

Rapport NF28

Projet P2MM

Laura Daras – Maëva Guerry – Valentin Hervieu – Vincent Mercier



P2013

Sommaire

Introduction	3
I - Présentation du projet	4
1) Contexte de l'étude	4
2) Kinect	4
3) Positionnement par rapport au projet « La Séparation » (PR/TX)	5
II - Etude Utilisateurs Et Concurrence	5
1) Etude Utilisateurs	5
2) Etude logiciel concurrent, Positionnement	6
The V Motion Project	6
Collaborative Kinect Puppet	7
Spectacle interactif avec enfants	8
III - Fonctionnalités	9
1) Gestes	9
2) Interface Graphique	9
3) Organisation de la scène	10
4) Relation Gestes-Action	11
IV- Architecture de l'application	12
1) Kinect	12
2) WPF	13
3) Structures de données et autres éléments importants	14
Architecture du code	14
4) Modifications apportées	16
V - Évaluation auprès des utilisateurs (User Feedback)	17
VI - Limites du prototype et améliorations possibles	17
1) Limites	17
2) Améliorations	18
Conclusion	19
Annexes	20
1) Sondage	20
2) Métadonnées sur les sondés	20
3) La Kinect	21
4) Monde du spectacle & Kinect	22

Introduction

L'UV NF28 présente les méthodes et les outils utilisés pour la conception, le développement et l'évaluation de systèmes interactifs. Elle aborde également les formes d'interaction homme-machine. C'est dans cette optique que nous avons la possibilité de mener un projet qui nous amènera à créer notre propre système interactif.

Dans le cadre d'une UV PR nous participons à un projet de création artistique et de développement technologique qui s'intitule « Poésie à 2 mi-mots » (P2MM). Lors d'une performance scénique, ce procédé permet notamment au performeur de jouer avec les mots sur scène, grâce à une tablette.

C'est dans ce contexte que nous avons eu l'idée de nous servir des gestes du performeur pour interagir avec les mots, au moyen d'une Kinect. D'une manière distincte de la PR, notre projet de NF28 consiste à évaluer les possibilités offertes par la Kinect pour créer des applications interactives. Doté d'une forte dimension exploratoire, l'application réalisée pourra servir de base au développement d'animations scénarisées dans le cadre de futurs projets étudiants.

Il s'agit à présent de rappeler le cahier des charges initial, afin d'exposer de quelle façon le travail réalisé a pu répondre aux exigences d'entrées, exigences ayant parfois dû être revues. Enfin, l'analyse de l'expérience de l'utilisateur permettra de mettre en place la passation du projet, en précisant les limites atteintes et des pistes afin de surmonter ces difficultés.

I - Présentation du projet

1) Contexte de l'étude

En 2013, un projet de recherche et création fait l'objet d'une collaboration entre la compagnie de spectacles ALIS (<http://www.alis-fr.com/>), le collectif numérique i-Trace (<http://i-trace.fr/>) et l'UTC. Ce projet a été labellisé par le Centre d'Innovation de l'UTC. Il s'agit de proposer une création sous deux modalités articulées (performance sur scène de poésie numérique et création en ligne).

Ce projet est fondé sur un procédé inventé par Pierre Fourny, fondateur d'ALIS, intitulé la Poésie à 2 mi-mots. Ce dernier consiste à couper les mots d'un trait horizontal et à associer les moitiés de mots obtenus à d'autres mots pour en former de nouveaux (<http://www.alis-fr.com/site/?q=node/26>).



Exemple: "mots" et "filtre" partagent la même moitié inférieure

Dans ce cadre, un groupe d'étudiants, participant à des UVs TX et PR travaillent sur la réalisation d'une application mobile permettant à Pierre Fourny de mettre en scène la poésie à 2 Mi-mots sur un support numérique, grâce à des interactions de type tactile. L'idée suivante a alors été évoquée : ne serait-il pas intéressant de pousser davantage l'expérience en introduisant une interaction avec les gestes du performeur? Le développement de cette idée dépassant les objectifs fixés pour la PR, nous avons proposé à Mr Fourny de nous y intéresser dans le cadre de NF28.

2) Kinect



La Kinect pour Windows, conçue par Microsoft, permet de contrôler des applications sans l'utilisation du moindre périphérique. Il est possible d'interagir avec les applications grâce à la détection de mouvements, de voix et d'images. Le SDK Windows, fournie par Microsoft, permet de développer de telles applications.

La Kinect se prête donc tout à fait au développement du projet. L'objectif sera donc d'utiliser les capteurs de profondeurs et de gestes afin de jouer avec des objets (mots) et leur appliquer les différents procédés de transformations utilisés par le performeur. Le projet aura avant tout un objectif exploratoire et permettra de poser les bases pour une exploitation ultérieure.

3) Positionnement par rapport au projet « La Séparation » (PR/TX)

A la différence du développement réalisé dans le cadre de ces UVs, le projet P2MM n'a pas pour but de développer une application prête pour le spectacle. Si le projet aboutit, une telle application apportera très certainement un « plus » à la représentation. En effet, nous pensons que le contrôle distant est un type d'interaction originale disposant d'une popularité toujours croissante. Le public est depuis toujours fasciné par les tours de prestidigitation, et autres effets « magiques », qu'il s'agisse de performances technologiques ou humaines (cf. sondage en annexe). L'utilisation de la Kinect pourrait rendre possible une interaction directe entre le public (voix) et l'application, et pourrait permettre de « chorégraphier » le spectacle, impliquant complètement le performeur dans les transformations qu'il présentera.

II - Etude Utilisateurs Et Concurrence

1) Etude Utilisateurs

Notre projet ne vise qu'un seul utilisateur : Pierre Fourny. Il est, à la fois, le commanditaire de notre projet et celui qui assure les performances sur scène et qui utilisera donc les fonctionnalités de notre interface.

Pierre Fourny est l'inventeur du concept de poésie à 2 mi-mots. Dans le cadre de ses spectacles, il est le seul performeur (avec toutefois une possibilité d'interaction avec le public). La mise en scène est simple, et légère. Elle est composée d'une succession de micro-spectacles, s'apparentant à des tours de magie durant lesquels le performeur présente et utilise le procédé de « mots coupés ». Il utilise les supports physiques à sa disposition pour expérimenter et jouer avec les mots, les libérer de leur sens premier, et faire des représentations visuelles parfois humoristiques. Avec sa dimension ludique, M.Fourny permet à son public de faire travailler son imagination et sa curiosité.

Il est possible de voir une présentation en ligne du projet, ainsi qu'un extrait d'une performance réalisée par Pierre Fourny à l'attention des étudiants, à l'utc, à cette adresse: http://webtv.utc.fr/watch_video.php?v=2M8DS67O9WHN

Dans le cadre de ce projet, M.Fourny pourra interagir avec les mots, sur scène, en utilisant la Kinect et les interactions exploitables sur scène. Ainsi, une partie de sa performance scénique pourra, à terme, s'appuyer sur les opportunités qu'offre la Kinect.

Nous avons eu l'occasion de nous entretenir avec lui par téléphone, pour connaître son point de vue sur le projet. Très enthousiaste et impliqué, il nous a expliqué ce qu'il attendait de nous et le côté exploratoire de notre étude de la Kinect. Il a également fixé une

liste (non exhaustive) des différents gestes, associés à une transformation de mots, qu'il souhaitait pouvoir utiliser avec la Kinect.

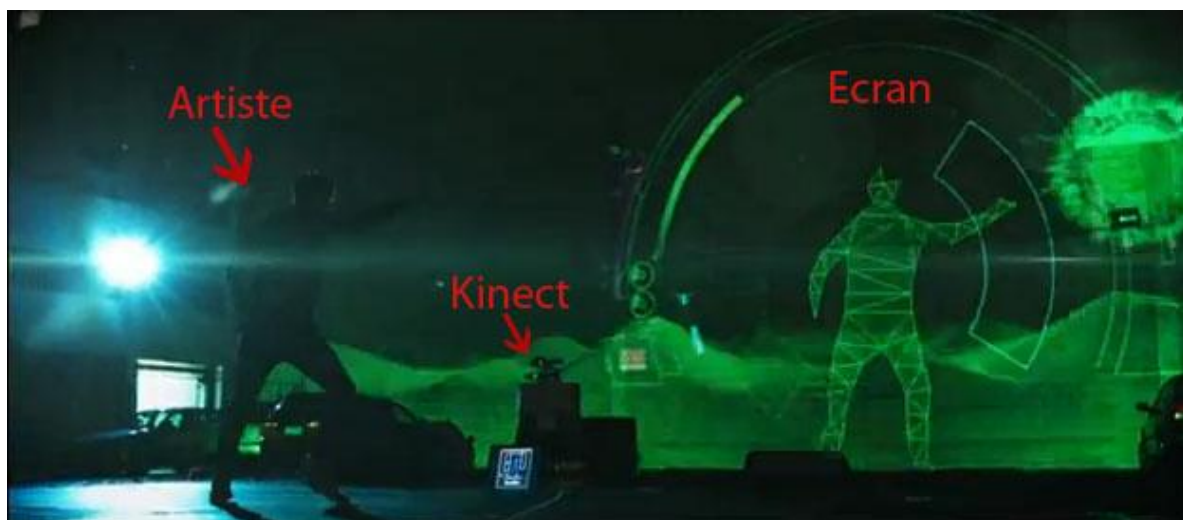
2) Etude logiciel concurrent, Positionnement

Le but de cette partie est d'étudier les logiciels concurrents afin de se rendre compte si notre projet a déjà été réalisé par d'autres équipes et donc si ce projet reste intéressant. Cependant, dû à la nature de notre projet, cette partie consiste plutôt en une étude de l'existant afin de voir ce qu'il est possible de réaliser en utilisant la Kinect dans le cadre d'un spectacle. En effet, on ne peut pas vraiment parler de concurrence dans le domaine du spectacle puisque qu'ici le logiciel développé ne sera qu'un outil d'aide à la performance.

Nous avons donc recherché différents exemples d'utilisation de la Kinect sur une scène de spectacle et comme nous l'imaginions, nous nous sommes aperçus qu'il n'y avait que peu d'exemples se rapprochant de notre idée. En effet, la Kinect est un petit peu utilisée pour certains spectacles, mais tous ces spectacles sont principalement basés sur des performances musicales et dansantes. Cependant, il est intéressant d'étudier la position de la Kinect sur la scène de ces différents exemples puisque c'est une question que nous allons devoir traiter pour notre projet. De plus, ces exemples permettent de se rendre compte des différents interactions possibles avec la Kinect.

Nous avons sélectionné trois exemples intéressants d'utilisation de la Kinect dans un cadre se rapprochant de celui de notre projet.

The V Motion Project



Architecture de la scène

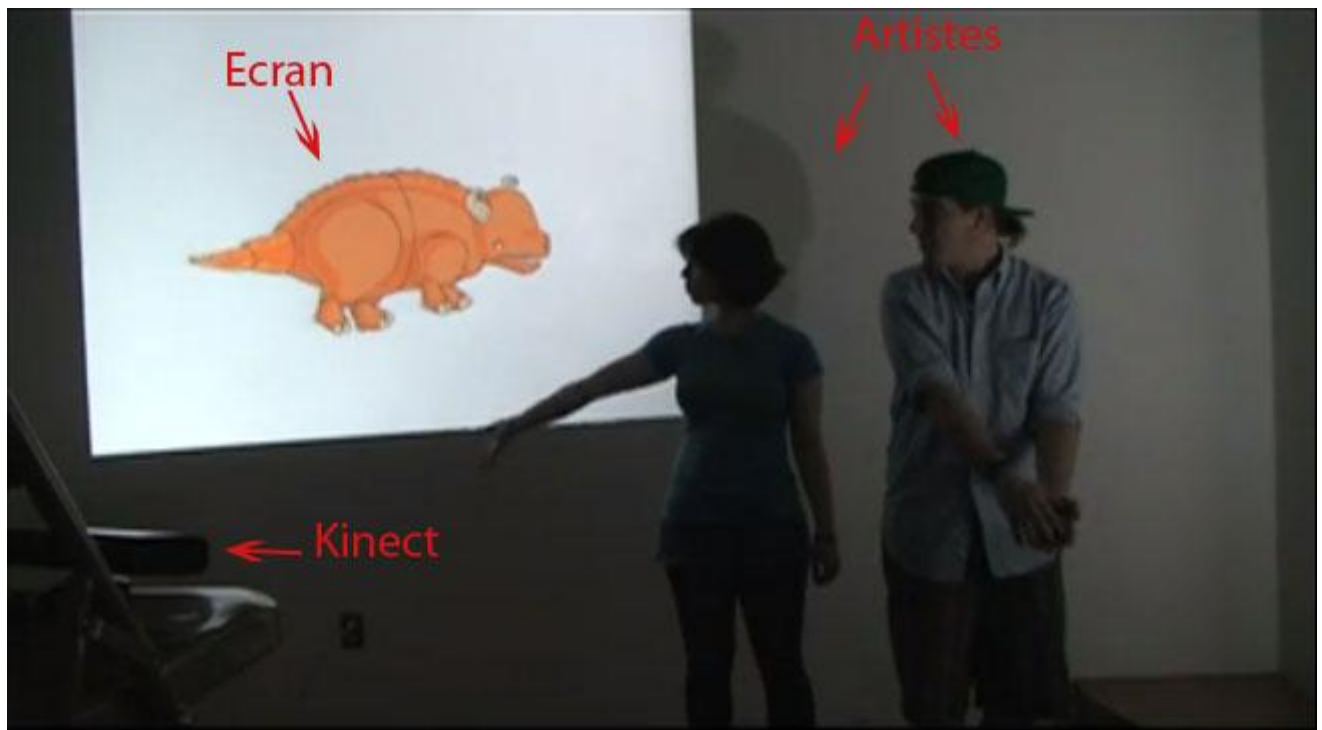
Lien du projet : <http://fugitive.co.nz/v-motion-project-tech-how-we-built-it>

Ce projet consiste à créer de la musique DubStep à partir des gestes de l'artiste. L'artiste a alors une interface à l'écran avec des zones sensibles qui réagissent lorsque ce dernier y place une des ses mains. De plus, il peut changer de panel de sons simplement en faisant un mouvement de balayage avec son bras. Il semble qu'il y ait également des interactions avec ses pieds afin de modifier les différents sons.

D'un point de vue technique, ce projet utilise en fait deux Kinects, une reliée à un PC qui s'occupe de la partie de génération de musique, et l'autre, reliée à un Mac qui s'occupe de la partie de génération des effets visuels.

Au niveau de l'architecture de la scène, l'artiste fait face à l'écran (et tourne donc le dos au public), et la Kinect se trouve face à lui.

Collaborative Kinect Puppet



Architecture de la scène

Vidéo du spectacle: http://www.youtube.com/watch?v=OA_4rAWwYdq

Dans ce spectacle, deux artistes contrôlent une marionnette virtuelle à travers leurs gestes de façon collaborative.

Au niveau de l'architecture de la scène, les artistes sont dos à l'écran et la Kinect leur fait face en étant décalée sur le côté. On voit d'ailleurs sur l'image que les artistes sont obligés de tourner la tête pour suivre la marionnette du regard, ce qui n'est pas très agréable pour le public.

Spectacle interactif avec enfants



Vidéo du spectacle: <http://techcrunch.com/2011/02/23/little-magic-stories-interactive-art-with-the-kinect/>

Ce spectacle met en scène des enfants qui interagissent avec un environnement virtuel qui est représenté grâce à une projection d'hologrammes sur un écran de verre transparent. Les enfants peuvent créer (en dessinant sur des feuilles qui seront numérisées) et interagir avec leurs propres personnages, attraper des œufs de Pâques, embrasser des poissons ou courir sur la scène.

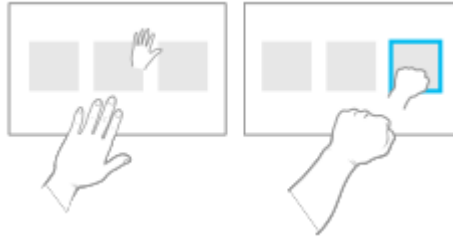
Au niveau de l'architecture, ce spectacle est un peu différent de ceux présentés précédemment et du spectacle qui amène notre projet, à cause de la nature de l'écran. Cependant, on peut noter que les enfants font face au public et que la Kinect est face à eux (même si celle-ci est dissimulée).

Finalement, on peut voir qu'il existe des spectacles utilisant la Kinect et présentant un artiste interagissant avec un environnement virtuel. Il est intéressant de voir que les trois exemples utilisés ici utilisent une architecture de scène différente (position artiste - écran - Kinect). Notre projet pourrait se rapprocher du deuxième exemple puisque l'artiste devra interagir avec des objets à l'écran. Cependant, il sera intéressant de prendre en compte les différents gestes utilisés par l'artiste sur le premier spectacle afin de naviguer dans la scène virtuelle. Enfin, la position des enfants sur la scène se rapproche de celle que nous aimerions mettre en place dans le cadre de ce projet.

III - Fonctionnalités

1) Gestes

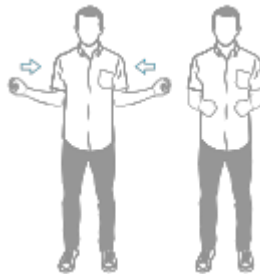
Plusieurs gestes de base dont l'exploitation semble intéressante à Pierre Fourny, notre client, ont été retenus :



- sélection et distinction d'un objet



- translation horizontale et verticale
- rotation



- jeu avec la profondeur, zoom (effets de perspective)

Il ne s'agit pas d'une liste exhaustive de gestes, mais plutôt de gestes de bases pour effectuer des manipulations de bases. Ainsi, de nouveaux gestes, qui seront potentiellement des gestes composés des gestes définis ci-dessus, feront leur apparition pendant la réalisation du projet.

2) Interface Graphique

L'idée est de centrer le développement sur les interactions, essentiellement parce que le développement ne doit pas aboutir à une application finie et exploitable pour la scène, mais consiste plutôt en une étude exploratoire de ce qu'il est possible de faire. Il convient de réfléchir au bon usage de l'interface graphique dans la mesure où notre client envisage ce projet comme un moyen de considérer l'apport de la technologie Kinect à son travail de performeur sur scène. Dans ce cadre, consacrer une partie importante du temps de développement à des problématiques purement esthétiques ou plastiques semble peu utile. Quoi qu'il en soit, l'idée est de centrer la concentration du spectateur sur l'action du performeur sur l'écran video-projeté sur scène. C'est cette interaction et sa sémantique qui donnera une dimension divertissante au spectacle.

Il convient donc de mettre en avant deux choses: le geste du performeur (jeu de lumière sur le performeur) et l'action qui en résulte sur le mot. De ce fait, l'interface graphique de l'application video-projetée sera minimaliste. Elle ne contiendra pas de boutons, pas de clavier virtuel, juste un écran noir et éventuellement un ou deux pointeurs discrets permettant de visualiser les positions relatives des mains de l'artiste afin de faciliter l'action du performeur.

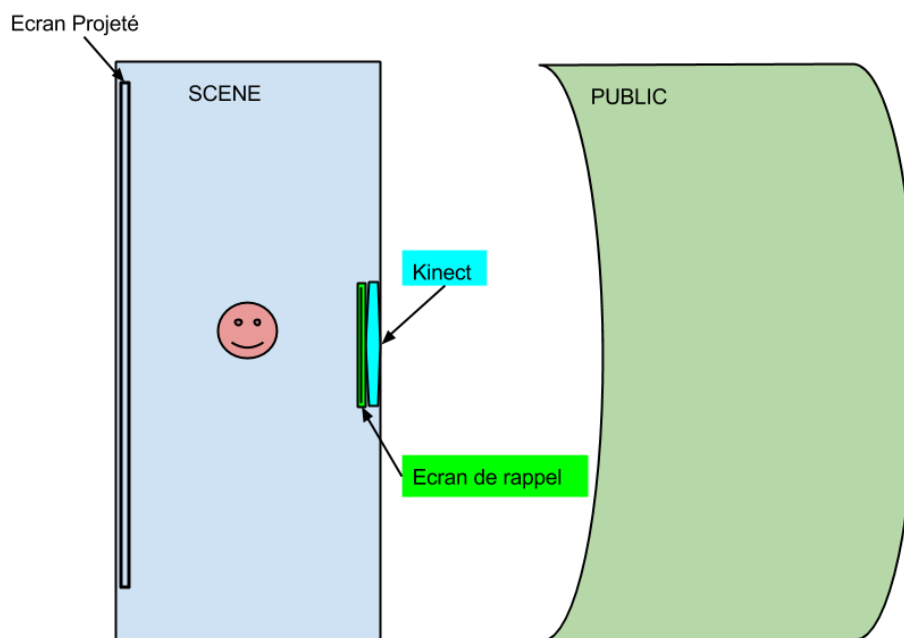
3- Organisation de la scène

Les trois contraintes majeures pour l'utilisation de la Kinect pour interagir sur une image projetée sont:

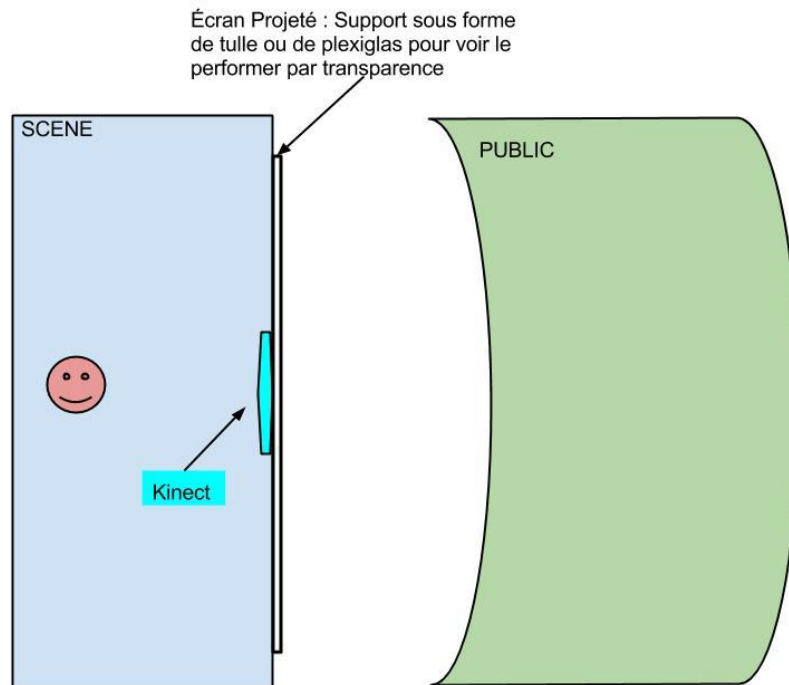
- la Kinect doit être placée devant le performeur pour capter ses gestes.
- le performeur doit être devant l'écran pour que le spectateur voie le lien direct entre le geste et l'action sur le mot. Dans la mesure où l'écran est projeté sur le mur du fond de la scène, cela implique que le performeur soit dos au mur.
- un écran de rappel doit être placé au-dessus de la Kinect pour que le performeur ait un retour sur ce qu'il fait s'il ne voit pas directement l'écran principal (notamment s'il est dos à l'écran principale)

De ces trois contraintes découlent plusieurs modèles d'organisation spatiale de la scène.

Modèle n°1



Dans ce cas de figure, le performeur fait dos à l'écran principal. Il lui faut donc un petit écran de rappel dont l'affichage a été inversé pour correspondre à ce que le performeur réalise à l'aide de ces gestes.



Dans ce cas de figure, le performeur est derrière l'écran projeté. Il a donc un regard direct sur ce qu'il fait. Il n'y a pas besoin d'écran de rappel.

4) Relation Gestes-Action

Premières propositions:

Lors de sa représentation, l'artiste, passera d'une action à l'autre. Cette transition pourrait s'effectuer par un geste de translation horizontale.

Les mots manipulés pourraient être effacés par un geste de balayement de la main comme pour éponger ou gommer de la main le mot.

Le pointeur permettra de savoir la position relative de la main du performeur sur l'écran. En fermant le point, le performeur pourra sélectionner le mot pour le déplacer.

Les gestes nécessaires pour manipuler les mots selon les différents processus (retournements, polices miroirs, polices coupables...) sont encore à définir. Il s'agit en réalité d'une partie du sujet.

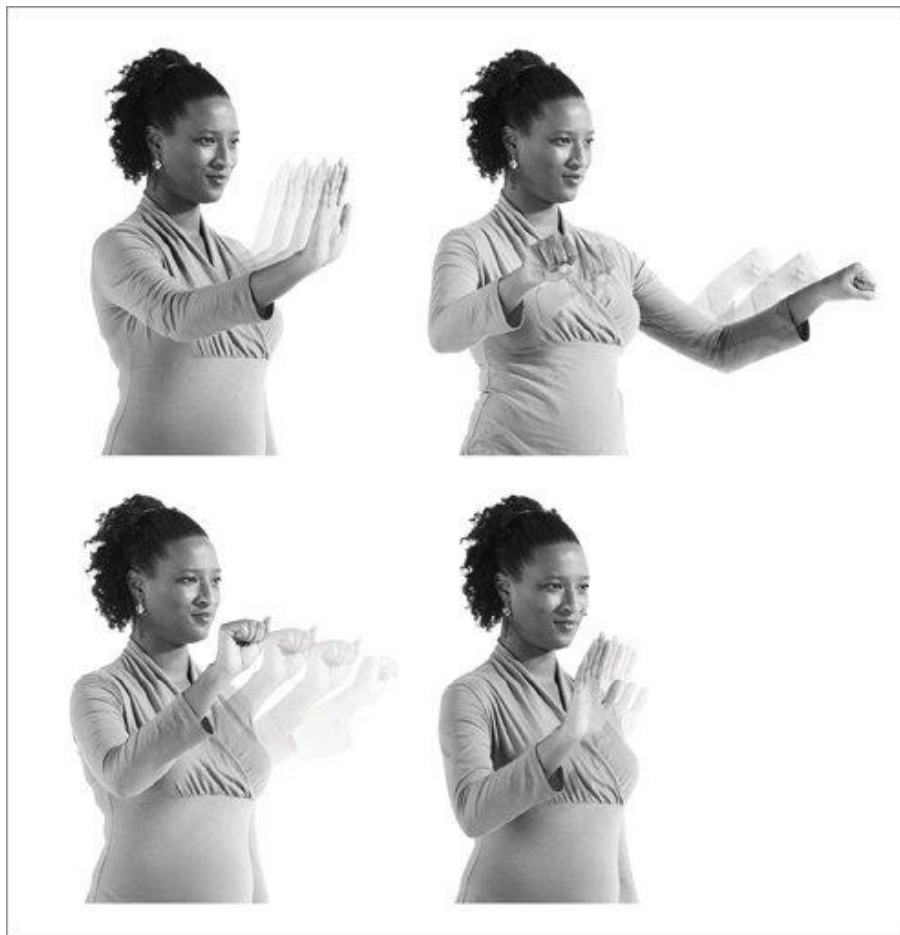
IV- Architecture de l'application

L'application que nous avons développée est une application Windows. En effet, nous utilisons la Kinect pour détecter les mouvements de l'utilisateur et la bibliothèque WPF (Windows Presentation Foundation) qui permet de créer des interfaces graphiques pour Windows. Nous allons ici détailler l'architecture de notre application par rapport à ces deux composants.

1) Kinect

Afin de pouvoir utiliser la Kinect depuis notre application, nous devons utiliser le SDK Kinect (version 1.7) afin d'accéder aux interactions de base. De plus, nous avons utilisé le Developer Toolkit pour la Kinect afin d'accéder à des bibliothèques de plus haut niveau.

En effet, de base, le SDK permet seulement de suivre des points-clés du corps de l'utilisateur tels que les mains, les coudes, les hanches ou les genoux mais ne permet pas de détecter des gestes plus complexes tels qu'une main fermée. Le Toolkit fournit alors une bibliothèque nommée Kinect Interaction (présente sous l'espace de nom Microsoft.Kinect.Toolkit.Interaction) qui permet de différencier une main ouverte d'une main fermée, ainsi qu'une main avancée vers l'écran ou non. Ce sont d'ailleurs ces gestes de base qui ont dicté l'ergonomie de notre application.

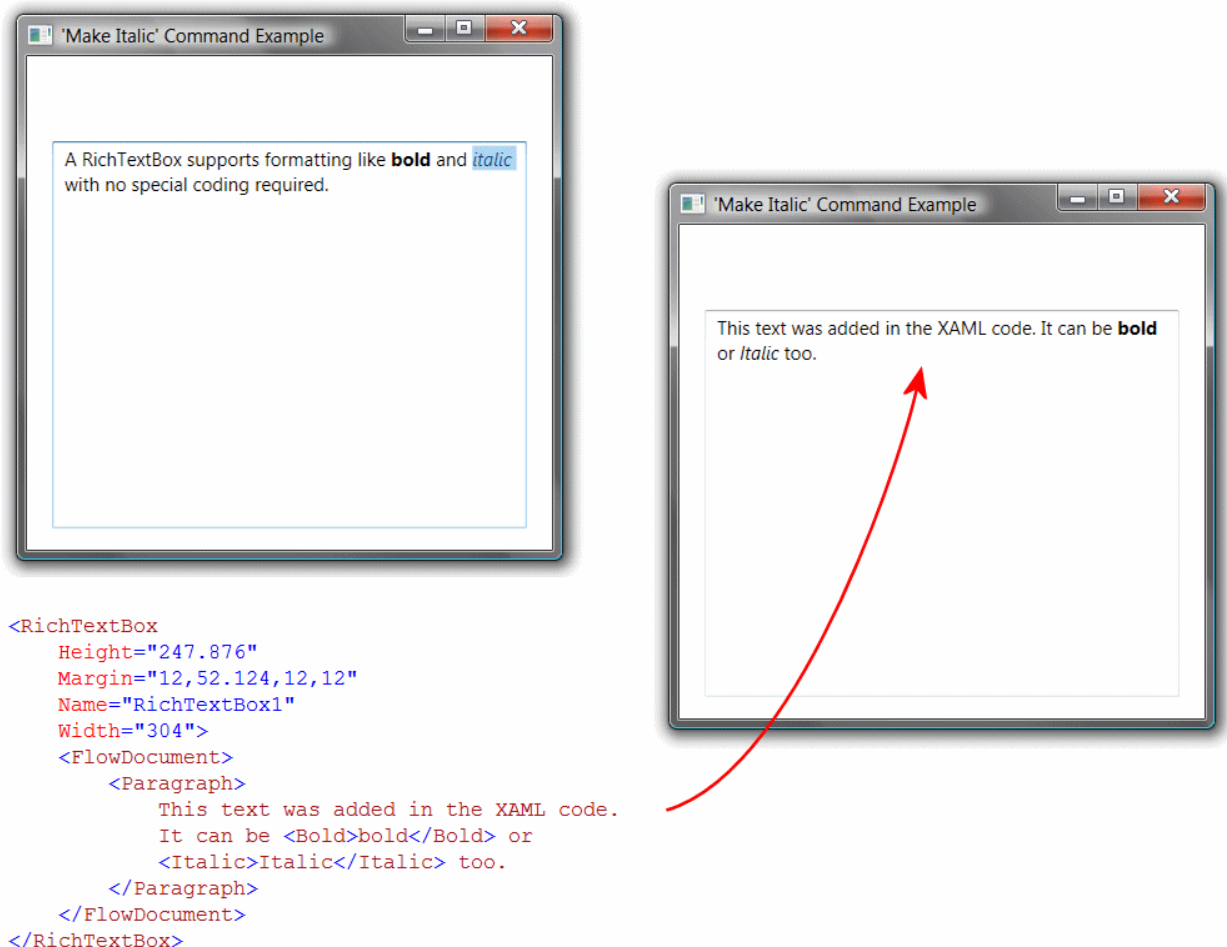


Finalement, afin de réaliser notre application, nous n'avons eu qu'à suivre les gestes des mains et non toutes les parties du corps.

2) WPF

Notre application a été développée en C# au sein du framework .NET de Microsoft et en utilisant la bibliothèque d'interface graphique WPF. Cette bibliothèque a la particularité de permettre de créer des interfaces graphiques facilement et sans coder en passant par une description XML (fichier .xaml dans WPF). Ensuite, chaque fichier XAML a une classe C# attachée (héritant de la classe mère Window) afin d'écrire la partie contrôleur et modèle de l'application. La classe C# est d'ailleurs appelée "classe partielle" car elle est complétée à l'exécution par les objets décrits dans le XAML.

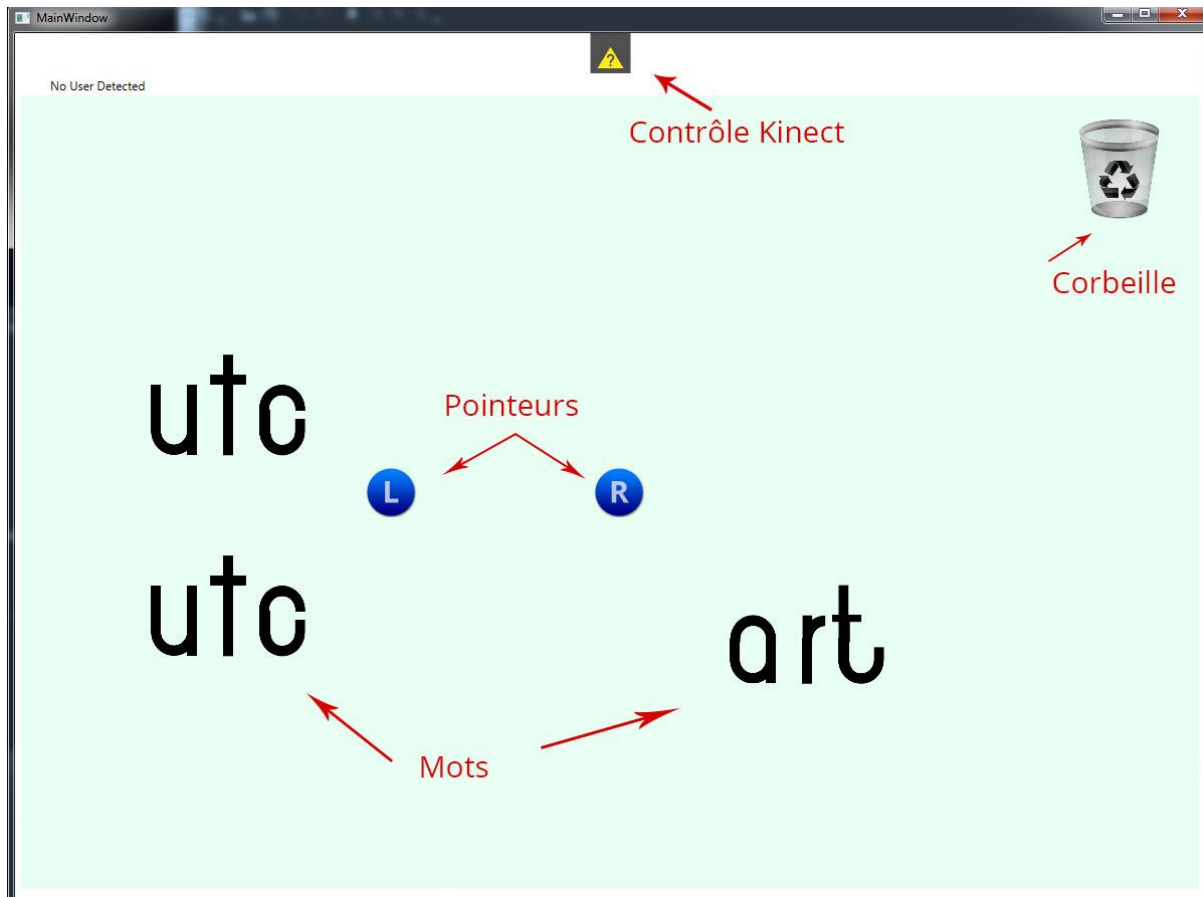
Nous avons donc accès aux éléments de la fenêtre à partir de l'attribut "this" dans la classe partielle en passant par le nom des éléments graphiques. Par exemple, si on a un simple texte nommé "monTexte" décrit dans le XAML, on pourra y accéder simplement en écrivant "this.monTexte".



Nous allons maintenant présenter succinctement l'interface de notre application.

Tout d'abord, il y a un contrôle spécifique à la Kinect placé en haut et au centre de l'écran qui permet de donner des informations sur la Kinect (si elle n'est pas branchée par exemple) et qui disparaît au bout de quelques secondes après le lancement de l'application (si la Kinect est connectée et en bon état de fonctionnement).

Ensuite nous avons un canvas qui contient l'ensemble des objets graphiques manipulés via la Kinect. On retrouve alors deux pointeurs permettant à l'utilisateur de situer ses mains sur l'écran, les mots manipulables par l'utilisateur et une corbeille permettant la suppression d'un mot. Ces trois objets graphiques (Pointeur, Mot et Corbeille) héritent d'une classe GUIObject que nous avons développée et dont le fonctionnement sera décrit dans la partie "Structures de données et autres éléments importants".



3) Structures de données et autres éléments importants

Architecture du code

Pour développer cette application, nous avons essayé de séparer les parties propres à l'utilisation de la Kinect de l'interface graphique. De plus, nous avons créé une classe GUIObject dont tous nos objets graphiques héritent afin de pouvoir écrire une librairie de manipulation graphique (fournissant des méthodes de zoom ou de rotation par exemple) qui puisse manipuler n'importe quel type d'objet graphique.

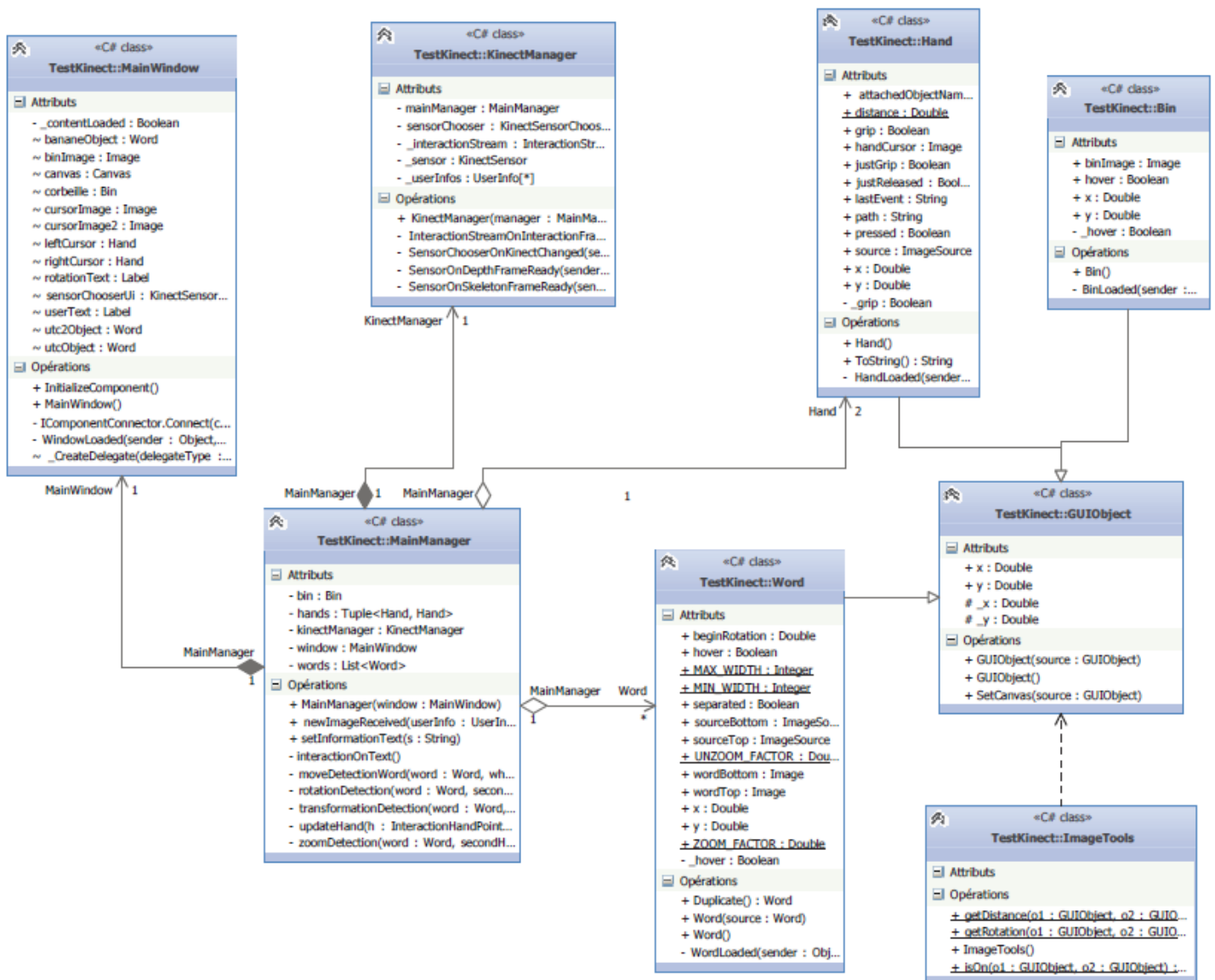


Schéma UML

Nous avons donc une classe `MainWindow` qui hérite de la classe `Window` et qui permet de pouvoir créer la fenêtre principale de notre application. Ensuite nous avons une classe `MainManager` ayant pour but de faire l'interface entre la Kinect et les composants graphiques. Les événements Kinect sont gérés par la classe `KinectManager` qui les simplifie et les transmet à la classe `MainManager` (un objet de la classe `UserInfo` est transmis lors de chaque nouvelle image envoyée par la Kinect).

Une fois l'objet `UserInfo` reçu, la classe `MainManager` se charge d'en extraire les informations propres à chaque main de l'utilisateur. Ensuite, différentes méthodes permettent de déterminer si l'utilisateur est en train d'interagir avec des objets de notre application. Pour ce faire, deux objets de la classe stockent à tout moment les informations sur les mains de l'utilisateur, telles que s'il a la main ouverte ou fermée. Une fois les informations sur les mains de l'utilisateur calculées, la classe `MainManager` fait appel à des méthodes vérifiant si l'utilisateur saisit un mot, s'il effectue une rotation, s'il "coupe" un mot ou bien s'il fusionne deux parties de mot.

Comme nous l'avons expliqué dans la partie "Architecture de l'application", nous avons créé une classe `GUIObject` dont tous nos objets graphiques héritent. Cette classe hérite elle-même du contrôle WPF `Grid` qui est un conteneur de base. `GUIObject` contient alors des attributs `X` et `Y` correspondant à la position de l'objet sur le canvas (De base, la classe `Grid` ne fournit pas de positions `X` et `Y` mais une position relative par rapport au canvas). Ainsi, comme tous nos objets auront les mêmes attributs `X` et `Y`, nous avons pu créer une classe `ImageTools` qui contient des méthodes statiques permettant de faire des opérations géométriques. Pour l'instant, nous avons des méthodes permettant d'effectuer un zoom, une rotation et de calculer une distance entre deux objets. Par exemple, nous sommes donc capables de calculer de la même manière la distance entre les deux pointeurs de main que la distance entre un pointeur et un mot.

Les objets héritant de `GUIObject` étant de simple conteneurs, ils doivent contenir des images afin d'être dessinés à l'écran. Les objets des classes `Hand` et `Bin` contiennent simplement une image représentant l'objet souhaité. Par contre, les objets de la classe `Word` contiennent en fait deux images dues à la nature du spectacle. En effet, nous devons être capables de couper un mot en deux parties puis de fusionner deux parties de mots pour recréer un mot complet.

4) Modifications apportées

Au fur et à mesure du développement de l'application, nous avons apporté des modifications au produit du projet initial.

Dans un premier temps, nous nous sommes rendu compte que certains gestes n'étaient pas détectables par la Kinect. Par exemple, dans l'objectif de couper des mots en deux, nous aurions souhaité placer notre main horizontalement et couper le mot (virtuellement) dans sa largeur. Cependant, la Kinect ne peut détecter la position précise des doigts d'une main, ou son orientation (verticale/horizontale). Nous avons donc dû repenser ce geste, pour qu'il soit conciliable avec les gestes de base détectés par la Kinect. Ainsi, pour couper un mot, l'utilisateur doit fermer ses deux mains sur l'objet et écarter ses mains, ce qui « déchirera » automatiquement, et à l'horizontale, le mot en 2 parties. De la même façon, il n'est pas possible de passer d'un écran à l'autre (à la manière du défilement de slides), mais cette option n'est pas souhaitée par le client.

Une autre modification a été effectuée suite à une des réunions avec notre client. En effet, ce dernier nous a indiqué précisément ce qu'il voulait pouvoir faire avec les mots sur l'écran. Ainsi, il ne souhaite pas avoir la possibilité de zoomer (fonctionnalité que nous avons développée, et qui est disponible dans le code source au cas où le client souhaiterait en disposer dans le futur).

De même, il a été intéressant et instructif de définir précisément de quelle façon un mot pourrait se transformer en un autre (grâce à la moitié commune aux deux mots). Alors que nous avons initialement songé à une animation qui assemble directement les mots entre eux, il s'avère que le client préfère assembler les mots par lui-même, et maîtriser ainsi la scénarisation et la vitesse des objets mis en scène.

V - Évaluation auprès des utilisateurs (User Feedback)

Le développement a été jalonné de rencontres avec l'artiste (les Mercredis midi) pour éclaircir les éléments d'incompréhension et valider le travail réalisé (gestes et interactions associées). Ces réunions ont permis à l'utilisateur final (Pierre Fourny) de tester l'application. Nous pensions que l'artiste mettrait un certain temps avant de réussir à manipuler les mots, étant donné le côté déroutant de manipuler des objets dans le vide (les mains ne tiennent rien). A notre grande surprise, ce dernier a très rapidement compris le fonctionnement des différentes interactions et a eu beaucoup d'aisance dans l'utilisation de l'application dès les premières minutes. Pour d'autres acteurs qui ont testé l'application, le temps d'adaptation a été plus long. Il semble donc que le temps de prise en main varie en fonction des utilisateurs bien qu'il soit assez court pour la majorité des gens.

Il est à noter que l'utilisation du prototype a donné beaucoup d'idées au principal intéressé. En effet Pierre Fourny souhaiterait se servir de la bibliothèque que nous avons développée pour exploiter le procédé de la police de l'ombre (procédé décrit plus haut dans ce même rapport) ; il voudrait aussi donner une dimension didactique et ludique au projet en envisageant son utilisation lors de ses interventions dans les établissements scolaires afin de sensibiliser les jeunes à la "poésie numérique".

VI - Limites du prototype et améliorations possibles

1) Limites

Une première limite de notre projet découle de ce qui a été décidé dans le cahier des charges : l'application réalisée n'est pas directement exploitable par Pierre Fourny. Il ne s'agit que d'un prototype exploitant la bibliothèque développée en vue de manipuler des mots sur un écran à l'aide de gestes.

Par exemple, il faut pour l'instant passer par le code pour ajouter des mots dans l'interface graphique. L'artiste a dit qu'il ferait certainement développer une interface d'ajout de mots en WPF dans le cadre d'une UV Projet à l'UTC.

Par ailleurs, le fait que l'image ne soit pas inversée constitue une autre raison pour laquelle l'exploitation sur scène n'est pas possible à l'état actuel.

Ainsi, nous expliquions dans le cahier des charges que, pour que l'artiste puisse faire face au public, il serait nécessaire que la caméra et un petit écran de rappel soient face à lui. Il pourrait ainsi voir les actions qu'il réalise sur l'interface. Une extension de cet écran de rappel doit être projetée derrière lui (scène disposant d'un écran de projection en fond) ou devant lui (scène disposant d'un voile en tulle pour une projection en transparence). L'image projetée sur cet écran doit être inversée de sorte que les actions sur l'écran soient en correspondance avec les mouvements du performeur. Or, nous n'avons pas testé cette inversion (réalisée à l'aide du driver de carte graphique) avec le performeur.

2) Améliorations

La première amélioration à envisager lors de la poursuite de ce projet sera la scénarisation. Elle comporte plusieurs éléments que l'artiste aimerait voir intégrés dans la version à venir de l'application:

- 1- La possibilité d'ajouter des mots via une interface à définir. En "attrapant" des mots dans un panneau sur la gauche de l'écran et en les faisant glisser sur l'écran principal.
- 2- L'intégration dans la Bibliothèque déjà développée d'une interaction permettant de mettre en scène le procédé de la police de l'ombre (procédé utilisant une autre police que la police coupable, utilisé dans notre application, par lequel deux mots superposés forment un nouveau mot dont la sémantique est en rapport avec les deux mots constituant). Il s'agirait ici de définir une interaction, un geste afin que deux mots utilisant ce procédé puissent s'aimer de sorte à faire apparaître le troisième mot.
- 3- Des tests sur scènes avec la mise en place de la Kinect, de l'écran de rappel et de la projection inversée (configuration de la carte graphique de la machine sur laquelle l'application s'exécute). L'objet de ces tests est, entre autres, de visualiser la surface sur laquelle le performeur peut se déplacer sur la scène en ayant l'assurance que la Kinect détectera ses gestes.
- 4- Réflexion sur l'aspect didactique de l'application en vue d'une utilisation dans les établissements: installation rudimentaire dans une salle de classe avec un dispositif Kinect + vidéo-projection.

Ces quatre axes de progression feront l'objet d'une proposition de PR (destinée à des GIs pour s'assurer de la bonne compréhension du code).

Conclusion

Ce projet nous a permis de mesurer les enjeux de la conception de systèmes interactifs. Lors de telles réalisations, l'utilisateur doit être au centre de la réflexion. L'implication de Pierre Fourny, qui sera le principal utilisateur des débouchés de ce prototype était donc très appréciable. Finalement, la convivialité ou l'aspect intuitif d'un système dépend en grande partie des utilisateurs, ce qui induit donc la nécessité de nombreux tests auprès de ces derniers, pour que l'interface réalisée leur corresponde.

Par ailleurs, nous avons pu découvrir le développement pour Kinect. L'utilisation de la Kinect dans le cadre de projets artistiques est une idée émergeant progressivement. Ce mode d'interaction plaît, du fait de son côté magique, donc ludique, et très intuitif: les gestes réalisés peuvent se rapprocher au plus près de la réalité. La Kinect pour Windows étant très récente, on peut s'attendre à ce que des améliorations de cet outil ouvrent de nouvelles possibilités de développement pour les applications à visée artistique telles que la nôtre.

Enfin, les objectifs fixés en début de semestre ont pu être atteints: Pierre Fourny dispose d'un prototype illustrant les gestes utilisables pour présenter une animation ou un spectacle. Il reste à présent à définir les aspects liés au scénario et à la présentation sur scène, aspects qui seront développés lors de projets proposés le semestre prochain.

Annexes

1) Sondage

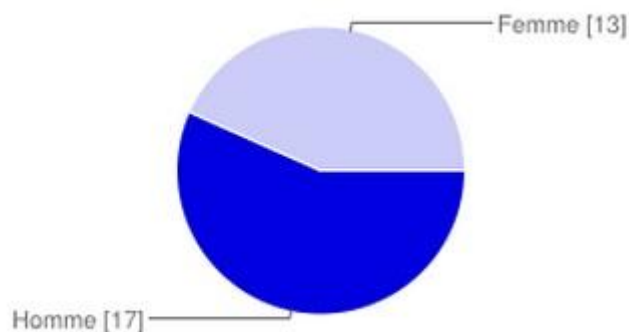
Résultat du sondage portant sur l'intérêt pour les technologies liées au Kinect dans le cadre de l'univers du spectacle:

Notre sondage a pour but de savoir assez grossièrement si l'idée d'animer un spectacle en utilisant la Kinect est intéressante mais aussi si la culture Kinect est déjà présente dans les esprits.

Il est à noter que ce sondage a été publié parmi nos connaissances ce qui implique qu'une bonne partie des sondés sont jeunes et intéressés par les nouvelles technologies (souvent UTCéens).

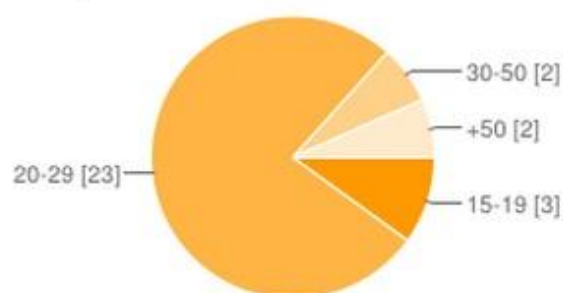
2) Métadonnées sur les sondés :

Sexe



Homme	17	57%
Femme	13	43%

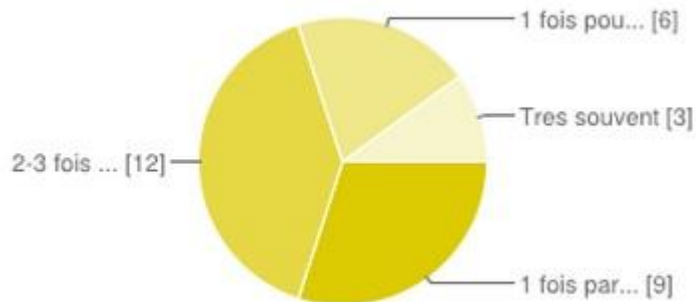
Tranche d'âge



15-19	3	10%
20-29	23	77%
30-50	2	7%
+50	2	7%

Comme prévu, l'âge des sondé est plutôt compris entre 20 et 30 ans.

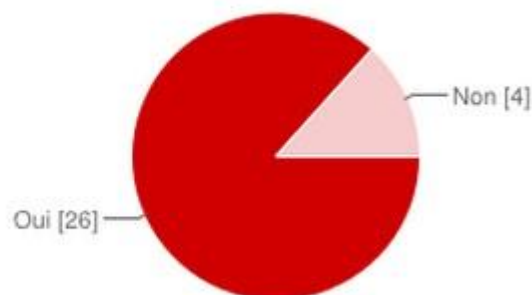
Vous allez souvent voir des spectacles?



1 fois par an	9	30%
2-3 fois par an	12	40%
1 fois pour mois	6	20%
Tres souvent	3	10%

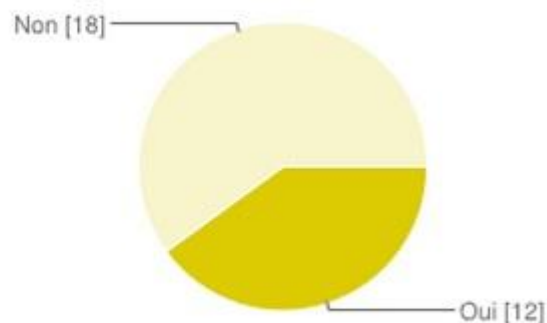
3) La Kinect:

Savez-vous ce qu'est une kinect



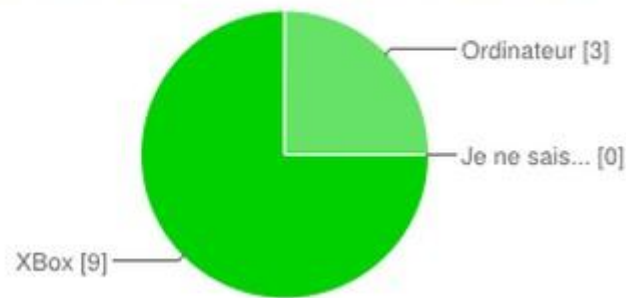
Oui	26	87%
Non	4	13%

Avez-vous déjà utilisé la kinect?



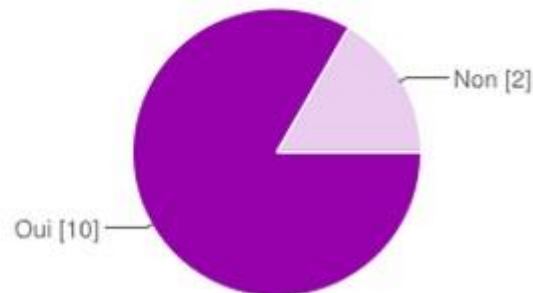
Oui	12	40%
Non	18	60%

Sur quel support avez vous déjà utilisé une kinect?



XBox	9	75%
Ordinateur	3	25%
Je ne sais pas	0	0%

Avez vous trouvé l'expérience intéressante?

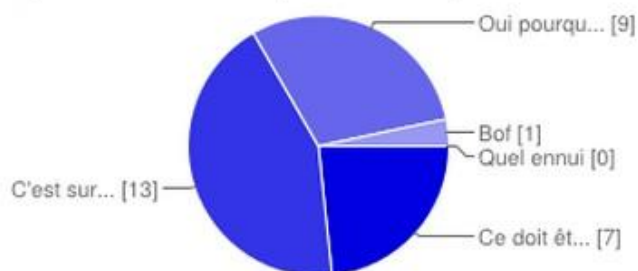


Oui	10	83%
Non	2	17%

Peu de gens ont trouvé l'expérience de l'utilisation de la Kinect inintéressante. Cela induit que le performeur appréciera sûrement l'utilisation de la Kinect dans son spectacle.

4) Monde du spectacle & Kinect:

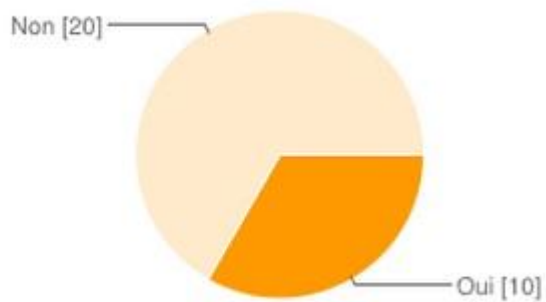
Que pensez vous d'un spectacle ou le performeur interagit avec des objets affichés sur un mur via la gestuelle?



Ce doit être fabuleux	7	23%
C'est sûrement intéressant	13	43%
Oui pourquoi pas	9	30%
Bof	1	3%
Quel ennui	0	0%

Il est important de noter que la plupart des sondés pense qu'un spectacle utilisant la Kinect serait intéressant.

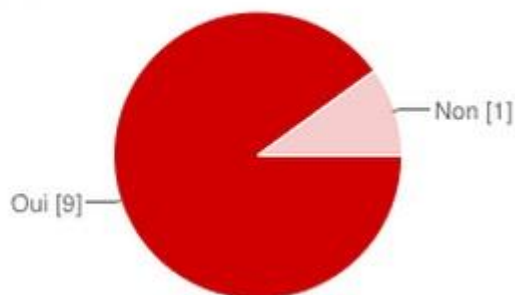
En avez vous déjà vu?



Oui	10	33%
Non	20	67%

Peu de sondés ont déjà vu ce type de spectacle.

Etait-ce bien?



Oui	9	90%
Non	1	10%

Presque tous ceux qui en ont vu un ont trouvé l'expérience intéressante.