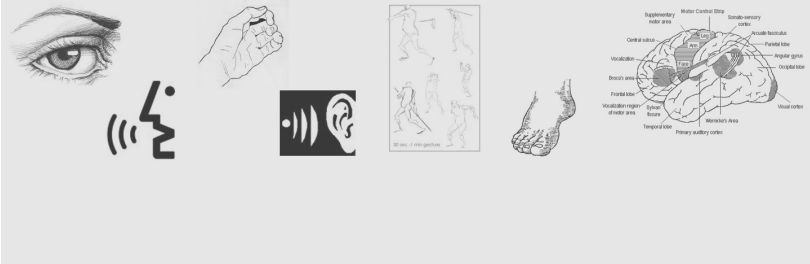
EIA-FR
HTA-FR



Multimodal Interfaces

2010

Adrien Giner
Frédéric Nell
Mathieu Barras

Date : 04/19/2010

Sommaire

1 Introduction

Dans le cadre du cours MMI, nous avons du réaliser un petit projet. Ce projet avait pour but d'intégrer deux modalités.

La durée du projet était de 8 semaines, au rythme de 1 après-midi de cours par semaine, le reste de l'application étant développée durant notre temps libre

Nous avons donc choisi de faire un pictionnary, qui utilise la voix pour deviner quel mot on est en train de dessiner et le geste (wii-mote) pour dessiner. Nous avons aussi intégrer la reconnaissance de gestes à notre application.

2 Présentation du groupe

Notre groupe est composé de 3 personnes de la classe I3 de l'École d'Ingénieurs et d'Architectes de Fribourg:

Barras Mathieu
Giner Adrien
Nell Frédéric

Mathieu Barras s'est occupé de la partie qui concernait la voix.

Adrien Giner s'est occupé de la manette Wii et de son utilisation.

Frédéric Nell s'est occupé de la conception de l'interface et des diverses interactions et fonctionnements du jeu et de réaliser un petit film.

3 Description du projet

L'application que nous avons décidé d'implémenter sera une sorte de *Pictionary* interactif.

3.1 Règles de jeu

- Deux équipes de minimum 2 joueurs s'affrontent.
- A tour de rôle, chaque équipe élit un dessinateur qui devra dessiner une suite de mots aléatoirement choisis par l'ordinateur en les dessinant à l'aide d'une wii-mote.
- Les coéquipiers du dessinateur doivent dire le mot représenté par le dessin dès qu'ils pensent l'avoir deviné.
- Le but est, pour chaque équipe, de trouver le plus de mots possible dans un temps donné (1 minute).
- Chaque équipe joue alors à tour de rôle un certain nombre de manches et obtient un score équivalant à la somme des mots trouvés durant chaque manche.
- L'équipe ayant trouvé le plus de mot à la fin du jeu gagne.

3.2 Déroulement du jeu

Chaque équipe désigne un dessinateur, celui-ci met le casque qui lui permet d'entendre le mot à dessiner. Puis il dessine les mots durant le temps qui lui est imparti (une minute) et les autres membres du groupe peuvent énoncer le mot, lorsqu'il pense l'avoir deviner. Les interactions du dessinateur sont de deux types :

- Vocales : permet de passer à un autre mot, effacer la zone de dessin, répéter le mot, etc.
- Gestuelles : permet de dessiner les mots à deviner ou effacer la zone de dessin au moyen d'un geste pré-défini.

Les autres membres du groupes peuvent interagir avec l'application avec la voix. En énonçant les mots qu'ils pensent avoir deviner.

A la fin de la minute de dessin, les points ayant été comptabilisés, l'autre équipe prend possession du système et joue.

3.3 Modalitées d'interaction

La donnée du projet spécifie que nous devons utiliser au moins 2 modalités différentes dont l'une doit être le geste. Afin de réaliser notre jeu, nous utilisons donc:

- Le geste (wii-mote): Dessin du mot caché à l'aide d'une wii-mote, détection de geste pré-enregistré et lié à une action
- La voix (microphone): Identification du mot découvert par les joueurs, interaction avec le jeu pour le dessinateur

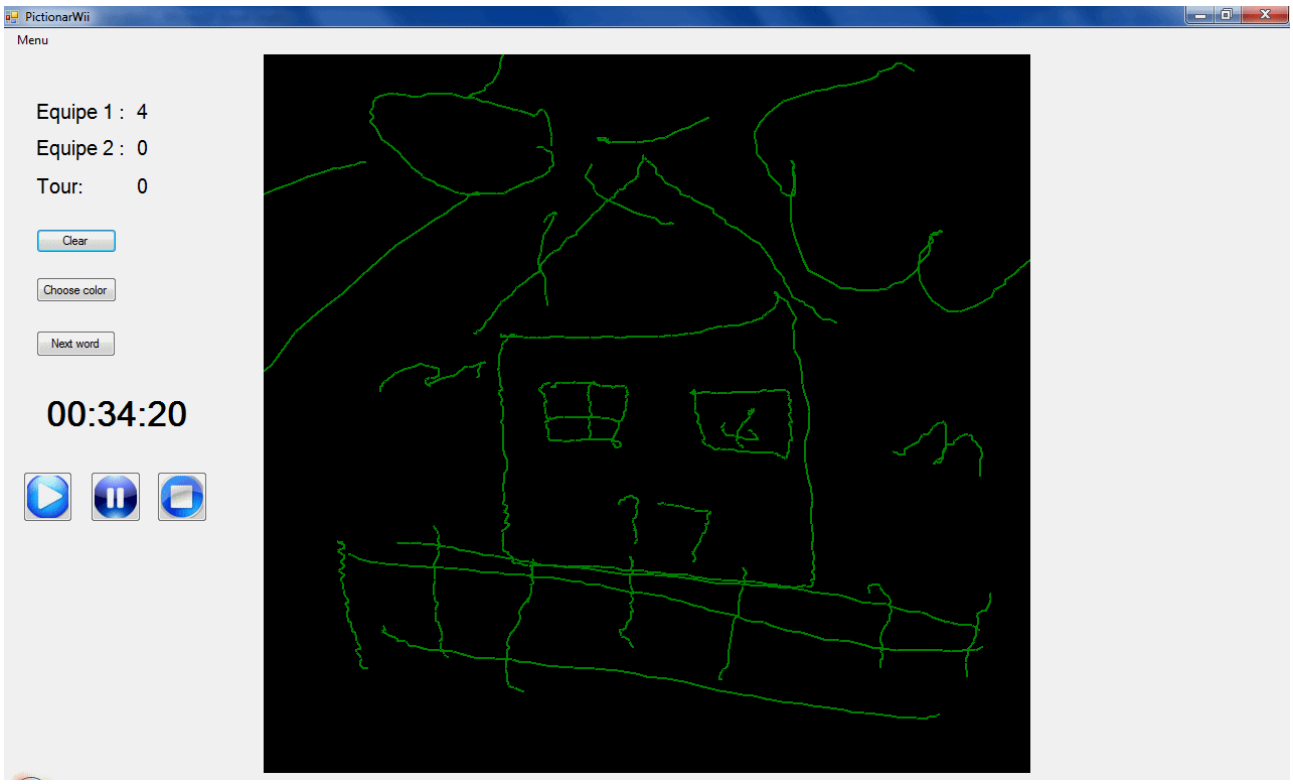
Ces éléments sont mis à disposition par l'école dans le kit multimodale.

Nous allons vous décrire ces modalités et comment nous les avons implémentées.

3.4 Fenêtre principale

La fenêtre principale de l'application est divisé en deux parties :

- La partie droite est un « PictureBox » permettant de dessiner les mots à deviner.
- La partie gauche permet de visualiser les points des deux équipes, de démarrer/arrêter/pauser manuellement le timer, de visualiser le temps restant, de modifier la couleur et d'effacer la zone de dessin.



Nous avons aussi un menu nous permettant de configurer les différentes modalités de l'application :

- Configuration de la wii-mote.
- Configuration de la détection de gestes

De plus le menu permet de quitter l'application ou de redémarrer un nouveau jeu.

4 Geste

Nous avons choisis d'utiliser la wii-mote afin d'intégrer le geste dans notre application.

Il existe plusieurs librairies disponibles afin de développer une application en C# pour la wii-mote cependant nous avons décidé d'utiliser la librairie fournie par M.Nicolas Pittet. Ceci afin de pouvoir aussi intégrer la détection de la gestuelle dans notre application car à ce jour il n'existe pas de librairie complète permettant de l'intégration en C#. Le développement de notre propre algorithme de détection de mouvement n'était pas envisageable dans le temps imparti pour le projet.

La wii-mote est simplement utilisé comme une souris, en modifiant la position du pointeur pour le mouvement et la génération de différents éléments de clique au moyen de dll de Windows.

De plus la wii-mote nous permet de faire de la détection de gestes. Dans notre application nous avons lié le geste d'un « L » à l'effacement de la zone de dessin.

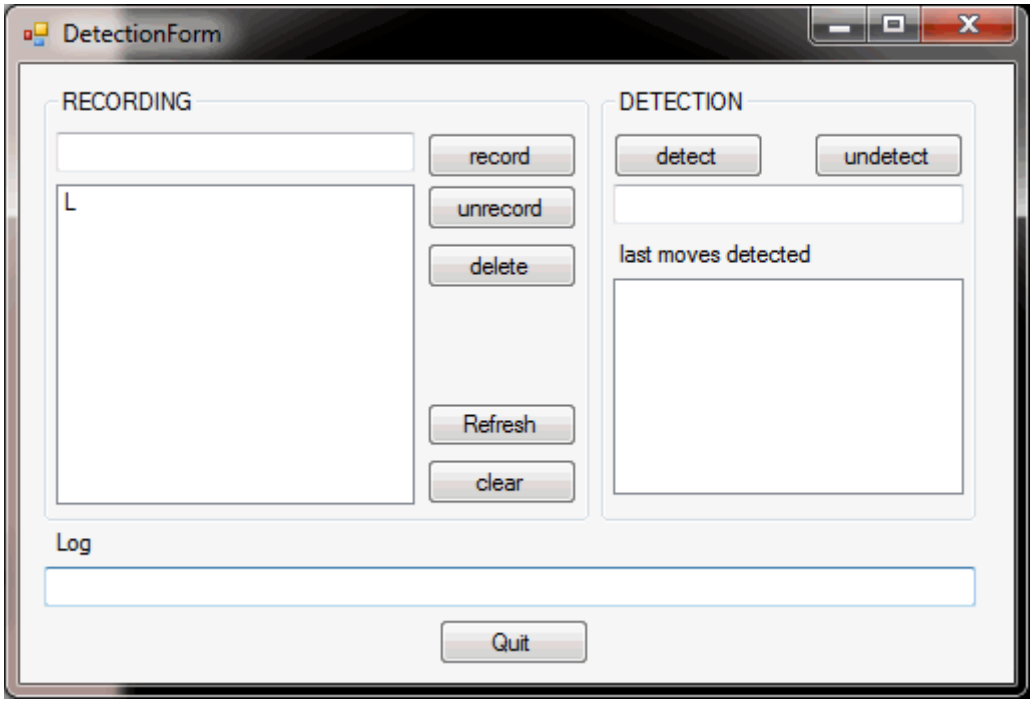
4.1 BkWorkerForm

Cette interface d'arrière-plan permet d'utiliser la wii comme une souris. Ceci est effectué en parsant les senseurs de la wii-mote et en déterminant sa position sur l'écran puis en changeant le curseur de la souris (*Cursor.Position*). De plus, à l'aide de la classe Mouse.cs, cette interface génère des évènements de clique droite/gauche.

Cette interface appelle la fonction *public void CatchMvt(String mvt)* de la classe Conteneur.cs (fenêtre de dessin) afin de l'avertir qu'un geste a été détecté.

4.2 DetectionForm

Cette interface permet d'enregistrer et de de détecter les différents mouvements enregistrés.



La partie gauche permet d'enregistré un mouvement. Il faut entrer le nom du mouvement dans le textbox puis cliquer sur « record ». Le geste est ensuite exécuté avec la touche « Minus » de la wii-mote pressée. L'ensemble des mouvements enregistrés sont listés dans la listebox.

La partie droite permet de détecter les mouvements préalablement enregistrés. A chaque fois qu'un nouveau mouvement est détecté, celui-ci est ajouté dans la liste de droite « last moves detected ». Comme pour l'enregistrement de mouvements, la détection doit se faire avec la touche « Minus » de la wii-mote pressée.

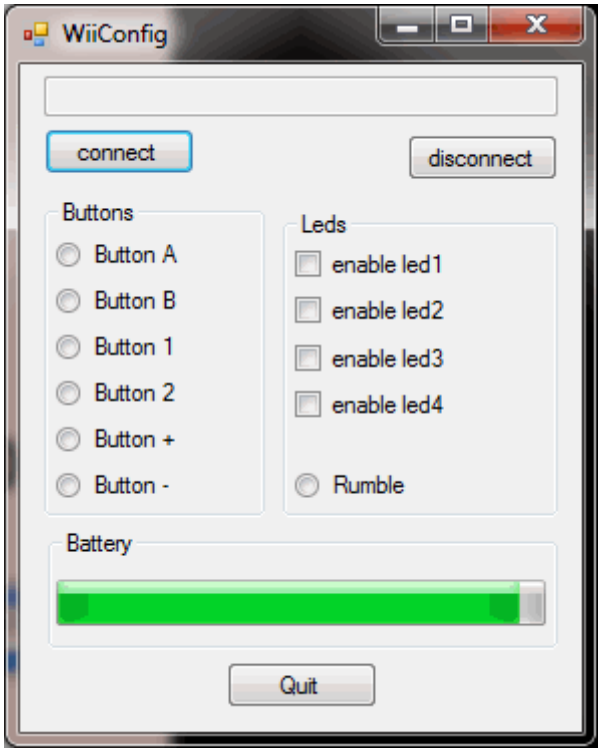
Dans notre application nous n'avons enregistré qu'un seul geste, un « L » qui nous permet d'effacer le dessin.

4.3 WiimoteConfig

Cette interface permet de configurer la wii-mote.

Lorsque l'utilisateur appuie sur un bouton de la wii-mote, les différents radio bouton s'active. On peut aussi activer les différentes leds au moyen des check box.

On peut faire vibrer la wii-mote au moyen du radio button « Rumble » et voir l'état de la batterie au moyen de la barre de progression.



4.4 Mouse

Cette classe permet simplement de faire des appels à la librairie Windows *User32.dll* afin de générer des cliques de souris.

4.5 Package WiiGestureLib

Ces classes nous ont été fournies par M. Nicolas Pittet. Elles permettent de gérer la wii-mote et de faire de la reconnaissance de gestes.

Seule la classe *wiimoteManager.cs* a été modifiée afin de l'adapter à notre application.

- Suppression des fichiers xml de mapping des touches et des méthodes/attributs y relatifs
- Ajout de la possibilité d'avoir plusieurs « Observer » sur les différents événements de la wii-mote.

- Modification de la façon dont les touches de la wii-mote sont gérées. A chaque appui ou relachement de l'utilisateur sur un bouton de la wii-mote, un événement est généré et envoyé à l'ensemble des « observers » enregistrés.
- Plusieurs autres modifications mineurs ont été effectués (ajout de getters, setters, refactoring, ...).

Liste des classes :

- AccValue.cs
- Axle.cs
- Movement.cs
- wiimoteManager.cs
- WindowHoldingAWiimote.cs

4.6 Package WiiLib

Ces classes nous ont été fournies par M.Nicolas Pittet. Elles permettent d'interagir avec la wii-mote. Seul quelques modifications mineures ont été apportées.

C'est la librairie bas-niveau permettant de récupérer les différentes informations des senseurs sur la wii-mote à l'aide de la connection bluetooth.

Liste des classes :

- DataTypes.cs
- Events.cs
- HIDImports.cs
- wiimote.cs

4.7 Package WiiXMLHandler

Ces classes nous ont été fournies par M.Nicolas Pittet. Elles permettent de travailler avec des fichiers XML pour l'enregistrement des mouvements.

Liste des classes :

- XMLReader.cs : lecture de fichiers XML
- XMLValidator.cs : validation de fichiers XML
- XMLWriter.cs : écriture dans un fichier XML

5 Voix

Par rapport à une application standard de type GUI, l'utilisation de la reconnaissance et de la synthèse vocale peut rendre une application beaucoup plus facile et confortable à employer.

5.1 Reconnaissance vocale

Dans un système comme le notre ou plusieurs utilisateurs interagissent en même temps sur la même application, cette modalité permet d'éviter à chaque joueur de devoir accéder à un clavier ou à une souris pour interagir avec le système. Le simple fait de dire un mot suffit à communiquer avec la machine. Quoi donc de plus naturel !

De plus, nous avons définit certaines commandes qui permettent d'interagir avec le système. Les interactions que nous avons implémentées sont les suivantes:

- Demander un nouveau mot (dire « mot »)
- Demander de répéter le mot (dire « Répète »I
- Effacer le dessin (dire « efface »)
- Démarrer le chronomètre avant de commencer à dessiner (dire « démarre »)

D'un point de vu purement technique, nous pouvons mettre en avant les paramètres suivants:

La reconnaissance est de type « Keyword Spotting » elle ne reconnaît que des mots simples et non des phrases entières.

La reconnaissance est « User-Independent » la voix de tous les joueurs doit pouvoir être reconnue (pour autant qu'elle ne soit pas trop particulière).

Les mots connus par le système sont codés en dure dans le programme. C'est une façon de faire très simpliste et peu flexible mais qui à été simple et rapide à mettre en place. Une amélioration que nous aurions pu apporter aurait été de stocker les mots à chercher ainsi que les mots utilisés comme commande dans un fichier XML. Nous avons décider de ne pas le faire, faute de temps.

5.2 Synthèse vocale

Dans notre jeu, ce système met en avant un aspect un peut inattendu de cette modalité en permettant de communiquer une information à un joueur tout en la cachant aux autres utilisateurs. En effet, l'utilisation d'un casque par le dessinateur empêche tout autre

personne de connaître le mot à dessiner.

Comme nous utilisons la synthèse vocale de Windows, suivant la version installé, la langue français n'est pas reconnue. La version utilisée pour le développement, les tests et la démonstration du jeu étant une version française de Windows 7, la sythèse vocale est en français.

6 Configuration

6.1 Voix

Bien que l'outil de reconnaissance et de synthèse vocale de Windows 7 est totalement fonctionnel pour notre application, il ne dispose que d'une synthèse vocale en anglais. Il est donc nécessaire, avant de commencer, d'installer deux programmes qui permettront d'obtenir une voix de synthèse en français.

Par la suite, pour utiliser les librairies de reconnaissance et de synthèse vocale de Windows 7 il faut importer les librairies dans le projet.

Les deux points suivants décrivent comment procéder pour obtenir un l'environnement nécessaire au bon fonctionnement de notre jeu.

6.1.1 Installation des programmes

Pour pouvoir utiliser la synthèse vocale de Windows 7 avec une voix française, il faut installer les deux programmes suivants:

- **Microsoft Reader 2.1** : MSReaderSetupFRA.exe (disponible sur le CD dans le dossier « Programmes » ou sur le site internet de Microsoft à l'adresse suivante: *http://www.microsoft.com/reader/fr/downloads/pc.mspx*
- **Microsoft Reader TTS Français :** ReaderTTSInstallFRA.exe (disponible sur le CD dans le dossier « Programmes » ou sur le site internet de Microsoft à l'adresse *suivante:*http://www.zebulon.fr/files/ReaderTTSInstallFRA.exe

Une fois ces deux programmes installés, vous pourvez vous rendres dans le panneau de configuration sous « Voice recognition » et sélectionner « Voice Synthesis » dans le menu de droite.

Dans ce menu, cliquer sur l'onglet « Voice synthesis » et sélectionner la voix « LH Pierre ».

Désormais, la vois de synthèse de l'ordinateur sera une voix française.

6.1.2 Installation des librairies

La librairie utilisé pour la synthèse et pour la reconnaissance vocale est la librairie fournie par Windows 7:

System.Speech

Pour l'utiliser dans l'application, il faut ajouter une référence vers cette librairie en procédant de la manière suivante:

1. clic droit sur le nom du projet dans l'explorateur de solution
2. Sélectionner « Ajouter une référence »
3. Dans l'onget « .Net »
4. Retrouver la librairie « System.Speech »
5. Cliquer sur « OK »

Dans les classes utilisant la synthèse ou la reconnaissance vocale (classe Voice.cs dans le projet Pictionnarwii) il faut importer les objet suivants:

- System.Speech.Recognition;
- System.Speech.Synthesys;

Les objets de reconnaissance vocale sont désormais accessibles.

6.2 Configuration de l'oreillette-micro

L'oreillette – micro est fournie avec une clé USB qui permet à l'oreillette d'être reconnue. La communication entre l'ordinateur se fait par bluetooth.

Pour commencer, brancher l'adaptateur Bluetooth – USB dans un port USB de l'ordinateur.

Le PC reconnaît et installe automatiquement les drivers.

Une fois que l'appareil est installer, il faut configurer le PC pour qu'il utilise uniquement l'oreillette pour la sortie audio (de manière à empêcher les joueurs d'entendre le mot caché) et qu'utilise les entrées audio de l'oreillette et du micro de l'ordinateur pour permettre à tous les joueurs de parler.

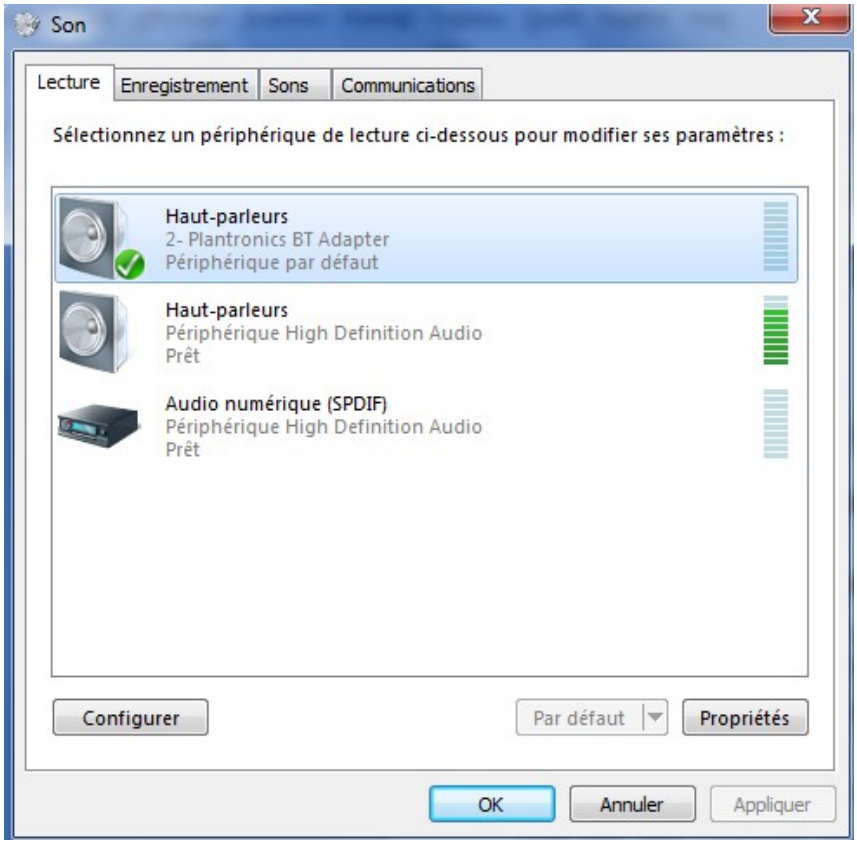
Ouvrir le panneau de configuration

Aller Sous le panneau « Reconnaissance vocale »

Dans le menu de droite se trouve un lien vers la fenêtre des « options vocales avancées »

La fenêtre qui s'ouvre, cliquer sur le bouton « Entrées audio »

La fenêtre suivante s'ouvre:



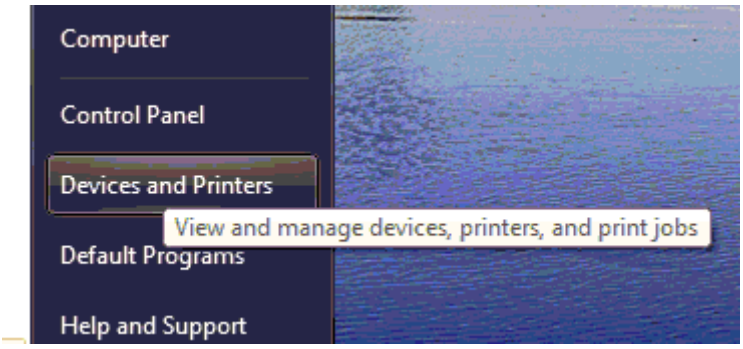
Le haut parleurs BT Adapter doit être configurer comme étant le périphérique à utiliser pour la voix. Si vous disposez d'un micro sur l'ordinateur, il peut également être utiliser comme entrée vocale pour améliorer la reconnaissance des autres joueurs.

Enfin déterminer l'oreillette comme sortie par défaut en coupant le son de la sortie de l'ordinateur.

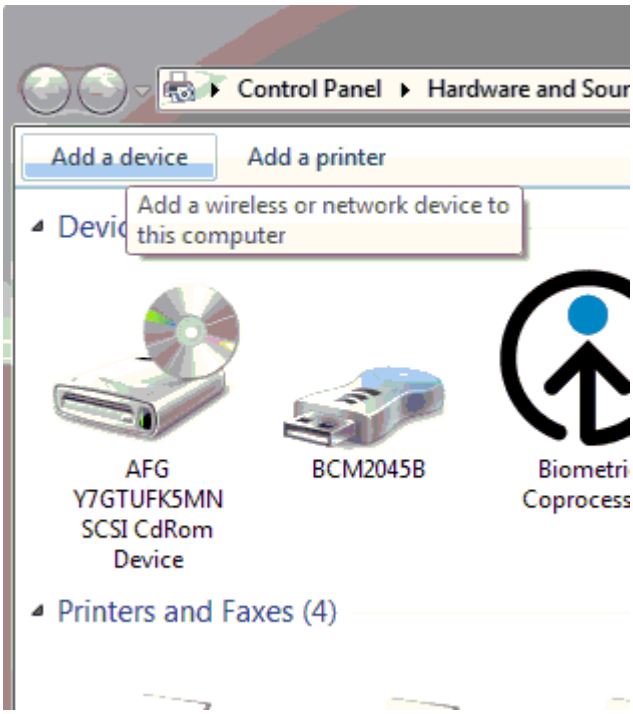
6.3 Geste – Wii-mote

Aucune librairie particulière n'est requise pour utiliser la wii-mote avec notre application. Cependant une interface *bluetooth* est requise pour se connecter à la wii-mote.

Avec Windows 7, il suffit d'ouvrir les périquériques depuis le menu démarrer :

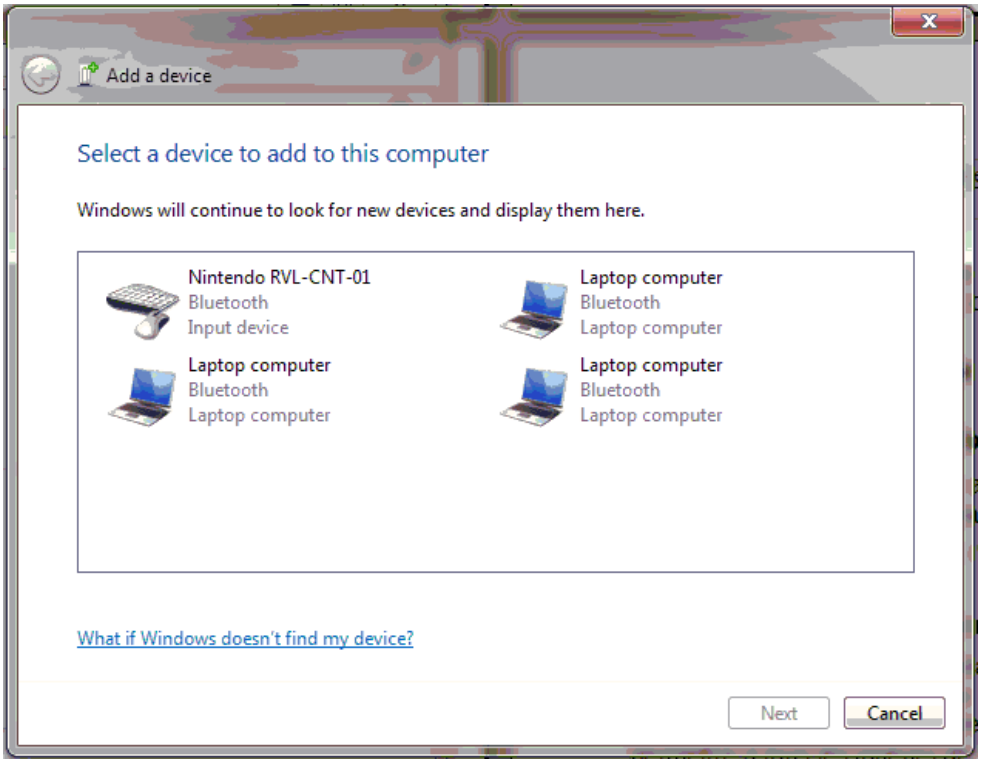


Puis de cliquer sur « Add a device » :

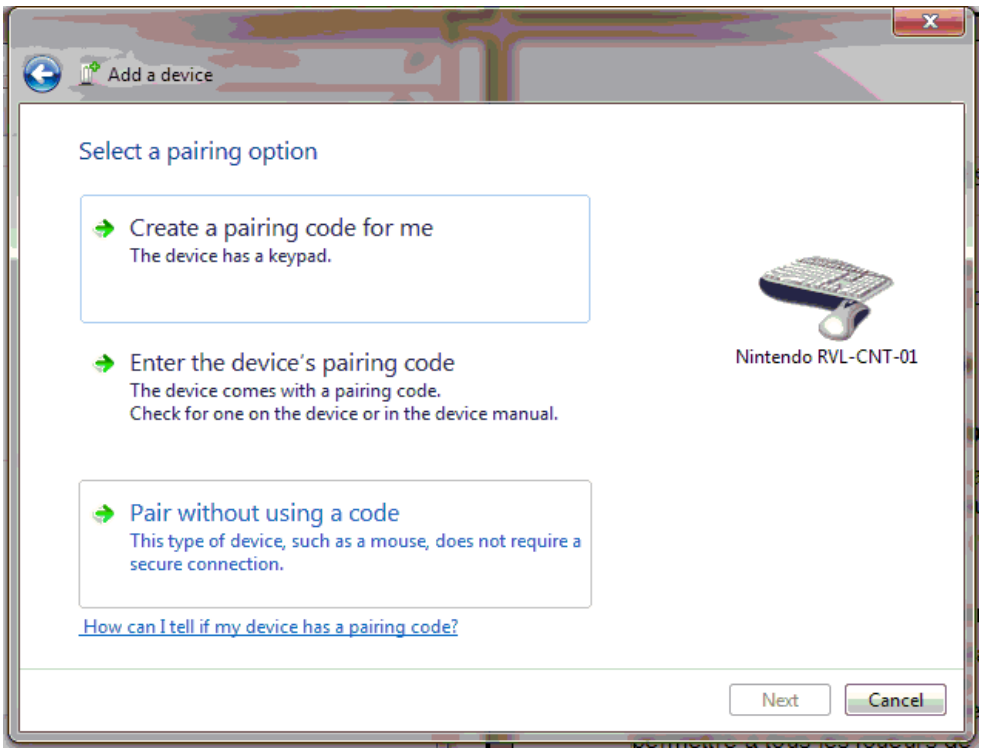


Ensuite il suffit de laisser les deux boutons « 1 » et « 2 » de la wii-mote appuyer. La wii-mote est ensuite détectée. Il faut cependant ne pas relâcher les deux boutons avant la fin de l'opération complète.

Sélectionner la wii-mote et cliquer sur « Suivant »:



Il faut ensuite sélectionner « Pair without using a code» et attendre que le périphérique soit installé.



Relacher les boutons « 1 » et « 2 » et la wii-mote est maintenant operationnelle.

7 Choix de conception

Nous avons longtemps réfléchis sur quel framework nous allions développer notre application multimodale. Ayant tous de bonnes connaissances en Java nous aurions pu choisir ce langage afin de développer notre application. De nombreuses librairies y étant disponibles, notre projet aurait donc été tout à fait réalisable dans ce language.

Nous avons eu une introduction au langage C# et au framework .Net et effectué quelques travaux pratiques dans ce langage durant le cours « Systèmes d'information » du premier semestre de la troisième année. Nos connaissances étaient donc très limitées dans ce framework et dans le langage C#.

C'est ce qui nous a motivé à utiliser ce framework, l'envie d'apprendre et de découvrir un nouveau language et environnement de développement. Nous avons donc choisi d'utiliser le framework .Net et de développer notre application sous visual studio 2008 en C# dans le but d'étendre nos connaissances.

En plus en se penchant sur les librairies que nous devrions utiliser, nous avons constaté que la librairie pour la reconnaissance vocal disponible en C# était plus performante que la librairie disponible pour Java.

Notre choix de se risquer dans une nouvelle technologie à donc été judicieux.

8 Configuration du matériel

Les appareils externe utilisés par notre jeu sont:

Une manette Wii (wiimote)

Une oreillette-micro

Ces dispositifs nécessite d'être configurer pour être reconnus par le système. Ce chapitre décrit pas à pas comment les configurer

9 Conclusion

Nous avons vraiment eu beaucoup de plaisir autant à travailler ensemble qu'à utiliser ce nouveau Framework avec lequel nous manquions de pratique. Le projet c'est donc bien déroulé puisque que nous avons pû trouver des solutions à tous les problèmes rencontrés et car l'application finale répond à la déscription qui en avait été faite tout en étant très simple et conviviale à utiliser pour des utilisateurs inexpérimentés.

10 Remerciements

Merci à M. Nicolas Pittet pour son aide pour le développement et l'utilisation de la librairie pour la wii-mote.