RAPPORT DE PROJET DE CREATION GANT HAPTIQUE DESTINÉ A LA REALITE VIRTUELLE

Professeurs encadrants : Patrick Saint-Denis et Martin Marier

Table des matières

Introduction	3
Artistique	4
Seeds	4
Immersion	5
Technique	7
Côté Microcontrôleur	7
Côté Ordinateur	9
Bracelet connecté : Rythme cardiaque	9
Activation des vibreurs : détection des collisions	10
Activation des vibreurs : envoi du MIDI BLE	11
Flux de données	11
Développement	13
Technique	13
Artistique	13
Annexes	14

Introduction

Pour ce projet, j'ai volontairement choisi de parler du fond de mon projet avant la forme. C'est-à-dire que je préfère présenter les aspects sémantiques de celui-ci avant de parler de la technique, notamment parce que mon projet de lutherie numérique s'imbrique au sein d'un projet global en répondant à une contrainte par une solution technologique.

Ce projet s'appelle *Seeds*, et c'est celui que je développe pendant ce D.É.S.S., et que j'ai commencé depuis maintenant deux ans, en binôme avec Ombeline Vautelet. L'une des lignes de conduites du projet, bien que parfois dure à tenir, nous tient très à cœur : Ne pas laisser la technique prendre le dessus. Nous voulons notre projet poétique, attirant, envoutant, et nous voulons qu'il ait du sens, que ce ne soit pas juste une démonstration technique.

Je vais donc commencer par vous présenter le projet assez succinctement, pour se rapprocher de son rapport avec mon projet de lutherie numérique.



Vous trouverez en annexe la page de mon site montrant les vidéos de Seeds

Artistique

Seeds

Seeds est une expérience de réalité virtuelle poétique, imaginée d'après un double postulat contemporain : L'humanité témoigne aujourd'hui d'un détachement de la nature, en se rapprochant de plus en plus des nouvelles technologies ; Nouvelles technologies qui, par leur existence, participent entre autres à l'exploitation non réversible des ressources naturelles.

Le projet a donc pour objectif de répondre à la problématique suivante :

Par quels moyens peut-on sensibiliser véritablement le public à l'écologie au sein d'une installation utilisant les technologies immersives ?

Les questions écologiques étant vastes, nous avons choisir de nous rapprocher de l'agriculture, et des nouvelles méthodes et connaissances autour de celle-ci, que l'on peut retrouver en agroécologie ou encore en permaculture. Nous avons commencé un partenariat avec la ferme de permaculture de Rennes, en France, afin d'avoir un suivi de fond sur ce que l'on raconte dans notre projet.

L'expérience Seeds invite un specta(c)teur au sein d'une ferme de permaculture, à suivre un atome d'azote dans un cycle complet : de l'air vers la terre, puis de la terre vers l'air. Nous avons voulu que

cette expérience soit en réalité virtuelle pour immerger le spectateur au plus près du sujet, car l'immersion est une méthode intéressante, sinon efficace, pour conscientiser un public.

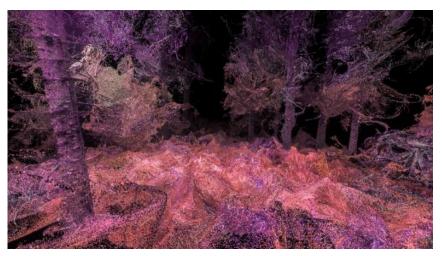
Il n'est pas simple de recréer de façon réaliste un espace naturel dans un environnement en images de synthèses. Et ça l'est encore moins en temps



We Live In an Ocean Of Air, Marshmallow Laser Feast, 2018

réel, et encore moins pour une expérience de VR fluide. Après avoir vu une expérience du

Marshmallow Laser Feast « We Live In An Ocean Of Air » dans lequel l'environnement naturel est représenté sous forme de nuage de points, nous avons revu nos critères de réalisme. Il est clair que nous ne pourrions jamais atteindre un niveau de détails suffisant en tentant le photoréalisme. D'un autre côté, l'idée d'avoir du réel, et de le réinterpréter nous allait.



In The Eyes Of The Animal, Marshmallow Laser Feast, 2016

Nous avons alors eu la chance de trouver des géomètres acceptant de nous réaliser pour nous un scan laser au LiDAR de la ferme de permaculture. Nous avons aujourd'hui un nuage de 124 millions de points que nous sommes en train d'optimiser, de

travailler et d'animer pour l'afficher dans une expérience de réalité virtuelle réalisée dans Unity.

Immersion

Nous avons cherché à explorer les différentes méthodes d'immersion afin « d'augmenter » la VR. Le plus dérangeant est souvent de ne pas voir ses mains. Un Leap Motion sur le casque permet de résoudre ce problème assez facilement. Il n'est plus supporté officiellement, mais il est encore possible de le faire fonctionner avec les dernières versions de unity. Il en résulte un maillage 3d de nos mains, animé en temps réel. Parmi l'ensemble de nos 9 sens (admis scientifiquement), le toucher nous a alors semblé le premier à développer. C'est d'ailleurs un de nos sens les plus important que l'on apprend à développer dès la naissance, bien avant la vue.

La question des retours haptiques en VR est complexe. Une expérience d'immersion dite « totale » supposerait d'être entièrement attaché à un exosquelette qui pourrait synchroniser notre corps virtuel et notre corps réel, pour éviter de passer à travers un mur (virtuel), ou pour attraper et

ressentir un objet réel. Il y a aujourd'hui des prototypes de gros gants avec exosquelette qui permettent de s'en approcher. Ils seront bientôt commercialisés, mais sont très chers (5000 USD).

Etant donné que Seeds sera entièrement modélisée sous forme de particules, nous n'avons pas vraiment besoin de retour haptique moteur.



Gants haptiques de TeslaSuit

Nous avons plutôt imaginé ressentir une petite vibration à chaque fois que l'on touche une particule.

C'est ainsi que l'idée de mon projet de lutherie numérique est née : proposer au spectateur de vivre parmi les particules, de les suivre, d'en faire partie, et même de les toucher.

Toujours dans un but d'immersion, nous voulons que l'avatar virtuel du spectateur se synchronise avec lui, qu'ils aient un lien. Nous avons donc pensé à synchroniser le rythme cardiaque grâce à un capteur, lequel viendra modifier l'apparence des particules de l'avatar.

Les gants n'ont pas besoin d'être trackés dans l'espace via des capteurs de flexion, mais le seront de manière fluide grâce au Leap Motion, ce qui permet d'alléger leur conception.

Technique

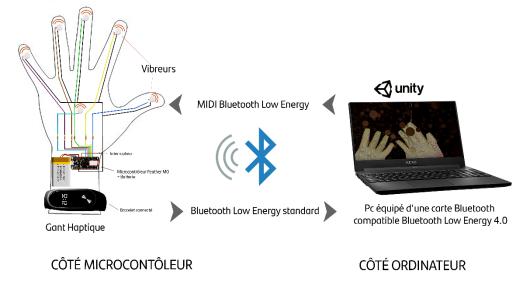


Schéma général

Côté microcontrôleur

Mon projet est composé des éléments suivants :

- Adafruit Feather M0 Bluefruit (Bluetooth Low Energy)
- Batterie
- 6 vibreurs + 6 résistances
- Un interrupteur
- Un bracelet connecté Xiaomi Mi Ban 2 qui mesure le rythme cardiaque

Chacun des vibreurs est relié à une sortie PWM du M0. Une résistance à chacun d'eux est recommandée pour éviter de griller la carte.

La batterie 3.7V est reliée directement à celle-ci et s'éteint grâce à un interrupteur reliant la sortie « EN » à la terre. Elle se recharge via le port USB.

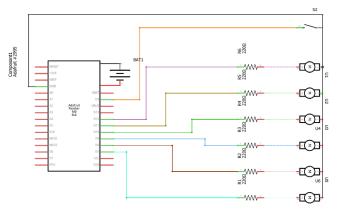
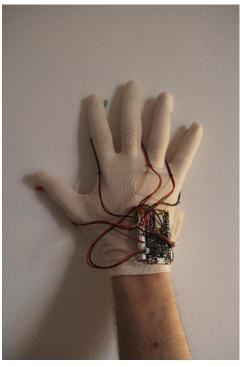


Schéma électrique

Le microcontrôleur est cousu au gant, et scotché à la batterie. Les fils sont cousus au gant et ajustés pour permettre une mouvance optimale. Le gant ci-dessous en test a été équipé de DEL pour la vidéo de projet, en parallèle des vibreurs. Il n'y a sur cette version du gant ni résistances, ni interrupteur.



Face postérieure du gant



Face antérieure du gant

Le but du gant est de générer des vibrations sur la main de l'utilisateur lorsque celui-ci touche une particule dans l'environnement en réalité virtuel. Le gant reçoit le numéro de doigt à faire vibrer, puis génère une impulsion au vibreur correspondant. La programmation du microcontrôleur est faite dans l'environnement de développement Arduino. Pour cela j'utilise un tableau avec les numéros des sorties, et un tableau avec leur valeur. Si le M0 reçoit en MIDI BLE un chiffre entre 0 et 5, il attribue 254 comme valeur à la sortie associée. A chaque fin de boucle, si une valeur est supérieure à 0, on la décrémente de 2 et on l'écrit grâce à « analogWrite ». Pour plus de détails, voire le croquis depuis le lien en annexe.

Côté ordinateur

Le Bluetooth Low Energy (abrégé "BLE") est un protocole particulier, dédié aux objets connectés. Son principal intérêt est de consommer peu d'énergie en ne créant pas de liaison permanente pour dialoguer mais en n'envoyant et en ne recevant un signal qu'en cas de besoin, à l'inverse du Bluetooth « standard » qui permet une émulation de port série. J'ai choisi le BLE afin de minimiser la taille de la batterie pour le confort de l'utilisateur, mais j'ai expérimenté pendant mes tests un temps de latence conséquent. Bien qu'inférieur à une seconde, il est suffisamment important pour gêner l'expérience de retour haptique.

Il est nativement très bien géré par les téléphones intelligents, mais un peu moins par les ordinateurs. Sur MacOS, il est bien géré, mais le MIDI Bluetooth demande à l'utilisateur de se reconnecter manuellement à chaque fois, et n'est pas des plus pratiques. Sur Windows, c'est encore moins simple puisque le BLE est nativement pris en compte par peu d'applications. Le Bluetooth MIDI lui ne l'est pas dans Windows. A noter que cela ne fonctionne qu'avec un pc équipé d'une carte Bluetooth 4.0/4.2 compatible BLE, et à partir de Windows 10 v.1703.

Bracelet connecté : Rythme cardiaque

Pour le bracelet connecté, la démarche est assez simple : un petit logiciel trouvé sur github « mibandheartrate » permet de récupérer très simplement les données cardiaques lues par le bracelet et de les écrire dans un fichier texte directement dans le projet Unity. J'ai attaché un script dans la scène Unity qui va lire le rythme cardiaque du fichier et l'envoyer dans le générateur de particules en bpm. Il arrive que la lecture soit erronée et que la fréquence soit d'un seul coup inférieure à 50bpm (battement par minute) ou supérieure à 120bpm. Il y a donc un filtre qui ignore les valeurs n'étant pas dans cette gamme, ainsi que celles qui sont trop différentes de la valeur précédente. Une fois au générateur de particules, le rythme cardiaque est divisé par 60 pour les convertir en Hz, qui détermine alors la fréquence d'une sinusoïde agissant sur l'aspect des particules.

Activation des vibreurs : détection des collisions

Nous avons choisi d'utiliser un outil assez récent dans unity : le VFX graph. C'est un moteur de particules sur processeur graphique (GPU) optimisées, programmable dans un environnement nodal. L'avantage de faire exécuter de tels calculs est de pouvoir afficher beaucoup plus de particules : sur ma machine j'ai pu passer de 50 000 sur processeur (CPU) contre environ 50 000 000 sur GPU. Ensuite, le nombre dépend de la fluidité recherchée, et des opérations réalisées sur les particules. Le graph n'est aujourd'hui plus en bêta et est stable, il permet de nombreuses possibilités et est scriptable... dans un seul sens! C'est là l'inconvénient de faire exécuter des calculs sur le GPU: il est très complexe de récupérer des informations une fois qu'elles y sont. Il n'y a à ce jour aucun moyen d'obtenir des informations de collisions entre un maillage 3D et des particules GPU (voir sujet en annexe) dans le vfxGraph. Mais il est possible de créer ces collisions. Voici la routine dans le graph que j'ai imaginé pour répondre au problème : Des objets vides sont placés au bout de chaque doigt ainsi que sur la paume des mains. Une échelle spatiale fixe de 1 à 12 leur est attribuée, et est envoyée, ainsi que leur position vers le graph. Chaque position devient le centre d'une sphère qui se cogne aux particules. Les échelles quant à elles influent sur la couleur des particules en fonction de la distance avec les positions : si une particule est très proche de la sphère, elle change de couleur. Le fait d'avoir des échelles différentes permet de séparer les doigts, et donc de générer une couleur différente par doigt. Ensuite, il est possible de faire mourir une particule lorsqu'elle rentre en collision avec une de ces sphères GPU. Lorsqu'une particule meurt, il est possible de générer un événement GPU pour créer un autre système de particules. Ici, ce système ne comporte qu'une particule et hérite de la couleur de la particule parente lors de son décès (oui, c'est triste). Cette nouvelle particule apparait devant une caméra placée loin au-dessus du joueur, pour qu'il ne la voit pas. Au lieu de s'afficher sur un écran, la caméra écrit sur une texture sur le disque dur, une « render texture ». Elle comporte très peu de pixels, puisque c'est seulement sa couleur qui m'intéresse. Un script attaché à la caméra va alors lire un pixel de la texture pour avoir son niveau de gris. Ce niveau de gris sera alors différent pour chaque doigt qui aura touché une particule.

On notera qu'il est alors possible de ne détecter qu'une collision maximum par image. La fréquence recherchée en réalité virtuelle étant de 90 images par secondes, c'est une « latence » que j'ai jugé acceptable !

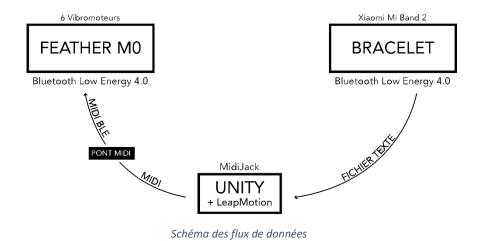
Ensuite, le script de la caméra attribue le niveau de gris de chacun des doigts et le transforme en entier de 1 à 12 et donne cette valeur à une variable. Pour finir, un second script récupère cette variable et l'envoie directement en MIDI vers le gant.

Activation des vibreurs : envoi du MIDI BLE

Comme expliqué plus haut, le MIDI BLE n'est pas nativement reconnu dans Windows. Mais au vu du nombre grandissant de contrôleurs l'utilisant, deux logiciels sont disponibles gratuitement et permettent de contourner le problème : Le premier, loopMIDI créé un instrument virtuel MIDI détectable par les logiciels supportant le MIDI sur Windows. Ensuite, il existe deux logiciels identiques, MidiBerry et WindSYNTHBerry qui permettent de détecter à la fois le contrôleur virtuel et le Feather M0 en BLE pour créer un pont entre les deux. Lorsque l'on veut pouvoir aller dans les deux sens, c'est-à-dire récupérer des données du Feather et lui en envoyer, il faut utiliser deux contrôleurs virtuels, ainsi que les deux logiciels permettant de créer deux ponts. Dans mon cas, un seul pont suffit : depuis l'ordinateur vers le microcontrôleur.

Flux de données

Afin de récapituler, voici l'ordre dans lequel les données transitent : Le bracelet connecté capte les pulsations cardiaques de l'utilisateur, et les écrit dans un fichier qui est lu en temps réel par unity et synchronisé avec l'apparence des particules. De son côté le leap motion permet de tracker la position des mains (réelles) de l'utilisateur, qui vont entrer en collision avec une particule. Le numéro du doigt est envoyé en MIDI depuis Unity vers l'instrument virtuel, puis transite dans le pont MIDI vers le Feather M0. Ce dernier associe le numéro de doigt à une sortie pwm, et génère une impulsion électrique vers le vibreur associé au doigt de l'utilisateur.



Le développement technique de ce projet aura pris le plus de temps sur la partie avec unity, ainsi que sur le BLE. Ce sont les deux parties les plus complexes et les moins optimisées du projet. Cependant, elles sont fonctionnelles, et le gant vibre correctement lorsque qu'une particule virtuelle est touchée!

Développement

Technique

Comme cité précédemment, le prototype est fonctionnel. Il demande cependant à être optimisé sur plusieurs points :

Premièrement, les soudures cassent souvent, même si c'est en partie à cause des DEL. Mais je dois être très vigilant à chaque fois que je l'enfile. Je n'imagine pas le faire utiliser dans son état actuel aux personnes qui utilisent le casque. Je vais donc prospecter premièrement du côté des câbles flexibles et techniques de vêtements connectés.

Ensuite, le fonctionnement du Bluetooth Low Energy complique beaucoup de choses. Je voudrais passer soit au Wifi, soit au Bluetooth « standard ». Un des éléments à prendre en compte est d'ailleurs de pouvoir en faire fonctionner deux en même temps.

Les particules GPU n'optimisent pas l'expérience! Je réfléchis donc soit à fouiller plus en détails du côté de Unity, soit à faire la concession de ne pouvoir « toucher » seulement certaines particules, qui seraient calculées sur le CPU.

Artistique

Nous réfléchissons premièrement à apporter d'autres éléments de synchronisation entre l'utilisateur et son avatar. Comme Marshmallow Laser Feast l'a fait, il peut s'agir de la respiration, ou bien de stimuler d'autres sens, comme l'odorat.

Si nous avons appris une chose en faisant des expériences de VR dans des salons, musées, galeries, etc., c'est l'importance de l'expérience collective : l'œuvre prend une autre dimension lorsqu'elle est partagée. Nous pensons donc, non pas à faire vêtir un casque à plusieurs utilisateurs en même temps, mais à créer de l'interaction entre ceux qui attendant, et celui vit l'expérience. Lorsque le projet sera terminé, nous espérons pouvoir le présenter dans des galeries, des festivals, etc.

Annexes

Le projet Unity avec les scripts en C#, le projet Arduino ainsi que miband-heartrate :

https://github.com/Levrden/HapticGloveUnity VfxGraph LutherieNumerique

Les sources de miband-heartrate : https://github.com/Levrden/miband-heartrate

La page de mon site regroupant les videos de Seeds : http://3d.hardyjonathan.fr/project/seeds/

Forum Unity traitant de la récupération des données GPU : https://forum.unity.com/threads/vfx-

 $\underline{graph\text{-}accessing\text{-}particles\text{-}data\text{-}in\text{-}scripts.651709}$

Loopmidi: https://www.tobias-erichsen.de/software/loopmidi.html

Midiberry: https://www.microsoft.com/store/productId/9N39720H2M05

WindSynthberry: https://www.microsoft.com/store/productId/9PFQ7B9P492X