Super Camera Project

Rapport

Guillod Fabien
Clément Valentin
Menoud Alain

Table des matières

1. Description du projet

Dans le cadre du cours MMI, nous avons vu du imaginer un projet libre mais avec les contraintes suivantes :

- Utilisation d'au moins 2 modalités. Une modalité doit être le geste, avec au moins 2 gestes différents à reconnaitre. La deuxième modalité est à choix
- Le matériel mis à disposition est un kit phidget, une manette Wii, un microphone et une caméra.

Notre idée de projet consiste à réaliser une application qui permet de gérer une web-cam motorisée.
La webcam est posée sur 2 moteurs afin de pouvoir se diriger de gauche à droite et de haut en bas. L'application sera également capable de réagir à la voix de son utilisateur grâce à des commandes telles que « Start », « Stop » ou « Save » par exemple.

Nous avons donc utilisé les modalités suivantes :

- Reconnaissance vocale : La reconnaissance est utilisée pour les commandes de l'application

- Motricité de la webcam : La webcam est posée sur deux moteurs afin de pouvoir être libre de ses mouvements. Un accéléromètre sera utilisé pour générer les mouvements de la caméra. (Possibilité de le mettre sur un gant ou éventuellement un casque).

2. Modalités utilisées

2.1 Reconnaissance de la voix

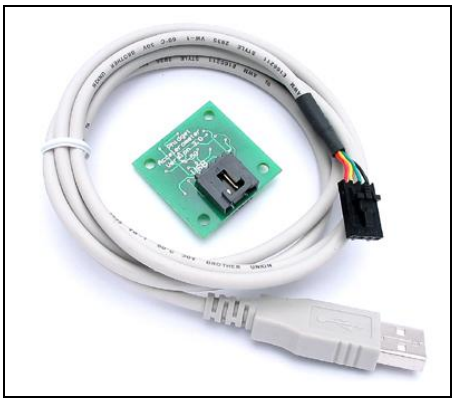
Pour la reconnaissance vocale, nous avons utilisé la librairie gratuite pour Java qui se nomme Sphinx. Cette librairie permet de définir notre propre grammaire que l'on souhaite faire reconnaître, et nous pouvons donc être plus précis. Cependant, le "recognizer" de cette libraire peut être configuré de manière pointue, afin de s'adapter au mieux à l'accent ou la prononciation de l'utilisateur.

Pour ce projet, nous n'avons pas changé les paramètres par défaut, car l'objectif était surtout de découvrir les différents moyens d'intégrer la multimodalité dans les applications. C'est pourquoi le "recognizer" a un peu de peine à reconnaître notre beau suisse-romand ☺

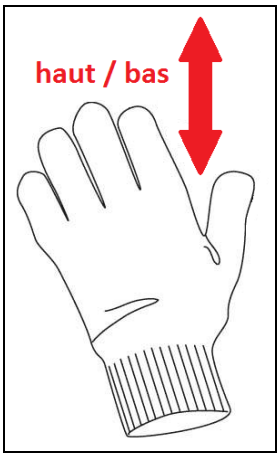
Pour faciliter l'utilisation de la voix, nous avons utilisé l'oreillette bluetooth fournie :

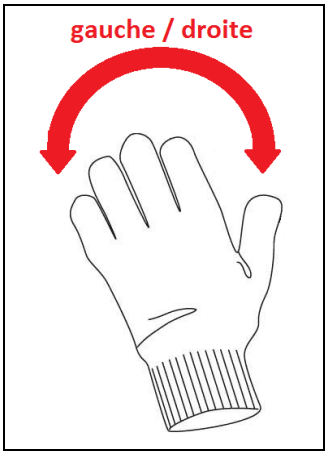
2.2 Mouvement par accélèromètre

Pour la modalité des gestes, notre dispositif utilise un accélèromètre qui fonctionne sur deux axes. Voici le phidget utilisé que l'on branche directement sur le port USB:



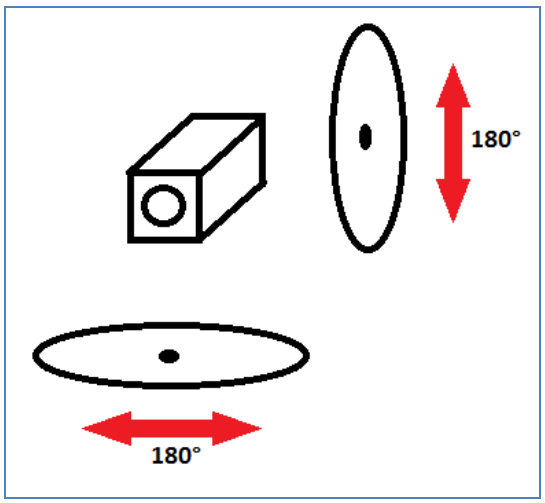
Afin d'obtenir un fonctionnement simple et évident pour l'utilisateur, nous avons fixé le phidget sur un gant. Les mouvements de la main de haut en bas et de la gauche vers la droite permettent de faire bouger notre camera.





2.3 Controle moteur

Notre dispositif utilise une structure en bois faite par nos soins afin de pouvoir faire pivoter la camera dans les deux directions (haut-bas et gauche-droite). Afin de faire bouger cette structure, nous avons utilisé deux moteurs phidget avec un rayon de 180° chacun. Grâce au kit phidget, ces moteurs sont contrôlables via l'API fourni. Nous utilisons donc les positions données par l'accéléromètre afin de mettre à jour nos moteurs. Les données fournit par l'accéléromètre nous permettent également de gérer la vitesse de rotations des moteurs se qui permet un déplacement plus précis pour l'utilisateur.



Afin de développer cette partie, nous avons utilisé le langage de programmation Java qui semblait être le plus facile à mettre en place. Pour cela, nous avons développé une classe MotorControl qui nous permet de gérer toutes les opérations liées à ces moteurs.

2.4 Capture vidéo

Après avoir fait plusieurs essais infructueux avec un Framework destiné à Java (JMF), nous avons constaté que notre webcam était trop récente pour être utilisée avec ce Framework qui n'a plus été mis à jour depuis un certain temps. De plus, nous n'avons pas trouvé d'autre alternative en Java.
Pour pallier à ce problème, nous avons donc développé une petite application de capture vidéo en C#. Cette application acquiert les images directement de la webcam pour les afficher sur notre interface. Afin de créer des fichiers vidéo, nous avons utilisé une librairie gratuite (SWFSlideShowsCout) qui nous permet de générer un fichier SWF (vidéo flash) à partir de nos images sauvées en tant que fichiers temporaires.
Afin de contrôler cette interface, nous avons mis en place un petit serveur socket du côté de la capture vidéo et un socket client du côté de l'application principal. Ceci nous permet d'envoyer des commandes à notre application de capture lorsque l'application générale reçoit les commandes vocales.

2.5 Fusion des modalités

Dans notre projet, les deux modalités sont indépendantes et peuvent s'exécuter en parallèle. Par conséquent, nous ne fusionnons pas les modalités dans le programme.

3. Librairies

Pour l'utilisation des phidgets (les moteurs et l'accéléromètre), il faut récupérer la librairie Phidget21 qui se trouve à cette adresse ou sur le cd :

http://www.phidgets.com/drivers.php

La reconnaissance de la voix s'effectue avec la librairie Java Sphinx, téléchargeable à cette adresse ou sur le cd :

http://cmusphinx.sourceforge.net/sphinx4/

La création de vidéo au format SWF se fait à l'aide de la librairie SWFSlideShowCout, cette libraire est disponible sur le cd fourni.

4. Conclusion

Nous avons trouvé ce projet intéressant. Il nous a permis de mettre en pratique la théorie vu en cours à travers ce projet que nous avons dû imaginer de toutes pièces.
Par ce projet, nous avons pu constater que l'avenir des applications se fera de plus en plus avec des interfaces multimodales. L'être humain utilise dans la vie de tous les jours et naturellement ses 5 sens. En conclusion, il sera judicieux pour nous à l'avenir d'utiliser ses modalités afin de simplifier, rendre évidente et naturelle l'utilisation de nos applications.