Rapport de projet de 4ème année :

Tableau Blanc Interactif

POLYTECH LILLE

Université de Lille 1 SCIENCES ET TECHNOLOGIES

Table des matières

Remerciements

Nous tenions à remercier dans un premier temps toute l'équipe pédagogique de Polytech Lille ainsi que les responsables de la formation Informatique Microélectronique Automatique pour leur aide dans la réalisation de ce projet.

Nous voulions également remercier nos tuteurs de projet à savoir Mme Pichonat, M. Boé et M. Vantroys mais aussi M. Redon pour leur disponibilité, leur aide et leurs conseils notamment au début pour nous aiguiller dans nos choix.

Enfin, nous souhaiterions aussi remercier M. Bernard Carré pour son aide dans la programmation du code JAVA ainsi que M. Thierry Flamen pour son aide dans la fabrication du stylet.

1. Introduction

Les objets connectés et les nouvelle technologies font aujourd'hui partie intégrante de nos vies. On les retrouve dans nos téléphones, dans nos maisons comme avec les consoles de jeu.
Cependant le domaine de l'enseignement semble encore assez peu touché par ce phénomène car le « bon vieux tableau » reste très présent et même si les vidéo-projecteurs se multiplient, l'interaction reste limitée.

Nous avons choisi de travailler sur ce sujet car c'est un projet dont nous pourrions profiter en tant qu'étudiant. Nous tenterons alors de réaliser un dispositif d'interaction via un tableau blanc facilement utilisable par n'importe qui. Cette réalisation nous permettra ainsi de projeter un écran d'ordinateur au tableau et de pouvoir le modifier directement au tableau sans devoir rester en permanence devant de son ordinateur.

Nous commencerons donc par présenter le projet mais aussi le cahier des charges puis nous développerons nos réalisations techniques, en détaillant partiellement notre programme mais aussi les réalisations de chaque structure utile ainsi que les résultats que nous avons obtenus. Enfin, nous terminerons en présentant les problèmes que nous avons rencontrés ainsi que les améliorations auxquelles nous avons pensé.

2. Contexte

Les tableaux blancs à craies ont été pendant longtemps la seule manière d'enseigner, que ce soit à l'école jusqu'aux universités, c'est d'ailleurs encore le cas dans beaucoup de bâtiments, ici-même sur le campus.
Les tableaux blancs ont été vus comme une amélioration singulière de l'ancienne méthode mais cela fait maintenant environ 30 ans qu'il n'y a pas eu de réelle innovation.
En tout cas jusqu'au tableau blanc interactif.
Cependant ces tableaux sont assez onéreux, il fallait donc trouver une alternative peu coûteuse et innovante à ces « vieux » tableaux.

L'idée de ce projet a donc germé lors de la rénovation d'une salle du bâtiment C : la salle C201.
En effet s'il était question d'en faire une salle fonctionnelle pour les TP, pourquoi ne pas lui intégrer un tableau plus évolué qui serait plus fonctionnel pour les explications donnés lors des séances.

En effet, comme dans la plupart des salles de l'école, le tableau était à feutres, cela n'était donc pas optimal pour rendre le TP compréhensible.

3. Mise en place du cahier des charges

La mise en forme de notre cahier des charges n'a pas été simple car notre sujet était très ouvert. Ainsi, nous pouvions choisir de faire ce qui nous plaisait. Cependant il fallait réfléchir à la réalisation matérielle qui nous attendait. Nous avons réfléchi à toutes les technologies à notre disposition et, comme beaucoup de personnes, nous avons pensé aux jeux vidéos.

L'objectif final du projet est d'obtenir un dispositif permettant d'utiliser un tableau comme un écran d'ordinateur et donc d'améliorer les interactions entre les professeurs et les étudiants.

Avant divers entretiens avec nos tuteurs de projet, nous n'avions pas vraiment d'idée pour démarrer ce projet. Or nous avons repensé à l'exemple des consoles de jeu, l'utilisation que nous en faisons reste ce qui nous a été dit de faire : jouer. Mais avec un peu d'imagination, nous pouvons les utiliser pour faire beaucoup d'autres choses.

Suite à cette réunion, nous avons pu établir un cahier des charges qui prenaient en compte les principales idées de notre projet :
- Utilisation d'une manette de Wii afin de capter les mouvements faits au tableau.
- Création d'un stylet infrarouge pour ce contrôle.
- Conception d'un programme réalisant cela.
- Regrouper les données grâce à une Raspberry -pi

4. Découpage des tâches

Pour permettre une dissociation de ces objectifs et donc d'optimiser notre travail d'équipe, nous avons divisé ces objectifs en tâches :

- Comprendre le fonctionnement de la Wiimote
- Se familiariser avec la reconnaissance infrarouge
- Concevoir un algorithme de notre programme
- Se familiariser avec la bibliothèque dédiée à la Wiimote
- Écrire le programme
- Concevoir le stylet
- Programmation de la Raspberry -pi pour le regroupement des données
- Ajout de fonctionnalités au programme si le temps nous le permet

Pour faire cela, nous avons établi un calendrier que nous avons essayé de respecter au maximum :

	Novembre 2015	Décembre 2015	Janvier 2016	Février 2016	Mars 2016	Avril 2016	Mai 2016
Elaboration du cahier des charges							
Documentation sur la réalisation du projet							
Analyse des données transmises par la caméra							
Réalisation du programme et configuration de la Raspberry							
Phases de test							
Adapter les modifications aux résultats des tests							
Test Final dans une salle de cours							13/05 : Soutenance

Échéancier prévu

Pour mener à bien ce projet, nous avons utilisé différents éléments que nous connaissions pour la plupart grâce à notre formation :

Pour le stylet :
- Un stabilo pour la structure
- Une led infrarouge
- Un bouton poussoir
- Une pile 1,5V

Pour le projet :
- Une Wiimote
- Une Raspberry -pi
- Un vidéo-projecteur

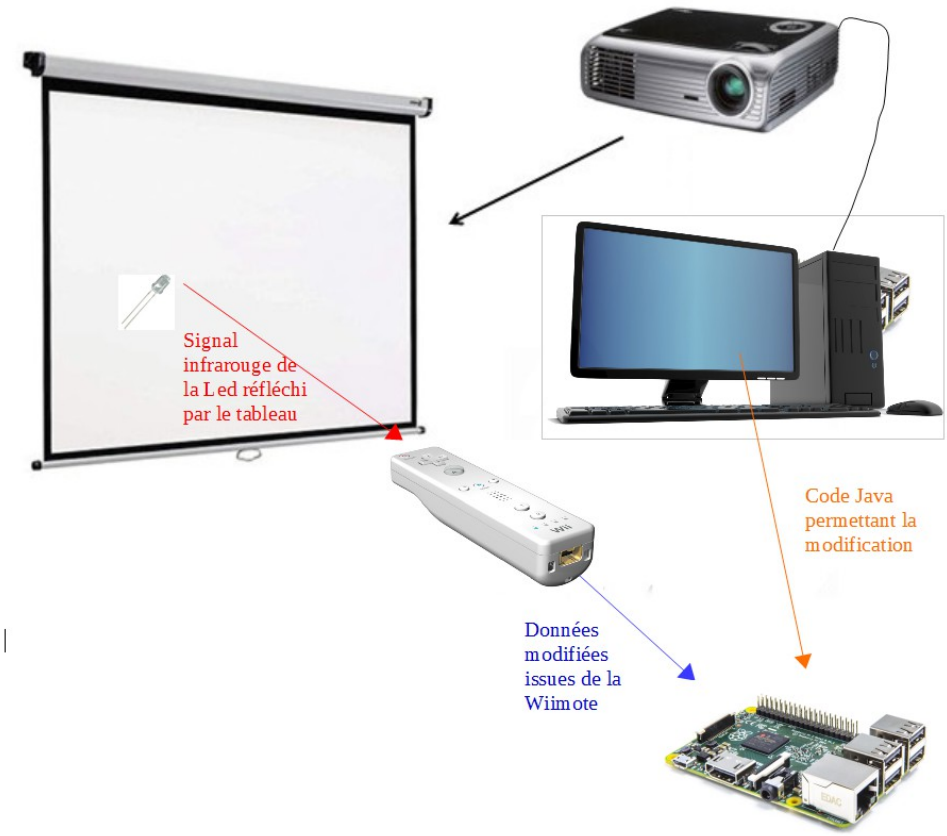


Schéma de base de notre projet

5. La Wiimote

Nous avons décidé d'utiliser une Wiimote car il s'avère que c'était la solution la plus simple pour mener à bien notre projet. Après avoir fait quelques recherches, nous avons appris que ce projet avait déjà été réalisé, cela nous a donc conforté dans notre idée.

La wiimote est normalement utilisé comme émetteur d'un rayon infrarouge étant capté par une balise directement connectée à la console de jeu.
Dans notre cas, la manette ferait office de manette : elle serait la balise immobile qui capterait les mouvements de notre stylet.

Nous allons donc utiliser seulement cette fonctionnalité de la wiimote.
Il faudra la placer à une distance convenable mais aussi modifier l'angle entre elle et le tableau afin qu'elle puisse couvrir toute la surface du tableau.Pour cela il a fallu se renseigner sur la documentation technique associée à la Wiimote.

La caméra de la Wiimote possède un angle de vision de 30° à la verticale et 40° à l'horizontale. Grâce à ces données nous pouvons calculer et donner un ordre d'idée de la distance à laquelle doit se trouver la Wiimote pour couvrir l'intégralité de la surface projetée au tableau.

En partant du principe qu'un vidéoprojecteur diffuse une image sur une surface de 2m x 1.50m afin que cela concorde également avec les dimensions d'un tableau (notamment pour la hauteur) :

Pour faciliter les calculs nous supposerons la Wiimote centrée verticalement et horizontalement.

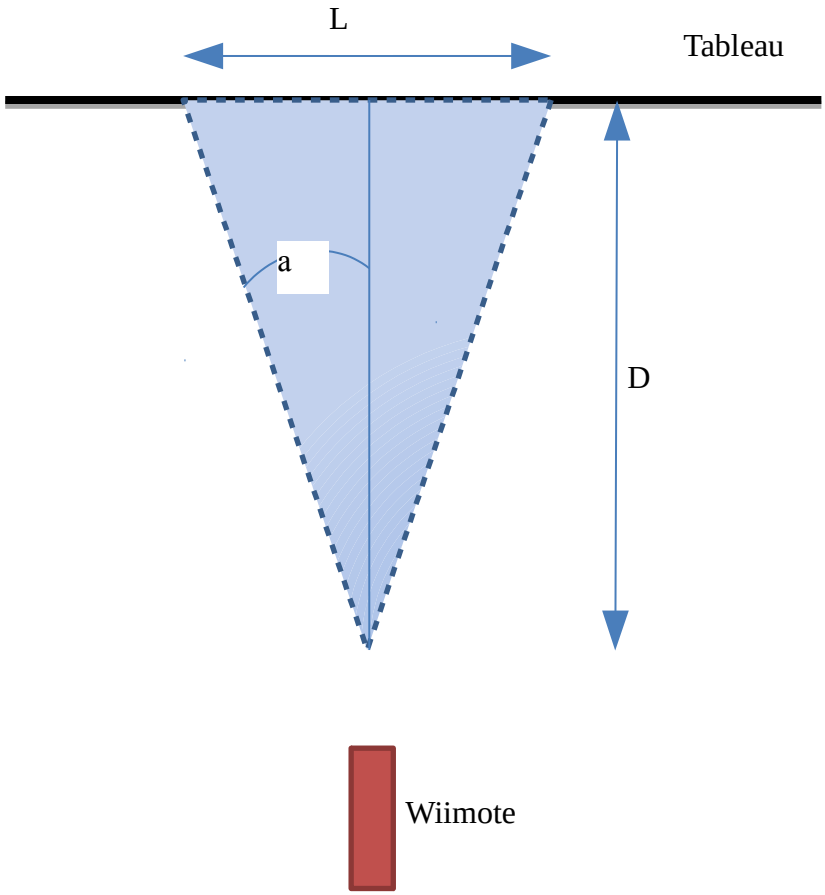
<u>Calculons la distance en fonction de l'axe horizontal :</u>

$$\tan a = \frac{\frac{L}{2}}{D}$$

$$D'où \quad D = \frac{\frac{L}{2}}{\tan a}$$

$$avec\ L = 2m\ et\ a = \frac{40°}{2}$$

Ce qui donne D = 2,8m.



Effectuons désormais le calcul en fonction de l'axe vertical :

$$\tan b = \frac{\frac{H}{2}}{D}$$

$$D'où \quad D = \frac{\frac{H}{2}}{\tan b}$$

$$avec\ H = 1.5m\ et\ b = \frac{30°}{2}$$

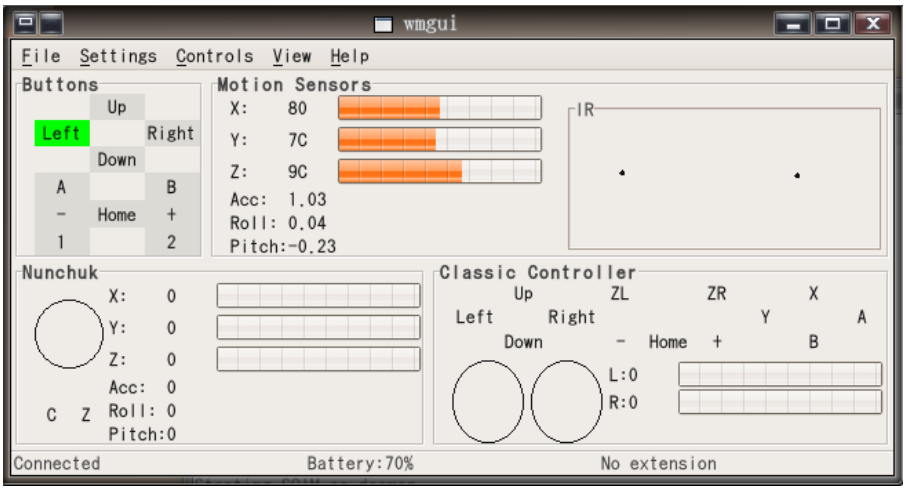
Ce qui donne D = 2,8m.

Cette distance peut être légèrement amélioré en plaçant la Wiimote en hauteur ou sur un côté, Cependant il faut bien diriger la caméra infrarouge afin qu'elle couvre tout le tableau.

Problèmes rencontrés :

Le premier problème que nous avons rencontré en ce qui concerne la Wiimote fut la version. En effet, nous avions reçu une Wiimote Motion Plus, or les informations que nous avons pu recueillir quant à son fonctionnement étaient moins intéressantes que la manette « classique » que nous avons eue plus tard. En effet, le logiciel que nous avons utilisé (wmgui) ne fonctionnait pas avec ce type de manette.

Wmgui est un logiciel qui permet de détecter une manette Wiimote connectée en bluetooth à proximité. Ce logiciel nous permet dans un premier temps de savoir si la Wiimote se connecte bien en bluetooth et si nous communiquons avec elle mais également quelle sont les données qui sont transmises. Ces données sont représentées sous forme d'icône et de jauge afin d'en faciliter la lecture :



L'autre problème associé à la Wiimote fut la batterie car ce genre de manette fonctionne avec des piles et pour les phases de tests, nous avons dû recharger plusieurs fois ces piles. Or cela n'est pas pratique pour l'utiliser en tant que support de cours. Il fut évoqué de relier l'alimentation de la manette directement sur un port USB du vidéoprojecteur ou sur le réseau électrique tout en adaptant les caractéristiques électriques afin de correspondre avec celles de la Wiimote (3.5V, 20mA). Cependant, faute de temps nous n'avons pas pu mettre en place ce moyen.

6. Programme

1. Le programme principal

Développement :

Pour la réalisation de notre programme, nous avons commencé par nous documenter sur les différentes librairies existantes pour la Wiimote. De plus, nous avons décidé de coder ce programme en JAVA dans un soucis de simplicité graphique. Il nous semblait alors plus simple de réaliser une interface de contrôle en JAVA plutôt qu'en C. Nous avons donc étudié la librairie WiiRemoteJ qui est la libraire JAVA dédiée à la Wiimote afin de rassembler tout ce qui pourrait nous être utile.

Puis nous avons essayer d'établir un algorithme, il nous a donc fallu réfléchir aux fonctions qui nous seraient utiles et qui sont réalisables ou déjà existantes dans la librairie. Dans cette optique, nous avons étudier le fonctionnement d'un tableau blanc intéractif pour établir une suite d'action à réaliser.

Tout d'abord il faut gérer l'état de la Wiimote : si elle n'est pas allumée, il n'y a aucune chance pour notre programme fonctionne. Ainsi nous avons décidé d'utiliser une classe qui nous donnerait ces informations sur la connexion de la manette ou sur l'état de la batterie. Une fois cette partie fonctionnelle, nous pouvons nous concentrer sur l'interaction entre la Wiimote et la source infrarouge.

Nous allons donc commencer par la détection du stylet par la manette. En effet, dès que la manette est allumée, il faut qu'elle parcourt sa fenêtre de détection jusqu'à détecter un signal infrarouge. Une fois cela effectué, la manette va se synchroniser sur cette source et cela va nous permettre de synchroniser la souris avec le stylet.
Il faut cependant gérer le fait qu'il puisse y avoir un problème dans le maintien de la connexion (une panne de batterie ou un dysfonctionnement du stylet par exemple) et donc effectuer cette recherche en permanence dès que la Wiimote n'est plus connectée.

Puis il y a la définition du tableau et donc le calibrage. Cette partie est essentielle pour le bon fonctionnement du programme. Il faut en effet que le stylet reste dans un espace prédéfini. Pour cela, il faut définir 4 points (un dans chaque coin de la fenêtre) que le stylet validera afin de définir le tableau.
Ainsi nous pouvons maintenant créer les fonctions de mouvement du stylet et la récupération des données issues de ces mouvements.

Enfin, il y a l'interface graphique à configurer. Cette partie fut assez compliquée car ce n'est pas le point le plus développé du cours que nous avons pu suivre ce semestre. Il a donc fallu faire plus de recherche afin d'avoir une interface assez esthétique mais surtout fonctionnelle :

Interface graphique du programme principal

<u>Problèmes rencontrés :</u>

Le principal problème que nous avons rencontré lors de cette étape fut la compréhension de la librairie utilisée, en effet le JAVA n'est pas un langage que nous maîtrisions parfaitement surtout au début du projet quand nous n'avions que quelques heures de cours. Il a donc fallu être patient et recommencer plusieurs fois les mêmes tests pour ne plus générer d'erreurs.

Malgré tout, nous avons aussi remarqué que,chaque ordinateur ne possède pas forcément toutes les libraires dont notre programme a besoin, il a donc fallu les réinstaller lorsque nous changions de machine.

Le click fut aussi un problème pour nous car nous voulions utiliser seulement un bouton mais nous avions deux actions différentes à faire : le click droit et le click gauche. Nous avons alors décidé de nous concentrer sur le click gauche qui nous semblait être le plus important pour modifier l'écran projeté au tableau. Nous avons donc essayé d'ajouter une temporisation pour le click droit : il fallait rester appuyé 1s sans bouger pour déclencher le click droit, malheureusement, après plusieurs tests, nous n'avons pas réussi à le faire fonctionner.

2. Le JPAINT

Développement :

Nous avons donc un programme qui nous permet d'interagir avec notre écran d'ordinateur directement au tableau. Cependant rien ne nous permet d'éditer cette projection...

Nous avons alors décidé d'ajouter une application à notre programme, pour cela et au vu du temps restant pour la faire, nous avons décidé de faire « une sorte de Paint » et donc d'utiliser les mêmes outils afin d'annoter une page projetée au tableau et de pouvoir la sauvegarder.

Pour cela nous avons divisé le programme en 2 sous-parties : l'initialisation et la sélection des outils et la modification de la page chargée.
Il faut en effet initialiser la fenêtre du programme, notamment les outils qu'on voudra utiliser.

On utilise donc plusieurs classes pour les couleurs mais aussi pour les différents types de modification du fichier : le stylo, le feutre par exemple sans oublier la gomme, notamment la classe ToolBox (voir annexe 1) qui initialise les objets puis dans une autre fonction instancie dans une boucle l'utilisation des outils.

Une fois cette initialisation faite, il ne reste plus qu'à attendre que l'utilisateur choisisse un élément ou une action. Cette partie repose essentiellement sur l'utilisation de différents types de « listeners » et sur les actions à réaliser si l'un d'entre eux se déclenche. Alors dès qu'un événement d'un type voulu survient, on aura une suite d'action qui va être réalisée. Ensuite nous réunissons toutes ces fonctionnalités dans une seule classe : ControlClass qui créé les différentes palettes, le menu, les outils... Cette classe permet donc de modifier le fichier charger ou la page blanche initiale.
Il ne reste ensuite plus qu'à créer une variable dans la classe Main que nous lancerons une fois compilée.

Problèmes rencontrés :

Pour coder ce programme, le principal problème fut de l'adapter à notre projet. En effet un « Paint » n'a pas forcément d'intérêt lors d'un cours, mais nous avions prévu de l'utiliser pour charger l'écran projeté au tableau et ainsi pouvoir le modifier. C'est en réalité un problème de fenêtre car il nous a été impossible de charge la fenêtre représentant l'écran : la fenêtre de base.

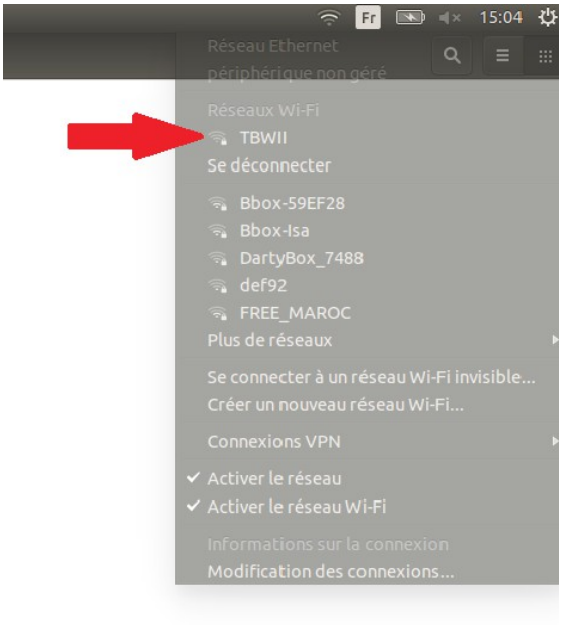
7. La Raspberry –pi

Développement :

Concernant la Rapsberry-pi nous nous sommes d'abord tournés sur le fait qu'elle serait au centre de notre projet. C'est-à-dire que ce serait via la Raspberry-pi que nous lancerions le programme et qu'elle serait reliée au vidéoprojecteur.

Nous avons donc commencé par configurer le Bluetooth dans cette optique-là. En effet la manette Wiimote qui récupère le signal infrarouge communique cette données via Bluetooth, d'où la nécessité d'établir une connexion Bluetooth. Cette configuration est possible grâce à diverses librairies associées : libbluetooth, bluez, bluez-utils, hcitools.

Ensuite il a fallu rendre la raspberry-pi en point d'accès Wifi :



Attention, le mot de passe sera demandé lors de la connexion Wi-Fi, le mot de passe étant : « raspberry ».

Ainsi toute personne présente dans la salle de cours où se trouve la raspberry-pi peut seconnecter au réseau TBWII et ainsi peut avoir accès à la raspberry via la commande suivante :

```
root@mazierleo-Satellite-C660:~# ssh pi@192.168.100.1
```

Arrivé à ce stade nous avons remarqué que nous ne pouvions pas bloquer les utilisateurs de la raspberry-pi de s'identifier en tant que root. En effet nous pouvons bloquer les connexions entrantes identifier comme root à l'aide du fichier sshd_config situé dans /etc/ssh/sshd_config. Mais une fois connecté à la raspberry-pi nous n'avons pas trouvé le moyen de bloquer les utilisateurs de se connecter en tant que root car le nom d'utilisateur utilisé pour se connecter, « pi », est un super-utilisateur.

Nous avons alors décidé de transformer la raspberry-pi en simple espace de stockage des fichiers à projeter au vidéoprojecteur. Le principe est que seul l'odinateur du professeur sera désormais relié au vidéo-projecteur et que, via la commande scp, celui-ci peut copier des fichiers sur la raspberry-pi. Ainsi chaque étudiant peut récupérer les fichiers distribués par le professeur. Les étudiants peuvent également tranférer leur propres fichiers sur la raspberry-pi pour qu'ils puissent être repris par le professeur et projetés au tableau grâce à la même commande scp :

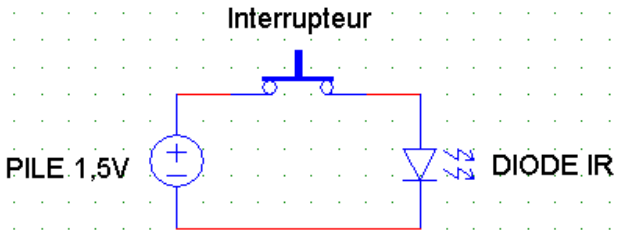
```
root@mazierleo-Satellite-C660:~# scp -r WiimoteWhiteboard pi@192.168.100.1:/home/pi
```

Ainsi seul l'ordinateur du professeur est relié au vidéo-projecteur et au logiciel.

8. Stylet

Développement :

Concernant le stylet, nous avons eu besoin d'un phototransistor infrarouge et d'un bouton poussoir. Ensuite nous avons donc créé un circuit reliant une pile 1.5V, 2600 mAH au bouton-poussoir puis au phototransistor comme indiqué ci-dessous :



Nous avions dans un premier temps opté pour une structure à base d'un stabylo, puis pour des raisons esthétiques et pratiques nous avons décidé de fabriquer un stylet à l'aide du fablab (voir Annexe 2). Nous avons placé le phototransistor à l'extrémité la plus proche du tableau car cela permet une meilleure précision. Mais c'est ce détail qui nécessite la présence d'un tableau blanc « à feutres » car il faut que le signal émis par le phototransistor se reflète sur celui-ci afin d'être capté par la caméra infrarouge de la Wiimote située devant le tableau.

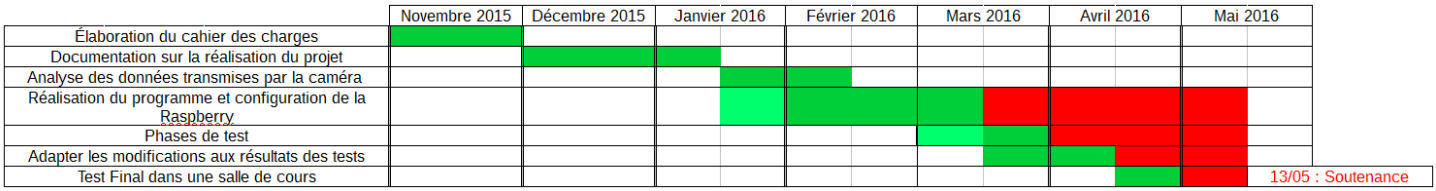
Problèmes rencontrés :

La composition du stylet est assez simple cependant, sa confection fut assez délicate car nous avions eu l'idée d'utiliser un feutre de tableau comme structure mais notre bouton poussoir était trop grand, de plus le feutre n'était pas assez profond pour contenir la pile et la led. Nous avons donc dû réaliser une structure adéquate pour pouvoir contenir tous les éléments du stylet.

Or, une fois cette structure imprimée, malgré quelques problèmes d'impression, nous nous sommes rendus compte qu'elle était maintenant trop large et donc la pile ne cessait de bouger, cela générait donc des court-circuits.

9. Résultats

Tout d'abord, nous allons détailler l'échéancier que nous avons utilisé bien qu'il diffère un peu de celui que nous étions fixé :



Échéancier réel

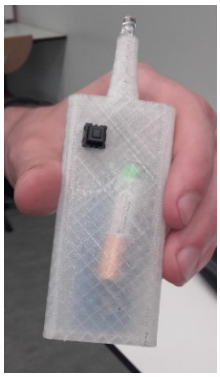
En effet, au vu des problèmes que nous avons évoqués précédemment dans nos différentes parties, nous avons pris du retard et ce dès la première semaine à cause du type de la manette puis le programme nous a retardé. Enfin, le programme d'édition n'était pas fonctionnel comme nous le désirions, nous avons donc continué de travailler dessus jusqu'à la dernière semaine.

Même si nous n'avons totalement respecté notre échéancier initial, nous avons tout de même certains aboutissements.

Tout d'abord le programme qui nous permet de nous servir d'une led infrarouge comme d'une souris fonctionne, on peut naviguer sur le tableau comme sur un écran d'ordinateur.

Pour l'améliorer, il faudrait retravailler la partie sur le click droit qui n'est pas encore fonctionnelle. Il suffirait peut-être de synchroniser les deux clicks sur deux périodes distinctes afin que le programme puisse les distinguer.

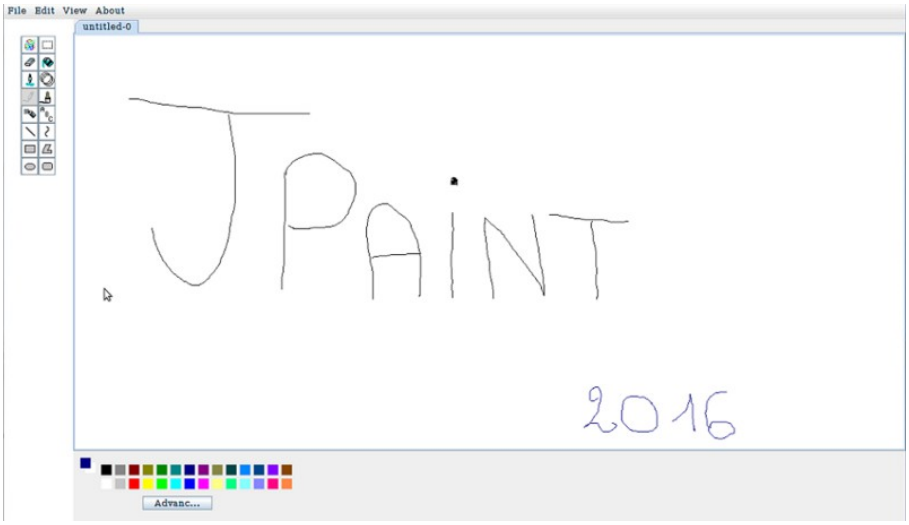
Le stylet fonctionne donc lui aussi parfaitement bien, il remplace l'ancien feutre comme nous le souhaitions :



Pour l'améliorer, il faudrait revoir sa structure afin de ne pas avoir à souder la pile et plutôt pouvoir la remplacer. Il suffirait de « remplir » la structure afin que la pile ne bouge pas.

La raspberry fonctionne elle aussi bien que son utilisation ne soit pas celle prévue, nous lui avons trouvé une nouvelle utilité qui permet au professeur d'interagir plus facilement avec ses étudiants.

Le programme d'édition quand à lui fonctionne aussi si nous voulons modifier des images (donc fichiers PNG, JPG, GIF). Cependant on ne peut pas s'en servir pour modifier des fichiers PDF ou de présentation tels que les fichiers ODP.



Interface de notre programme d'édition : le JPaint

Pour l'améliorer, il faudrait réussir à charger l'écran. Mais après plusieurs essais non aboutis, nous pensons qu'il faudrait peut-être trouver une autre idée pour l'édition ou alors choisir un logiciel qui existe déjà.

En ce qui concerne la manette, nous avons utilisé des piles rechargeables mais il serait plus intéressant de la relier directement en USB au vidéo-projecteur. Il faudrait juste faire attention à la tension délivrée par ce port.

Conclusion

Après ces 13 semaines de travail, nous pouvons dire que nous avons réalisé quelque chose de fonctionnel mais qui n'est pas encore terminé comme nous l'avons expliqué précédemment, cependant, il est utilisable en l'état.
Ce projet nous a permis de développer nos connaissances en JAVA, et donc de mieux comprendre le cours que nous avons suivi. Cela nous a aussi permis d'approfondir les connaissances acquises en TP et de ne pas se limiter au programme.

De plus, ce projet nous a permis de revoir la configuration d'une RaspberryPi et d'utiliser l'imprimante 3D, chose que nous n'avions pas l'habitude de faire.
Ce projet nous a aussi permis de nous documenter sur une technologie déjà existante mais aussi de faire des choix dans la liste de composants afin d'obtenir le projet le plus fonctionnel possible dans le temps qui nous était donné mais aussi le moins cher possible.

Ce projet nous a donc permis d'améliorer nos compétences en travail de groupe mais aussi nos compétences en programmation.

Ce fut donc un projet enrichissant par les compétences que nous avons pu développer mais aussi par sa gestion, il fallait en effet établir notre propre cahier des charges ainsi qu'un échéancier auquel il fallait se tenir.

ANNEXE 1:

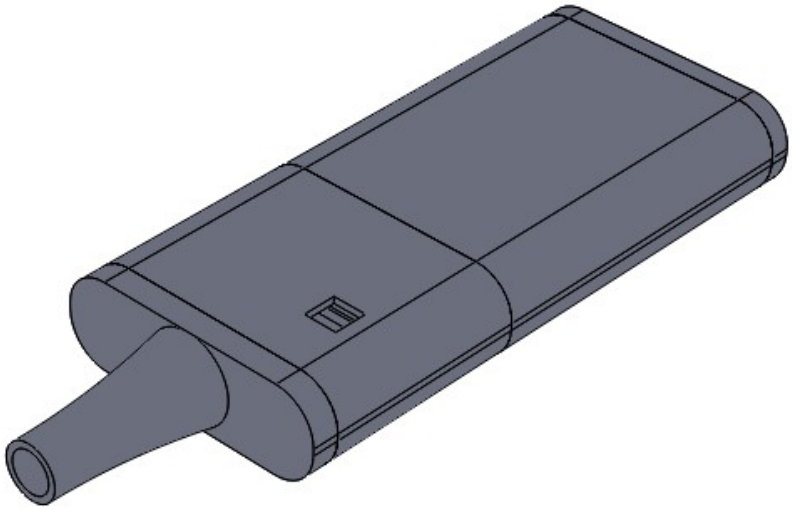
Classe ToolBox.java

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Image;
import javax.swing.JPanel;

public class ToolBox extends JPanel
{
        public static int RANDOMDRAW = 0, ERASER = 1, EYEDROPPER = 2, PENCIL = 3,
        SPRAY = 4, LINE = 5, RECTANGLE = 6, OVAL = 7, ROUNDRECTANGLE = 8,
        POLYGON = 9, CURVELINE = 10, TEXT = 11, BRUSH = 12, DRAG = 13, FILL = 14,
        SELECT = 15;
        public static ToolIcon[] tools;
        public static int toolSelected;

        /** Creates a default ToolBox. **/
        public ToolBox()
        {
                // sets the normal settings and calls the run method
                setLayout(null);
                setSize(200,300);
                run();
        }

        public void run()
        {
                tools = new ToolIcon[16];
                setBackground(Color.white);
                for(int i=0;i<tools.length;i++)
                {
                        String iconPath = "Images/"+i+".gif";
                        tools[i] = new ToolIcon(iconPath,i);
                        if(i<8)
                                tools[i].setBounds(25,i*25+25,25,25);
                        else
                                tools[i].setBounds(50,(tools.length-i)*25,25,25);
                        add(tools[i]);
                        repaint();
                }
                // sets the initial default tool the pencil tool
                toolSelected = PENCIL;
                tools[toolSelected].setEnabled(false);
        }
}
```

ANNEXE 2 :

Voici les schémas SolidWorks :

La structure :



Le cache pour changer la pile :