



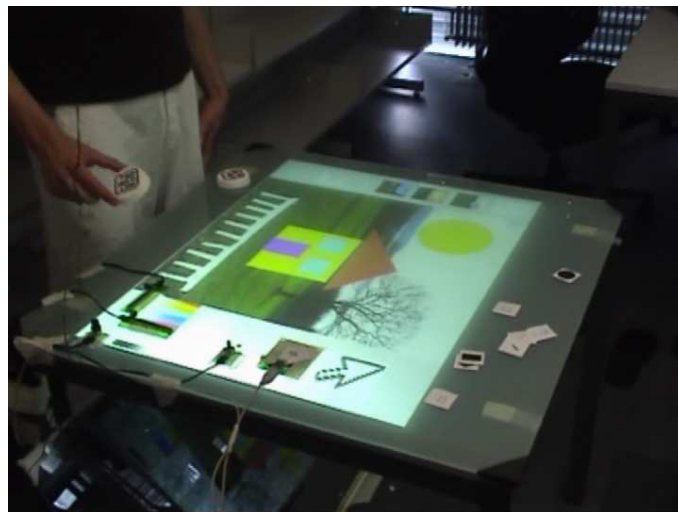
Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg



Phidgets Contest

Table de dessin

X-Paint



Auteurs :	Luca Zingg, Daniele Della Bruna et Yannick Thiessoz
Superviseur :	Denis Lalanne, Omar Abou Khaled et Jacques Bapst
Classe :	Master en Informatique université de Fribourg
Type :	Projet de phase II du cours MultiModal Interfaces
Début :	08.05.2006
Fin :	19.06.2006

1	INTRODUCTION	3
2	LE PROTOTYPE	3
3	LES MODALITÉS	4
3.1	LA VUE	5
3.2	L'OUÏE	5
3.3	MODALITÉ PHYSIQUE	6
4	ARCHITECTURE	9
5	IMPLÉMENTATION	10
5.1	LE CONTROLLER	11
5.2	LES INPUTS	12
5.3	LE GUI	13
6	INSTALLATION	14
6.1	LA MISE EN PLACE MATÉRIELLE	14
6.2	LA MISE EN PLACE LOGICIELLE	15
7	UTILISATION	15
8	TESTS	17
9	EVALUATION DE L'INTERFACE	17
10	AMÉLIORATIONS	18
11	CONCLUSIONS	18
12	RÉFÉRENCES	18
13	ANNEXES	19

1 Introduction

Le but de ce projet (Phidgets contest) est, sur la base des librairies que nous avons étudiés lors de la première partie du cours, de développer une interface multimodale tangible. Pour cela nous avons à disposition un starter-kit de phidgets et nous pouvons utiliser une ou plusieurs librairies vues durant la première partie du cours. Cette interface doit répondre à trois critères déterminants qui sont ; l'originalité, l'utilisabilité et l'utilité.

Durant la première partie du cours MMI nous avons réalisé une petite interface tangible représentant une table de dessin. Pour cela nous avons utilisé une librairie **ReactiVision** qui permet de traquer des marqueurs. Ces marqueurs représentaient dans notre exemple des stylos de différentes couleurs.

Nous avons décidé de reprendre ce concept pour le phidgets contest. Nous avons fait ce choix car ce système remplissait déjà parfaitement le critère d'originalité. Et nous pouvions améliorer son utilisabilité en intégrant d'autres modalités. Nous avons donc ajouté le starter-kit de **phidgets** pour réaliser certaines opérations et nous avons aussi intégré la librairie de reconnaissance de la voix **Sphinx** pour d'autres opérations. Ces deux ajouts améliorent nettement l'utilisabilité de la table de dessin. La partie utilité est aussi traitée dans ce projet. En effet, sa réalisation démontre les possibilités d'utilisation d'une telle table dans le domaine du dessin mais aussi pour d'autres domaines.

Dans les prochains chapitres nous allons faire une petite introduction à notre prototype ainsi qu'à son fonctionnement, puis nous allons décrire l'intégration des différentes modalités, l'architecture de notre prototype, les différents tests que nous avons effectués, l'installation du prototype et pour finir une conclusion.

2 Le Prototype

Pour réaliser notre prototype de table de dessin nous avons donc utilisé la librairie de reconnaissance de formes **ReactiVision**, la librairie de reconnaissance de la voix **Sphinx** et le starter-kit de **phidgets**.

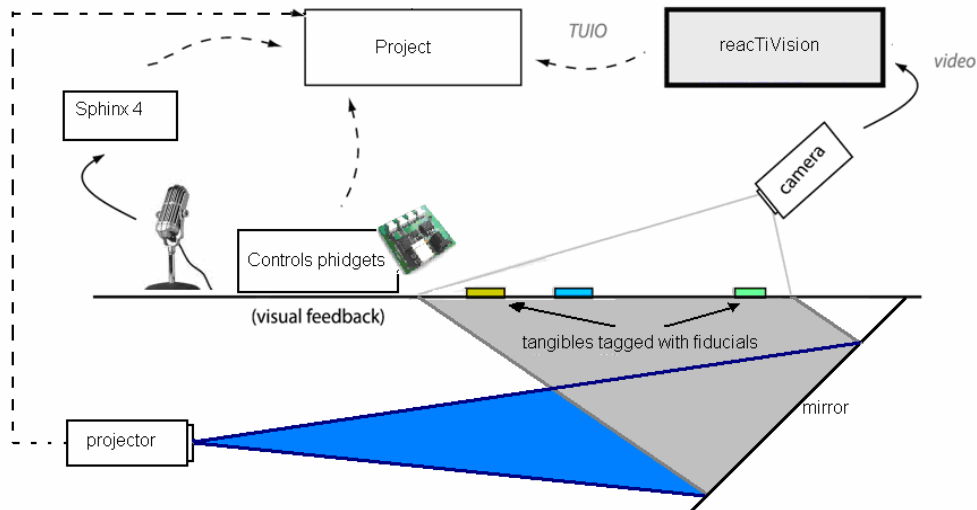
La librairie **ReactiVision** a besoin, pour son fonctionnement, d'une caméra pour traquer les marqueurs. Nous utilisons ici une camera Sony mise à disposition par l'université.

Pour utiliser la librairie **Sphinx** notre prototype est aussi équipé d'un micro.

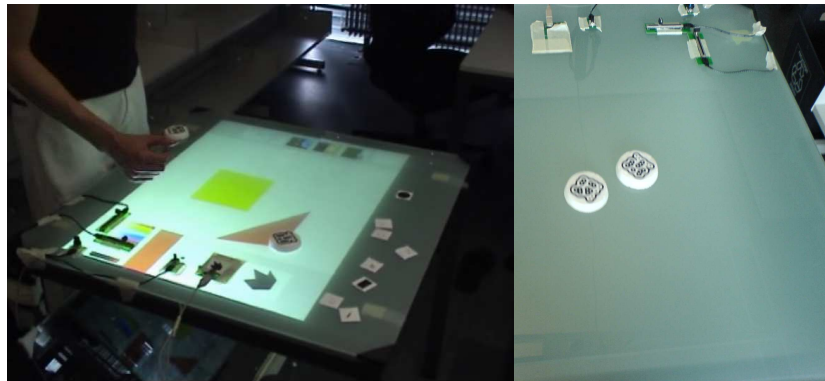
Pour certaines commandes, nous utilisons une partie des phidgets disponible dans le starter-kit.

Nous utilisons aussi une table semi transparente, cette table est composée d'une plaque de verre dont l'un des deux cotés a été sablé. Ce sablage nous permet de projeter une image sur cette table depuis le bas avec un rendu des couleurs très agréable. Ce principe de rétroprojection était nécessaire pour éviter une projection et une capture d'image depuis le même côté. En effet, sans cela la projection viendrait se superposer sur les marqueurs empêchant leurs détections. Pour permettre la mise en place de la rétroprojection, nous avons dû intégrer un miroir à notre prototype car la hauteur de la table ne permettait pas une projection directe. Nous aurions aussi pu faire l'inverse, à savoir placer la caméra en dessous de la table et le projecteur en dessus. Malheureusement cette configuration aurait introduit deux problèmes. Le premier étant que la distance (caméra-table) aurait été petite et nous aurions donc été dépendants de l'angle de vue de la caméra. Le deuxième étant la moins bonne qualité d'image. Les couleurs sont plus vives de l'autre côté de la table par rapport au point d'entrée de la projection.

Ce prototype est représenté dans le schéma suivant :



Une autre vue de notre prototype est la table elle-même. Voici une image la représentant.



Dans la première image nous pouvons voir les trois parties de notre application de dessin. A gauche de l'utilisateur, nous découvrons les différents fonds disponibles ainsi que le log d'acquisition des commandes vocales. Au centre nous avons notre aire de dessin sur lequel se déplacent les marqueurs. Et à droite se trouvent les différents phidgets utilisés par notre prototype.

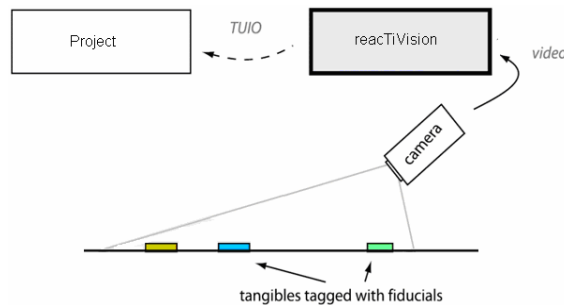
3 Les modalités

Comme nous en avons parlé dans l'introduction, notre prototype inclut plusieurs modalités. Chacune d'elles est traitée dans un sous-chapitre. Ce sous chapitre met en avant la modalité utilisée et son utilité au sein du prototype. Bien évidemment, nous utilisons certaines modalités dues aux cinq sens de l'homme de manière naturel, par exemple, nous utilisons la vue pour savoir ce que nous dessinons. Ici nous ne décrivons que les modalités que notre prototype utilise pour recevoir des informations.

3.1 La vue

Présentation

La vue est l'un des cinq sens dont nous disposons. Elle nous assure une image en permanence sur le monde qui nous entoure. Nous utilisons la vue en tant que modalité grâce à la librairie **ReactiVision**. Cette librairie nous permet de suivre les déplacements des marqueurs visuels à l'aide d'une caméra. Ensuite nous pouvons récupérer et traiter ces informations au sein d'une application de niveau supérieur. Nous allons donc utiliser cette librairie comme entrée de notre prototype pour connaître les marqueurs que nous utilisons et leurs déplacements. Ce fonctionnement est représenté par le schéma suivant :



Intégration à la table de dessin

Nous utilisons ici un nombre restreint de marqueurs (au maximum 2). Nous avons fait ce choix pour être le plus près possible de la manière dont un être humain dessine. En effet, il est plutôt peu courant de voir une personne dessiner avec plusieurs stylos. Nous avons tout de même permis l'utilisation de deux marqueurs en simultanés pour certaines opérations. De cette manière, la création d'un cercle est simplifiée en permettant à un marqueur de définir le centre de celui-ci et à l'autre d'exprimer son rayon. Les marqueurs nous permettent donc, dans notre prototype, de réaliser les opérations suivantes :

- Dessin de divers formes (ligne, cercle plein, cercle vide, rectangle plein, rectangle vide, polygone plein, polygone vide).
- Dessin à main levée
- Effacement d'une forme
- Rotation et déplacement d'une forme
- Sélection de formes
- Sélection d'une image de fond

Ces marqueurs ont été fixés sur de petits supports. Pour que l'utilisateur ne puisse pas entraver la détection en mettant ses mains entre la caméra et les marqueurs.

3.2 L'ouïe

Présentation

L'ouïe fait aussi partie des cinq sens dont nous disposons. Nous utilisons l'ouïe en tant que modalité grâce à la librairie **Sphinx**. Cette librairie nous permet d'associer certaines opérations à des commandes vocales. Nous allons donc utiliser cette librairie comme entrée de notre prototype pour réaliser certaines commandes.

Intégration à la table de dessin

L'intégration de la voix dans la table de dessin permet de nettement faciliter son utilisation. En effet, nous avons pu combiner l'association de commandes vocales simples à l'utilisation de marqueurs de la librairie **ReactiVision** pour réaliser certaines opérations. Par exemple, lorsque l'utilisateur veut dessiner un polygone il faut qu'il donne un certain nombre de points. Dès lors, quoi de plus simple que de combiner la librairie précédente à une commande vocale pour créer la liste des points. En effet, il suffit de capturer la position d'un marqueur de la librairie **ReactiVision** et de l'associer à une commande vocale pour créer un point. Voici la liste des commandes vocales de la table de dessin ainsi que leurs utilités :

- **Select Shape** : Permet de sélectionner une des formes déjà présentes sur le dessin. De là, il est possible d'effectuer les trois opérations suivantes : erase shape, move shape et rotate shape.
- **Erase Shape** : Permet d'effacer une des formes déjà présentes sur le dessin. Nous ne laissons pas la possibilité à un utilisateur d'effacer qu'une partie d'une forme car pour nous la forme est un tout.
- **Move Shape** : Si une forme dessinée n'est pas à l'emplacement désiré. Il est raisonnable de permettre son déplacement, c'est ce que permet cette commande.
- **Rotate Shape** : Permet la rotation d'une forme déjà présente sur le dessin
- **Return** : Permet de clore un select Shape et de libérer la forme sélectionnée. A partir de là les opérations suivantes ne sont plus disponibles : erase shape, move shape et rotate shape.
- **Begin Draw** : Lorsque l'on désire dessiner une forme, cette commande commence le dessin.
- **Point** : Permet de créer des points supplémentaires dans la réalisation d'un polygone ou d'une ligne.
- **End Draw** : Permet de conclure le dessin d'une forme.
- **Select Background** : Permet de sélectionner le fond de l'image.

L'intégration d'une interaction vocale améliore donc l'utilisabilité de notre prototype. Malheureusement, nous nous sommes rendus tributaires de la librairie **Sphinx** qui n'est disponible gratuitement qu'en anglais et qui ne garantit pas une excellente reconnaissance des mots lorsque l'on n'a pas un bel accent anglophone.

3.3 Modalité physique

Présentation

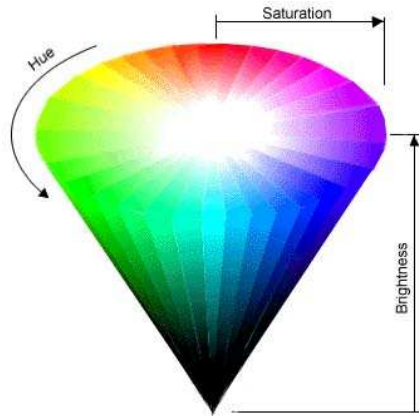
Les modalités physiques d'entrée d'un système informatique les plus connues et utilisées sont sans aucun doute la souris et le clavier. Ici pour réaliser notre interface tangible nous avons intégré des phidgets. Les phidgets sont des composants hardware (USB). Il en existe de plusieurs sortes et ils sont faciles à implémenter grâce à leur API. Nous les avons utilisés comme entrée de notre projet pour réaliser certaines commandes.

Intégration à la table de dessin

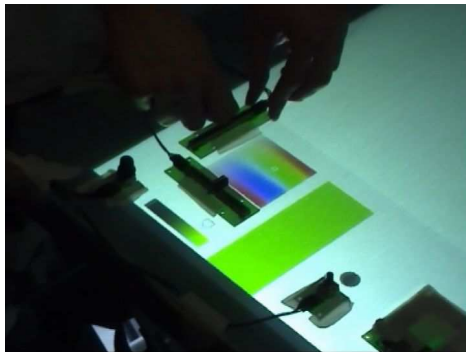
Nous disposons d'un starter-kit de phidgets mais nous n'avons eu besoin que d'une partie d'entre eux. Ils nous permettent de définir trois informations importantes de notre application de dessin, le choix de la couleur, le choix de l'épaisseur du trait et le type d'opération à effectuer.

Le choix de la couleur

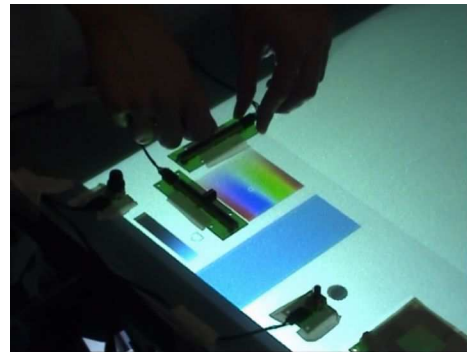
Pour cela nous utilisons un phidget de type PhidgetInterfaceKit 8/8/8 et trois senseurs. Cela nous permet de définir la couleur à l'aide d'un modèle de type HSB. La méthode de définition de couleur en HSB peut être représentée de la manière suivante :



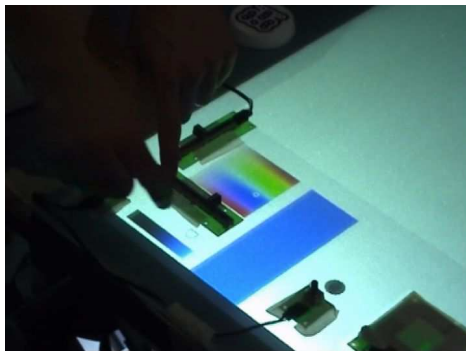
Voici une image représentant la manière dont le choix d'un des composants HSB de la couleur est réalisée :



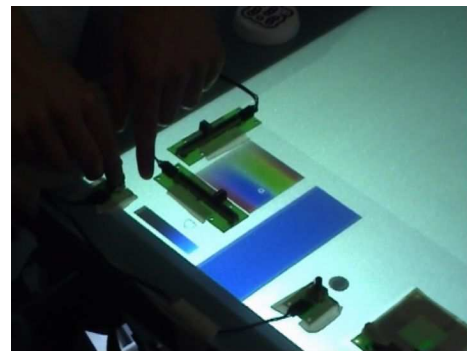
Couleur de départ



modification de la tonalité H



Modification de la saturation S



modification de l'éclat B

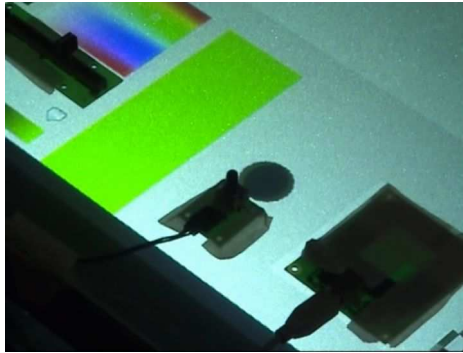
Sur cette image nous pouvons voir que le slider qui se trouve en bas permet de définir la composante H (tonalité) cette composante correspond à l'angle du cône dans la figure de représentation des couleurs en HSB.

Le slider vertical permet de modifier la composante S (saturation) qui est représentée horizontalement dans la figure de représentation des couleurs en HSB

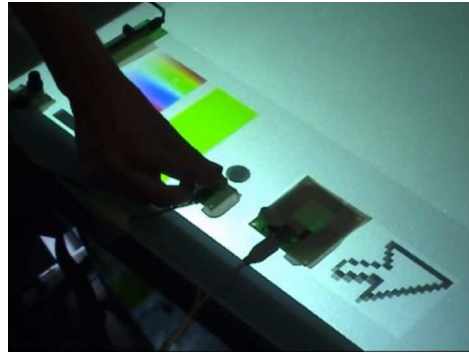
Et le potentiomètre permet de définir la composante B (éclat) grâce à laquelle il est possible d'avoir le noir. Cette composante correspond à l'axe vertical de la figure de représentation des couleurs en HSB.

L'épaisseur du trait

Là nous utilisons le même phidget de type PhidgetInterfaceKit 8/8/8 avec un autre senseur de type potentiomètre. Ces images permettent de décrire la modification de la taille du trait :



Taille de départ



modification de la taille

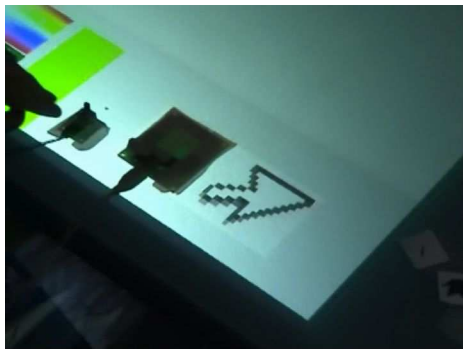
En tournant le potentiomètre on augmente/diminue la taille du trait.

Le choix de l'opération

Nous utilisons ici un phidget de type RFIDPhidget. Ce phidget permet d'identifier des objets par fréquence radio. Nous avons donc attribué une opération à chaque identificateur. En plaçant un identificateur sur la borne, notre prototype sait alors quelle opération il devra effectuer.

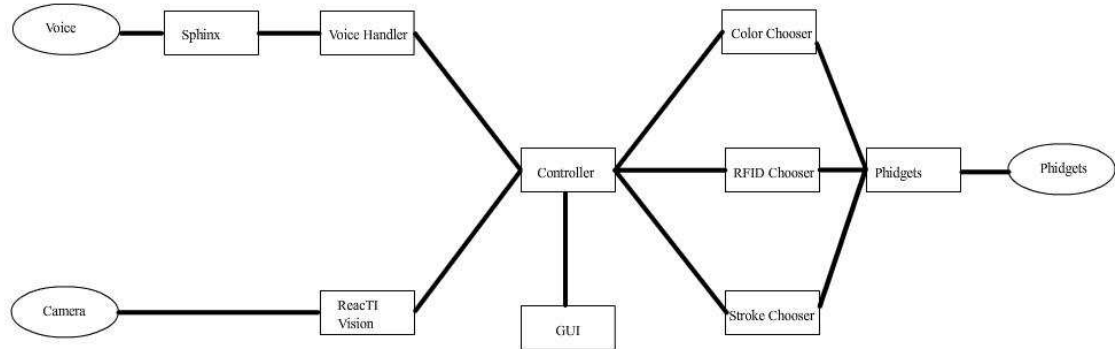
Nous avons déterminé les opérations suivantes :

- sélectionner
- ligne
- cercle plein
- cercle vide
- rectangle plein
- rectangle vide
- polygone plein
- polygone vide
- main levée



4 Architecture

Comme nous pouvons le voir dans le schéma suivant, notre architecture est clairement centralisée autour d'une partie que nous avons appelée **Controller**. Le **Controller** permet de gérer l'arrivée d'informations des trois sous parties d'input. Dès qu'il reçoit des informations en input, il les traite et les redistribue en output si nécessaire.

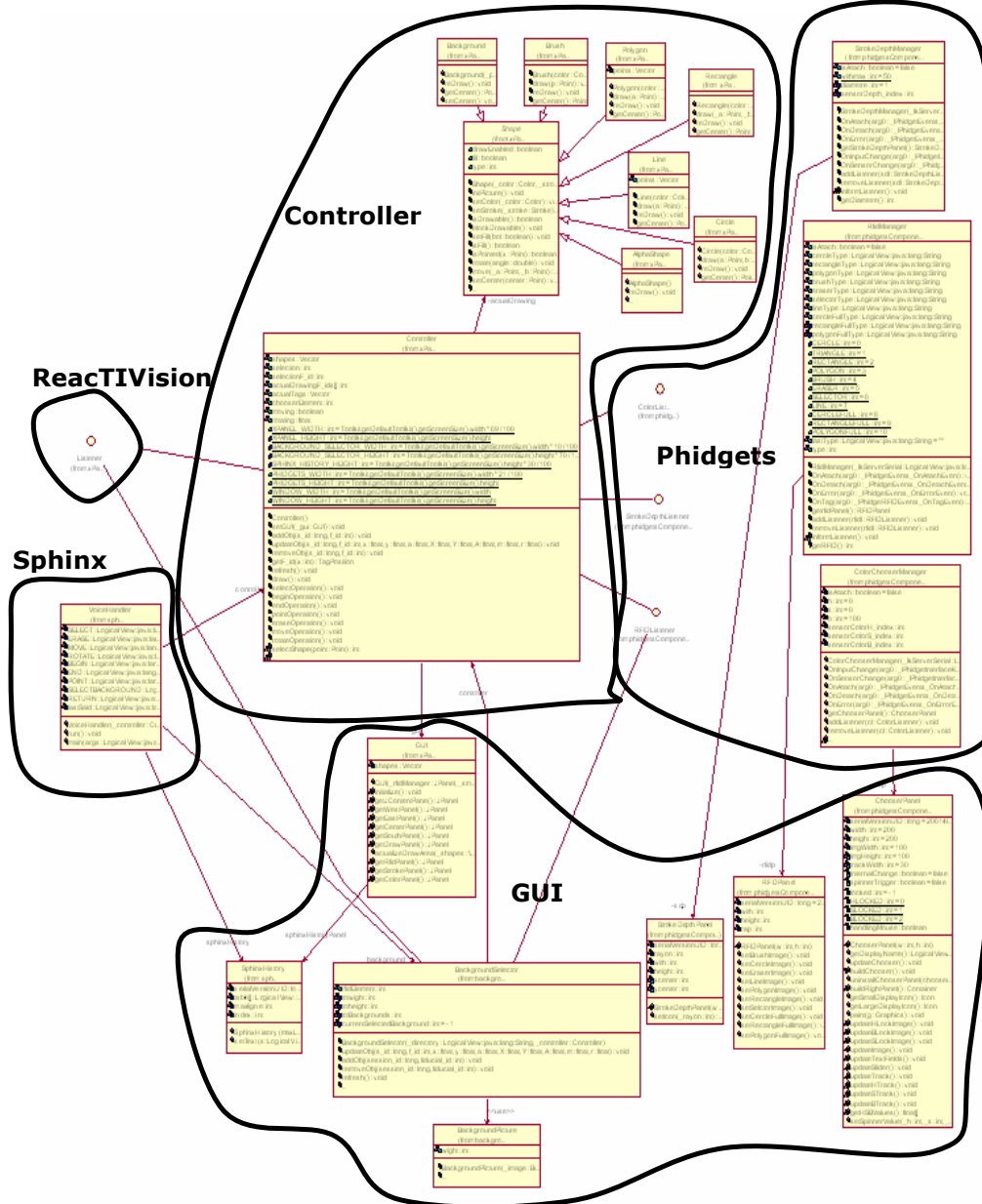


Les trois parties d'input sont la voix via la librairie **Sphinx**, la vue via la librairie **ReactIVision** et en hardware les **phidgets**. Ce sont ces parties d'input qui fournissent les informations au **Controller** quand un utilisateur interagit avec la table de dessin.

La dernière partie du schéma ci-dessus est le GUI. Le Gui représente l'output de notre système. C'est ce GUI qui sera projeté sur la table rend les interactions de l'utilisateur avec le système visible.

5 Implémentation

Après avoir décrit l'architecture de notre système dans son ensemble, nous allons montrer, dans les diagrammes de classes suivants, l'implémentation de cette architecture que nous avons mise en œuvre en Java.

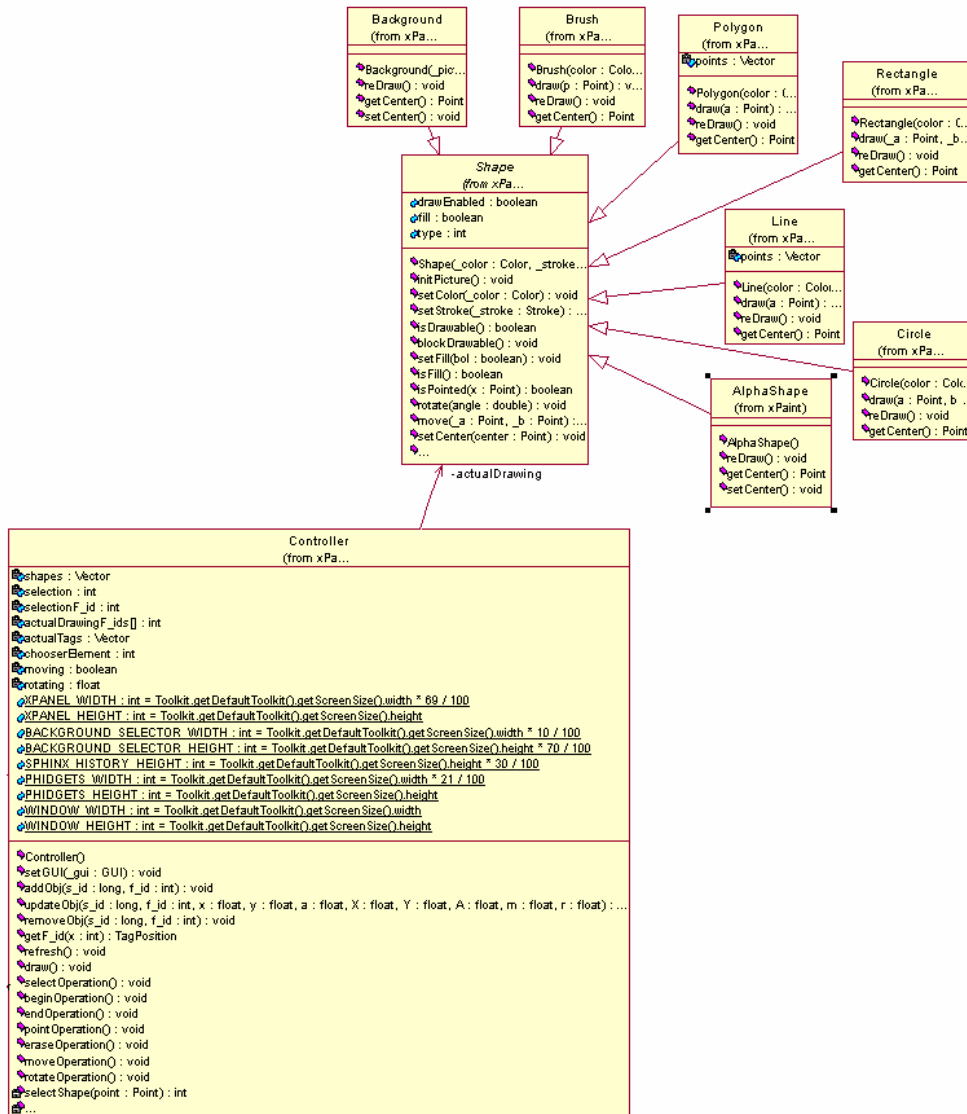


Le premier diagramme représente l'architecture totale de notre système après implémentation. Le but étant de montrer les différentes parties présentées dans le chapitre précédent.

Comme nous pouvons le voir, nous avons bien le contrôleur, nos trois parties d'input et le GUI. Chaque partie est traitée plus en détail dans les sous chapitres suivants.

5.1 Le controller

Le controller est composé d'une classe principale Controller auquel vient s'ajouter une classe représentant la forme que nous sommes en train d'éditer. Cette classe est une généralisation des différents types de formes qui sont représentables dans la table de dessin.

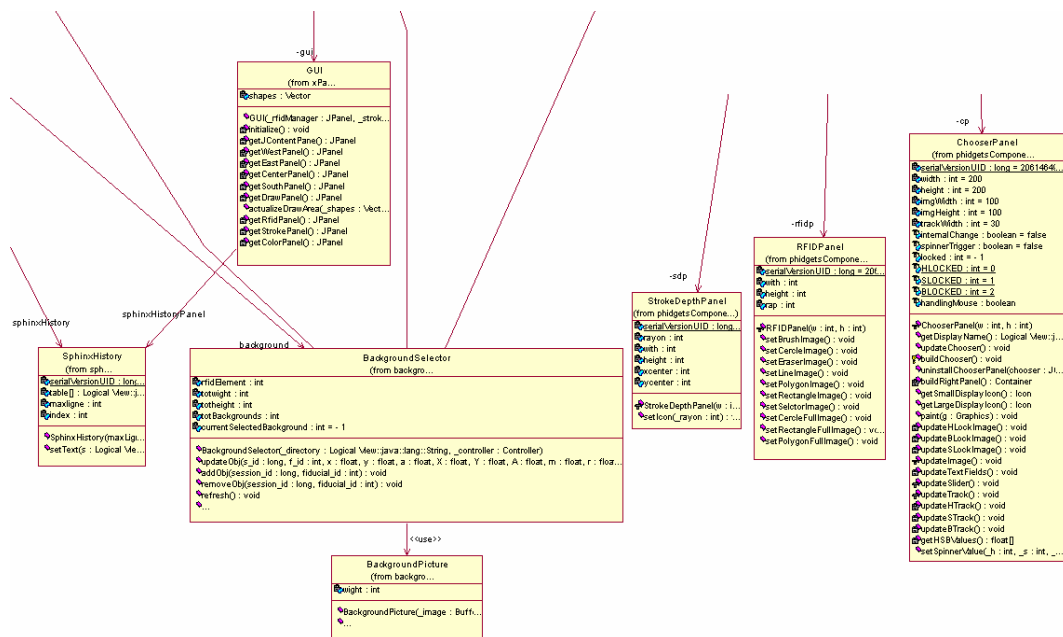


La classe principale contient entre autre un vecteur contenant toutes les formes d'un dessin. Elle contient aussi les différentes méthodes des différentes interfaces de récupération des inputs qu'elle étend. Pour finir elle est aussi composée de champs permettant de connaître l'état général du système.

5.3 Le GUI

Le GUI représente la partie qui sera projetée sur la table pour permettre à l'utilisateur de visualiser le résultat des interactions qu'il réalise. Le Gui est représenté par une classe principale à laquelle vient se rajouter les classes suivantes :

- SphinxHistory : Permet de représenter la liste des dernières utilisations de la librairie vocale faite par l'utilisateur
- BackgroundSelector : Permet de représenter le fond de table sélectionné par l'utilisateur.
- StokeDepthPanel : Permet de visualiser le changement de largeur de trait effectué à l'aide des phidgets.
- RFIDPanel : Permet de visualiser le changement de type d'opération courante exigée par les phidgets RFID.
- ChooserPanel : Permet de visualiser le changement de la couleur effectué à l'aide des phidgets.



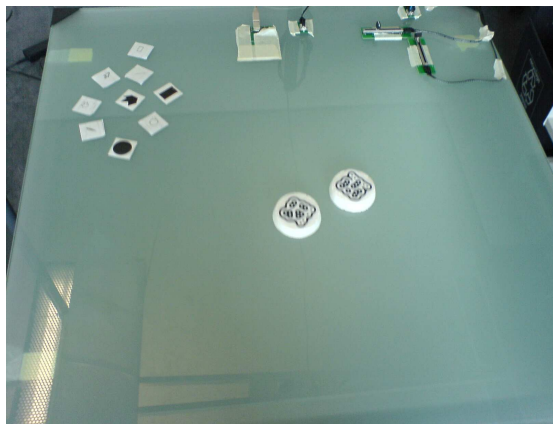
6 Installation

L'installation de notre projet peut être découpée en deux phases.

6.1 La mise en place matérielle

Cette partie comprend :

- Branchement du projecteur, du micro et de la caméra à l'ordinateur qui contient notre projet et les librairies.
- La mise en place de la table, du miroir et du projecteur pour que l'image projetée soit de la bonne dimension et au bon endroit sur la table.
- La mise en place de la caméra pour avoir une bonne reconnaissance des cibles par rapport à l'image projetée
- Le choix de la taille des cibles en fonction de la détection de celles-ci par la caméra
- Le branchement des phidgets, ainsi que la mise en place de ceux-ci sur la table au bon emplacement par rapport à la projection.



6.2 La mise en place logicielle

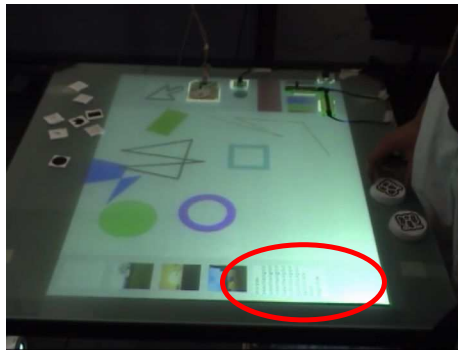
Cette partie comprend :

- Lancement de la librairie **ReactiVision**.
- Lancement du Webservice pour les **phidgets**
- Lancement de notre application Java.

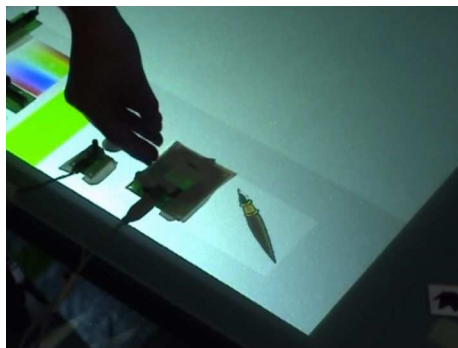
7 Utilisation

Lorsque l'installation et le lancement de notre application sont effectués, on peut utiliser la table de dessin. Ce chapitre permet de voir les différentes utilisations possibles de la table.

Avant de passer aux différentes opérations possibles nous allons clarifier quelques points. Tout d'abord pour s'assurer que les commandes vocales ont été interprétées correctement, un petit historique de ces commandes est disponible dans la partie inférieure gauche.



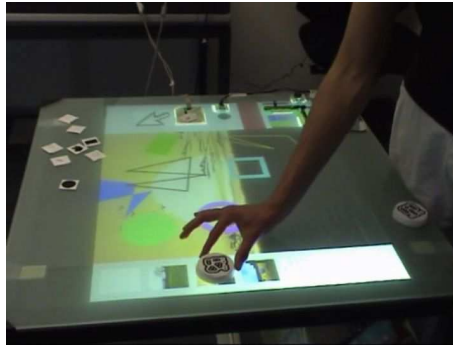
Ensuite pour s'assurer que la bonne opération de dessin est activée, il suffit de regarder le coin supérieur droit où l'image de l'opération active est affichée. Si ce n'est pas l'opération désirée, il faut placer le **rfid** de l'opération sur le capteur.



Ces différents points étant réglés nous pouvons passer à l'utilisation de la table.

La première opération qui peut être réalisée consiste à choisir un fond de dessin. Par défaut, le fond est totalement blanc. Mais sur la gauche de la table se trouve une liste de fonds disponibles. Pour sélectionner un de ces fonds, il faut tout d'abord s'assurer que le **rfid** de sélection soit bien activé.

Ensuite il suffit de ce déplacer avec un des marqueur de la librairie **ReactiVision** sur le fond désiré et de dire **select background**.



La deuxième opération qui peut être réalisée consiste à choisir une couleur pour dessiner. Cette opération peut être réalisée avant ou pendant le dessin d'une forme mais dès que le dessin est terminé le choix de la couleur ne peut plus être modifié.

La troisième opération qui peut être réalisée consiste à choisir l'épaisseur du trait de dessin. Cette opération peut aussi être réalisée avant ou pendant le dessin d'une forme mais dès que le dessin est terminé ce choix ne peut plus être modifié.

Ces opérations basiques étant décrites, nous allons nous intéresser maintenant au dessin lui-même. Comme décrit précédemment un certain nombre de forme sont disponibles pour le dessin. Pour réaliser ces formes, il faut s'assurer qu'elle soit correctement activée via **rfid**. Pour commencer à dessiner la forme, il faut dire **begin draw**. Dans le cas d'un dessin avec l'opération **main levée**, il suffit de bouger le marqueur de la librairie **ReactiVision**. Pour certaines formes (cercle plein, cercle vide, rectangle plein, rectangle vide) l'utilisation de deux marqueurs de la librairie **ReactiVision** s'avère utile. Chacun de ces marqueurs définit une caractéristique de la forme. Quand au dessin de ligne ou de polygone il demande uniquement un marqueur mais l'insertion d'un point se fait à l'aide de la commande vocale **point**. Enfin pour terminer un dessin il faut dire **end draw**.

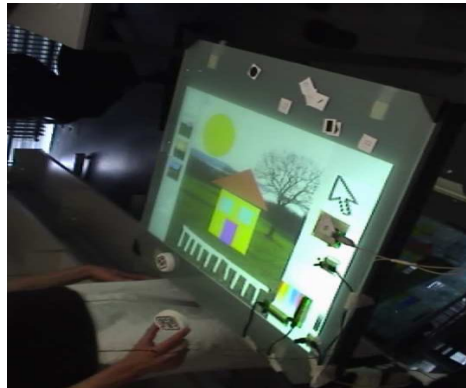
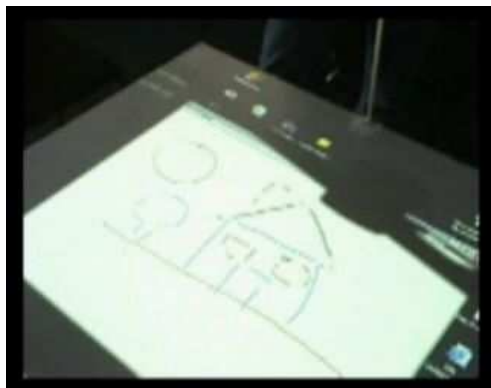
Après avoir dessiné une forme plusieurs opérations sont encore possibles sur celle-ci. Tout d'abord pour accéder à ces opérations il faut sélectionner la forme. Pour cela, il faut que le **rfid** de sélection soit activé. Ensuite il faut pointer sur la forme à l'aide du marqueur de la librairie **ReactiVision**. Puis pour finir il faut utiliser la commande vocale **select shape**. Lorsque la sélection a été effectuée, la forme sélectionnée est mise en évidence. De là, plusieurs opérations sont effectuables, il est possible d'effectuer une rotation de la figure en associant la commande vocale **rotate shape** à une rotation du marqueur de la librairie **ReactiVision**. Il est aussi possible de déplacer la forme en associant la commande vocale **move shape** à un déplacement du marqueur de la librairie **ReactiVision**. Pour finir il est aussi possible de supprimer cette forme en utilisant la commande vocale **erase shape**. L'opération de rotation et de déplacement d'une forme se termine avec la commande vocale **end select**.

Voici donc les différentes possibilités d'utilisation de cette table de dessin.

8 Tests

Nous avons testé toutes les commandes vocales et toutes les options de dessins de manière séparée et/ou conjointe. Toutes les options présentées précédemment fonctionnent correctement.

Après avoir testé toutes les options, nous avons repris le concept de dessin déjà abordé dans l'étude de la librairie **ReactiVision**. Le but étant de dessiner une maison. Cette démonstration du fonctionnement de la table est fournie en annexe dans une petite vidéo. Voici les images du résultat avec la table de dessin et sans :



Nous pouvons voir les nettes améliorations quand à la qualité du dessin. Par contre ce qui est moins évident à voir sur ces photos c'est l'amélioration de l'utilisabilité de cette application de dessin.

9 Evaluation de l'interface

Nous avons basé notre évaluation de l'interface selon l'une des méthodes vues aux cours et plus principalement celle de l'interviews. Nous avons effectué notre évaluation avec la participation de deux de nos collègues Timothée Maret et Nicolas Martenet. Nous les avons suivis durant leurs utilisations du système en les ayant informés au préalable des différentes opérations réalisables. Ils ont très rapidement pris en main la table de dessin et leur avis a été positif quand à l'utilisabilité de celle-ci.

Mais le point intéressant de cette évaluation a été la détection des lacunes qu'ils ont permis de mettre en avant. Grâce à eux nous avons pu voir l'intérêt de l'ajout d'un historique des commandes vocales reconnues par le système. Cet ajout permet d'améliorer considérablement l'utilisabilité du système. Un autre point qui est ressorti de cet interview est la qualité de reconnaissance vocale donnée par la librairie **Sphinx** qu'ils n'ont pas jugée bonne.

10 Améliorations

Malgré le fonctionnement avéré de la table de dessin, un certain nombre d'améliorations peuvent encore y être ajouté. Voici une petite liste non exhaustive des améliorations possibles :

- Une gomme.
- Amélioration de la détection des marqueurs pour pouvoir en utiliser des plus petits. Actuellement légèrement limité par l'intensité lumineuse de la projection. Voir même changement de la librairie de détection de mouvement.
- Ajout d'images de fonds disponibles
- Sélections multiples
- Ajout du concept des layer
- Sauvegarde des dessins.
- ...

11 Conclusions

Notre application réunit plusieurs modalités afin de créer une table de dessin attrayante. Son utilisation s'avère simple et conviviale. Nous sommes néanmoins un peu déçus de la librairie sphinx qui ne dispose pas d'une possibilité d'utilisation en français. De plus, elle est très pointilleuse sur la prononciation ce qui ne nous a pas facilité la tâche. Nous nous attendions aussi à une meilleure reconnaissance des marqueurs, qui est principalement limitée par le fait que la projection sature (trop de luminosité globale) l'image capturée par la caméra. Ce qui nous aurait permis d'utiliser des marqueurs plus petits, mais ce qui n'était pas forcément nécessaire car le rapport taille/utilisabilité des marqueurs courants est très bon. Malgré ces petits inconvénients son utilisabilité et sa simplicité nous ont permis de faire des dessins de meilleure qualité.

12 Références

Couleur HSB :

<http://www.cecs.csulb.edu/~jewett/colors/hsb.html>

Le site de la librairie ReactiVision:

<http://sourceforge.net/projects/reactivision/>

Le site de la librairie Sphinx:

<http://cmusphinx.sourceforge.net/html/cmusphinx.php>

Le site des Phidgets :

<http://www.phidgets.com/>

13 Annexes

Vous trouverez sur le cd les documents suivants :

- Code source
- Vidéos (présentation, concours, dessin vitesse réelle)
- Application reactIVision
- Executable du webservices des phidgets
- Framework .net
- Jdk 5.0
- Executable de notre application (JAR)