

GRAILLOT Youé  
DOUSSE Robin

Partenariat :  
Laboratoire De Recherche  
Canadien

# OMNI LOW-COST MAISON



I.U.T. Clermont-Ferrand I, Département  
Informatique Graphique  
Année Universitaire (2015 – 2016)

**IUT** **département**  
**informatique graphique**

# Omni Low-Cost Maison

Rapport de projet tuteuré de seconde année Imagerie numérique

Graillet Youé

Dousse Robin



## Remerciements :

Nous tenons à remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet et plus particulièrement :

- Simon Marchand : Un membre de l'équipe à la naissance du projet qui partira quelques semaines plus tard au Canada. Il aura contribué à sa mise en place et au bon commencement de ce travail.
- Vincent Sauvage : Notre enseignant responsable qui aura beaucoup contribué à ce projet, toujours à l'écoute de nos besoins matériels, toujours très dynamique et sympathique.
- Pascal Pothée : La chef de département imagerie numérique.
- L'université d'Auvergne : Qui aura permis l'achat du matériel utilisé durant ce projet.
- Toute l'équipe pédagogique de l'IUT du Puy-En-Velay : Pour avoir répondu à certaines de nos questions, nous avoir fourni un matériel de qualité, et nous avoir donné un peu de leur temps.

## Résumé

En tout premier lieux, il est important de rappeler ce qu'est le Virtuix Omni et plus généralement les problématiques soulevés par l'arrivée imminente de la **réalité virtuelle** dans nos salons. Ce produit permet donc de compléter les **casques de réalités virtuels** actuels (ex : Oculus Rift) dans le but de nous immerger encore plus dans le monde virtuel. En effet, le joueur qui utilisera un casque de réalité virtuel doit tout de même se doter d'un contrôleur externe, un gamepad<sup>1</sup> la plupart du temps, pour interagir avec le monde virtuel. Le Virtuix Omni **remplace le contrôleur** (partiellement) et il est donc possible de reproduire la marche, la course et le saut sans appuyer sur aucun bouton. Cela contribue au réalisme de l'expérience et répond donc à l'une des principales problématiques de la réalité virtuelle : l'immersion de l'utilisateur.

Cependant, ce confort se paye au prix de 750\$ (sois moins de 700€), et sachant qu'un casque de réalité virtuel coûte autant, il est d'autant plus difficile d'imaginer voir ce dispositif dans le salon d'un particulier dans le but de jouer. En conséquence, notre objectif a donc consisté à reproduire ce système pour un prix plus modeste. Il nous a donc fallu **concevoir un modèle permettant de retranscrire les mouvements du joueur** dans un monde virtuel. Pour cela, nous avons donc décidé d'utiliser quatre cartes programmables, qui avec une douzaine de capteurs ultrasons, permettent de détecter la position en 2D de l'utilisateur. Ce matériel placé sur une plateforme constitue donc **notre alternative au système Omni** pour un peu plus d'une centaine d'euros sans compter la structure.

Pour mener à bien notre projet, nous avons dû en premier lieux nous mettre d'accord avec M. Sauvage quant à la forme que prendrait le produit fini. Celle-ci a changé avec l'avancement du projet pour finalement être constituée de deux rangées de capteurs soutenus par une structure solide. Le traitement des données de sorte à obtenir une bonne approximation de la position du joueur ainsi que l'utilisation de celle-ci dans une application interactive ont constitués les principaux défis à relever. Une fois cela fait, nous avons finalement pu adapter la valeur de la position pour que le jeu y réagisse.

Nous avons pu finir le projet à temps pour faire une démonstration du prototype à la journée portes ouvertes de l'IUT du Puy-En-Velay. Notre système, environ 6 fois moins cher que le produit original, permet de déplacer l'avatar virtuel mais est tout de même limité. Il n'est en effet pas encore possible de sauter ou de s'accroupir sans utiliser un contrôleur conventionnel en plus de notre système. Finalement, sans connaissance préalable sur l'utilisation de cartes programmables, nous avons pu apprendre beaucoup sur l'intégration dans Unity3D de matériel externe. Travailler à deux (puisque Simon a étudié son semestre 4 à l'étranger) fut une expérience riche et productive.

---

1 Gamepad : Contrôleur, souvent sous la forme d'une manette ou d'un clavier

## Sommaire :

### Rappel Du Projet Et De Ses Problématiques :

- L'Omni original
- La conception physique
- Le matériel
- La conception logicielle

### Réalisation :

- Arduino
- Unity

### Critique et ouverture

### Bibliographie

### Annexes

## **Rappel Du Projet Et De Ses Problématiques :**

Nous aborderons dans ce chapitre les grandes problématiques posées par ce sujet, en commençant tout d'abord par le cœur même du sujet, l'Omni original, en parlant de ses caractéristiques et définir les bases de ce qu'est un « Omni ». Après avoir découvert la réelle problématique, nous nous intéresserons à la conception physique du projet, les premières maquettes, et la structure et de ce qui sera plus tard la plat forme.

### **L'Omni original:**

Nous allons ici parler de la thématique essentielle du projet « Omni Lowcost », pour se faire rappelons ce qu'est l'Omni ([Cf annexe n°1](#)), de la société « Virtuix » :

« Le Virtuix Omni est la première plate forme de mouvement en réalité virtuel. RV<sup>2</sup> active, où vos actions dans le monde virtuel sont contrôlés par une navigation à la première personne comme marcher ou courir, qui crée une sensation d'immersion sans précédent qui ne peut être expérimenté en étant assis. Le Virtuix Omni vous permet de marcher, courir, vous accroupir, et vous pencher avec une liberté de mouvement à 360 degrés qui vous permet de contrôler votre avatar sans contrainte. ».<sup>3</sup>

Ce produit a été disponible à la précommande à partir du 30 avril 2015, et a été présenté dans de grands nombres de conventions informatiques et sélectionné comme produit phare. Cependant, ce produit a un coût et est aujourd'hui disponible pour la somme de 700 dollars ( sans livraisons ) en ne comptant que la plat forme, la ceinture, et les chaussures, sachant qu'il faut en plus un casque de réalité virtuel qui est indispensables à la bonne immersion. Ce produit n'est donc pas accessible pour le grand public mais à un public assez aisé, surtout pour une innovation assez récente. Ce qui nous amènera à notre sujet :

**« Comment réaliser un Omni Lowcost ? »**

---

2 RV : Réalité Virtuel

3 Société Virtuix, « [Produit Omni](http://www.virtuix.com/products/) » <http://www.virtuix.com/products/>

## La Conception Physique :

Ici sera abordé la conception de la plat forme (cf [Annexe 2](#)), son élaboration, son suivi depuis le début du projet (prototypage, premières esquisses, et stratégies pour répondre à la problématique).

Lors des premières réunions, nous nous sommes entretenues avec Mr.Sauvage pour concevoir un prototype de plat forme. Les premières buts de cette plat formes était de :

- Supporter le poids d'une personne
- Pouvoir se mouvoir
- Être capable de « ressentir » le mouvement
- Pouvoir récupérer la position de l'utilisateur sur celle-ci
- Facultatif : Être démontable

Point par point, la structure devait être assez solide pour supporter le poids d'une personne, nous nous sommes donc penchés sur les matériaux qui vont être utilisés, à commencer par le bois ou un métal léger. Le coût d'un métal assez résistant et léger, de plus à façonner, ne serait pas adapté au budget puisque les prix décollent pour ce genre de réalisation. Nous nous sommes donc penchés sur le bois.

Une fois le matériau choisi, vient les premières esquisses du support, à commencer par les dimensions. Faudra-t-il que l'utilisateur soit suspendu par la taille comme l'omni original, ou simplement se déplacer sur une plat forme ? Pour encore une fois par soucis de budget, un utilisateur en suspens nécessite une architecture solide et élaborée. Nous avons donc préféré choisir une plat forme où le joueur pourra se déplacer sur celle-ci de dimension 150\*150cm.

Ensuite vient sa forme : forme rond ? Forme carré ? La question est resté sans réponse quelques temps avant de choisir une structure carrée. Nous avons choisi cette forme pour sa simplicité, et l'espace plus grand pour manœuvrer.

L'omni suspend le joueur pour qu'il puisse marcher dans le vide et donc avoir la sensation de se déplacer. Pour avoir cette même sensation nous avons choisi de placer une ceinture reliée à des élastiques robustes pour que le consommateur ressente l'effort de son mouvement et puisse rendre l'immersion encore meilleur. Nous avons donc poster un pilonne à chacun des quatre coins du support pour supporter l'élastique.

Et enfin viens l'aspect démontable puisqu'un tel dispositif nécessite de la place. Ce souci sera résolu par un dispositif qui a été monté avec plusieurs pièces de bois, mis à part les poteaux supportant les élastiques qui seront en métal. Le tout vissé, donc facilement démontable par le moindre novice comme un meuble ikea.

Après avoir fait les premiers schémas et décidé de la forme et des dimensions de la structure vient le choix du matériel électronique à utiliser.

## **Le matériel :**

Ensuite comment pouvoir récupérer la position du joueur ? Nous nous sommes penchés d'abord sur l'utilisation de cartes programmables Arduino. Ces cartes matériellement libres ont l'avantage d'être peu coûteuses vu le nombre d'utilisation qu'il est possible d'en faire. Aucun membre de l'équipe n'avait eu d'expériences préalables quant à leur utilisation mais l'utilisation du langage C++, lui bien connu, nous a conforté dans notre choix. De plus la communauté qui utilise ce type de carte étant très présente sur internet, nous savions qu'il nous serait facile de nous documenter sur le sujet.

Nous nous sommes ensuite renseignés sur les différents capteurs que nous pourrions utiliser pour renseigner la position de l'utilisateur. Plusieurs options s'offraient à nous et deux ont retenus notre attention : les capteurs de distance infrarouge (15€) ainsi que les capteurs de distance à ultrasons (2€). La première est plus coûteuse et nous nous sommes donc penchés sur la deuxième solution.

Les capteurs à ultrasons fonctionnent en deux temps. Celui-ci peut envoyer un signal ultrasons face à lui avec son émetteur qui va être réfléchi par le premier obstacle se trouvant en face de lui, dans notre cas : l'utilisateur. C'est à la réception du retour de ce signal que nous pouvons déduire la distance qu'il a parcouru à partir du temps qu'il aura fallu entre l'envoi et la réception. Cependant, l'angle d'action de ce type de capteur est limité et c'est pourquoi il a fallu en utiliser une douzaine (ainsi que quatre cartes programmables) dans le but de couvrir tout l'espace de la plat forme.

Le prix du matériel nécessaire avoisine donc les 100€, ce qui est largement moins que l'Omni de Virtuix proposé à 700€. L'utilisation de cartes programmables Arduino ainsi que de capteurs était inédit pour nous et nous ne savions pas encore comment les mettre à profit pour repérer la position de l'utilisateur.

Dans la prochaine section, nous décrirons en détails les solutions logicielles que nous avons utilisé avec le matériel cité précédemment pour approximer la position de l'utilisateur et ainsi simuler le mouvement de son avatar.

## **La Conception Logicielle :**

Le choix des logiciels à utiliser fut relativement simple. La programmation des cartes Arduino passe par l'intermédiaire de l'application officiel java, libre et multiplat-forme téléchargeable sur le site [arduino.cc](http://arduino.cc). Simple d'utilisation, il sert d'éditeur de code et de compilateur, et peut transférer le firmware et le programme au travers de la liaison série (USB dans notre cas). Le langage de programmation utilisé est le C++ (compilé avec `avr-g++`) et est lié à la bibliothèque de développement Arduino pour permettre l'utilisation de ses entrées/sorties. Puisque le moteur de jeu Unity3D, que nous avons utilisé plusieurs fois, est compatible avec la communication cartes Arduino, nous avons décidé de l'utiliser pour tester notre Omni Low-Cost dans une scène en 3D.



## **Réalisation :**

### **Arduino :**

La première étape qu'il nous a fallu franchir fut la connexion des cartes programmables avec les capteurs. Même si nous n'avions quasi aucune connaissance dans le domaine, la documentation et les exemples trouvés sur internet nous ont permis de très vite pouvoir tester le matériel. Nous avons ainsi rapidement obtenu la distance d'un objet face au capteur en s'appuyant sur le tutoriel du site d'Arduino (schéma et code source en [Annexe 3](#) et [Annexe 4](#)). Il ne nous restait plus qu'à adapter le code de sorte à recevoir les données de plusieurs capteurs à la fois et de transmettre l'information à l'application via le port USB.

La partie suivante, plus longue, fut l'exclusion de certaines des valeurs. En effet, les capteurs ultrasons, imprécis, envoyaient souvent une mauvaise valeur de la distance. Il a donc fallu les trier avant de les utiliser et ce, en faisant attention à ne pas trop augmenter de temps de réactivité dans Unity3D. En calculant la moyenne directement dans le programme Arduino et en excluant les valeurs qui s'en éloignaient trop nous avons réussi à réduire le nombre de mauvaises valeurs. Malgré cela, les capteurs ont continué, mais dans une moindre mesure, à envoyer régulièrement des valeurs erronées. C'est pourquoi la position utilisée pour déplacer l'avatar aura tendance à ne pas être stable et par conséquent faire trembler le mouvement. Nous avons aussi remarqué le fait de placer un obstacle tel un drap en face d'une rangée de capteurs diminuaient sensiblement les mauvais résultats.

Une fois que nous avons réussi à obtenir des résultats satisfaisants il a fallu trouver le meilleur moyen d'utiliser les capteurs de sorte à obtenir une position en 2D du joueur. Nous avons tout d'abord pensé à placer les capteurs en cercle de sorte à obtenir l'accélération du joueur dans plusieurs axes. Cependant ce système eu l'inconvénient de ne pas pouvoir s'adapter à la plateforme. Nous nous sommes ainsi décidé à les placer sur deux rangées perpendiculaires, une pour l'axe horizontale et une pour l'axe vertical. En récupérant les valeurs provenant de chaque capteur, il suffit de choisir la valeur minimale pour obtenir le capteur en face de l'utilisateur et ainsi, sa coordonnée sur un des axes.

## Unity :

Toutes ces opérations se déroulent à l'intérieur d'Unity3D au travers de deux game-objects, un pour chaque rangée de capteurs. Ces game-objects configurables doivent avoir des variables correctement définies par l'utilisateur pour bien fonctionner : le nom des ports utilisés, l'axe correspondant à la rangée de capteurs et la force du mouvement à appliquer. Les ports utilisés par les cartes Arduino sont des ports série sont définis en suivant la règle de nommage : "COMX" avec X le numéro du port.

Une fois cela fait, il ne reste plus qu'à interpoler la valeur de manière à obtenir une vitesse qui peut être négative pour simuler la marche arrière et à l'appliquer sur un objet comportant un rigidbody et la caméra pour finalement reproduire le mouvement à l'intérieur du jeu. Nous avons choisi de proposer une expérience de jeu contemplatif en raison du manque de réactivité du système.

Unity3D rend très facile l'intégration du casque de réalité virtuel et il a suffi de trouver la bonne configuration pour l'utiliser dans notre jeu. Nous n'avons eu la possibilité d'utiliser le matériel qu'en fin de projet mais l'installation de celle-ci ainsi que la calibration des capteurs n'a pas mis longtemps avant que nous puissions tester notre premier jeu avec le produit final (cf [Annexe 2](#)).

## **Critique et ouverture :**

Notre but principal était de retranscrire les déplacements du joueur dans une expérience de réalité virtuelle immersive pour un prix plus modeste. Notre alternative low-cost à l'Omni est environs 6 fois moins chère que son modèle et permet de reproduire le déplacement de l'utilisateur. En ce sens nous avons atteints nos objectifs de départ mais il est cependant encore possible d'améliorer le système pour se rapprocher encore plus du modèle original. En effet, ce dernier propose la possibilité de sauter ainsi que de s'accroupir et utilise des capteurs plus précis et immersifs de par leur utilisation, les travaux de Simon décrits dans la partie suivante pourraient permettre ces actions. Tandis que sur la plat forme originale il faut bouger ses pieds de la même manière que si l'on marchait, il suffit de s'approcher des bords de notre plat forme pour mouvoir son Avatar.

Aussi, l'utilisation d'un autre type de capteurs de distance (ex : capteur de distance infrarouge), plus précis mais aussi plus coûteux, permettrait d'obtenir un meilleur résultat en ce qui concerne l'approximation de la position 2D du joueur. Cela permettrait aussi de réduire la latence en utilisant moins de capteurs (2 pourraient suffire) et aussi le nombre de cartes Arduino.

Pour aller plus loin il serait aussi possible d'ajouter un dispositif audio 3D à l'ensemble pour apporter encore plus à l'immersion de l'utilisateur.

Finalement, solidifier la structure permettrait d'augmenter la force que l'on applique sur le joueur en réponse à son mouvement. Sa ceinture peut elle-aussi être améliorée en réduisant le frottement qu'elle exerce lorsque le joueur veut se tourner. Celle-ci lui rappelle qu'il se trouve sur une plat forme et limite ainsi son expérience.

## **Répartition de la charge de travail et conclusion :**

Si initialement nous devions être trois, nous nous sommes finalement retrouvés à travailler à deux pour la majorité du projet. Le travail avec Simon était compliqué compte tenu du fait qu'il soit parti étudier son semestre 4 à l'étranger, il a donc été décidé qu'il travaillerait à part sur un autre projet pouvant éventuellement servir à améliorer notre alternative de l'Omni. Son but a consisté à exécuter une action en fonction de la position et des gestes d'un utilisateur pas l'intermédiaire de la Kinect de Microsoft.

Nous avons donc divisé le travail en deux. Robin était chargé de la programmation des cartes Arduino dans le but d'obtenir des valeurs approximant la position du joueur tandis que Youé a travaillé sur leur intégration dans Unity3D pour déplacer l'avatar virtuel. Il aura fallu apprendre certaines pratiques nous-mêmes comme la programmation de carte Arduino ou encore la lecture de port série mais la documentation et les exemples trouvés sur internet ont suffi. Les travaux variés que nous avons réalisés nous ont permis d'apprendre comment se servir de matériel externe comme d'un périphérique. Nous avons ainsi pu nous familiariser avec l'utilisation de matériel de réalité virtuelle récent.

Cela aura été une très bonne expérience qui nous aura permis de nous ouvrir à une informatique que nous n'avons pas l'occasion d'aborder en cours. Nous avons pu finir une démonstration jouable pour la journée portes ouvertes de l'IUT du Puy en Velay ce qui nous a permis de montrer au public présent nos résultats. De plus, travailler dans le but de rendre l'expérience de réalité virtuelle encore plus immersive nous a passionné et a constitué une expérience enrichissante dont nous sommes fières et qui pourra potentiellement nous être utile dans de futurs projets en rapport avec le jeu-vidéo ou autre.

## **Bibliographie :**

Auteurs multiples, « Arduino », <https://en.wikipedia.org/wiki/Arduino>  
Société Virtuix, « Produit Omni » <http://www.virtuix.com/products/>

## Annexes :

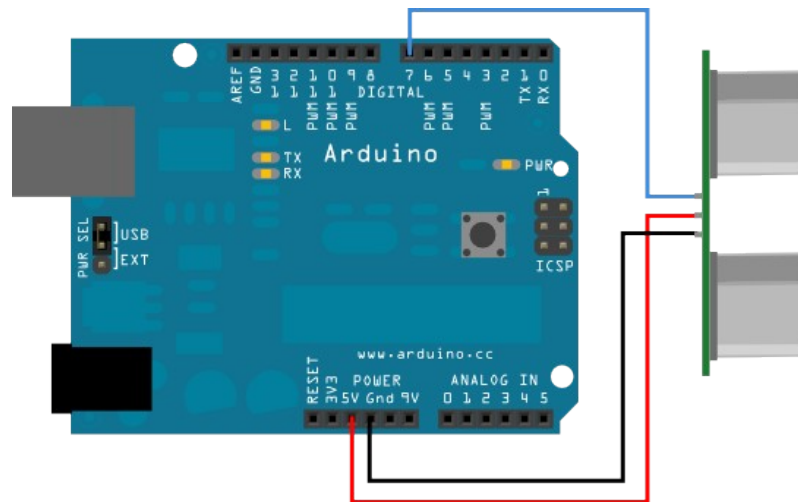
### Annexe n°1 : Omni Virtuix



## Annexe n°2 : La plat forme finale



## Annexe n°3 : Rendu Fritzing du branchement d'un capteur ultrason à une carte Arduino Uno





## Annexe n°4 : Code source exemple de l'utilisation d'un capteur ultrason

```
// this constant won't change. It's the pin number
// of the sensor's output:
const int pingPin = 7;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop() {
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))) : a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);
}

long microsecondsToInches(long microseconds) {
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second). This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```