ici décidé de nous baser sur la vitesse moyennée de chaque marqueur pour travailler avec une unique valeur. Cette vitesse a également été lissée (voir partie sur le traitement des données).

7.3.3 Définition du problème d'un point de vue musical

Pour chaque mesure musicale on affecte une valeur d'intensité moyenne et constante (parmi différents seuils d'intensité prédéfinis) qui dépend des valeurs des vitesses des marqueurs : à ces seuils sont associées des caractéristiques musicales précises (nombre d'instruments, solos, ...). Une version plus détaillée du fonctionnement est fournie à l'Annexe C.

Voici la représentation graphique de la vitesse en fonction des mesures avant et après traitement, l'échelle des ordonnées a été ramenée sur 100 :

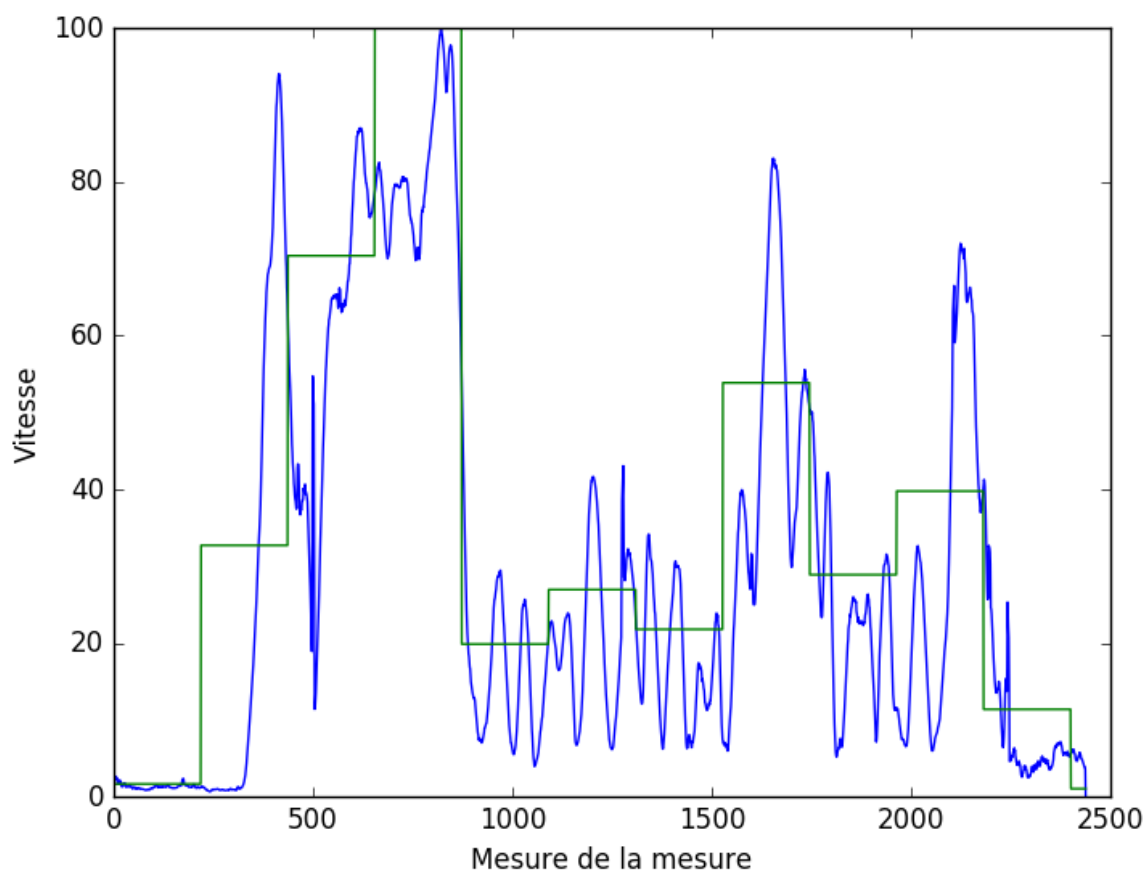


FIGURE 7.1 – Représentation de la vitesse avant (bleu) et après (vert) traitement

7.3.4 Automatisation

Un des enjeux majeurs de la Motion Capture et de notre projet est l'automatisation, cet algorithme de génération de rendu audio est particulièrement automatisé et correspond entièrement à l'objectif que nous nous étions fixé.

La seule étape à effectuer manuellement pour générer le rendu musical est d'importer les samples musicaux et de leur affecter une valeur d'intensité ; une fois cette étape d'initialisation effectuée les rendus sont automatiques.

7.3.5 Résultats

Le résultat est concluant et correspond bien à ce qui est attendu avec les données et samples que nous avons.

Pour aller plus loin dans les tests et confirmer le résultat, il faudrait disposer d'acquisitions plus longues et tester l'algorithme avec un plus grand nombre de sample, d'instruments (nous nous sommes limités à trois plus un soliste) et essayer des tempos différents.

7.4 Perspectives futures

Pour le futur de ce projet, deux voies sont possibles : soit approfondir et améliorer l'algorithme développé, soit essayer une des autres pistes présentées plus haut.

Pour améliorer cet algorithme, il peut être intéressant de retravailler les paramètres de l'acquisition utilisés, en effet, le fait de moyenner la vitesse de chaque marqueur est très restrictif et d'autres paramètres pourraient être pris en compte tel que l'accélération, la position des marqueurs dans l'espace ou l'espacement entre les marqueurs...

Il serait également intéressant de faire varier d'autres paramètres musicaux, tel que le volume, il n'est pas pris en compte dans l'algorithme actuel et serait un paramètre judicieux à exploiter.

Chapitre 8

Partage et pérennité du projet

8.1 Mise en open source de notre travail sur le site web Github

Suite à une suggestion de Romain Vuillemot, l'ensemble de nos codes et nos données d'acquisition sont disponibles sur la plateforme GitHub. Cette plateforme permet de laisser en libre accès l'ensemble de notre travail. D'une part, cela pourra bénéficier aux futures équipes amenées à travailler sur des projets similaires ou bien à travailler sur la plateforme Amigo ou avec la Motion Capture et Nexus.

D'autre part, cela permet les modifications de la part des membres du binôme et la mise en commun pratique des fichiers modifiés ou ajoutés, mais également la suggestion de modification par tout utilisateur.

Vous pouvez retrouver l'ensemble de nos codes à l'adresse : <https://github.com/AmigoCap/CaptureData/>

8.2 Les tutoriels d'utilisation de la plateforme

Si la deuxième partie du projet (travail de modélisation et d'animation sur le logiciel Blender) a été plus rapide et efficace, c'est notamment grâce à l'utilisation de tutoriels rédigés par une équipe de projet d'étude ayant déjà été confrontée à la découverte de l'animation 3D par export de données de Motion Capture.

Au delà des rendus visuels qui constituent les livrables du projet, nous tenons aussi à fournir aux prochaines équipes des tutoriels sur l'utilisation de la plateforme Amigo et le processus d'acquisition, afin qu'ils puissent faire face aux difficultés rencontrées plus rapidement.

Ces tutoriels seront sous la forme de :

- vidéos : captures vidéos de notre écran pendant l'acquisition et le traitement de l'acquisition : reconstitution du squelette, application de "pipelines", traitement des "gaps" d'informations...
- rendus écrits : documents pdf décrivant certaines étapes clés relatives à l'acquisition.

Chapitre 9

Conclusion

Une grande partie de notre projet a été consacrée à l'appropriation de la plateforme Amigo et du processus d'acquisition. Nous avons été confrontés à des difficultés techniques d'acquisitions (pertes de données lorsque des mouvements complexes cachent des marqueurs, erreurs de reconstitution du squelette...). Faire face à ces difficultés a permis de réduire considérablement les temps d'acquisition et de réaliser des tutoriels qui seront un réel outil pour les prochaines équipes travaillant sur la plate-forme Amigo. L'avancée dans l'appropriation de la plate-forme et plus spécifiquement du logiciel Nexus ouvre vers de nombreuses perspectives (possibilité d'acquisitions plus complexes, plus longues...).

La seconde partie du projet a permis de réelles avancées en ce qui concerne la modélisation, l'animation et le travail sur la partie auditive : contrairement à la première partie, notre travail s'est appuyé sur des tutoriels d'autres équipes et est devenu plus rapide et plus efficace. Les rendus générés correspondent à l'objectif premier qui est d'obtenir des rendus plus complexes et esthétiques. De nombreuses perspectives d'approfondissements ont vu le jour avec d'une part les différentes méthodologies de génération de bande sonore et d'autre part les voies d'optimisation de la construction des squelettes entre l'acquisition et l'animation.

Ainsi, avec un peu de recul, le déroulement de notre projet correspond à celui d'un premier projet de recherche sur une nouvelle plate-forme puisqu'il a permis de générer de nombreuses perspectives d'approfondissement ainsi qu'une base solide de compréhension du fonctionnement de la plate-forme. Cette base pourra être réexploitable et complétée par les futures équipes qui travailleront sur la Motion Capture à Centrale. Notre projet s'inscrit finalement dans une phase de découverte d'un outil ; si cette phase peut s'avérer longue et complexe, elle n'aura plus lieu d'être pour les prochaines éditions.

Annexe A

Les types de données

Les données Motion Capture ont été exportées de Nexus sous deux formats : « .csv » et « .c3d ». Ces deux formats permettent d'obtenir des fichiers qui contiennent les positions des marqueurs ainsi que leurs labels (noms).

- Le format « .csv » [6] est le format utilisé pour le traitement des données sous Python. Comma-separated values, connu sous le sigle CSV, est un format informatique ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules. Un fichier CSV est un fichier texte, par opposition aux formats dits « binaires ». Chaque ligne du texte correspond à une ligne du tableau et les virgules correspondent aux séparations entre les colonnes.
- Le format « .c3d » [7] a été utilisé pour importer sous Blender les points animés. Les fichiers « c3d » sont binaires, ils sont notamment utilisés en Biomécanique, en Animation et dans des laboratoires d'analyse du mouvement pour enregistrer des données analogiques 3D synchronisées depuis le milieu des années 1980. Ce format est soutenu par tous les fabricants majeurs de Systèmes de Motion Capture, ainsi que par d'autres entreprises dans la Biomécanique, la Capture de Mouvement et des Industries d'Animation.

Annexe B

Les modules de Python

Voici une liste des trois principaux modules de Python utilisés dans le cadre de notre projet :

- Le module Plotly [8] de Python permet de générer des graphiques interactifs et de qualité, en ligne. Quelques exemples possibles avec ce module : diagrammes en bande, histogrammes, affichages parallèles, axes multiples, diagrammes polaires et diagrammes en bulles, diagrammes en secteurs,...
- Matplotlib [9] est probablement l'un des packages de Python les plus utilisés pour la représentation graphique 2D. Ce package nous a également permis de générer des rendus 3D. Ce module fournit à la fois une façon rapide de visualiser des données utilisant le langage de programmation Python, et de générer des illustrations de grande qualité dans des formats divers.
- Le module Pydub [10] permet d'effectuer différentes opérations avec les pistes audio comme la superposition ou la concaténation. Il est également possible d'effectuer des traitements directement sur la piste audio comme le changement de gain ou le fait de renverser la piste.

Annexe C

Détails du traitement audio

La longueur des samples musicaux utilisés est forcément proportionnelle à la longueur d'une mesure (car un sample doit pouvoir être bouclé et le résultat doit être cohérent musicalement). On peut donc prendre la longueur minimale des samples T_{min} et diviser le problème en un certain nombre de petits problèmes en divisant l'acquisition par T_{min} . Si l'on note la durée de l'acquisition T_{acq} , on aura donc T_{acq}/T_{min} problèmes à régler tout en s'assurant de la cohérence lorsqu'on l'on va les relier. Ce principe rappelle et s'inspire fortement de la méthode des éléments finis que l'on retrouve dans d'autres domaines.

Lorsque la division temporelle est effectuée, on peut moyenner la vitesse sur toute la longueur de cette période, ainsi nous obtiendrons une unique valeur d'"intensité" pour cette période qui représentera l'"intensité" de la période. À partir de cette valeur, que nous pouvons comparer à la valeur des autres périodes, nous pouvons alors décider de la valeur des différents paramètres musicaux variables pour que le segment étudié soit à la bonne intensité musicale, par comparaison avec le reste de l'acquisition.

Bibliographie

- [1] Leray P. “Analyse numérique : problèmes pratiques; Support de cours INSA de Rouen”. In : (). <http://asi.insarouen.fr/enseignement/siteUV/anamum/12deriv-integr.ppt>.
- [2] Francine et MARC DIENER. “<http://math.unice.fr/~diener/StatL2/COURS6.pdf>”. In : (2004). <http://math.unice.fr/~diener/StatL2/COURS6.pdf>.
- [3] Clément VERMOT-DESROCHES. *Générer un dégradé en arc-en-ciel*. <http://blog.vermot.net/2011/11/03/generer-un-degrade-en-arc-en-ciel-enfonction-d-une-valeur-programmatio/>. Nov. 2011.
- [4] Jake VANDERPLAS. *Tutoriel d'animation Matplotlib*. <https://jakevdp.github.io/blog/2012/08/18/matplotlib-animation-tutorial/>.
- [5] Antoine VEYRAT. <https://openclassrooms.com/courses/debutez-dans-la-3d-avec-blender>. Mai 2015.
- [6] *Documentation librairie CSV*. <https://docs.python.org/2/library/csv.html>.
- [7] *Documentation format C3D*. <https://www.c3d.org/>.
- [8] *3D Python Scatter Plots*. <https://plot.ly/python/3d-scatter-plots>.
- [9] *Documentation librairie Matplotlib*. <http://matplotlib.org/>.
- [10] *Documentation librairie Pydub*. <https://github.com/jiaaro/pydub/blob/master/API.markdown>.