

Mini-projet : projet KiBoom

Cours d'interfaces multimodales, rapport de projet

Bachelor EIA-FR, I3

Frédéric Monney, Guillaume Schneuwly

Février - Avril 2011

Table des matières

1.	Contexte	3
2.	Description du mini-jeu	3
3.	Modalités.....	3
4.	Maquettes	4
5.	Modèle CASE	4
6.	Fusion	5
7.	Fission.....	5
8.	Choix de conception.....	6
9.	Caractéristiques techniques	6
10.	Librairies utilisées	7
11.	Le jeu	7
12.	Conclusion	10
13.	Annexes	10

1. Contexte

Le projet KiBoom est né suite à la demande des professeurs du cours d'interfaces multimodales de réaliser un mini-projet. Ce dernier a pour but de mettre en pratique la théorie du cours. Ce projet doit respecter le fait d'utiliser plusieurs modalités pour interagir avec le système, qui, cette année, doit être un mini-jeu (ou serious game).

Les auteurs de ce projet sont deux étudiants de l'EIA-FR en dernière année d'étude de Bachelor en informatique.

Le matériel est mis à disposition par l'école.

2. Description du mini-jeu

Notre application fait apparaître des formes virtuelles sur un écran de jeu. Ces formes sont d'une certaine couleur et ont également des designs différents.

Le joueur pointe du doigt une forme et dit sa couleur pour la faire disparaître (exploser). Chaque disparition donne un certain nombre de points et le but du jeu est d'amasser un maximum de points.

Une fois qu'un certain nombre de formes sont apparues, le jeu se termine. Les formes disparaissent d'elle-même au bout d'un moment.

3. Modalités

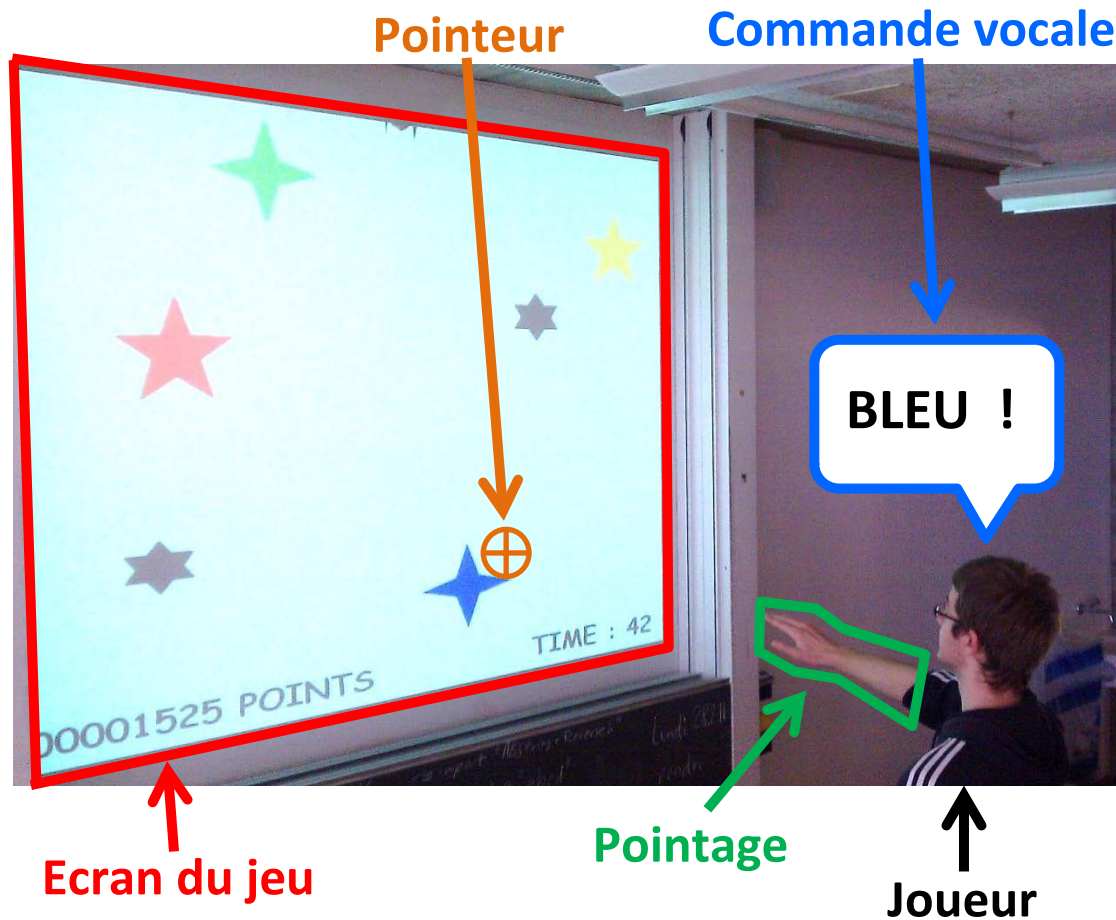
Nous avons décidé d'utiliser les deux modalités suivantes pour contrôler notre application :

- Voix (pour dire le nom des couleurs)
- Gestes (pour le dispositif de pointage)

La combinaison des deux modalités crée un événement (explosion d'une forme).

4. Maquettes

Voici une maquette qui nous donne une bonne idée du résultat finale de l'application :



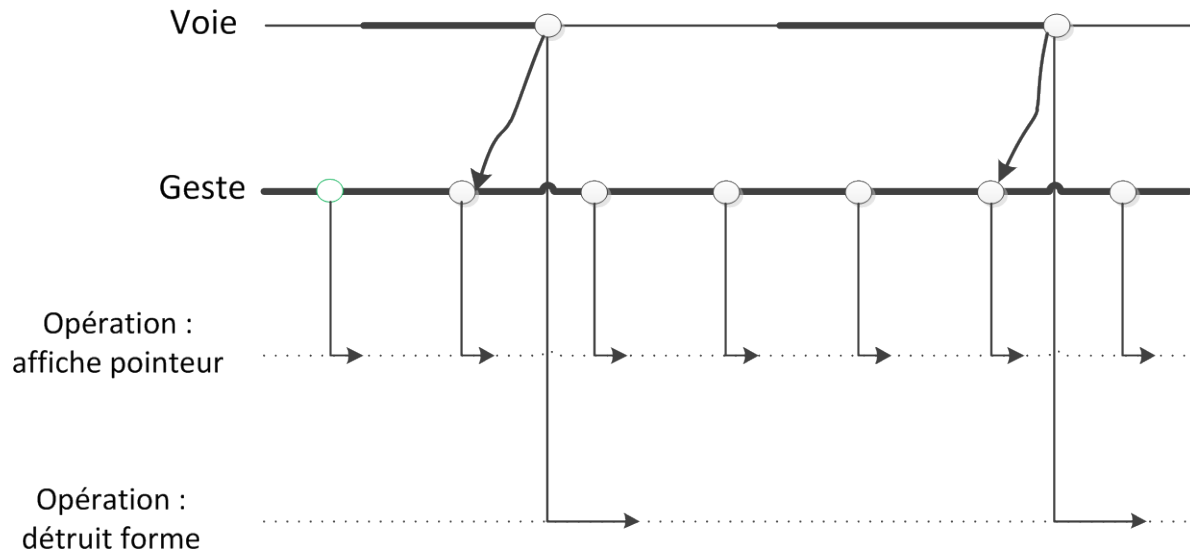
Ici le joueur manque la forme (le pointeur n'est pas dessus).

5. Modèle CASE

Selon le modèle CASE (réf. cours), notre application se classe dans le domaine synergique. En effet, nous capturons constamment la position du pointeur ET nous capturons sans cesse la voie pour une éventuelle reconnaissance.

6. Fusion

Si un nom de couleur est reconnu par notre application, elle va rechercher la position courante du pointeur pour proposer la destruction d'une forme au contrôleur du jeu. L'information de la couleur combinée à celle de la position donne une commande de destruction.



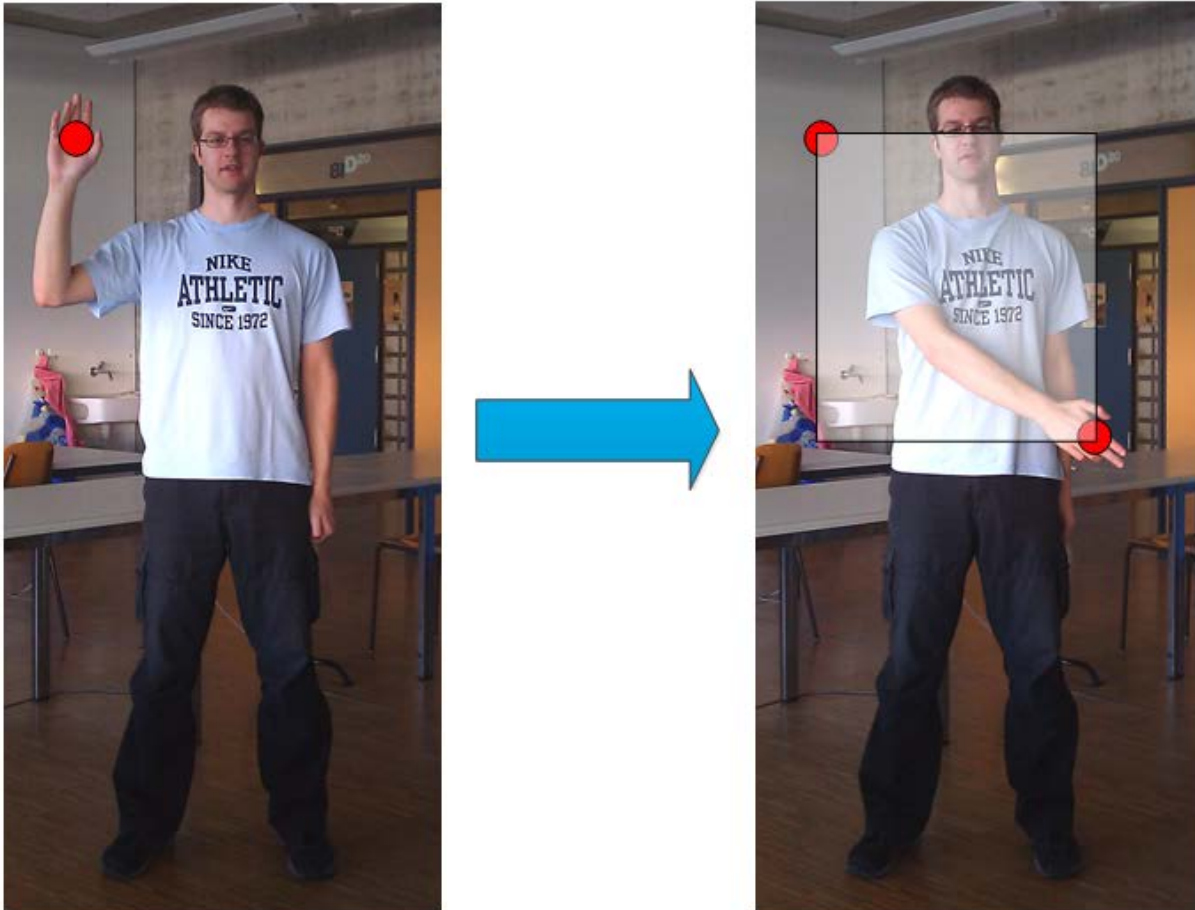
7. Fission

Nous utilisons uniquement un écran pour l'affichage de l'information.

8. Choix de conception

Pour la **voie**, nous utilisons un vocabulaire très restreint (6 noms de couleur + 2-3 commandes pour les menus). Il est ainsi plus rapide et efficace de reconnaître la voie.

Pour le **pointage**, nous utilisons la position de la main relative à une zone d'action préparée lors de la phase de calibration qui précède la phase de jeu.



9. Caractéristiques techniques

- Utilisation de la reconnaissance vocale de Windows 7, matériel : micro.
- Utilisation de la reconnaissance des mouvements d'openKinect, matériel : kinect.
- Programmation C#, environnement de développement : Visual Studio 2010.

10. Bibliothèques utilisées

OpenNI : cette bibliothèque offre une détection des membres avec la Kinect. Elle nous impose également une phase de calibration pour notre application (pour reconnaître une personne).

<http://www.openni.org/>

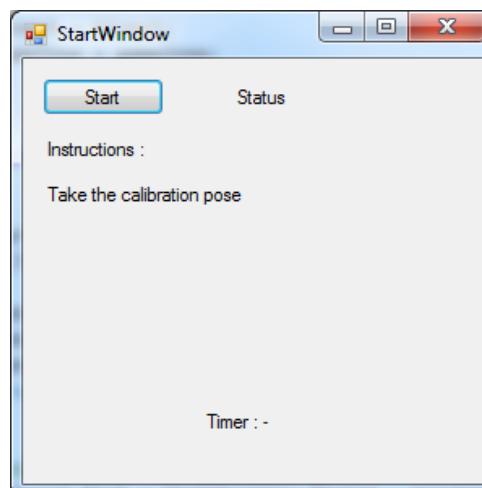
Windows 7 Speech : cette bibliothèque nous permet d'effectuer de la reconnaissance vocale dans l'environnement .Net.

[http://msdn.microsoft.com/fr-ca/library/ms695836\(v=VS.85\).aspx](http://msdn.microsoft.com/fr-ca/library/ms695836(v=VS.85).aspx)

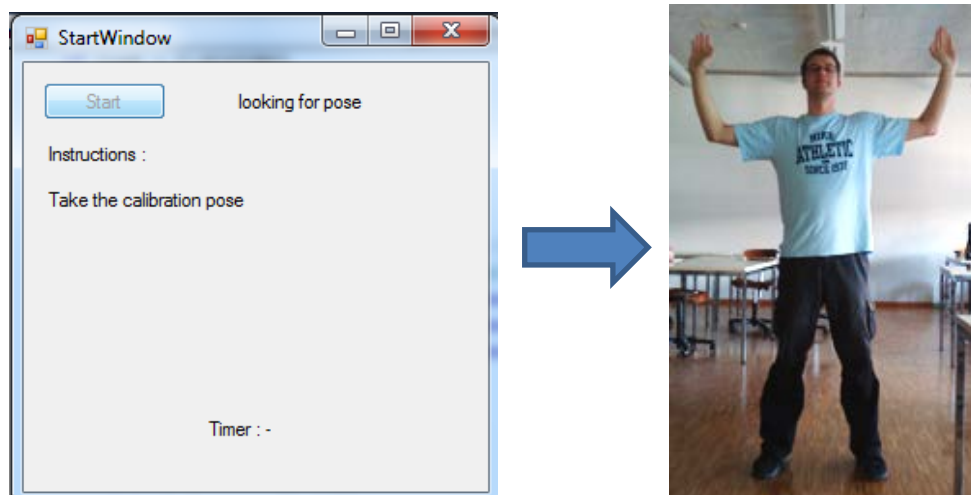
11. Le jeu

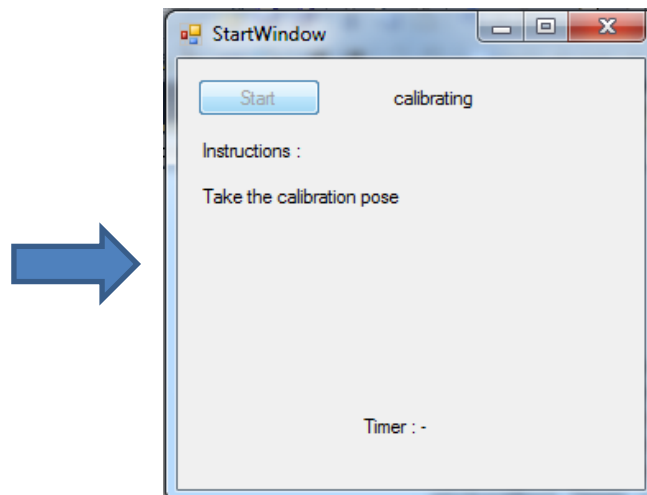
Pour exécuter l'application, il suffit d'exécuter le programme 'KiBoom.exe' qui se trouve dans le répertoire installé (voir 'Manuel d'installation').

Le premier écran attend que l'utilisateur appuie sur start.

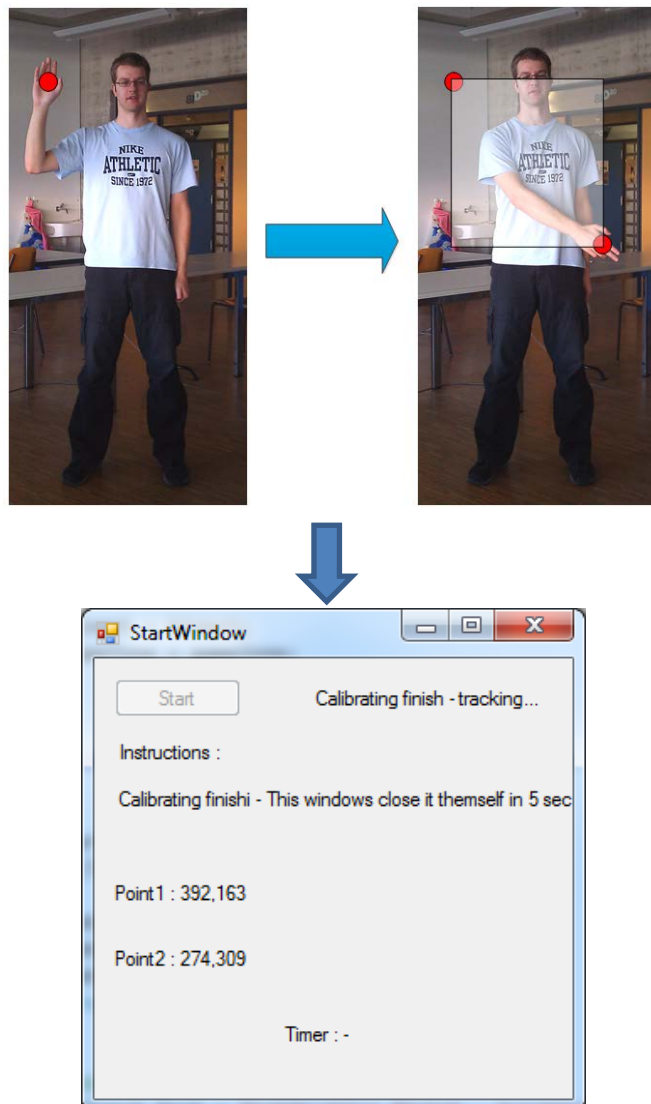


La phase suivante est celle de calibration, l'utilisateur doit se mettre dans une position bien précise et attendre que le système le reconnaisse.

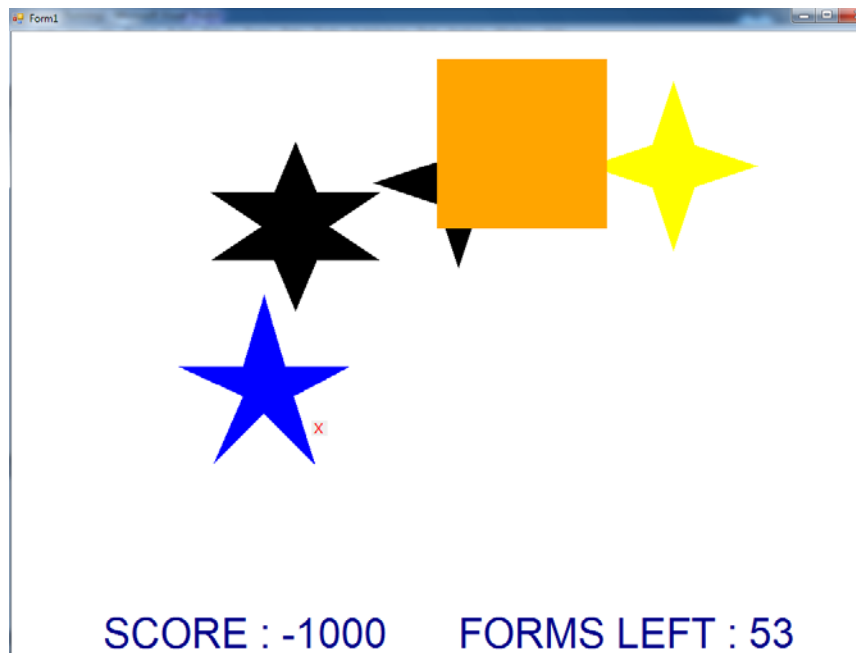




Une fois l'utilisateur reconnu, la phase de détermination de la zone d'action suit. Pour déterminer cette zone, l'utilisateur met sa main droite en haut à droite, puis en bas à gauche.



Une fois la zone d'action utilisateur définit, le jeu commence !



L'utilisateur pointe les formes avec sa main droite et dit leur couleur pour les éliminer et faire le maximum de points, génial !



Une fois le jeu terminé, il est possible de relancer une partie.

12. Conclusion

Nous avons réalisé la phase de notre application sans trop de problème. Cependant, il reste quelque point à améliorer :

- Les threads, pour augmenter la réactivité
- Gestion du joueur (voir plusieurs joueurs)
- Meilleure gestion du temps
- Meilleure gestion des formes
- Ajout d'animations
- Gestion des scores
- Développer pour console XBOX 360

Ce projet était intéressant car il nous a permis de découvrir de nouvelle façon d'interagir avec un programme. Par contre nous trouvons dommage que le temps mis à disposition ne soit pas plus grand. Nous pensons qu'avec plus de temps les projets pourraient être mieux développés et donc plus intéressants.

13. Annexes

- CD avec l'ensemble des fichiers du projet
 - Rapport (Rapport_KiBoom.pdf)
 - Manuel D'installation (Manuel_installation.pdf)
 - Vidéo du programme (video_KiBoom.avi)
 - Descriptif de la vidéo (Descriptif_video.pdf)
 - Présentation du projet (KiBoom_presentation.pdf)
 - Code du projet (dossier sources)
 - Autre documentation (dossier doc)
 - Jeu compilé (dossier jeu)