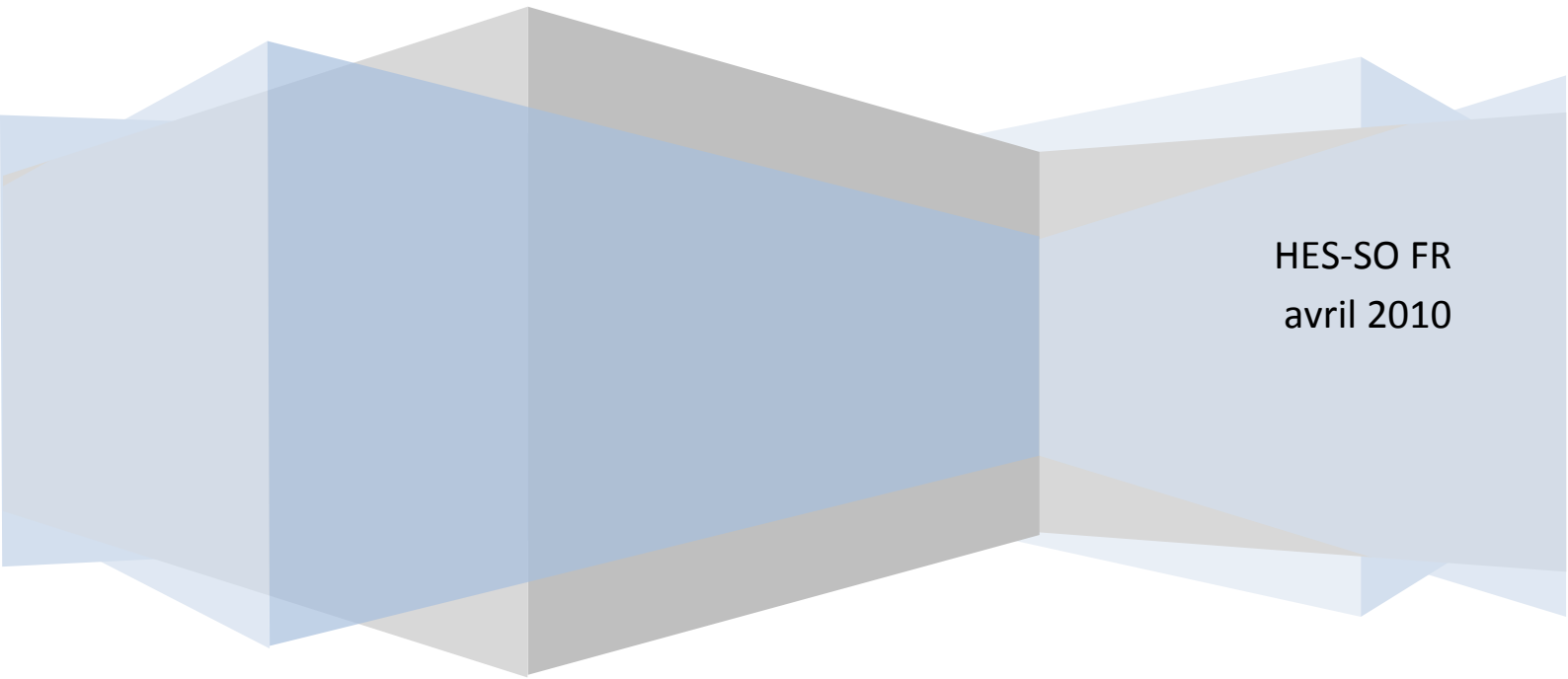
Projet MMI

WiiPaint

Constantin Cédric
Demierre Cédric
Tscherrig Julien



HES-SO FR
avril 2010

Table des matières

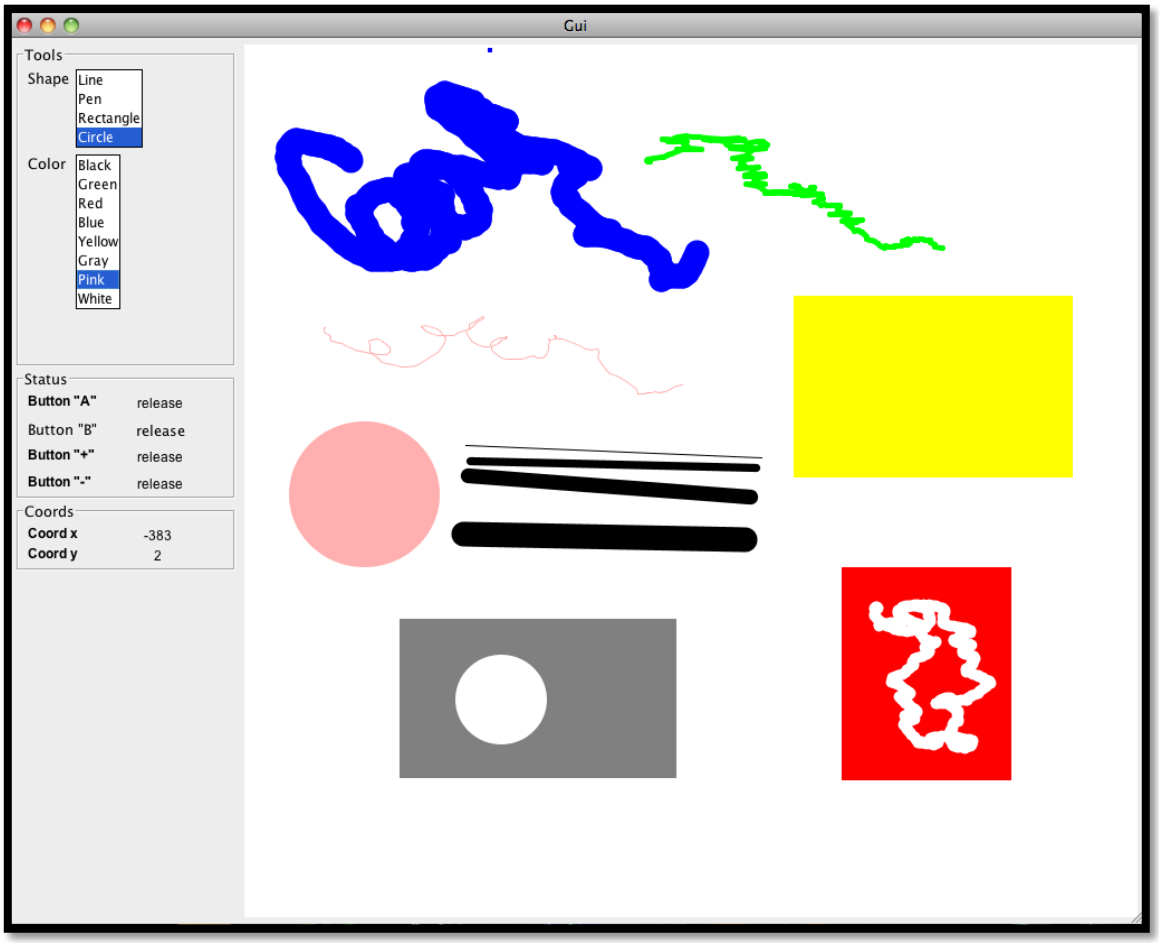
1. Description du projet

Le projet WiiPaint s'inscrit dans le cadre du cours « Interface multimodales ». Le but du projet est de créer un petit programme donnant à l'utilisateur la possibilité d'utiliser le geste et une ou plusieurs autres modalités. Notre programme utilise donc la voix et le geste, il s'agit d'un outil de dessin simple dans lequel on utilise une Wiimote comme pinceau, ou comme outil de dessin de formes. La voix sert alors à changer d'outil ou de couleur des traits.

1.1. L'interface

Comme on peut le voir ci-dessous, l'interface est très simple. La partie de gauche contient des informations concernant l'outil actuellement sélectionné, la couleur des traits ainsi que l'état des différents capteurs de la Wiimote. La partie de droite quant à elle est une grande zone graphique où l'on va dessiner.

Comme on peut le voir, plusieurs outils et plusieurs actions peuvent être effectués par l'utilisateur. Il est possible de utilisé un pinceau, de dessiner des cercles ou des carrés. Il à également une belle palette de couleur et la possibilité de gommer, avec la couleur blanc. Touts les actions sont décrites plus en dans le chapitre suivant « modalités d'interaction ».



2. Modalités d'interaction

Voici la liste de toutes les actions possibles avec les modalités utilisés.

2.1. Dessiner

L'utilisateur peut dessiner sur la surface blanche du programme à l'aide de gestes. Pour cela, il devra prendre la Wiimote et l'utiliser comme un pointeur laser. Pour déposer la couleur, il faudra alors maintenir appuyer sur la touche 'B' (gâchette) puis simplement dessiner. Comme sur un logiciel de dessin, plusieurs outils sont à disposition. Il est possible de peindre librement avec « Pen » ou alors de faire des rectangles, des cercles qui sont tout les deux pleins ou des lignes.

- CASE : Il n'est pas possible d'effectuer d'autres taches lorsqu'on dessine, c'est donc un cas d'utilisation exclusif.
- CARE : Pour l'utilisateur, la seule possibilité de dessiner est d'utiliser le geste donc nous avons affaire ici à un assignement. Le geste est assigné à l'action de dessiner.

2.2. Changer d'outil ou de couleur

L'utilisateur peut à l'aide de sa voix et de la Wiimote changer simplement l'outil sélectionné mais également la couleur des traits. Pour cela, il lui faut maintenir la touche 'A' enfoncée et prononcer le nom de l'outil ou de la couleur désirée.

- CASE : Au niveau logiciel dans ce cas, nous utilisons deux flux de données distincts pour effectuer l'action. Dans un thread on récupère le fait que la touche 'A' est pressée et dans un autre, que l'utilisateur prononce un le nom d'une couleur. Ce cas peut donc être considéré comme étant une utilisation synergétique de la multimodalité.
- CARE : L'utilisateur doit effectuer deux actions différentes pour arriver au résultat. Le geste est donc ici complémentaire à la parole.

2.3. Changer la taille du trait

Il est possible de changer la taille du trait (pour les lignes et le pinceau) en appuyant sur les touches '+' et '-' de la Wiimote.

- CASE: le changement de la taille du trait est une utilisation exclusive. Elle ne peut être effectuée que par le geste et doit être effectuée individuellement
- CARE: La modification de la taille du trait est assignée au geste.

2.4. Quitter l'application

Il est également possible de quitter l'application simplement en appuyant sur la touche 'home' de la Wiimote.

- CASE : Quitter l'application peut être effectué à n'importe quel moment et est réservé au geste. On pourrait la considérer comme concurrent mais comme elle coupe définitivement toute autre action, elle est plutôt de type exclusif.
- CARE : Le geste est assigné à cette action.

3. Choix de conception et d'implémentation

3.1. MVC simplifié

Pour notre projet, nous avons choisi de faire au plus simple en n'utilisant que partiellement le modèle MVC. Comme il n'y a pas de donnée ou de calculs excessifs, nous n'avons pas implémenté de Model distinct. C'est la vue qui va directement dessiner les formes, lorsqu'elle reçoit un événement de la Wiimote. C'est elle également qui stocke l'information de couleur et d'outil sélectionné directement dans les listes correspondantes. Lorsqu'elle reçoit des événements de la reconnaissance vocale, elle va également changer directement ces valeurs.

3.2. Passage d'événement

Comme la reconnaissance vocale et l'utilisation de la Wiimote sont basées sur des librairies externes, nous avons été limités dans nos possibilités d'implémentation. De ce fait, ces deux parties utilisent des techniques différentes.

- **Reconnaissance vocale**

Puisque la classe principale utilisée avec la librairie Sphinx n'étend aucune classe, nous avons choisi la technique de passage d'éléments classique et simple de java « Observable, Observer ». La classe de reconnaissance vocale est donc observée par la vue et indique le mot qui vient d'être prononcé avec la méthode `notifyObservers(String motdetecter)`.

- **Wiimote**

Avec la Wiimote et l'utilisation de la librairie WiiRemoteJ, il nous a été impossible d'utiliser cette technique car la classe principale génératrice d'événements devait déjà étendre une classe. Nous avons donc décidé d'utiliser la méthode « Custom Events ». Lorsqu'une action est faite avec la Wiimote, un évènement est lancé sur un listener prévu à cette effet et un objet de type WiimoteState contenant tous les états de la Wiimote est transmit à la vue.

3.3. Choix du vocabulaire

Pour la reconnaissance vocale, nous avons choisi d'utiliser un vocabulaire très simple, un seul mot au lieu d'une phrase. Cela pose passablement de problèmes car Sphinx n'est que peu fiable pour la reconnaissance de mot seul. Il faut pouvoir l'utiliser dans un lieu calme et prononcer les mots dans un bon anglais si l'on veut une reconnaissance correcte. Nous avons fait ce choix car c'est la manière la plus naturelle de parler à l'ordinateur. Il serait stupide d'utiliser une longue phrase pour changer de couleur car soit les phrases n'auraient aucun sens, soit seul le nom de la couleur changerait dans la phrase, ce qui n'améliorerait pas la reconnaissance.
Pour éviter que des mots ne soit reconnus sans qu'on l'ait voulu, nous avons décidé de coupler l'action de parler avec celle de maintenir la touche 'A' de la Wiimote enfoncée. De ce fait, il est possible de parler lors de l'utilisation du programme et de changer de couleur uniquement lorsqu'on le désire réellement.

4. Librairies utilisées

4.1. Sphinx-4 1.0beta

Pour la reconnaissance vocale, nous avons utilisé la librairie sphinx-4 en version 1.0Beta. C'est une librairie gratuite entièrement codée en java qui permet de faire énormément de choses au niveau de la reconnaissance vocale. Plusieurs dictionnaires sont à dispositions et les réglages possibles sont nombreux. La création d'un petit vocabulaire plus ou moins bien reconnu est facile. Mais pour arriver à une bonne reconnaissance, il faudrait avoir la possibilité de tester ce produit plus en détail.

http://cmusphinx.sourceforge.net/

4.2. WiiRemoteJ

Pour l'utilisation de la Wiimote, nous avons choisit d'utiliser la librairie WiiRemoteJ. C'est une librairie gratuite entièrement codée en java qui permet de connecter et de lire l'état d'une Wiimote à l'aide de BlueTooth. L'utilisation en est très simple et les informations qu'il est possible de récupérer sont nombreuses. Les boutons actuellement pressés et les coordonnées en sont un exemple.

http://www.world-of-cha0s.hostrocket.com/WiiRemoteJ/

4.3. Bluecove-2.0.1

Pour la connexion Bluetooth, nous avons utilisé la librairie Bluecove qui est une librairie pour la connexion Bluetooth pour JAVA. Nous avons eu beaucoup de problème avec cette librairie, il nous a été impossible de l'utilisé correctement avec Windows. Nous nous somme rencontrer devant un problème de stack Bluetooth incompatibles avec la librairie. Nous avons donc élaboré notre projet sur une plateforme MAC.

http://sourceforge.net/projects/bluecove/files/

5. Conclusion

Ce projet nous a éclairé sur la difficulté à créé un projet multimodale et aussi toutes les possibilités qu'un offre a l'utilisateur l'utilisation des gestes et de la voix. Nous avons également appris que malgré la grande promesse de JAVA d'être cross-plateforme, les technologies telles que Bluetooth reste indépendante du système.

Au final ce projet reste simple mais illustre parfaitement l'utilisation des modalités dans une application.