

Développement d'une solution de contrôle par capture de mouvements dans un logiciel de cartographie 3D de l'hôpital



Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES

par :

Charles MITTEMPERGER

Conseiller au travail de Bachelor :

Peter DAEHNE, professeur HES

HEG - Genève, 09.12.2013

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de Gestion

Déclaration

Ce travail de diplôme est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre de Bachelor en Informatique de Gestion. L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de diplôme, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de diplôme, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 30.09.2013

Charles Mittempergher

Remerciements

Je tiens à remercier en premier lieu Peter Daehne, qui a accepté de me soutenir dans ce travail. Mes remerciements suivants vont à divers collaborateurs des Hôpitaux universitaires de Genève, avant tout au professeur Christian Lovis, dont l'imagination et l'audace ont permis à un tel sujet de prendre forme. Il a également fait tout le nécessaire pour que mon travail puisse être fait au sein des HUG, et a montré une grande confiance en mes capacités.

Merci à Cyrille Duret, qui travaille sur le projet *POS3D* et qui m'a fortement aidé à la compréhension initiale de ce projet. Son expertise était essentielle afin de diriger mes recherches.

Finalement, merci à tous ceux qui ont contribué de près ou de loin au bon déroulement de ce travail.

Résumé

Les HUG¹ font partie des grandes institutions genevoises reconnues nationalement et internationalement pour leurs prestations médicales et leurs travaux de recherche à la pointe de l'innovation. Leurs recherches dans le milieu informatique ne sont également pas en reste. Le service de l'information médicale (SIMED), dirigé par le professeur Christian Lovis, travaille quotidiennement sur des projets qui font partie des prestations technologiques de demain.

L'un de leur projets, le *POS3D*², a pour but d'étudier les diverses transmissions épidémiologiques qui peuvent survenir à l'intérieur des HUG, un bâtiment dans lequel travaillaient quotidiennement en 2012 plus de 10'000 collaborateurs qui ont effectué plus de 900'000 prises en charge ambulatoires (1)

En parallèle à ce projet, la question d'amélioration de l'hygiène demeure un point permanent dans la stratégie de gouvernement de l'hôpital. L'intégration croissante d'outils informatiques tels que ceux développés par le SIMED ne fait que renforcer ce besoin de travail dans un environnement stérile.

Partant de ces constatations, ainsi qu'en constatant une évolution technologique et technique incessante, notamment par l'apparition d'appareils de contrôle par capture de mouvements, démocratisés dans le milieu ludique des jeux vidéos, envisager un changement progressif dans les moyens d'interactions homme-machine est légitime. Le but de ce travail est une intégration d'une telle technologie dans le *POS3D*, qui n'est pas directement destiné aux milieux exigeants d'hygiène mais qui permettra de faire le point sur la pertinence et les possibilités offertes par la capture de mouvements dans le milieu hospitalier.

¹ *Hôpitaux universitaires de Genève*

² *Projet Organigramme Structurel 3D*

Table des matières

Déclaration.....	1
Remerciements	2
Résumé.....	3
Table des matières	4
Liste des Tableaux	6
Liste des Figures.....	6
Introduction	7
1. La technologie de capture de mouvements	8
1.1 Fonctionnement général	8
1.1.1 Capture optique	8
1.1.2 Capture mécanique.....	9
1.1.3 Capture visuelle	9
1.2 Appareils existants.....	9
2. Pourquoi choisir Leap Motion ?	12
2.1 Fonctionnement hardware	13
2.2 Outils pour développeurs	13
2.3 Domaines d'applications.....	15
2.4 Utilisation médicale	15
Avis scientifiques et populaires sur le produit.....	18
3. Changement de paradigme.....	19
4. Les degrés de liberté.....	20
5. Le POS3D	21
5.1 Histoire du projet.....	21
5.2 Architecture et technologie du logiciel	24
5.3 Pertinence pour une intégration Leap Motion	25
6. Réalisation de l'intégration.....	26
6.1 Choix des manipulations implémentées	31
6.2 Détails techniques de l'implémentation (partie 1)	33
6.2.1 Rotations et zoom.....	34
6.2.2 Changement de bâtiment.....	35
6.2.3 Mode pointeur.....	36
6.2.4 Changement de zone.....	37
6.2.5 Mode temporel.....	37
6.2.6 Translations	38
6.2.7 Résumé des contrôles	39

6.3	Détails techniques de l'implémentation (partie 2)	41
6.3.1	<i>Navigation générale</i>	41
6.3.2	<i>Contrôles du modèle 3D</i>	42
6.3.3	<i>Timeline</i>	43
6.3.4	<i>SERMEDS</i>	46
6.3.5	<i>Détails</i>	48
7.	Evolution du projet	51
8.	Utilisation dans d'autres projets	51
	Conclusion	52
	Bibliographie	53
	Annexes	57
	Code source	57
	Description officielle du POS3D	57

Liste des Tableaux

Table 1: Comparaison des alternatives	12
Table 2: Comparaison des gestes pour le contrôle.....	31
Table 3: Changement de mode	39
Table 4: Contrôles avec deux mains	39
Table 5: Actions en mode 3D	39
Table 6: Actions en mode <i>Timeline</i>	39
Table 7: Actions en mode pointeur	40

Liste des Figures

Figure 1: Vue des composants du Leap	13
Figure 2: Les six degrés de liberté.....	20
Figure 3: Vue du logiciel POS3D	23
Figure 5: Vue conceptuelle d'un objet <i>Frame</i>	26
Figure 4: Système de coordonnées Leap Motion	26
Figure 6: Données d'un objet <i>Frame</i>	27
Figure 7: Données d'un objet <i>Hand</i>	27
Figure 8: Données d'un objet <i>Finger</i>	28
Figure 9: Données d'un objet <i>Gesture</i>	28
Figure 10: Vue du logiciel <i>Cyber Science 3D</i>	30
Figure 11: Navigation entre les bâtiments des HUG	32
Figure 12: Illustration d'un <i>keyTap</i>	35
Figure 13: Diagramme d'état de l'application	36
Figure 14: Gestion du zoom	44
Figure 15: Etapes de zoom	44
Figure 16: Défilement des services médicaux	47
Figure 17: Détails d'une contamination.....	50

Introduction

La présence de l'informatique se renforce au quotidien et les Hôpitaux Universitaires de Genève n'échappent pas à l'augmentation de la présence d'ordinateurs et de tablettes. Les possibilités offertes contribuent continuellement à l'amélioration des prestations et des traitements au sein des HUG.

En parallèle à cet accroissement du besoin informatique, la capture de mouvements, à savoir l'enregistrement des mouvements dans le but d'une utilisation informatique, se démocratise et se répand, à l'aide notamment des jeux vidéo et de la téléphonie mobile. On peut penser à la Wii ou la Xbox, mais également à plusieurs téléphones portables qui intègrent désormais une telle technologie.

Est-il possible de combiner l'informatique médicale et la motion capture ? Le but de ce travail est une analyse de possibilités offertes par un de ces appareils, le Leap Motion. Proposée à un prix abordable et dotée de caractéristiques techniques fortes, cette technologie a été choisie par le professeur Lovis comme seconde approche de la capture de mouvements, après la Kinect de Microsoft, utilisée dans un projet antérieur. La première partie de ce travail est un état des lieux du concept de *motion capture* et ses diverses technologies.

Après avoir également abordé la pertinence de la capture de mouvements dans un milieu hospitalier, deux étapes d'intégration avec un logiciel existant, le *POS3D*, seront effectuées. La première consiste en une « prise en main » du contrôleur, afin de voir si les promesses annoncées sont utilisables dans ce projet. Il s'agit donc d'une intégration basique pour déterminer la pertinence d'un tel produit.

La deuxième étape est l'intégration forte avec le *POS3D*, en se liant avec les manipulations et les spécifications actuellement offertes. Divers contrôles seront mis en place afin de retrouver les mêmes actions qu'avec des périphériques standards.

Finalement, un constat sur l'intérêt d'une telle technologie ainsi que ses limitations sera tiré. La plus-value éventuelle du Leap dans ce travail sera déterminée, et la question sur les possibilités d'utilisation d'un tel contrôleur dans d'autres projets – médicaux ou non – sera ouverte.

1. La technologie de capture de mouvements

La capture de mouvements (« motion capture ») est un procédé qui permet d'enregistrer les déplacements d'objets, généralement des parties du corps, pour ensuite les traiter informatiquement. Son avantage principal est le respect de la fidélité du mouvement. Grâce à cela, le résultat obtenu représente des mouvements naturels et non rigides, comme ce fut le cas pendant longtemps. Ce procédé, initialement réservé au cinéma, a depuis été popularisé par les jeux vidéo, qui font régulièrement usage de cette technologie pour augmenter le réalisme des déplacements des personnages.

1.1 Fonctionnement général

Il existe plusieurs méthodes de fonctionnement qui permettent à un appareil d'effectuer de la capture de mouvements. Grâce à cette panoplie d'alternatives, chaque besoin peut être assouvi. Les techniques les plus populaires sont les suivantes (2):

1.1.1 Capture optique

La capture optique peut se subdiviser en deux sous-catégories

- Caméras infrarouges et marqueurs passifs

Le sujet est équipé de marqueurs réfléchissants, situés à des endroits spécifiques sur le corps ou la partie du corps que l'on veut capturer.

Par la suite, plusieurs caméras (au minimum deux, mais une augmentation de ce nombre augmente également la précision) identifient chaque marqueur et une triangulation est faite pour déterminer sa position précise dans l'espace. Un logiciel reconstruit ensuite le corps à l'aide de tous les points localisés.

- Caméra et marqueurs actifs

La capture de mouvements par marqueurs actifs utilise quant à elle des marqueurs alimentés qui produisent à tour de rôle une lumière brève. Couplés à un logiciel, ce dernier peut, grâce à plusieurs caméras, déterminer la position du marqueur et sait précisément à quel emplacement du corps chaque marqueur correspond. Ce procédé ressemble au marquage passif, mais nécessite un équipement plus évolué, car chaque marqueur doit être alimenté. Son avantage est la portée de capture, qui est limitée lors de l'utilisation de marqueurs passifs par la réflexion naturelle de ces derniers. Les marqueurs actifs permettent donc un champ de capture qui est limité uniquement par la puissance des LEDs des marqueurs.

1.1.2 Capture mécanique

Basée sur l'utilisation d'un appareil fixé le long du corps, cette méthode utilise des capteurs gyroscopiques ainsi que des potentiomètres aux articulations pour reconstituer la position du sujet. Il est ainsi possible de saisir des mouvements sans utiliser de caméras, ce qui peut être utile dans certaines conditions – un espace restreint par exemple.

1.1.3 Capture visuelle

Cette technologie consiste en un appareil qui contient une ou plusieurs caméras. Ces dernières détectent les différents membres du corps et l'appareil génère une représentation 3D de l'utilisateur. L'exemple le plus répandu est la Kinect de Microsoft, mais il existe d'autres produits, notamment le produit de la société Leap Motion, qui sera utilisé dans ce travail. Cette méthode se différencie des précédentes par l'absence d'accessoires nécessaires pour la capture. Cet avantage en fait également un inconvénient, car la précision des mouvements est réduite.

1.2 Appareils existants

Grâce au cinéma, le contrôle d'une machine par des gestes est devenu un rêve pour toute personne technophile. Un des plus célèbres contrôles gestuels est représenté dans le film futuriste *Minority Report*, qui est depuis 2002 un des principaux exemples du potentiel d'une telle technologie. Cette technologie a depuis été réalisée par la société Oblong Industries, (3) mais reste de par son domaine d'utilisation, sa complexité de prise en main et sa mise en place réservée à une élite. Ce produit, baptisé g-speak™ est donc un produit destiné entre autres à « des agences gouvernementales » (4).

Afin de considérer les alternatives du même type que le Leap, nous élaborons une liste qui ne prend en compte que les appareils de la même catégorie, c'est-à-dire ceux qui remplissent les critères suivants :

- Portables
- Possédant des API³s et des possibilités de développement personnalisé
- Fonctionnant sans accessoires supplémentaires

³ *Application Programming Interface* –Interface de programmation

Hormis l'appareil Leap Motion, trois produits émergent :

- Microsoft Kinect (CHF 120.00)

Disponible sur Xbox 360 depuis novembre 2010, cet appareil détecte les gestes et offre également une reconnaissance vocale. Conscient des possibilités offertes par leur produit au-delà du monde des jeux vidéo, Microsoft propose depuis février 2012 un SDK⁴ pour Windows. La portée du capteur s'étend de 1.2 à 3.5 mètres. La technologie utilisée consiste en un projecteur infrarouge et d'une caméra pour calculer la distance des objets, et d'une caméra VGA classique. Un moteur permet à l'appareil de pivoter pour suivre les utilisateurs et ainsi augmenter l'espace utilisable. (5) (6)



- Asus Xtion (CHF 137.00)

Ce produit d'Asus utilise la technologie PrimeSense, tout comme la Kinect. (7) La différence notable est l'absence de motorisation de l'appareil. Il est à noter que PrimeSense a depuis sorti deux appareils utilisant sa technologie, l'un offrant un champ de vision de 0.8 à 3.5 mètres, et l'autre étant destiné à un usage plus proche (de 0.35 à 1.4 mètre). (8)



⁴ *Software Development Kit* – Kit de développement de logiciel

- Airstrike (96cm : \$5869 et 120cm : \$6327)

Visant un public différent (musées, expositions, lieux publics), la société Im3labs a développé « Airstrike », un produit d'apparence simple. Une vitre émet une lumière infrarouge, et une caméra détecte toute intrusion dans la zone située au dessus de l'objet. La précision de l'objet n'est pas évoquée sur le site du fabricant, mais on peut imaginer qu'elle est inférieure aux produits précédents, car la zone captée est uniquement bidimensionnelle. (9)



2. Pourquoi choisir Leap Motion ?



Le choix du produit de la société Leap Motion (appelé désormais « le contrôleur » ou le « Leap ») est principalement motivé par le mandant. Cependant, cette décision peut être soutenue par une multitude de points. Tout d'abord, la simplicité d'utilisation est un facteur capital dans ce projet. Les utilisateurs cibles finaux étant des personnes dont le domaine d'activité n'est pas l'informatique, il est nécessaire de leur fournir une solution peu contraignante. Certaines autres technologies listées au point précédent demandent des accessoires spécifiques pour pouvoir fonctionner, que ce soit des pastilles réfléchissantes, des LEDs à fixer sur les membres, un exosquelette, ou encore un gant à enfiler.

Les trois autres appareils retenus au point précédent ont été éliminés car leur précision n'est pas suffisante pour satisfaire l'objectif du projet, ou leur coût est trop important. La technologie PrimeSense, utilisée par la Kinect et le Xtion nécessite des mouvements amples, adaptés dans un grand espace ou pour une reconnaissance globale de la totalité du corps, mais qui ne sont pas adaptés pour une navigation précise telle qu'on la souhaite dans le logiciel *POS3D*. Pour un usage dans un logiciel différent, il serait tout à fait possible de choisir le produit de Microsoft ou un autre produit offrant des caractéristiques similaires.

Produit	Raisons de l'élimination
Leap Motion	Aucune
Microsoft Kinect	Sensibilité non adaptée au besoin
Asus Xtion	Champ de vision trop grand
Lm3labs Airstrike	Prix trop élevé Taille de l'appareil

Table 1: Comparaison des alternatives

2.1 Fonctionnement hardware

Pour des raisons de protection de technologie en attente de brevet, la société ne fournit pas d'informations officielles précises sur le fonctionnement du contrôleur.

Le contrôleur utilise deux caméras et trois LED infrarouges. Celles-ci génèrent une zone de points, que les caméras vont capter et transformer en données concrètes. Lorsqu'un objet se trouve dans l'espace illuminé, l'appareil

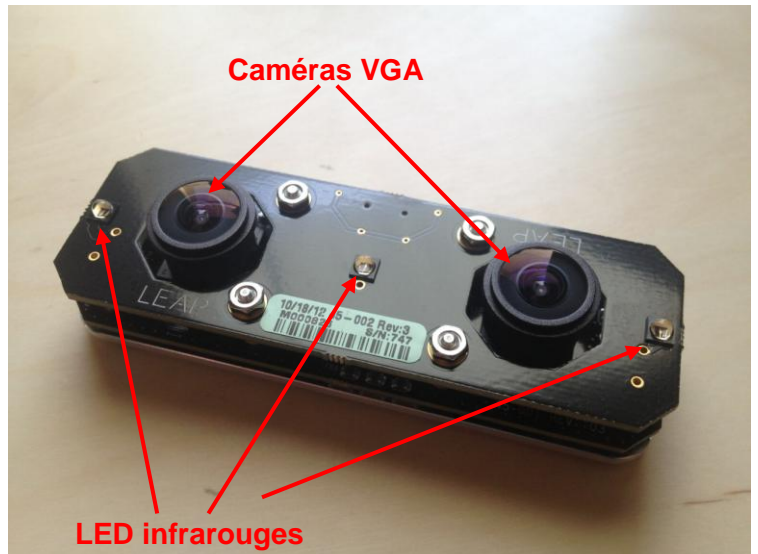


Figure 1: Vue des composants du Leap

détermine le nombre de mains et de doigts présents. La précision annoncée est de l'ordre de 0.01mm, et le volume de l'espace est un cube d'environ 60 cm de côté. Le taux de rafraîchissement varie en fonction de la génération de port USB utilisé, et pour une connexion USB 2.0, ce taux oscille autour de 60 FPS (frame per second), ce qui offre une réactivité élevée (10)

2.2 Outils pour développeurs

Un des arguments de vente du produit est sa versatilité. La société met en avant son contrôleur comme une porte d'entrée à un nombre incalculable d'applications. Ajouté à cela, une liberté de développement permet à ses clients d'obtenir un produit qui correspond complètement à leurs besoins. L'étendue des possibles est résumée sur la page de produit de Leap Motion (11):

« Do almost anything without touching anything »

(« Faites presque tout sans rien toucher »)

« So almost anything's possible. »

(« Donc presque tout est possible »)

En exprimant de tels propos, il est nécessaire pour la société de compter sur une participation de programmeurs et de spécialistes de domaines techniques afin de proposer l'utilisation la plus vaste possible. La technologie fournit les données

nécessaires, et des dizaines d'entreprises de développement répondent présent pour obtenir une part du gâteau d'un domaine encore peu répandu mais très prometteur.

Sur le modèle d'Apple, Android et Microsoft, respectivement propriétaires des *AppStore*, *Google Play* et *Windows Phone Store*, Leap Motion a créé un « store » en ligne, baptisé *Airspace Store*.

Contenant à la fin septembre une centaine d'applications, dont une trentaine gratuite, ce magasin en ligne permet à tout développeur désireux de partager ses œuvres de les mettre à disposition gratuitement ou à un prix fixe. Le prix doit être compris entre \$0.99 et \$999.99, même si pour l'instant le prix le plus élevé est de \$4.99.

Lors d'une mise de l'application sur l'*Airspace Store*, le développeur touche 70% des revenus issus de la vente de ses produits. Il est utile de préciser que cette démarche n'est en aucun cas restrictive du point de vue de la distribution et qu'une application peut être à la fois disponible sur le store et vendue ou partagée selon d'autres moyens au choix du développeur, tels que sur des forums, le site de l'entreprise qui l'a créé ou sur un site personnel. La mise à disposition gratuite ou la vente sur l'*Airspace Store* permet à son développeur de jouir d'une exposition plus grande ainsi que d'un gage de qualité, car les applications doivent être validées par Leap Motion. (12)

Visant des domaines dans l'unique limite de l'imagination de ses utilisateurs, il est judicieux de la part de Leap Motion de proposer une multitude d'outils de développement. Contrairement à des applications mobiles comme celles proposées par les stores cités plus haut, le contrôleur Leap Motion permet de par son fonctionnement une utilisation par des technologies adaptées au but de l'application. Ces technologies sont les suivantes :

- C++
- Java
- Python
- C# and Unity
- JavaScript

Des bibliothèques de ces différents langages sont disponibles pour tous les systèmes d'exploitation, et contiennent divers exemples d'utilisation.

2.3 Domaines d'applications

Etant un produit ouvert et jouissant d'un nombre élevé de langages supportés, Leap Motion est destiné à tout domaine lié à l'informatique. Les différentes catégories proposées sur l'*Airspace Store* donnent une idée non exhaustive de l'étendue des possibilités :

- Business
- Education
- Finance
- Games
- Graphics & Design
- Health & Fitness
- Medical
- Music & Entertainment
- News & Weather
- Plugins
- Productivity & Utilities
- Reference
- Science
- Social & Web
- Sports & Lifestyle
- Travel & Local
- Video & Photography
- Websites

2.4 Utilisation médicale

Citée dans la liste des catégories, l'utilisation du contrôleur pour une utilisation dans le monde médical est un choix logique, qui peut être justifié par plusieurs arguments.

Tout d'abord, et ce point est valide indépendamment du domaine d'utilisation, l'intégration de contrôles par le mouvement des mains fait partie des prochaines évolutions. Les géants du monde informatique s'engouffrent l'un après l'autre dans la brèche. Après Microsoft, déjà bien implanté dans le créneau avec sa gamme Kinect, le fabricant de processeurs Intel a fait l'acquisition d'Omek Interactive, une société israélienne spécialisée dans le contrôle gestuel pour 40 millions de dollars. (13) A la fin du mois de novembre 2013, la société Apple a quant à elle racheté PrimeSense pour un montant estimé à plus de 350 millions de dollars. Cet intérêt généralisé pour de telles technologies démontre que la capture de mouvements est une des futures tendances. (14)

L'apparition de tels contrôles sur certains téléphones portables contribuent également à leur démocratisation. On peut évoquer certains modèles Samsung, comme le Galaxy S4, qui proposent des contrôles par mouvements. Un capteur dédié sur la face avant permet des possibilités d'interactions étendues. Il est possible de répondre à un appel, de naviguer dans les images, et de faire défiler un texte d'un simple mouvement de la main, pour ne citer que les fonctions les plus importantes. (15)

Cette apparition démontre la pertinence du contrôle par gestes également dans le domaine de la téléphonie et sa démocratisation constante.

Le second point concerne l'environnement particulier du monde de la médecine. Ce dernier est entravé par des contraintes d'hygiène qui sont difficiles à appliquer lorsqu'on sait qu'un clavier d'ordinateur peut facilement contenir 400 fois plus de bactéries qu'un siège de toilettes (16). Plusieurs études démontrent également qu'il existe un risque de contamination non négligeable dans les environnements hospitaliers. La présence régulière de bactéries résistantes aux antibiotiques est confirmée dans le milieu hospitalier. Le risque de transferts de celles-ci lors de l'utilisation d'un clavier peut augmenter jusqu'à 92%, dépendant de son type. (17)

Face à un tel problème, des solutions émergent, comme la technologie antibactérienne Silver Nano, développée par Samsung. Conçue pour réduire la propagation et l'existence de germes, cette innovation est implantée par la firme sud-coréenne dans une majorité de ses domaines d'activités, comme les lave-linges, les réfrigérateurs, les purificateurs d'air, et plus récemment les équipements informatiques. Une fine couche d'argent et appliquée sur les touches, et son contact avec l'air ambiant crée un dégagement d'ions qui étouffent les bactéries potentielles. Une suppression de 99.999% des germes est annoncée lorsque cette technologie est utilisée dans un lave-linge. (18)

Cependant, une étude réalisée en 2009 par le réseau d'organisations environnementales Friends of the Earth (19) dénonce fortement l'utilisation d'une telle technologie. Selon cette étude, il existe un manque de régulation concernant l'usage de nanoparticules justifié par les réglementations existantes concernant la forme brute du matériel. Des nouvelles réglementations sont également souhaitées pour des appareils qui, de par leur usage de nanoparticules à but antibactérien, devraient faire partie de la catégorie des pesticides.

Toujours selon cette étude, l'utilisation de telles particules provoque également une augmentation de leur présence dans l'eau, et les conséquences d'une telle croissance sont encore inconnues. Cependant, on peut concevoir que relâcher dans la nature une forte quantité de nanoparticules – utilisées pour lutter contre les bactéries – pose un problème d'équilibre environnemental.

Ces deux aspects, l'apparition croissante d'usages de capture de mouvements et sa précision toujours plus élevée ainsi que la réticence aux produits chimiques, suggèrent fortement que les appareils de capture de mouvements constituent ici un bon

compromis. Utiliser un contrôle sans contact peut également réduire le nombre des procédures actuelles de nettoyages de périphériques peu pratiques à effectuer ou chronophages.

La situation actuelle pour limiter le nombre de contacts entre le médecin et son ordinateur est contraignante. Le chirurgien affiche les diverses radios nécessaires à son opération, puis délègue les manipulations à un assistant.⁵ Cette tâche provoque un ralentissement, car il est difficile pour l'assistant de savoir précisément le niveau de zoom ou la position de l'image souhaité par le chirurgien. A l'aide d'un moyen de contrôle direct, les résultats seraient plus rapidement ceux désirés, et cette interaction pourrait être supprimée. Un gain de temps serait également perçu. L'accroissement de l'utilisation de documents radiologiques informatiques ne va qu'augmenter la nécessité d'un ordinateur, et à terme, une autre solution devra être envisagée.

Plusieurs exemples d'utilisations dans un environnement médical sont déjà disponibles sur le web, principalement dans des applications où le contrôleur remplace une souris pour le déplacement du curseur ainsi que pour effectuer des sélections. (20) (21)

Une compagnie espagnole active dans le milieu médical, TedCas, se penche activement sur le contrôle sans contact dans les environnements hospitaliers où l'hygiène est capitale. Après avoir proposé une interface à l'aide de la Kinect de Microsoft, elle s'est intéressée au produit de Leap Motion. Le résultat est une application permettant un contrôle des différentes images ou radiographies présentes sur l'écran du chirurgien. Testée dans six hôpitaux et deux centres de recherche médicale autour du monde, cette intégration confirme le potentiel d'une telle technologie et la pertinence de sa présence dans les salles d'opérations. Simon Karger, un consultant de Cambridge Consultants, résume la situation actuelle:

« Les chirurgiens actuels ont grandi avec des jeux vidéos et utilisent régulièrement des interfaces avancées. Les écrans tactiles, le contrôle vocal [...] sont ordinaires dans les produits de consommation – et les professionnels de la médecine les veulent aussi dans leur sphère professionnelle. [...] Les interfaces offrant au chirurgien un contrôle en maintenant la stérilité ont toujours été un challenge. Les dernières technologies ont le potentiel pour changer ça, et finalement mettre à portée de main des chirurgiens la technologie du 21^{ème} siècle ». (22)

⁵ Entretien avec Mme Viviane A., instrumentiste en salle d'opération, Clinique Bois-Cerf, Lausanne, le 8 octobre 2013

Avis scientifiques et populaires sur le produit

La levée de fonds effectuée en 2010 par Leap Motion a suscité curiosité et intérêt de la part de la communauté technophile du web, de par son prix, sa taille, sa précision, et les possibilités d'utilisation vantées par la société.

Début janvier 2013, un article paru sur le site engadget.com annonçait un partenariat avec ASUS (23), suivi quelques mois plus tard d'un article de la BBC dans lequel le président de Leap Motion, Inc. Andy Miller dévoilait que Hewlett-Packard (HP) serait également un futur partenaire de la firme. (24) Ces deux partenariats ont été confirmés par des annonces officielles effectuées sur le site de Leap Motion (25) en juillet de la même année, ainsi que sur le site de HP (26).

En mars, la NASA démontre également son intérêt pour les appareils de capture de mouvements, et plus précisément pour le contrôleur Leap Motion, en développant un logiciel de contrôle d'un robot situé à plus de 600 kilomètres. L'usage des gestes pour contrôler précisément un robot ayant six degrés de liberté semblait évident pour l'équipe en charge de ce projet (27)

En juillet 2013, le contrôleur est disponible à grande échelle, et les avis sont divers. Bien que la qualité du produit soit reconnue, le principal défaut relevé par les critiques est l'utilisation de l'appareil. Si la reconnaissance d'une ou plusieurs mains, des doigts ou d'un objet pointu se fait aisément lorsqu'on est immobile, en cas de mouvements il arrive fréquemment que des objets disparaissent (deux doigts sont par exemple confondus en un seul). Il est facile d'imaginer que certaines applications se comportent de ce fait de manière non désirée par l'utilisateur. (28) (29)

L'autre reproche, inhérent au concept de capture de mouvements, concerne l'usage actuel d'une telle technologie. Beaucoup parlent d'un « gadget », qui aurait du « rester en mode beta jusqu'à ce qu'il puisse commencer à amener le changement nécessaire pour montrer son potentiel complet » (traduction personnelle). (30)

En Suisse, les avis commencent également à apparaître, et les reproches formulés par les utilisateurs américains se retrouvent dans la presse helvétique. Une fois de plus, ce n'est pas la technologie qui est décriée, mais son utilisation pour l'instant limitée. (31)

3. Changement de paradigme

L'objectif de ce travail et de manière générale celui de l'utilisation d'appareils de capture de mouvements n'est pas de remplacer une souris ou un clavier. La souris restera le moyen le plus précis pour déplacer un curseur et son utilisation généralisée depuis des décennies en fait le meilleur périphérique pour interagir avec les ordinateurs actuels. Il n'est donc pas pertinent de vouloir utiliser un appareil tel que le Leap pour déplacer un curseur ou sélectionner un bouton, car ceux-ci ont été conçus pour être atteints avec une souris ou un touchpad. Une des caractéristiques du paradigme sur lequel sont basées les interfaces classiques est l'immobilisation des éléments lorsque le contact avec les périphériques cesse. Lorsqu'un curseur se trouve à une position donnée, il y restera tant que la souris ne se déplace pas.

Avec une technologie qui utilise en permanence des informations sur la position ou les gestes non plus d'un appareil mais de l'utilisateur lui-même, cette immobilité ne peut être réalisée. Bien évidemment, celle-ci est effective lorsqu'on sort de l'espace de détection des appareils de capture de mouvements, mais la transition entre les deux espaces – de détection et hors champ – provoque inévitablement une interprétation par les appareils et conduit le programme à des variations « parasites » non-désirées.

Il faut donc changer de point de vue et revoir complètement la conception visuelle des applications utilisant des appareils de capture de mouvement. Ce changement de paradigme est nécessaire car des composants graphiques construits pour être utilisés avec une souris ne peuvent pas l'être de manière optimale lorsqu'une technologie différente entre en jeu. Cela est confirmé lors de l'utilisation de *Touchless*, disponible sur l'*Airspace Store*. Cette application remplace la gestion de la souris par le contrôleur, et son utilisation, bien que réactive, est contraignante et une fermeture de fenêtre qui prend moins d'une seconde à l'aide d'une souris prend au moins trois fois ce temps à l'aide du Leap. Sa précision est paradoxalement ce qui pose problème : comme évoqué plus haut, le moindre déplacement est interprété et il est donc difficile de conserver le curseur sur une zone de la taille d'un bouton conventionnel.

Cependant, on peut imaginer que l'apparition d'ordinateurs hybrides (fonctionnant à l'aide de la souris ainsi que tactilement) et les systèmes d'exploitation conçus pour ces derniers, notamment Windows 8, faciliteraient leur utilisation avec le Leap en raison de la taille des contrôles logiquement agrandis pour une utilisation tactile. Des essais d'intégration impliquant un curseur ont été faits dans le cadre de ce projet (voir partie réalisation 1), mais à l'instar de *Touchless*, l'intérêt était limité et l'on se demande vite

pourquoi on a abandonné notre souris. Il est donc nécessaire de se séparer des habitudes actuelles, car « il n'y a pas de curseur de souris dans Star Trek »⁶.

L'intérêt de la capture de mouvement est l'apparition de nouveaux déplacements et de nouvelles gestuelles qui tirent profit de l'univers 3D – inexistant avec les périphériques standards. En repensant la navigation de manière plus intuitive, on arrive à imaginer non plus des endroits précis où interagir mais des zones spatiales qui ne sont plus clairement définies, mais suffisamment séparées pour qu'une compréhension naturelle prenne place. C'est cette approche qui sera envisagée dans la seconde partie de la réalisation de ce projet.

4. Les degrés de liberté

Lorsqu'on considère un corps dans un espace tridimensionnel, il possède des degrés de libertés (« degrees of freedom », souvent abrégé « DoF »). Ces différents degrés représentent les possibilités de mouvement que l'objet peut effectuer. Ils sont sous-divisés en deux catégories, la translation et la rotation.

Cela signifie qu'un corps peut se déplacer sur trois axes :

- de gauche à droite (jaune)
- en avant et en arrière (rouge)
- de haut en bas (bleu)

Il peut effectuer des rotations sur ces trois mêmes axes :

- le roulis, rotation sur l'axe horizontal z (vert)
- le lacet, rotation sur l'axe vertical y (orange)
- le tangage, rotation sur l'axe de profondeur x (bleu clair)

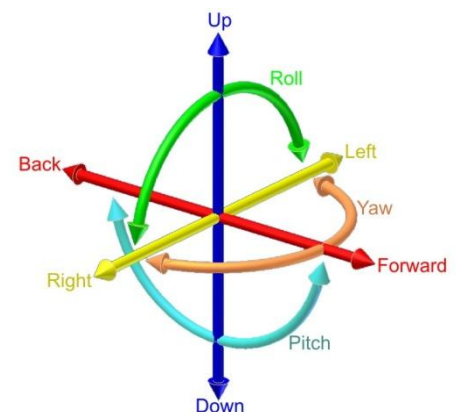


Figure 2: Les six degrés de liberté

Ces six degrés combinés permettent de décrire l'intégralité des mouvements d'un corps en liberté dans l'espace. Un objet complètement libre possède donc six degrés de liberté (6DoF). Lorsqu'un ou plusieurs de ces degrés sont supprimés, la capacité de mouvement est limitée et restreinte à 6-x degrés (x étant le nombre de degrés enlevés). (32) (33) (34)

Dans le cadre de ce travail, ces différents termes seront évoqués pour décrire les différents mouvements possibles qui seront effectués à l'aide du Leap.

⁶ Christian Lovis, 17 octobre 2013

5. Le POS3D

5.1 Histoire du projet

Ce projet utilise des données provenant de différents services de l'hôpital : les données architecturales proviennent du service de cartographie des HUG, les données relatives aux infections sont fournies par le SPCI (Service de prévention et contrôle de l'infection).

- Cartographie
AutoCAD 2D
- Données d'infections
Fichier Excel manuel

En combinant les entrées du fichier d'infections avec les noms des unités répertoriées grâce aux fichiers architecturaux, il est possible de visualiser la localisation d'un patient infecté à un temps donné. C'est cette fonctionnalité qui est au cœur du logiciel. Elle permet de reconstruire les différentes contaminations inter-patients et de les visualiser de manière compréhensible et temporelle sur le modèle 3D. L'intérêt de ce logiciel est de mettre en place des méthodes de lutte contre les infections nosocomiales (survenant dans un milieu hospitalier) en identifiant clairement les foyers d'infection. Ces infections se divisent en plusieurs familles, la plus répandue et donc la plus problématique étant le MRSA (*Methicillin-resistant Staphylococcus aureus*).

Comme son nom l'indique, ce staphylocoque est résistant à la pénicilline ainsi qu'à d'autres antibiotiques usuellement utilisés dans les établissements de soins. Cette souche provoque des infections à fort taux de mortalité (septicémies), et son traitement nécessite un antibiotique plus puissant mais également provoquant plus d'effets secondaires néfastes, la vancomycine. L'accroissement de l'utilisation d'antibiotiques depuis leur découverte amène les staphylocoques à développer régulièrement des résistances, ainsi il existe désormais des souches résistantes à la vancomycine. Bien que ces dernières aient été identifiées en 1997, leur présence reste fortement inférieure au MRSA dans la majorité des pays industrialisés. (35) (36)

Les conséquences d'une infection par MRSA, hormis les risques sanitaires évoqués précédemment, sont un prolongement de la durée du séjour des patients - un coût supplémentaire pour l'hôpital - et également une dégradation de l'image de l'institution, qui doit assumer la responsabilité de l'infection. (37)

Un des problèmes majeurs que les HUG doivent affronter est la croissance des contaminations intra-hospitalières par ce type d'infections, et c'est pour lutter contre cela que le *POS3D* a été développé. Ce taux en 2012 s'élevait à 5%.

Les mesures de réduction des infections nosocomiales sont les suivantes :

- hygiène des mains : l'utilisation régulière d'une solution hydro-alcoolique permet de réduire la propagation de bactéries, les professionnels de la santé étant un vecteur de dissémination majeur. Une étude menée conjointement entre les HUG et l'OMS (Organisation Mondiale de la Santé) relève que l'usage de telles solutions réduit le risque d'infections nosocomiales de moitié. Cette procédure fait partie d'un modèle élaboré par le Dr. Didier Pittet aux HUG en 1995 sous le nom du « Geneva Hand Hygiene Model », qui est depuis appliqué par plus de 15'000 hôpitaux dans plus de 150 pays. (38) (39) (40)
- examens préalables lors de l'arrivée de nouveaux patients : cette mesure est appliquée dans les pays scandinaves, qui possèdent un des plus faibles taux d'infections hospitalières. Cependant, aucune étude de corrobore cette relation de manière certaine.
- Isolement des patients diagnostiqués positivement au MRSA

Ce projet offre également les mêmes prestations pour la grippe, une autre maladie virale fortement contagieuse. Ces deux infections sont d'autant plus transmises lorsqu'on se trouve dans un endroit fermé qui abrite des personnes contaminées. Les hôpitaux sont donc un environnement optimal pour de telles transmissions, et il est donc nécessaire de contrôler au maximum ces dernières. Le *POS3D* est donc une aide importante, qui offre des informations capitales sur les services les plus touchés ainsi que les risques de contaminations entre deux patients (respectivement contaminé et à risque).

On peut envisager une extension de son utilisation à tout type d'infections nosocomiales, en changeant la source de données. La description officielle du projet se trouve dans les annexes.

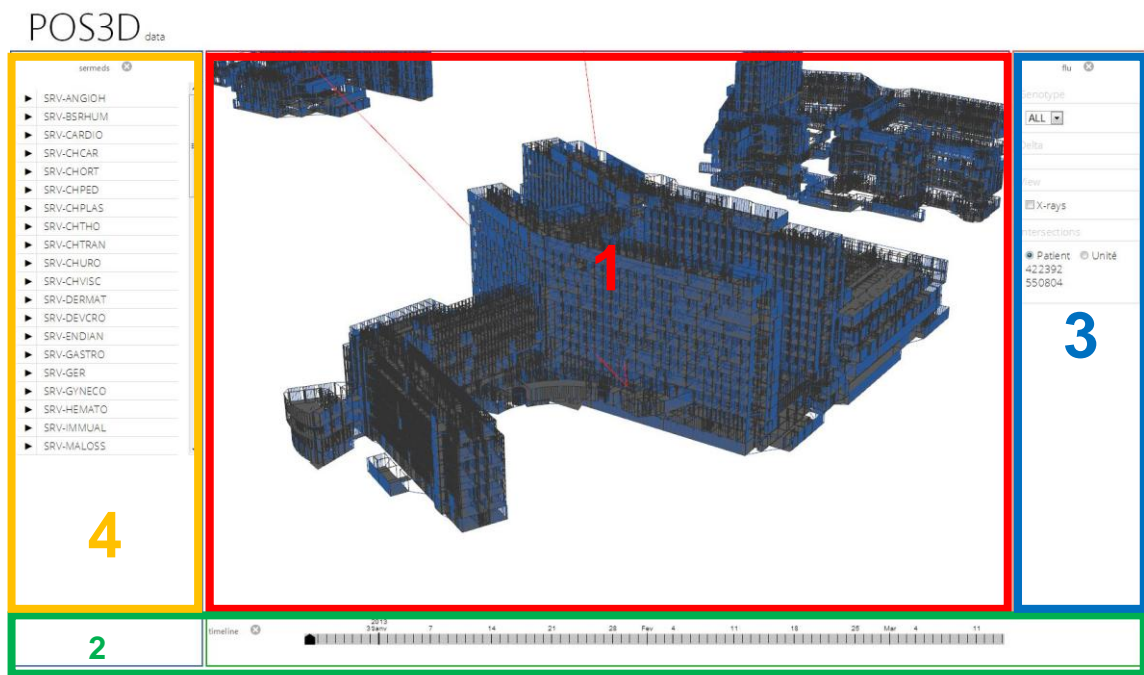


Figure 3: Vue du logiciel POS3D

La fenêtre du logiciel est divisée en plusieurs zones, identifiées sur l'image ci-dessus :

1. Visualisation
2. Timeline
3. Options d'affichage
4. Sermets

La zone 1 est la représentation en trois dimensions de l'hôpital. Il est possible de se déplacer, de tourner autour du modèle et d'effectuer un zoom.

La zone 2 représente le temps. Lorsqu'une option est sélectionnée sur le volet de droite, un déplacement du curseur de cette zone montre les données relatives à un instant précis. On peut par exemple visualiser les patients infectés dans l'hôpital à un moment donné.

La zone 4 contient la liste des services médicaux (*SERMEDS*) de l'hôpital. Il est possible de sélectionner l'un d'eux pour voir son emplacement sur le modèle.

La combinaison des différentes zones est donc capitale pour une utilisation optimale du logiciel. Ce sont les zones 1, 2 et 4 qui intégreront les capacités du contrôleur Leap Motion au terme de ce projet. La zone 3 sera masquée mais ses contrôles seront utilisés pour faciliter le développement des fonctionnalités.

5.2 Architecture et technologie du logiciel

L'application *POS3D* est réalisée dans le langage JavaScript, combiné avec du HTML5. La création et l'affichage des éléments 3D est effectuée à l'aide de l'API JavaScript WebGL. Cet API est supporté par les navigateurs web Google Chrome et Mozilla Firefox. Internet Explorer ne propose qu'une compatibilité expérimentale depuis sa version 11 et il est donc préférable de ne pas l'utiliser. .

Elle est déployée sur un serveur local, ce qui permet de pouvoir l'utiliser sans installation préalable. Les fichiers pertinents pour ce projet, ceux responsables de l'affichage dans un navigateur web, se trouvent à l'emplacement suivant :

```
SIMED\3D\pos3d-data\modules\urn.org.hug.simed.roc.mod.pos3d.webapp-0.1\www
```

Afin d'accéder au logiciel, il faut démarrer deux serveurs :

- Serveur architectural, la structure du logiciel

```
SIMED\3D\pos3d-archi\bin\netkernel.bat
```

- Serveur de données, les données du logiciel

```
SIMED\3D\pos3d-data\bin\netkernel.bat
```

Une fois ces serveurs démarrés, la visualisation se fait dans un navigateur, à l'adresse:

```
localhost:8181/pos3d/viewer/flu (ou /simple pour la première partie)
```

La plus grande partie du projet étant déjà développée et fonctionnelle, nous ne nous concentrerons sur la partie webapp, dans laquelle l'intégration prendra place.

La librairie Rx (Reactive Extensions) est une librairie qui permet de traiter les événements de manière asynchrone, en utilisant des sources diverses. Dans notre cas, cette librairie est utilisée pour gérer les différentes sources telles que la souris, le contrôleur, le clavier, mais on peut envisager d'autres générateurs d'événements comme par exemple l'utilisation d'un web service ou d'un objet tactile. Cette librairie permet aisément de rajouter des sources, puis de traiter les événements associés.

5.3 Pertinence pour une intégration Leap Motion

Le clavier et la souris ont fait leurs preuves dans la navigation plane, mais lors d'une navigation en trois dimensions, à l'instar de *Google Earth* pour prendre un exemple grand public représentatif, il est nécessaire d'intégrer des nouveaux moyens de déplacements. Une souris se déplace sur deux axes, et lorsqu'on se retrouve face à un élément 3D, une abstraction doit être réalisée par l'utilisateur pour associer un mouvement de souris sur une surface plane à un mouvement spatial. Une telle abstraction est contre-intuitive, et il semble donc évident qu'un mouvement spatial pour contrôler la navigation serait plus adapté.

Le même cas de figure se présente dans le cas du logiciel *POS3D*. Nous sommes face à une représentation en trois dimensions, dont la prise en main n'est pas immédiatement évidente. Cela n'est pas dû à la réalisation du *POS3D*, mais aux limitations de mouvements d'une souris et leur correspondances sur un objet tridimensionnel. Ce projet semble donc parfait pour intégrer le Leap. Un autre avantage de ce logiciel est l'étendue des fonctionnalités, qui permettent de rajouter des contrôles gestuels qui ne seront limités que par la complexité du logiciel.

6. Réalisation de l'intégration

L'application existante possède une navigation à l'aide de la souris, et se base sur des « events » lors du mouvement de cette dernière. Lorsqu'on la déplace, un signal est envoyé à la page, qui exploite les informations sur la position de la souris et son déplacement relatif par rapport à sa position précédente.

Afin d'intégrer le contrôleur au projet *POS3D*, il est nécessaire de comprendre avant tout comment son API JavaScript fonctionne. Evoquée plus en amont de ce document, cette API fournit un moyen simplifié de développement d'applications utilisant le contrôleur.

Le référentiel spatial du contrôleur contient trois axes qui sont utilisés pour décrire les différents mouvements. Les directions, les positions, les axes de rotations, ainsi que les déplacements sont fournies par l'API sous forme de vecteurs contenant les valeurs respectives de ces axes.

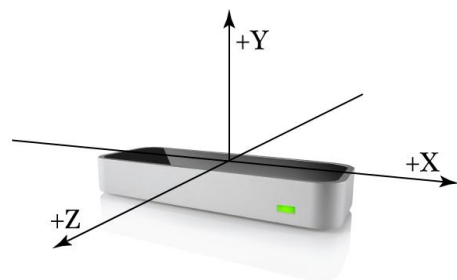


Figure 4: Système de coordonnées Leap Motion

Le boîtier détecte des états (appelés « frames »), et chaque frame contient des objets.

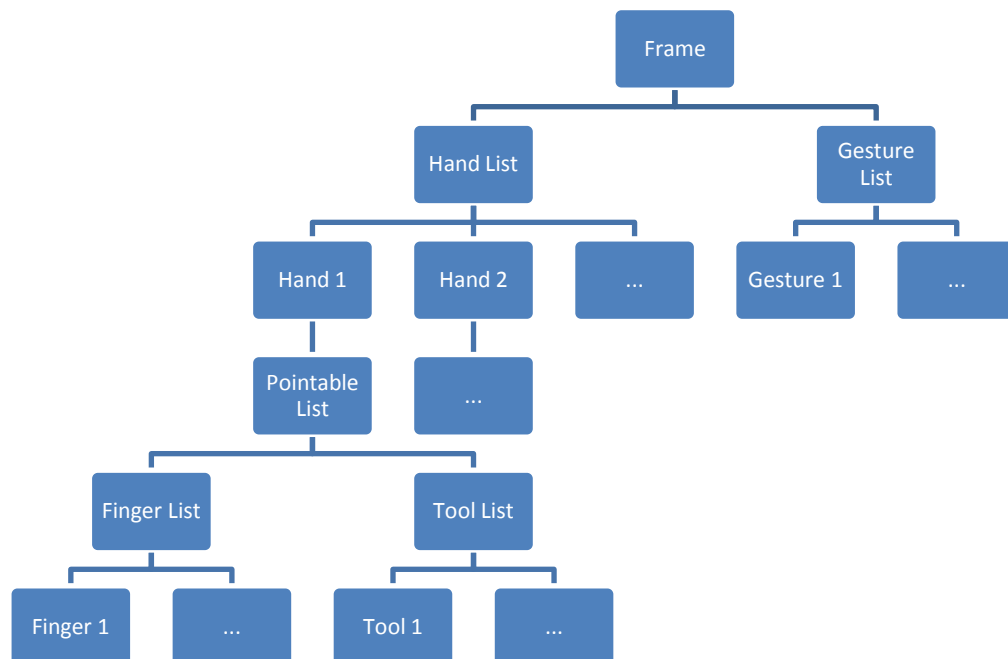


Figure 5: Vue conceptuelle d'un objet *Frame*

Chacun de ces objets possède des attributs spécifiques, qui sont accessibles de manière simple et bien documentée. Ci-dessous se trouvent les attributs principaux pour chaque objet. Ces différents attributs proviennent d'une capture du fichier d'exemple fourni par l'API:

```
Frame data:  
  
Frame ID: 80439  
Timestamp: 1250995448  $\mu$ s  
Hands: 1  
Fingers: 5  
Tools: 0  
Gestures: 1  
Translation: (-16.9, -3.6, 0.2) mm  
Rotation axis: (-0.98, -0.18, 0.02)  
Rotation angle: 0.06  
Scale factor: 1.00
```

Figure 6: Données d'un objet *Frame*

L'objet « Frame » est le conteneur principal. Ses informations principales sont le nombre d'objets présents ainsi que le déplacement global de ces derniers. Dans l'exemple ci-dessus, la frame contient donc une main et cinq doigts, et la main se déplace vers la gauche.

Chaque objet identifié dans une frame contient un identificateur (« ID »), qui restera le même tant que l'objet n'a pas quitté le champ de vision du contrôleur.

```
Hand data:  
  
Hand ID: 68  
Direction: (-0.02, 0.27, -0.96)  
Palm normal: (-0.48, -0.85, -0.22)  
Palm position: (-17.8, 185.7, 19.8) mm  
Palm velocity: (-761.9, -76.3, -89.2) mm/s  
Sphere center: (-33.9, 124.3, -9.7) mm  
Sphere radius: 103.2 mm  
Translation: (-16.9, -3.6, 0.2) mm  
Rotation axis: (-0.0, 0.6, -0.8)  
Rotation angle: 0.06 radians  
Scale factor: 1.00  
Fingers IDs: 11, 66, 37, 96, 54
```

Figure 7: Données d'un objet *Hand*

Les attributs d'un objet « Hand » fournissent beaucoup d'informations sur la position actuelle de la main ainsi que ses mouvements (translation et rotation). Elle contient en outre les identifiants de ses doigts.

Finger and tool data:

Pointable ID: 11
Belongs to hand with ID: 68
Classified as a finger
Length: 75.9 mm
Direction: (-0.19, -0.09, -0.98)
Tip position: (-27.5, 194.3, -84.0) mm
Tip velocity: (-1218.1, -254.1, -214.6) mm/s

Figure 8: Données d'un objet *Finger*

Un doigt ou un outil (« tool », un objet long et fin, un crayon par exemple), est identifié indépendamment des autres, et possède des caractéristiques propres. On notera principalement le déplacement et la position.

Gesture data:

Gesture ID: 64
type: swipe
state: start
hand IDs: 68
pointable IDs: 37
duration: 0 µs
start position: (92.2, 173.7, -67.0) mm
current position: (-101.6, 158.1, -64.3) mm
direction: (-1.00, -0.08, 0.01)
speed: 1954.6 mm/s

Figure 9: Données d'un objet *Gesture*

Un geste prédéfini par le contrôleur est reconnu et identifié. Dans l'exemple ci-dessus, un balayage est effectué par la main, et sa direction est déterminée par les valeurs (-1.00, -0.08, 0.01). Il s'agit donc d'un mouvement vers la gauche, car la première valeur – le déplacement horizontal – est négative (voir figure 4).

Les différents gestes reconnus par le Leap sont malheureusement difficiles à effectuer sans détection parasite. On notera notamment le swipe qui nécessite plusieurs tentatives avant d'être correctement identifié. Le keyTap et le circle sont quant à eux bien, voire trop bien identifiés. Il arrive que des déplacements soient interprétés comme des gestuelles, alors qu'ils n'étaient pas prévus. La sensibilité du contrôleur est à la fois une qualité et un défaut, et le principal défi dans ce projet est d'éviter des détections indésirables.

Pour cette intégration, l'objectif est de trouver une manière simple qui exploite intelligemment le travail déjà effectué jusqu'à présent, ainsi que les informations fournies par les différents objets. Avec la souris, il est possible de déplacer la vue du modèle de trois manières différentes. En utilisant le concept des degrés de liberté évoqués au point 1.3 de ce document, on peut classifier l'état actuel comme une navigation possédant **quatre** degrés de liberté :

- Bouton droit enfoncé et mouvement de souris
 - Déplacement gauche-droite
 - Déplacement avant-arrière
- Bouton gauche enfoncé et mouvement de souris
 - Rotation gauche-droite (axe y)
 - Rotation haut-bas (axe x)

Les deux derniers degrés sont ignorés, car leur implémentation n'apporterait aucun intérêt étant donné l'objet représenté. Une élévation (translation sur l'axe z) générerait une vue qui est quasiment atteignable à l'aide d'une combinaison d'une rotation haut-bas et d'une translation avant-arrière. Implémenter l'élévation à l'aide de la souris nécessiterait l'utilisation ponctuelle du clavier, car toutes les combinaisons bouton/déplacement offertes par la souris sont déjà exploitées.

Une rotation sur l'axe z provoquerait une bascule de l'hôpital, qui est à la fois inutile et dérangeante pour l'utilisateur.

Il existe encore une fonction de zoom, qui n'est pas un réel déplacement de l'objet mais une variation de sa représentation, effectuée à l'aide de la molette de la souris.

En gardant à l'esprit une idée de navigation simple et peu contraignante, la solution envisagée avec une seule main ignore le déplacement latéral. Ce choix est motivé par le fait qu'une gestion de trois déplacements différents à l'aide d'une main est difficilement appréhendable pour une personne non-expérimentée. Les degrés de liberté seront donc réduits à **deux** :

- La rotation gauche-droite (axe y)
- La rotation haut-bas (axe x)

Le zoom sera quant à lui conservé, car il représente un des intérêts majeurs du logiciel. En utilisant deux mains, on peut rajouter les **deux** degrés manquants, ceux des translations.

Ce choix de réduction des degrés de liberté pour un contrôle à une main est également motivé par l'essai d'une application existante, *Cyber Science 3D*, disponible sur l'*Airspace Store*, ainsi que par diverses expérimentations à l'aide de l'application *Google Earth*, qui propose une intégration du contrôleur depuis sa version 7.1.1.1580.

Cyber Science 3D est un outil représentant un crâne humain, et l'utilisateur peut le manipuler en utilisant des gestes d'une seule main. Lorsqu'un seul doigt est visible, un mode de sélection s'active et il est possible de pointer un os du crâne et de le déplacer. Cette application utilise **deux** degrés de liberté et démontre également la précision offerte par l'appareil. L'application est gratuite.



Figure 10: Vue du logiciel *Cyber Science 3D*

Le logiciel Google Earth est plus complexe et son utilisation à l'aide du contrôleur se révèle plus chaotique. Contrairement à *Cyber Science 3D*, il propose les **six** degrés de liberté, ainsi que le zoom avant-arrière. Toute la navigation se fait à l'aide d'une seule main, et le nombre élevé de déplacements différents compliquent lourdement la tâche. On peut résumer l'expérience en citant le site technofyi.com, « it takes a lot of getting used before you can use it without ~~rage-quitting~~ getting frustrated. », qu'on peut traduire par « il faut *beaucoup* de temps d'utilisation pour pouvoir l'utiliser sans ~~s'énerver~~ être frustré ». (41)

Même en tant que « connaisseur » du produit, de sa sensibilité ainsi que de ses défauts, ce sentiment a été présent immédiatement lorsque l'application a été testée dans les locaux des HUG. Difficile à imaginer comment s'en sortirait une personne dont c'est la première utilisation.

6.1 Choix des manipulations implémentées

Deux parties ont été envisagées dans ce travail. La première, qu'on peut considérer comme la prise en main, prend en compte le fonctionnement initial du *POS3D* et intègre le Leap comme substitut à la souris. En se basant sur les applications Leap Motion existantes, cette première étape est une expérimentation des différentes possibilités de manipulations spatiales et leurs implémentations dans le projet. Cette partie se base sur une version simplifiée de l'application, contenant uniquement le modèle 3D ainsi qu'un slider imitant le composant temporel de l'application dans sa version complète.

La seconde partie, rajoutée à la suite d'une discussion avec Christian Lovis et Cyrille Duret (jeudi 17 octobre 2013), met en place un nouveau concept d'utilisation. Le premier constat issu de cette entrevue est l'unanimité de tous ses protagonistes sur la complexité d'utilisation de l'application lors d'une substitution basique de la souris et sur le besoin de se séparer de la navigation suggérée par ce périphérique. Cette deuxième phase est intégrée à la partie du projet responsable du suivi de la grippe.

Chacune des manipulations développées est justifiée par des essais dans un but de simplicité d'utilisation, ainsi que de préférence par les personnes concernées.

Un grand nombre de gestes a été imaginé, puis chacun a été testé afin de déterminer s'il était satisfaisant. Plusieurs approches générales ont été étudiées. Tout d'abord, le choix du nombre de mains nécessaires, une ou deux, ajoutait ou retranchait des possibilités mais augmentait ou diminuait la complexité de prise en main. Une autre approche était le choix de mouvements amples – des déplacements complets de la main – ou uniquement de changements de position de la main. Ces deux alternatives offrent chacune ses avantages et ses inconvénients.

	Avantages	Inconvénients
Déplacements	Meilleur contrôle	Fatigue
Positions	Confort	Complexité

Table 2: Comparaison des gestes pour le contrôle

Une unanimité a chaque fois été trouvée, ce qui confirme que les différents gestes choisis sont intuitifs et que le contrôle final est agréable et fonctionnel. Pour une expérience optimale, c'est une combinaison des déplacements et des positions qui ont été privilégiés.

Il est nécessaire de pouvoir entrer dans l'espace de détection du contrôleur sans activer de déplacement. Pour ce faire, nous nous baserons sur la technique utilisée tant dans *Cyber Science 3D* que dans *Google Earth*, c'est-à-dire désactiver toute répercussion sur le programme lorsque le poing de l'utilisateur est fermé. Une fois ouvert, la navigation est activée et chaque mouvement est interprété par le programme. Il faut cependant faire attention de ne pas avoir la tête ou une autre partie du corps dans l'espace de détection, car cela perturbe le fonctionnement du contrôleur.

Actuellement, l'usage de la translation est utilisé principalement pour se déplacer d'un bâtiment à un autre. Les HUG possèdent un grand nombre de lieux, et la représentation 3D de ces derniers occupe un espace tridimensionnel important. Afin de simplifier l'usage du contrôleur à une main, la visualisation sera centrée sur un seul bâtiment, et la sélection de ce dernier se fera à l'aide d'un geste de la main. Quatre gestes sont reconnus par le contrôleur (balayage (« swipe »), cercle (« circle »), pression sur une touche (« key tap ») et pression sur écran (« screen tap »)).

A l'aide d'un geste prédéfini, le bâtiment suivant sera sélectionné et affiché dans la fenêtre principale. Une fois au bout de la liste, une vue globale sera présentée pour l'utilisateur.

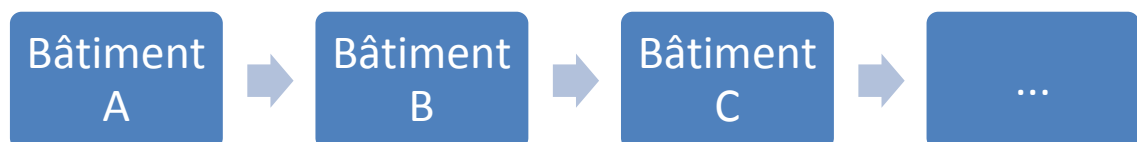


Figure 11: Navigation entre les bâtiments des HUG

Le logiciel *POS3D*, hormis sa navigation spatiale autour des bâtiments, offre d'autres fonctionnalités intéressantes pour une intégration avec le contrôleur Leap Motion. La partie inférieure de l'écran représente une navigation temporelle, qui permet de visualiser des événements en fonction de la date sélectionnée dans cette zone. Un des objectifs supplémentaires de ce projet est le passage entre ces deux modes à l'aide du contrôleur, ainsi que l'utilisation de mouvements pour faire varier la sélection temporelle.

La partie gauche contient une liste des différents services médicaux (SERMEDS) de l'institution. La sélection de l'un d'eux l'affiche sur le modèle 3D. Ce composant permet ainsi de connaître le service dans lequel une contamination s'est produite. La deuxième partie comportera une intégration avec ce composant.

6.2 Détails techniques de l'implémentation (partie 1)

En dépit de la complexité du projet, toute l'intégration de la première partie est réalisée en modifiant deux fichiers. En effet, il ne s'agit pas de développer de nouvelles fonctionnalités, mais d'intégrer des contrôles à celles déjà existantes.

simple.html : ce fichier est la page web qui gère l'affichage du logiciel. C'est la partie statique. La modification consiste à rajouter quelques éléments :

- La source de l'API du Leap (leap.min.js)
- L'image qui représentera le pointeur

pos3d-simple.js : ce fichier s'occupe des interfaces utilisateur. C'est là que la majorité de l'intégration prendra place. Afin de gérer les événements Leap Motion, il faut ajouter un objet MiniRx qui va les générer. En se référant au concept de la librairie Leap, on obtient un objet dont la structure est la suivante :

```
1  var mk_leap = function() {
2      var previousFrame = null;
3      return new MiniRx(function(cont) {
4          Leap.loop({enableGestures: true}, function(frame) {
5              var translation;
6
7              // Frame motion factors
8              if (previousFrame && previousFrame.valid) {
9                  translation = frame.translation(previousFrame);
10             }
11
12             if (situation qui doit déclencher un événement) {
13                 cont(objet) ;
14             }
15
16             // Store frame for motion functions
17             previousFrame = frame;
18         });
19     });
20 }
```

Explications des lignes :

2 et 17 : Le Leap se base sur des différences de *frame* pour détecter des changements. Pour garder une persistance, il est nécessaire de stocker la frame précédente.

4 : Le contrôleur effectue une boucle en permanence pour détecter les objets présents dans son champ de vision. *enableGesture* permet de considérer les gestes prédéfinis ou non, et est donc activé dans ce projet.

8-9 : Le contrôleur utilise deux frames (la précédente et l'actuelle) pour déterminer des changements.

12-14 : Exemple d'évènement : Lorsque le cas souhaité est présent, la méthode *cont(t)* lance un évènement MiniRx. Cet évènement transmet l'objet en paramètre à la méthode déclarée dans le *select* de la factory, instanciée dans le *run* de l'application.

```
var leap = mk_leap().select(dispatch_leapM);
```

L'objet *t* contiendra donc les informations pertinentes au traitement de l'évènement généré. C'est la méthode *dispatch_leapM* qui s'occupera de rediriger les évènements vers les bonnes fonctions.

Les différentes redirections sont développées dans les points des chapitres suivants, avec un extrait simplifié du code nécessaire lorsque cela est pertinent.

6.2.1 Rotations et zoom

Lorsqu'un mouvement valide pour une rotation survient, l'évènement suivant est généré :

```
cont({type:'mvt', t:translation, mode : control_mode});
```

type : caractérise le type d'évènement

t : contient un vecteur de translation de la frame, c'est-à-dire la direction du déplacement de la main

mode : spécifie si on est en mode 3D ou en mode temporel

La fonction *dispatch_leapM* transmet l'évènement à la fonction *translate_leap* (qui se charge de la conversion des données des mouvements du Leap en données de même ordre de grandeur qu'un mouvement de souris) et récupère le résultat. Les fonctions de déplacement du modèle (*mouse_wheel* et *mouse_drag*) sont ensuite effectuées.

Ces deux fonctions sont déjà implémentées et servent au déplacement du modèle à l'aide de la souris. Réutiliser ces deux fonctions est judicieux car les valeurs fournies par le Leap sont comparables à celles fournies par une souris.

- Mouvement gauche-droite → déplacement souris axe x
- Mouvement haut-bas → déplacement souris axe y
- Mouvement avant arrière → déplacement molette

Les méthodes pour la gestion de la souris qui sont réutilisées utilisent les paramètres suivants:

- *mouse_drag*: reçoit deux déplacements (shiftX, shiftY) et un bouton.
 - 1 : clic gauche - rotation
 - 3 : clic droit - translation
- *mouse_wheel*: reçoit un delta qui peut prendre deux valeurs
 - 1 : zoom avant
 - -1 : zoom arrière

6.2.2 Changement de bâtiment

En tirant profit des gestes reconnus par le Leap, on peut associer un de ceux-ci au changement de bâtiments. Les HUG en possèdent quatre, et chaque geste prédéfini passe au suivant. Afin de réaliser ceci, il faut créer un évènement lorsqu'un keyTap survient.



Figure 12: Illustration d'un *keyTap*

Un tableau contenant les informations des différents bâtiments est créé :

- Position centre affichage (x, y)
- Angle gauche-droite
- Angle haut-bas
- Distance

Ces quatre éléments permettent d'afficher n'importe quelle vue du modèle, et on s'en sert pour se centrer sur le bâtiment souhaité.

A chaque évènement, on reconstruit la vue à l'aide de ces valeurs. Cela est réalisé à l'aide la fonction `change_view()`, qui va actualiser les valeurs de la représentation en cours avec celles reçues en paramètre. On obtient donc un changement de vue, qui correspond successivement à chaque bâtiment des HUG, ainsi qu'à une vue d'ensemble du complexe.

6.2.3 Mode pointeur

Lorsqu'un ou deux doigts sont visibles – le Leap possède une sensibilité trop forte qui détecte souvent plus qu'uniquement le doigt tendu –, l'application quitte le mode 3D et passe en mode pointeur. Un curseur apparaît sur la fenêtre et l'utilisateur peut le déplacer en bougeant son doigt. Il est impossible en JavaScript de déplacer le vrai curseur de la souris, pour des raisons de sécurité évidentes.

Le mode curseur utilise l'image rajoutée dans le fichier *simple.html*. Elle est affichée puis déplacée en utilisant ses paramètres de style (`style.left` et `style.top`) et la position du doigt sur les axes X et Y (respectivement *positionX* et *positionY*). Lorsque plus de deux doigts sont visibles, l'application retourne en mode 3D. La portée des mouvements du doigt, c'est-à-dire l'espace à parcourir pour aller d'un côté de l'écran à l'autre, est actuellement réglée pour une utilisation assise, peut se modifier afin d'être facilement adaptée si on décide par la suite d'utiliser le Leap en étant debout (dans ce cas l'espace utilisé devient plus grand).

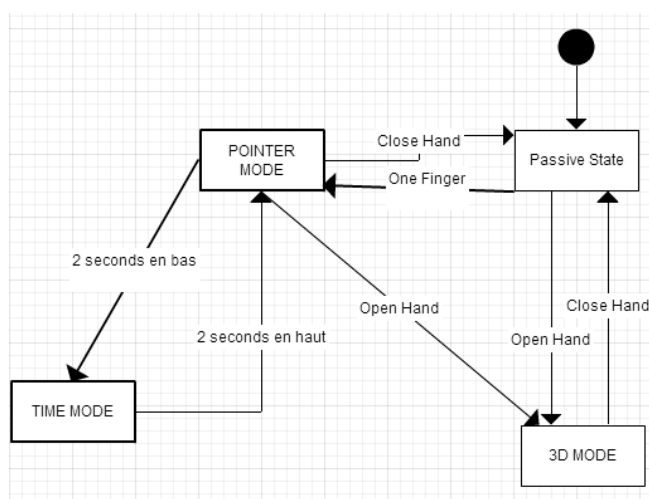


Figure 13: Diagramme d'état de l'application

Les deux évènements évoqués jusqu'à présent sont exclusifs, c'est-à-dire qu'il est conceptuellement impossible d'avoir le mode 3D et le mode pointeur simultanément. Cette exclusivité est garantie par les tests utilisés, qui différencient la présence de plus de trois doigts et de deux doigts ou moins (respectivement pour le déplacement et le pointeur).

6.2.4 Changement de zone

Afin de passer du mode 3D au mode temporel, l'utilisateur doit se mettre en mode pointeur, puis se placer sur la zone qu'il veut activer. En restant immobile durant deux secondes, le mode est sélectionné. De manière analogue, pour passer du mode temporel au mode 3D, la manipulation inverse est à faire, c'est-à-dire activer le mode pointeur et rester immobile deux secondes en ayant le curseur sur la zone de représentation en trois dimensions du complexe hospitalier. L'immobilité du curseur est déterminée à l'aide du paramètre *tipVelocity* du doigt, qui représente un vecteur de déplacement du bout du doigt. Etant donné qu'il est impossible d'être complètement immobile (sans parler de la sensibilité extrême du Leap), un seuil a été fixé après diverses tentatives afin de trouver un compromis entre une trop grande sensibilité et le fait de devoir trop se concentrer pour rester immobile.

Lorsque le pointeur est immobile, un chronomètre est lancé, et après le temps défini, le mode correspondant à la zone pointée est activé (*control_mode*). Si l'utilisateur bouge avant ce délai, le chronomètre est annulé. Un changement de couleur du pointeur confirme le changement de mode.

6.2.5 Mode temporel

Une fois le mode temporel activé, il est possible de déplacer le curseur de gauche à droite à l'aide d'un mouvement de la main ouverte. Il est nécessaire pour le programme de différencier les deux modes (3D vs temporel), car le même mouvement doit effectuer deux actions distinctes. Pour ce faire, un paramètre est inclus dans l'évènement produit par un mouvement de la main ouverte, pour préciser dans quel mode l'utilisateur se trouve. Pour le mode 3D, c'est le **déplacement** de la main qui produisait une rotation du modèle. Pour le mode temporel, ce paramètre est beaucoup trop sensible pour pouvoir être utilisé. On utilise donc la **position** de la main, qui offre une maîtrise supérieure. Il est possible à tout moment de « bloquer » la valeur sélectionnée en fermant le poing ou en repassant en mode pointeur.

La portée des mouvements, tout comme dans le mode pointeur, est déterminée par des valeurs limites qui peuvent être modifiées pour augmenter ou au contraire diminuer le mouvement nécessaire pour parcourir toute la ligne temporelle.

6.2.6 Translations

La translation, présente lors de l'utilisation de la souris, est utile pour naviguer plus précisément jusqu'à la vue que l'on souhaite atteindre. Cependant, comme évoqué au point 6.1, l'intégrer avec un mouvement d'une main n'est pas réalisable. Afin de ne pas perdre cette fonctionnalité, l'utilisation des deux mains est nécessaire pour utiliser ce mode. Le principe choisi est le suivant : lorsque les deux mains sont ouvertes, le mode translation est opérationnel, et un mouvement d'une ou des deux mains déplace le modèle. Il faut imaginer les bâtiments sur une table et le déplacement se fait sur le plan horizontal (axes x et z, voir figure 4). Afin de ne pas devoir sans cesse enlever ou replacer la deuxième main pour effectuer une rotation du modèle, une alternative a été mise en place. Lorsque l'utilisateur ferme une des deux mains, peu importe laquelle, le mode 3D est activé et la main ouverte se comporte exactement comme au point 6.2.1. Cela ajoute un confort, car il est possible de passer rapidement d'un mode à l'autre, le tout sans ôter la compréhension générale des mouvements. En ouvrant à nouveau les deux mains, le modèle repasse en translation. La translation est impossible lorsqu'on se trouve dans le mode temporel. Cette fonctionnalité utilise le jeu de conditions suivant :

```
if(nbHands==2 && nbFingers>3 && control_mode==0){
    var nbF1= frame.hands[0].pointables.length ;
    var nbF2= frame.hands[1].pointables.length;
    if(nbF1>3 && nbF2>3){
        // translation de la vue
    }else if(nbF1<=1 && nbF2>3){
        // rotation de la vue en utilisant la main 1
    } else if(nbF1>3 && nbF2<=1){
        // rotation de la vue en utilisant la main 2
    }
}
```

6.2.7 Résumé des contrôles

Changement de mode

Mains	1				
Doigts	1	2	3	4	5
Mode	Pointeur			3D/Timeline	

Table 3: Changement de mode

Mains	2			
Main 1	Ouvverte		Fermée	
Main 2	Ouvverte	Fermée	Ouvverte	Fermée
Mode	Translation	3D		

Table 4: Contrôles avec deux mains

Actions

Mode	3D			
Mains	1 ouverte 1 ouverte et 1 fermée			
Déplacement	Gauche-droite	Haut-bas	Avant-arrière	keyTap
Action	Rotation	Rotation	Zoom	Changement bâtiment

Table 5: Actions en mode 3D

Mode	Timeline			
Mains	1			
Déplacement	Gauche-droite	Haut-bas	Avant-arrière	keyTap
Action	Déplacement temporel			Changement bâtiment

Table 6: Actions en mode *Timeline*

Mode	Pointeur					
Mains	1					
Déplacement	Gauche-droite	Haut-bas	Avant-arrière	circle	keyTap	Aucun durant 2 secondes
Action	Déplacement curseur	Déplacement curseur		Zoom	Changement bâtiment	Sélection de zone

Table 7: Actions en mode pointeur

6.3 Détails techniques de l'implémentation (partie 2)

6.3.1 Navigation générale

A l'issue de la première partie de la réalisation, on peut constater que la navigation est fonctionnelle mais qu'elle ne permet pas de contrôler le programme avec aisance. L'utilisation d'un curseur offre à l'utilisateur une ressemblance avec les moyens de navigation classiques mais péjore fortement son confort. Il est donc nécessaire de repenser le choix de passage d'un mode à l'autre (du mode tridimensionnel au mode temporel). Un pas doit être franchi et il faut proposer quelque chose de nouveau. Pour ce faire, l'intégration du curseur ainsi que la sélection temporisée est oubliée, au profit d'une navigation plus floue mais également plus simple. A la place, le changement de mode est activé lorsqu'une main ouverte est placée dans une position verticale. A ce moment, le mode en cours s'arrête et les différentes options qui s'offrent à lui s'activent. En déplacement sa main dans la direction du contrôle souhaité, celui-ci est présélectionné et celui-ci est confirmé en fermant la main. Cette méthode est intuitive, car il suffit d' « attraper » le contrôle. En ouvrant à nouveau la main à l'horizontale, l'interaction selon le type de contrôle choisi est possible.

Pour adapter le programme au nouveau concept de navigation, il est nécessaire de rajouter des éléments, tels que les images représentant les différents contrôles. Ceux-ci sont ensuite déplacés sur le même principe que le pointeur de la première partie. Les éléments qui sont directement dépendants de la représentation 3D sont actualisés à chaque changement de celle-ci. La rotation de la boussole est effectuée à l'aide du plugin *jQueryRotate*, tandis que la position de l'échelle de zoom est réalisée grâce à un élément *range* offert par l'HTML5. Cet élément permet d'afficher une échelle de valeurs avec un curseur. En quittant les concepts classiques d'ergonomie et d'interface utilisateur, nous ouvrons la voie à d'autres idées qui sont appliquées dans cette deuxième partie. L'utilisateur ne doit pas faire d'analogie entre ce qu'il voit et ce qu'il connaît déjà. Un nouveau mode de contrôle implique donc une navigation inhabituelle.

Les objets qui sont désormais présents sur l'écran doivent se déplacer en fonction des gestes, mais uniquement pour informer du mouvement et non être perçus comme un contrôle utilisable avec une souris. En s'inspirant de « Star Trek », le choix se porte sur des symboles ne fournissant aucune information textuelle ou visuelle sur le contrôle activé.

Afin d'améliorer l'expérience de l'utilisateur, les différents contrôles et options disponibles ne sont visibles que lorsqu'une ou deux mains sont placées dans la zone de détection du Leap. Cela permet d'une part de décharger l'affichage mais également de proposer différents contrôles selon la direction d'entrée dans l'espace. Lorsque l'utilisateur entre par la gauche, l'option de gestion temporelle est visible, et lorsqu'il entre par la droite, c'est la gestion de l'affichage des différents services médicaux qui est proposée. On peut ainsi imaginer une augmentation du nombre de possibilités de manipulations offertes par le programme tout en conservant une espace simple et non surchargé. Une raison supplémentaire en faveur de ce choix est la réactivité du contrôleur, qui permet un changement d'affichage instantané. Cela offre à l'utilisateur un aperçu de la faible latence d'une telle technologie, qu'il peut utiliser sans devoir ralentir ses mouvements.

Les composants existants sont simplifiés, afin de contenir uniquement les éléments pertinents. Leurs structures sont définies à l'aide de fichiers HAML⁷. Il faut faire attention à ne pas simplement supprimer des éléments, mais à les cacher lorsqu'ils sont nécessaires, comme la *checkbox* du composant *timeline*, utilisée pour activer ou désactiver le mode *trail* (voir partie 6.3.3). L'objectif est de simplifier l'écran et enlever les objets qui perturberaient l'expérience de l'utilisateur. Des éléments du composant SERMEDS sont également enlevés.

6.3.2 Contrôles du modèle 3D

L'implémentation du mode 3D est reprise de la première partie de la réalisation. L'adaptation nécessaire en reprenant cette partie est la suppression du pointeur ainsi que de la suppression de la sélection des zones. Le choix initial des manipulations pour la rotation ainsi que le zoom sur le modèle est gardé pour sa simplicité d'utilisation et son efficacité. La translation à l'aide des deux mains est conservée, car elle offre un confort supplémentaire sans péjorer la facilité d'utilisation des autres manipulations.

⁷ *HTML Abstraction Markup Language*, permet de générer des documents HTML de manière minimaliste (45)

6.3.3 Timeline

Ce mode est partiellement repris de la première partie. Les gestes déterminés lors de la phase de test du contrôleur ont été gardés, mais la différence se situe au niveau du composant. Dans la partie 1, dans des soucis de simplification, le curseur temporel n'était qu'un simple élément HTML, le vrai objectif étant de voir si une gestion de déplacement de valeur sur une échelle graduée était possible et facile à prendre en main. Dans la deuxième partie, il faut intégrer ces gestes à un composant déjà existant du logiciel. Ce composant est construit à l'aide de deux fichiers :

- Component-timeline.js
- Component-timeline-internal.js

Le premier se charge de faire le lien entre les événements extérieurs provenant d'une modification des éléments de l'interface utilisateur et le fonctionnement interne du composant. Celui-ci est développé dans le second fichier. Lorsqu'un événement spécifique est envoyé au composant, celui-ci l'interprète et modifie les données du modèle en conséquence. Les étapes ci-dessous décrivent le processus lors du déplacement du curseur sur l'échelle temporelle (*event_cur_date_changed*).

Cet événement, généré par le déplacement manuel du curseur (à l'aide de la souris), invoque un événement dans la partie interne du composant (*cur_changed*), qui se charge de la représentation des données ainsi que de l'actualisation de la position du curseur sur l'axe.

L'intégration se fait de la manière suivante : on « trompe » le composant en lui faisant croire que le curseur a été déplacé, en lui transmettant une valeur fournie par le Leap.

L'événement utilisé est donc *cur_step_changed*, normalement déclenché par la partie externe du composant qui réagit à chaque changement de l'interface utilisateur. Pour envoyer un événement au composant, il existe en JavaScript (à l'aide de *jQuery*) une méthode *trigger* qui nécessite en paramètre un objet *Event* représentant le nom souhaité, dans notre cas *cur_step_changed*. Afin de pouvoir varier la valeur du curseur, on y ajoute la position de la main sur l'axe X.

```
$('#my-timeline').trigger($.Event('cur_step_changed',{value :t.t[0].toFixed(0)})) ;
```

Le composant va recevoir l'événement, déterminer la valeur à appliquer au curseur et modifier les valeurs du modèle affiché. La valeur du curseur temporel est exprimée en millisecondes depuis le premier janvier 1970 (Unix Time), il est donc nécessaire de recentrer la valeur de la position de la main pour qu'elle se trouve dans l'intervalle

affichée. Pour ce faire, nous utilisons les valeurs `model.min` et `model.max` qui sont respectivement la valeur du début et de la fin de l'axe. Par un calcul de proportions, ainsi qu'en incluant une marge de détection du contrôleur (les valeurs gauche et droite pour lesquelles le curseur est à ses extrêmes), on peut déterminer une nouvelle valeur qui sera transmise au modèle.

Pour la gestion du zoom sur l'axe temporel, en se basant sur l'implémentation du déplacement, nous créons dans la partie interne du composant un nouveau listener, qui va utiliser l'orientation de la main pour faire varier le zoom.

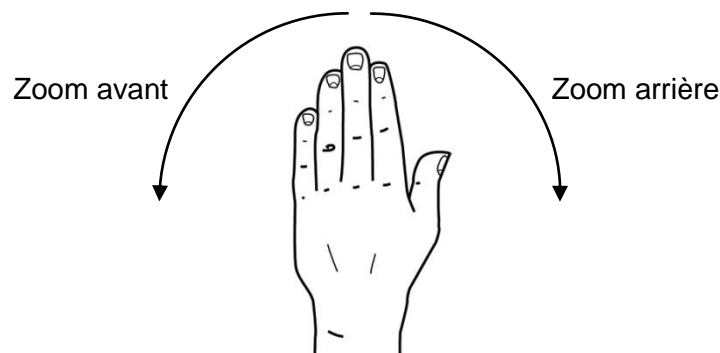


Figure 14: Gestion du zoom

De manière analogue, la valeur transmise par le Leap est calibrée pour fournir un changement progressif en fonction de l'angle de rotation. Ce listener est déclenché lors d'un événement de type `zoom_step_changed`, et va ensuite varier l'affichage du composant. Pour ce faire, la valeur maximale du modèle est incrémentée ou décrémentée selon l'orientation. Compte tenu des grandes valeurs de l'échelle, un facteur est ajouté pour accélérer la modification. Afin de visualiser le changement de zoom, le joystick change de taille lorsque cette situation se produit. En combinant une variation de la taille de l'image (`width`) et son emplacement (`position().top` et `position().left`), l'image est réduite en restant centrée. Les valeurs nécessaires sont calculées de la manière suivante : Lorsqu'on réduit la taille de l'image d'un nombre x de pixels, elle reste ancrée à gauche et en haut. Pour garder un emplacement centré, il faut déplacer l'image de $x/2$ pixels sur chaque axe. Ainsi, le même espace se trouve de chaque côté de l'image.

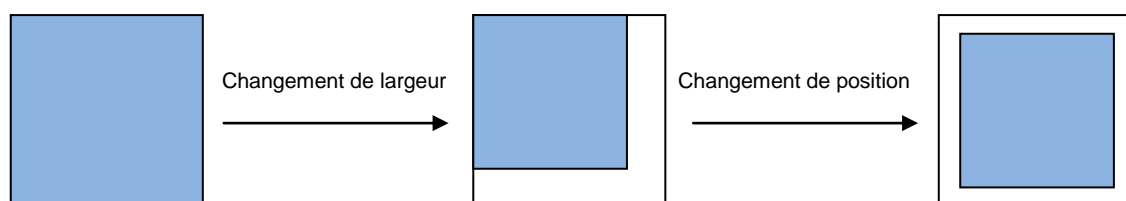


Figure 15: Etapes de zoom

L'agrandissement est effectué de la même manière, en utilisant des valeurs négatives. Lorsque le mode est quitté, la taille et la position sont réinitialisées.

Le passage au mode *trail*, qui permet d'afficher des éléments de plusieurs jours successifs simultanément, se fait initialement en cochant une case située dans le composant. Afin de simplifier l'implémentation de cette fonction avec le Leap, nous allons réutiliser les différents événements déjà existants, et les déclencher lorsque le geste souhaité est effectué. L'élément utilisé est donc la *checkbox*, et l'évènement associé est *'touch_or_click'*. Cependant, un problème persiste, car le composant utilise l'état actuel de la case pour activer ou désactiver le mode *trail*.

Une étape supplémentaire nécessaire est la modification de l'état de la case avant le déclenchement de l'évènement. Pour cela, nous utilisons la librairie *jQuery*, qui permet simplement de changer les valeurs d'un élément.

```
$('#chk-trail').prop('checked', !$('#chk-trail').prop('checked')) ;
```

Cette instruction inverse l'état de la case (la coche si décochée et inversement). On peut donc ensuite invoquer l'évènement sur le même élément :

```
$('#chk-trail').trigger($.Event('touch_or_click')) ;
```

Ainsi, nous réutilisons du code déjà existant et fonctionnel et gardons un niveau de complexité minimum. Le principe DRY (*Don't Repeat Yourself*, « consistant à éviter la redondance de code au travers de l'ensemble d'une application afin de faciliter la maintenance, le test, le débogage et les évolutions de cette dernière. » (42)) est respecté et la maintenance sera facilitée.

Une fois ce mode activé, l'intervalle est modifiée en plaçant les deux mains dans l'espace. Un déplacement de la main gauche change la valeur du *trail*. Il est utile de préciser que ce mode est très gourmand en ressources et qu'une sélection supérieure à une dizaine de jours provoque des ralentissements visuels. Le changement se fait en modifiant la valeur du curseur du *trail* et en invoquant un évènement de type *change* à l'élément *range-trail*. La valeur appliquée doit, de manière similaire au déplacement simple de l'échelle temporelle, être adaptée aux valeurs minimales et maximales de l'élément. Il est également possible de modifier la valeur du trail en ayant une seule main dans l'espace, avec un seul doigt tendu. Une fois de plus, une des difficultés de ce projet est le choix des mouvements pour implémenter une certaine fonction, et il est fort possible qu'une manipulation logique ou agréable pour quelqu'un ne le soit pas pour quelqu'un d'autre.

La différence entre les deux curseurs est le procédé de modification des valeurs. Dans la modification du curseur temporel, c'est le composant qui modifie l'affichage, et dans le cas du *range-trail*, le composant se base sur la valeur existante de l'objet.

Si l'on veut déplacer l'intervalle dans le temps mais conserver sa taille du trail (par exemple sept jours), il suffit d'ôter une main (ou de l'ouvrir à nouveau) et le déplacement se fait de manière similaire à celui du curseur lorsque le mode *trail* est désactivé.

Afin d'atteindre une date qui n'est pas représentée sur l'axe, on peut déplacer l'axe tout en conservant son échelle. La manipulation est une inclinaison de la main vers la gauche ou la droite. Un listener existe, mais doit être légèrement modifié afin de pouvoir utiliser l'angle de la main. Les modifications sont les suivantes (en rouge) :

```
var display_mode_changedE = $('#my-timeline').rx_event('display_mode_changed').select(function(e) {  
    var steps=e.new_steps;  
    if(steps.p) {  
        var FACTOR=100000000;  
        steps.max=(model.max)+(steps.p*FACTOR);  
        steps.min=(model.min)+(steps.p*FACTOR);  
    }  
    slider.attr('min',0);  
    slider.attr('max',steps.max-steps.min);  
    slider.attr('value',0);  
    return model.set_min(steps.min).set_max(steps.max).set_trail(-1)  
        .set_cur(steps.min).set_shift(steps.min).set_zoom(steps.max);  
}):
```

Lors de l'appel de l'évènement de manière « classique », le listener reçoit deux paramètres *min* et *max*, contenu dans le paramètre *new_steps* de l'évènement. Dans le cas du Leap, un seul paramètre est transmis, *p*. Il faut donc définir *min* et *max* avant d'utiliser la suite de la fonction. La dernière ligne met à jour les valeurs du modèle. La variable *FACTOR* a été définie après des tests comme offrant les meilleurs résultats pour une variation rapide mais néanmoins précise de l'axe temporel.

6.3.4 SERMEDS

L'affichage des services médicaux de l'hôpital se fait également à l'aide d'un composant. Tout comme pour la timeline, nous utilisons des événements adressés directement au composant. Le fonctionnement actuel est le suivant : lorsque l'utilisateur clique sur un élément de la liste, un événement est envoyé à la partie interne du composant. Celui-ci obtient la source de l'évènement, c'est-à-dire l'élément

HTML qui a été cliqué. Le composant se charge ensuite de l'ajouter à la liste des services qui doivent être affichés sur le modèle. Comme l'interaction avec le Leap n'utilise pas d'élément précis de la liste, il faut créer un nouveau listener afin de sélectionner l'élément de manière dynamique – passer au suivant ou au précédent. Pour faire ceci, nous transmettons avec l'évènement une valeur qui correspond à l'indice du service à afficher. Cette valeur est conservée à l'extérieur du composant car elle est liée directement à l'affichage. Lorsqu'on veut le service suivant ou précédent, elle est incrémentée ou décrémentée avant l'envoi de l'évènement. Le composant désélectionne l'élément actuel (s'il y en a un), et sélectionne le nouveau.

Une difficulté apparaît à la suite de cette mise en place : la partie de l'affichage dans le navigateur ne connaît pas le nombre d'éléments de la liste. La solution consiste donc, à l'intérieur du composant, à utiliser la fonction modulo pour être certain, peu importe la valeur reçue, d'obtenir un indice inférieur ou égal au nombre d'éléments de la liste. Lorsqu'on passe au service précédent, la valeur du service désiré peut également être négative. La fonction % offerte par le langage JavaScript fait l'affaire pour les nombres nuls et positifs, mais ne gère pas les nombres négatifs correctement.

Il faut donc user d'une astuce pour être certain, quel que soit le nombre reçu, qu'il sera converti en une valeur valide. La transformation finale est donc la suivante, où `num` représente le nombre transmis lors de l'évènement et `nb` le nombre d'éléments de la liste. Les nombres négatifs ne posent donc plus de problème.

```
num = (num%nb)+nb)%nb
```

L'interaction choisie pour faire défiler la liste est la position de la main sur l'axe Y, en considérant trois zones. Pour éviter des défilements involontaires, uniquement deux doigts doivent être identifiés.

Valeur	Contrôle
>170	PREVIOUS
100-170	
<100	NEXT

LEAP

Figure 16: Défilement des services médicaux

Lorsque la main se trouve à une distance correspondant aux valeurs seuils, un évènement est transmis au composant. En raison de la fréquence élevée de détection du Leap, un délai est fixé afin d'éviter un défilement trop rapide de la liste. Ce délai est fixé proportionnellement à la position de la main, c'est-à-dire qu'un positionnement à la limite du seuil générera un déplacement plus lent qu'un positionnement loin de cette valeur limite. En choisissant cette méthode, cela ajoute de la réactivité à l'utilisation de l'application et permet aux utilisateurs avertis d'obtenir des résultats plus rapidement.

En raison de l'architecture des bâtiments des HUG, il peut être difficile de percevoir l'emplacement de certains services. Pour lutter contre ceci, il existe un mode « x-rays », initialement activable à l'aide du panneau d'options situé à droite de l'écran. Ce panneau n'est pas affiché, mais ses contrôles sont malgré tout accessibles via HTML. Lors d'un geste prédéfini et reconnu par le Leap, un évènement est créé et simule simplement une coche de la case. Tout comme pour l'activation du mode trail dans le composant timeline, il faut changer l'état de la case avant de lancer l'évènement, car son statut est contrôlé par la partie interne du composant. Le geste sélectionné est également le `keyTap`, comme dans le mode timeline, ce qui permet à l'utilisateur de ne pas devoir mémoriser une multitude de gestes différents.

6.3.5 Détails

Ce mode permet d'afficher les différentes unités ainsi que les différents patients concernés par la sélection temporelle effectuée. Les deux listes se trouvent originalement dans la partie droite de l'écran, le composant *flu*. C'est notamment dans ce composant que se trouve le mode *x-rays* évoqué plus en amont dans la partie SERMEDS.

Lorsque le mode est activé (en attrapant le triangle situé en bas à gauche), un *keyTap* permet d'alternier entre les unités et les patients. Tout au long de ce projet, le principe DRY a été appliqué pour les fonctionnalités existantes, et une fois de plus il est possible de réutiliser le travail déjà fait. Tout comme les composants *timeline* et *sermeds*, le *flu* dispose d'évènements lors d'un changement d'état des composants visuels. Le choix de la liste à afficher se fait à l'aide de boutons radio, et c'est donc sur ces éléments que l'on va se focaliser. Ces éléments partagent logiquement le même nom. Par défaut, c'est la liste des patients qui est sélectionnée.

Lorsque d'un *keyTap*, la sélection est alternée. Les lignes suivantes sont effectuées :

```
var e=$.Event('touch_or_click');
$("input[name='intersection-mode']:not(:checked)").prop('checked',true).trigger(e);
```

Grâce à la librairie *jQuery*, il est possible de faire des sélections d'éléments correspondants à certains critères. Dans ce cas, l'élément sera un *input* dont le nom est *intersection-mode* et qui ne possède pas d'attribut *checked*. On lui rajoute cet attribut – l'autre bouton radio va donc se désélectionner – et on lance l'évènement qui simule un clic.

Une fois ce changement effectué, la liste est récupérée et affichée. Elle se trouve dans l'élément *intersections-db* et contient différents sous-éléments en fonction du nombre d'informations. Pour itérer à travers ceux-ci, *jQuery* offre une fois de plus des méthodes efficaces :

```
$('#intersections-db').children().each(function(index) {  
    $.notify(index,$(this).text());  
})
```

La méthode *children*, comme son nom laisse supposer, retourne la liste de tous les éléments enfants. On utilise ensuite la méthode *each* pour les parcourir et les afficher.

L'affichage se fait à l'aide d'un plugin développé par M. Cyrille Duret, qui offre un confort de lecture accru.

Cependant, cet affichage illumine sur le modèle tous les éléments correspondants, et il peut être intéressant de se focaliser sur un seul patient ou une seule unité. Pour répondre à cela, l'affichage individuel de chaque objet de la liste a été mis en place.

En déplaçant la main de gauche à droite, les différents patients (ou les différentes unités) sont affichés successivement. Le principe développé pour définir les changements est le suivant : un espace est défini par deux extrêmes comme étant la portée de la main pour aller du premier au dernier élément. Ces valeurs correspondent à la position de la main aux extrêmes gauche et droite.

Le nombre d'éléments -1 (car les indices commencent à 0) est divisé par la portée totale, puis multiplié par la position actuelle. On obtient donc une valeur (une fois arrondie) comprise entre 0 et le dernier élément. Ce nombre est utilisé pour afficher l'élément qui possède l'indice correspondant. La valeur de l'élément affiché est stocké dans la boucle principale du Leap, ainsi il y a une actualisation de l'affichage uniquement lorsque l'élément calculé est différent. Un exemple du processus de calcul se trouve à la page suivante :

A	-250		-125		0		125		250
<div style="display: flex; justify-content: space-between; align-items: center;"> ← G = -250 D = 250 → </div>									
B	0		125		250		375		500
<div style="display: flex; justify-content: space-between; align-items: center;"> ← → </div>									
C	0	1	2	3	4	5	6	7	8

A : Position de la main : x

B : Position positive : x-G

C : Élément à afficher : $nb-1 / (D-G) * (x-G)$

Exemple pour x = 172

$$e = (9-1) / (250-(-250)) * (172-(-250)) = 6.752 \rightarrow 7^{\text{ème}} \text{ élément}$$

En utilisant la méthode *eq(e)* sur l'ensemble des enfants de la liste, on récupère l'objet à la position e et on lance un évènement *touch_or_click*. L'élément s'affiche, ainsi que les détails associés. Ceux-ci sont récupérés de manière similaire dans l'élément *intersection-desc*. On obtient donc une vision complète de la contamination du patient.

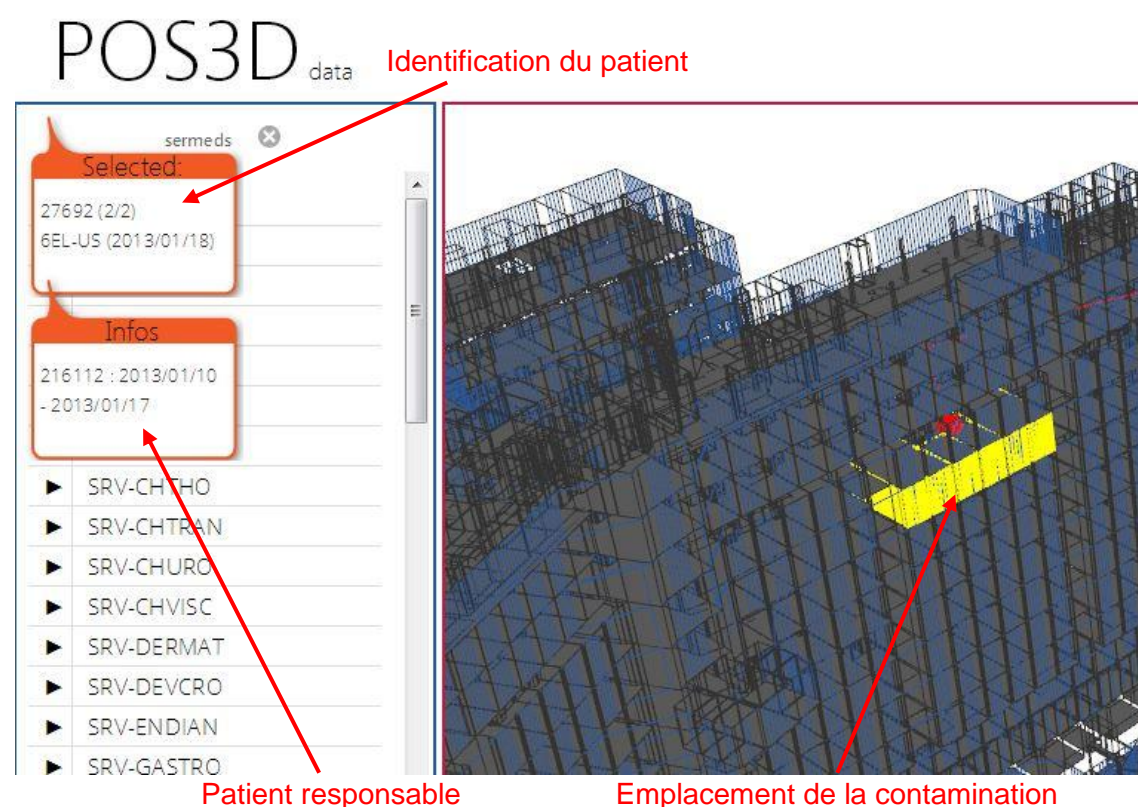


Figure 17: Détails d'une contamination

Dans l'exemple ci-dessus, le patient 27692 a été diagnostiqué positif dans l'unité 6EL-US le 18 janvier 2013. La source potentielle d'infection est le patient 216112, qui était présent dans cette unité du 10 au 17 janvier.

Avec cette fonctionnalité, l'utilisateur a accès de manière simple à tous le contenu présent dans le POS3D. Le tout de manière originale, intuitive et innovante.

7. Evolution du projet

Une fois ce travail terminé, il est légitime de se poser la question de la suite. Nous avons pu constater que l'intégration de la technologie Leap offre une nouvelle manière de contrôler le logiciel. Cependant, une refonte d'une partie majeure de l'application est planifiée, ce qui engendrera des changements inévitables pour conserver l'utilisation du Leap. Le logiciel actuel est complexe et ses performances ne satisfont pas M. Duret, qui compte l'optimiser en mettant en place le principe KISS⁸. Etant construite sur quelque chose d'existant, l'intégration du Leap se repose fortement sur les fonctions et concepts déjà présents, notamment la réalisation de la partie graphique et les évènements associés.

8. Utilisation dans d'autres projets

Au terme de ce travail, on peut constater que les possibilités offertes par le Leap sont uniquement limitées par l'architecture du logiciel et par l'imagination des développeurs.

Plusieurs projets déjà existants ailleurs montrent que la capture de mouvements n'en est qu'à ses débuts mais qu'un fort potentiel existe, également dans le milieu médical. La vraie difficulté est la pertinence de tels produits. Ceux-ci doivent apporter une vraie plus-value et ne pas simplement être un « gadget » dont on se lasserait vite.

En se basant sur la croissance continue d'appareils changeant les principes de navigation classiques, on ne peut qu'affirmer que la capture de mouvements va se populariser et renforcer sa présence sur le marché des nouvelles technologies. L'utilisation du Leap Motion dans d'autres projets au sein des HUG n'a pas été confirmée, car malgré sa précision et l'aide fournie par ses APIs, la justification d'un tel produit par rapport à d'autres périphériques n'est pas suffisante en dehors des endroits où la stérilité du matériel est capitale et où la gestuelle permettrait de s'affranchir de contraintes d'hygiène rébarbatives.

⁸ *Keep It Simple Stupid* : Rechercher la simplicité et éviter toute complexité non nécessaire

Conclusion

Le SIMED est continuellement à la recherche de nouvelles technologies intégrables dans un environnement médical, dans un but d'amélioration continue des services hospitaliers offerts. Ce projet confirme que ces nouveaux appareils peuvent être utilisés avec succès dans des applications existantes. Bien que le *POS3D* ne soit pas directement utilisés par les professionnels de la santé, son existence est légitime car il permet de réduire l'apparition et la transmission de maladies nosocomiales. Le Leap offre une interaction différente et futuriste, mais son utilisation à long terme n'est pas garantie. Les doutes initiaux concernant la pertinence de tels appareils ainsi que leur stade « expérimental » sont confirmés, et ce malgré un fonctionnement irréprochable. Le choix d'inclure cette technologie après le développement et non dès le début du projet peut être une raison de ce doute, car la comparaison avec les contrôles initiaux est inévitable.

Pour une optimisation de la technologie du Leap, il serait plutôt conseillé de développer l'intégralité du projet en ayant la capture de mouvements comme outil de navigation en tête, et ainsi repenser également les interfaces utilisateurs et les fonctionnalités de manière adaptée à une telle technologie.

Compte tenu de son statut de service de recherche, le SIMED peut se permettre de considérer des produits et des développements d'applications sans certitude de leur utilisation à terme par l'hôpital. L'acquisition du Leap Motion fait partie de ces « essais », ces intérêts pour de nouveaux produits potentiellement révolutionnaires ainsi que la recherche de leurs qualités et leurs limitations.

Sur le plan personnel, ce travail m'a permis de me confronter à un environnement de travail professionnel, qui se trouve être fortement différent du milieu plus « scolaire » de la HEG. Il m'a également confronté à un logiciel existant complexe, qui force à travailler de manière différente que lors d'un développement complet d'application, en misant sur la réutilisation maximale de fonctions existantes ainsi que sur les contraintes d'intégration inhérentes au projet. Des technologies jusque-là inconnues ou peu utilisées sont également entrées en jeu, notamment le JavaScript et la librairie *jQuery*.

Bibliographie

1. **Hôpitaux universitaires de Genève.** Les HUG en bref, chiffres-clés et faits marquants. [En ligne] 2013. [Citation : 3 Octobre 2013.] www.hug-ge.ch/sites/interhug/files/documents/les_hug/hug-en-bref.pdf.
2. Motion Capture. *Wikipedia, the free encyclopedia*. [En ligne] 10 Septembre 2013. [Citation : 30 Septembre 2013.] http://en.wikipedia.org/wiki/Motion_capture.
3. John Underkoffler: Pointing to the future of UI. *Video on TED.com*. [En ligne] Juin 2010. [Citation : 30 Septembre 2013.] http://www.ted.com/talks/john_underkoffler_drive_3d_data_with_a_gesture.html.
4. g-speak. *oblong industries*. [En ligne] 2013. [Citation : 30 Septembre 2013.] www.oblong.com/g-speak/.
5. How Microsoft Kinect Works. *HowStuffWorks*. [En ligne] [Citation : 10 Octobre 2013.] <http://electronics.howstuffworks.com/microsoft-kinect.htm>.
6. Kinect. *Wikipedia, the free encyclopedia*. [En ligne] 20 Août 2013. [Citation : 2 Octobre 2013.] <http://fr.wikipedia.org/wiki/Kinect>.
7. PrimeSense. *Wikipedia, the free encyclopedia*. [En ligne] 6 Octobre 2013. [Citation : 20 Octobre 2013.] <http://en.wikipedia.org/wiki/PrimeSense>.
8. Get your sensor. *PrimeSense*. [En ligne] 2013. [Citation : 9 Octobre 2013.] <http://www.primesense.com/get-your-sensor2/>.
9. Airstrike. *LM3LABS - computer vision-based interactive solutions*. [En ligne] 2013. [Citation : 9 Octobre 2013.] <http://2.lm3labs.com/technologies-2/airstrike/>.
10. Leap Motion. *Wikipedia, the free encyclopedia*. [En ligne] 30 Septembre 2013. [Citation : 30 Septembre 2013.] http://en.wikipedia.org/wiki/Leap_Motion.
11. **Leap Motion, Inc.** [En ligne] 2013. [Citation : 30 09 2013.] www.leapmotion.com/product.
12. —. Leap Motion Developer. [En ligne] 2013. [Citation : 30 Septembre 2013.] <http://developer.leapmotion.com/apps/airspace>.
13. Intel Buys Omek Interactive for \$40M: Raises the ante on Perceptual Computing. *GeekTime*. [En ligne] 16 Juillet 2013. [Citation : 10 Octobre 2013.] www.geektime.com/2013/07/16/intel-buys-omek-interactive-for-40m-raises-the-ant-on-perceptual-computing.

14. Apple buys motion sensor maker PrimeSense. *BBC News*. [En ligne] 25 Novembre 2013. [Citation : 25 Novembre 2013.] <http://www.bbc.co.uk/news/technology-25083914>.
15. How do I use Air Gestures to control my Samsung Galaxy S@4? *Samsung*. [En ligne] 26 Avril 2013. [Citation : 10 Octobre 2013.] www.samsung.com/us/support/howtoguide/N0000003/10141/120552.
16. Hygiène: Le bureau, pire nid à microbes que les WC. *24heures.ch*. [En ligne] 29 Septembre 2013. [Citation : 10 Octobre 2013.] www.24heures.ch/economie/bureau-pire-nid-microbes-wc/story/10687422.
17. Healthcare Providers' Computer Keyboards and Bacteria Hazards FactSheet. *Texas Department of Insurance*. [En ligne] Juillet 2005. [Citation : 2 Octobre 2013.] <http://www.tdi.texas.gov/pubs/videoresource/fskeyboardhygie.pdf>.
18. Silver Nano. *Wikipedia, the free encyclopedia*. [En ligne] 4 Août 2013. [Citation : 4 Octobre 2013.] http://en.wikipedia.org/wiki/Silver_Nano.
19. **Dr Rye Senjen, Friends of the Earth Australia, Ian Illuminato, Friends of the Earth United States**. Nano & Biocidal Silver: Extreme Germ Killers Present A Growing Threat To Public Health. [En ligne] Juin 2009. [Citation : 5 Octobre 2013.] nano.foe.org.au/sites/default/files/Nanosilver%20Report%202009.pdf.
20. Leap motion: the ideal mode of interaction in the operating room. *Inria*. [En ligne] 12 Juillet 2013. [Citation : 10 Octobre 2013.] <http://www.inria.fr/en/centre/lille/news/leap-motion-the-ideal-mode-of-interaction-in-the-operating-room>.
21. Leap motion hands free tech has potential for surgical uses within medicine. *iMedicalApps, the leading Physician review of medical & healthcare apps*. [En ligne] 30 Juillet 2013. [Citation : 10 Octobre 2013.] <http://www.imedicalapps.com/2013/07/leap-motion-tech-surgical-medicine/>.
22. **TedCas**. TedCas integrates Leap Motion Controller with Medical Imaging Systems. [En ligne] 15 Octobre 2013. [Citation : 30 Octobre 2013.] <http://www.tedcas.com/en/node/1562>.
23. ASUS partners up with Leap Motion, PCs with 3D motion control to debut in 2013. *engadget*. [En ligne] 3 Janvier 2013. [Citation : 1 Octobre 2013.] www.engadget.com/2013/01/03/asus-leap-motion-partnership.
24. Leap Motion seals HP deal to embed gesture control technology. *BBC News*. [En ligne] 16 Avril 2013. <http://www.bbc.co.uk/news/technology-22166424>.

25. **Leap Motion.** Leap Motion - HP Unveils World's First Computer Embedded with New Leap Motion Micro Sensor. *Leap Motion Blog*. [En ligne] 19 Septembre 2013. [Citation : 2 Octobre 2013.] blog.leapmotion.com/post/61666005204/hp-world-first-leap-motion-notebook.
26. **Sheila Watson, HP.** HP News - Hp Delivers Dynamic Holiday Portfolio with Industry-first Innovations. *HP*. [En ligne] 19 Septembre 2013. [Citation : 2 Octobre 2013.] www8.hp.com/us/en/hp-news/press-release.html?id=1488726#.Ukvag4Zi0Xx.
27. NASA JPL controls rover with Leap Motion, shows faith in consumer hardware. *Engadget*. [En ligne] 30 Mars 2013. [Citation : 7 Octobre 2013.] <http://www.engadget.com/2013/03/30/nasa-athlete-rover-leap-motion/>.
28. Leap Motion Controller Review. *Engadget*. [En ligne] 22 Juillet 2013. [Citation : 2 Octobre 2013.] www.engadget.com/2013/07/22/leap-motion-controller-review.
29. Leap Motion Controller Review. *Digital Trends Reviews*. [En ligne] 22 Juillet 2013. [Citation : 3 Octobre 2013.] <http://www.digitaltrends.com/pc-accessory-reviews/leap-motion-controller-review/>.
30. Leap Motion Launches With Limited Appeal, But It Could Be A Ticking Time Bomb Of Innovation. *TechCrunch*. [En ligne] 22 Juillet 2013. [Citation : 3 Octobre 2013.] <http://techcrunch.com/2013/07/22/leap-motion-launches-with-limited-appeal-but-it-could-be-a-ticking-time-bomb-of-innovation/>.
31. Les gestes ne domptent pas encore les ordi. *20 minutes*. [En ligne] 23 Octobre 2013. [Citation : 28 Octobre 2013.] <http://www.20min.ch/ro/multimedia/stories/story/18903177>.
32. Degrees of freedom (mechanics). *Wikipedia, the free encyclopedia*. [En ligne] 6 Octobre 2013. [Citation : 7 Octobre 2013.] [http://en.wikipedia.org/wiki/Degrees_of_freedom_\(mechanics\)](http://en.wikipedia.org/wiki/Degrees_of_freedom_(mechanics)).
33. Degré de liberté (mécanique). *Wikipedia, l'encyclopédie libre*. [En ligne] 24 Septembre 2013. [Citation : 7 Octobre 2013.] [http://fr.wikipedia.org/wiki/Degr%C3%A9_de_libert%C3%A9_\(m%C3%A9canique\)](http://fr.wikipedia.org/wiki/Degr%C3%A9_de_libert%C3%A9_(m%C3%A9canique)).
34. Six degrees of freedom. *Wikipedia, the free encyclopedia*. [En ligne] 28 Août 2013. [Citation : 7 Octobre 2013.] http://en.wikipedia.org/wiki/Six_degrees_of_freedom.
35. Résistance aux antibiotiques. *Wikipédia*. [En ligne] 13 Novembre 2013. [Citation : 18 Novembre 2013.] http://fr.wikipedia.org/wiki/R%C3%A9sistance_aux_antibiotiques.

36. *Staphylococcus aureus* résistant à la méticilline. *Wikipédia*. [En ligne] 18 Août 2013. [Citation : 14 Novembre 2013.] http://fr.wikipedia.org/wiki/Staphylococcus_aureus_r%C3%A9sistant_%C3%A0_la_m%C3%A9ticilline.
37. **Université de Genève**. MRSA. [En ligne] [Citation : 16 Novembre 2013.] http://medweb1.unige.ch/recherche/groupe/b_donnees/sujet_330_1.html.
38. Didier Pittet. *Wikipedia, the free encyclopedia*. [En ligne] 16 Novembre 2013. [Citation : 21 Novembre 2013.] http://en.wikipedia.org/wiki/Didier_Pittet.
39. WHO - Message from Professor Didier Pittet. *World Health Organization*. [En ligne] [Citation : 2013 Novembre 2013.] http://www.who.int/gpsc/pittet_message/en/index.html.
40. Infection nosocomiale. *Wikipédia*. [En ligne] 24 Juillet 2013. [Citation : 12 Novembre 2013.] http://fr.wikipedia.org/wiki/Infection_nosocomiale.
41. Review: Leap Motion. *TechnoFYI*. [En ligne] 8 Octobre 2013. [Citation : 11 Octobre 2013.] <http://technofyi.com/2013/10/08/review-leap-motion-controller/>.
42. **Leap Motion, Inc.** Leap Motion. *Leap Motion*. [En ligne] 2013. [Citation : 30 Septembre 2013.] www.leapmotion.com.
43. Motion capture hardware. *Meta Motion*. [En ligne] 2013. [Citation : 30 Septembre 2013.] www.metamotion.com/hardware/hands-motion-capture.html.

Annexes

Code source

L'intégralité du projet se trouve sur le DVD en annexe.

Description officielle du POS3D

Objectif : Un outil pour visualiser des données en temps réel dans l'hôpital

- *Créer un modèle 3D de l'hôpital*
- *Nommer le modèle selon le découpage institutionnel (POS)*
- *Visualiser des données médicales ou administratives*

Création du modèle 3D

- *Plan des architectes au format Autocad*
- *Stockage du modèle en SVG (2D)*
- *Assemblage dynamique (3D)*

Nommage selon le POS : Un lien entre le plan et le POS

- *Recherche de correspondances entre le nom des pièces du plan et le nom des chambres de soin dans le POS*

Améliorer la qualité du POS : Outil de contrôle du POS

- *Service web pour afficher dans le modèle 3D des éléments du POS*
- *Intégration possible dans l'outil d'édition du POS*

Données logistiques : Affichage de stocks de médicaments

- *Outil de visualisation de l'état des stocks de médicaments dans l'hôpital (format de données générique)*
- *Navigation interactive dans l'espace et le temps*

MRSA : Suivi des patients MRSA dans l'hôpital

- *Aperçu en temps réel du statut de l'infection MRSA*
- *Affichage des trajectoires des patients MRSA*
- *Analyse des intersections entre patients*