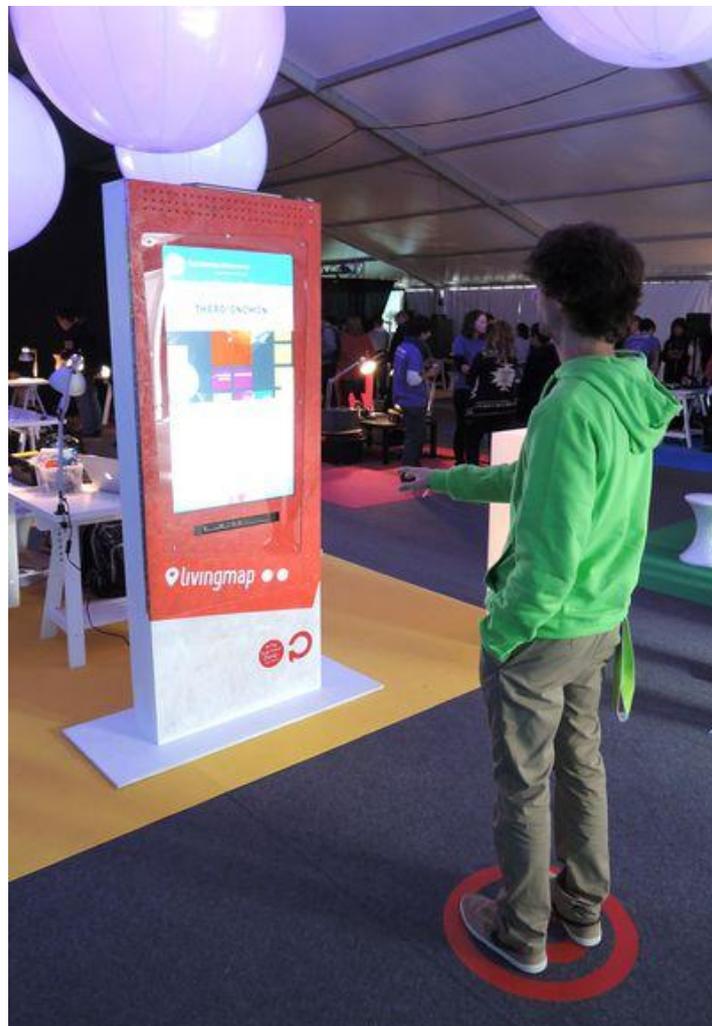


Christophe Adam  
Zhangmeng Zhang

# Rapport de projet

## Interactive Digital Signage (IDS)



Superviseurs : Amr Alyafi, Didier Donsez  
Département : RICM4, Polytech Grenoble

Année 2014-2015

# Présentation du projet

Le but du projet est d'améliorer la borne interactive qui a été commencée il y a deux ans par des élèves de RICM5. La borne aura pour but d'informer les étudiants à propos des dernières nouvelles sur Polytech et de leur permettre de consulter leur emploi du temps.

La borne doit pouvoir :

- afficher des informations sur Polytech de façon autonome (les informations seront dans un diaporama reveals.js).
- afficher l'emploi du temps des différentes filières.
- indiquer le prochain cours lorsqu'une personne passe une carte NFC devant le lecteur NFC.
- être autonome (en mettant les emplois du temps à jour régulièrement et automatiquement).

## Contexte du projet

Les bornes interactives à la pointe de la technologie commencent à apparaître dans nos villes. Au potentiel infini, elles vont révolutionner le monde de l'information et de la publicité.

**Voici quelque exemple :**

Dans les grands complexes pour s'orienter :



Dans les magasins pour présenter les produits :



Dans les centres commerciaux pour s'orienter :



Dans la rue pour faire de la publicité :



## Matériel

- Borne IDS contenant
  - Intel NUC D54250WYKH avec Windows 7
  - TV 37 Pouces Full HD
  - Verre trempé IKEA
  - Kinect Xbox
  - Haut Parleurs
  - Lecteur NFC (SCM Microsystems Inc. SCL3711 reader & NFC device 0)
- Tag NFC
- Téléphone Android (pour programmer les puces NFC)

Lecteur NFC SCL3711 :



Kinect:



## Logiciels et API utilisés

- <http://voxygen.fr/index.php> (synthèse vocale)
- javax.smartcardio (lecteur NFC)
- Reveals.js (affichage interactif des informations)
- HTML5 / CSS3 / PHP (page web)
- Webbit Server (seveur web)
- Java (emploi du temps et NFC)
- OpenNI (Kinect)
- WampServer (seveur web pour Voxygen)

## Logiciels réalisés

Le dépôt global des sources est sur **github**.

Voici la liste des différentes applications que nous avons réalisées dans le but de répondre au cahier des charges.

### 1) Emploi du temps et NFC

Le programme créé tourne en tâche de fond sur la borne. Dès qu'une puce NFC est détectée, une fenêtre s'ouvre avec les informations sur le prochain cours de la personne, la salle dans laquelle elle sera, son enseignant ainsi que les horaires du cours. Au moment de l'ouverture de cette fenêtre, la borne se met à parler à la personne qui est en face d'elle pour lui donner verbalement ces informations via la synthèse vocale. À la fin du speech, la fenêtre se referme.

#### Fonctionnement du programme :

Le schéma ci-dessous résume la façon dont le logiciel fonctionne de façon schématique. Tout d'abord, lors de son premier lancement, le programme télécharge les emplois du temps de chaque filière au format iCalendar. Ensuite, les fichiers iCalendar sont traités et les informations utiles sont stockées en mémoire dans des structures de données. Maintenant vient le début de la boucle principale de fonctionnement. Il ne sortira jamais de cette boucle sauf en cas d'arrêt du programme évidemment. Le programme est autonome. Il se mettra à jours automatiquement. Lors qu'une personne passe une puce NFC devant la borne, le programme interroge ses structures de données contenant les emplois du temps comme expliqué dans le paragraphe qui suit. Ensuite, il vérifie qu'il dispose bien des emplois du temps de la semaine courante et effectue la mise à jour si nécessaire. Il affiche alors le résultat de la requête dans une fenêtre et active la synthèse vocale qui fait parler la machine.



### Interrogation des structures de données :

Le schéma ci-dessous résume la façon dont le logiciel procède après identification de la personne qui a passé la puce NFC devant la borne. Ce qu'il faut retenir, c'est que si 'est un élève, c'est facile, il suffit d'aller interroger la liste correspondant à l'emploi du temps des la classe de l'étudiant. Ensuite on cherche une correspondance entre l'élève, la date et l'horaire. En revanche, dans le cas d'un enseignant, nous ne disposons pas de la classe avec laquelle il a cours puisque qu'un enseignant peut faire cours à plusieurs classes de différentes filières. Il faut donc chercher dans toutes les listes contenant les emplois du temps des différentes filières une correspondance entre l'enseignant, la date et l'horaire.

[04647989/eleve/Christophe/Adam/RICM4/RICM4 - Op. Multimédia](#)

[04645695/enseignant/DECHAMBOUX BOYER Fabienne](#)



## Mise à jour automatique des emplois du temps :

C'est une histoire de différence entre la date correspondant à la dernière fois que les emplois du temps ont été mis à jours et la date actuelle en tenant compte du jour de la semaine où la demande est effectuée. Un code tel que celui-ci est robuste et fonctionnera toujours. Dans le cas présent, les emplois du temps sont mis à jour chaque semaine.

```
/******téléchargement des emplois du temps automatiquement*****/  
Date now = new Date();  
String today = now.toString();  
today=today.substring(0,3);  
long diff = now.getTime( ) - datePre.getTime( );  
diff=(diff/(1000*60*60*24));  
  
switch(today)  
{  
case "Mon" : if(diff>0){Download(); datePre=now;}  
break;  
case "Tue" : if(diff>1){Download(); datePre=now;}  
break;  
case "Wed" : if(diff>2){Download(); datePre=now;}  
break;  
case "Thu" : if(diff>3){Download(); datePre=now;}  
break;  
case "Fri" : if(diff>4){Download(); datePre=now;}  
break;  
case "Sat" : if(diff>5){Download(); datePre=now;}  
break;  
case "Sun" : if(diff>6){Download(); datePre=now;}  
break;  
}  
}
```

## Info sur le programme

Le programme a été écrit intégralement en Java.

Les emplois du temps de chaque filière sont téléchargés automatiquement dès la première demande qui est effectuée.

La synthèse vocale fonctionne grâce au site web de l'entreprise Voxygen qui permet d'utiliser des voix à la pointe de la technologie qui ont été mise en ligne comme exemple.

Le texte du speech est créé dynamiquement par le programme pour s'adapter à chaque utilisateur. Ensuite, il est envoyé sur le serveur de Voxygen qui va alors renvoyer un fichier audio au format mp3 avec le contenu vocal souhaité. Enfin, le programme lance la lecture du fichier son pour faire parler la borne.

Les adresses web pour télécharger les emplois du temps de chaque filière se trouve dans un fichier nommé "adweb.txt". Il est donc très simple d'ajouter des classes, d'en enlever ou de mettre à jours une adresse qui aurait changé.

Le fichier est construit de la façon suivante :

```
RICM4 http://ade6-ujf.grenet.fr/jsp/custom/modules/plannings/anonymous_cal.jsp?resources  
TIS4 http://ade6-ujf.grenet.fr/jsp/custom/modules/plannings/anonymous_cal.jsp?resources=  
PRI4 http://ade6-ujf.grenet.fr/jsp/custom/modules/plannings/anonymous_cal.jsp?resources=  
GEO4 http://ade6-ujf.grenet.fr/jsp/custom/modules/plannings/anonymous_cal.jsp?resources=  
MAT4 http://ade6-ujf.grenet.fr/jsp/custom/modules/plannings/anonymous_cal.jsp?resources=  
3I4 http://ade6-ujf.grenet.fr/jsp/custom/modules/plannings/anonymous_cal.jsp?resources=9
```

Les puces NFC contiennent simplement un numéro d'étudiant (numéro d'étudiant par exemple). Une fois le numéro lu, il suffit d'aller chercher l'étudiant correspondant dans la base de données qui est en fait un simple fichier texte nommé "nfc.txt". Une des améliorations de notre projet sera d'aller chercher les informations dans la base de données en ligne qui a été réalisée par l'université. Ce sera ainsi plus simple que de rentrer tous les étudiants avec leur numéro d'étudiant, leur nom, leur prénom et leur classe dans un fichier.

Le fichier est construit de la forme suivante :

```
1 Christophe Adam RICM4  
2 Zhengmeng Zhang RICM4
```

Pour ce qui est de la voix choisie, c'est également dans un fichier que l'on peut la choisir. Dans le fichier "speak.txt", il faut donc indiquer la voix que l'on veut utiliser et également indiquer le chemin du dossier qui contient la partie du code PHP qui est utilisé par WampServer chaque fois qu'une requête est effectuée.

Le fichier est construit de la forme suivante :

```
Yeti  
C:/wamp/www/PHP-Voxygen-master/
```

Fenêtre du programme (cas d'un élève) :



La machine dira par exemple : "Bonjour Christophe, vous avez cours de (CAR) Applications réparties en salle 052 avec Dechamboux Boyer Fabienne de 8h00 à 9h30. Je vous souhaite une excellente journée."

Fenêtre du programme (cas d'un enseignant) :



La machine dira par exemple : "Bonjour Didier, vous avez cours de projet en salle TP259IIA ou peut être en salle S125 avec les RICM4 de 13h30 à 17h30. Je vous souhaite une excellente journée."

### Info sur la partie en PHP Voxygen

Le code PHP est exécuté par WampServer. Le dossier "PHP-Voxygen-master" doit donc se trouver dans le dossier "www" de WampServer pour que tout fonctionne correctement. Il faut donc bien mettre le lien vers le dossier "PHP-Voxygen-master" dans le fichier "speak.txt" comme expliqué un peu plus haut sinon ça ne pourra pas fonctionner.

## 2) Kinect et Reveals.js

Pour utiliser la Kinect sur Windows il faut installer :

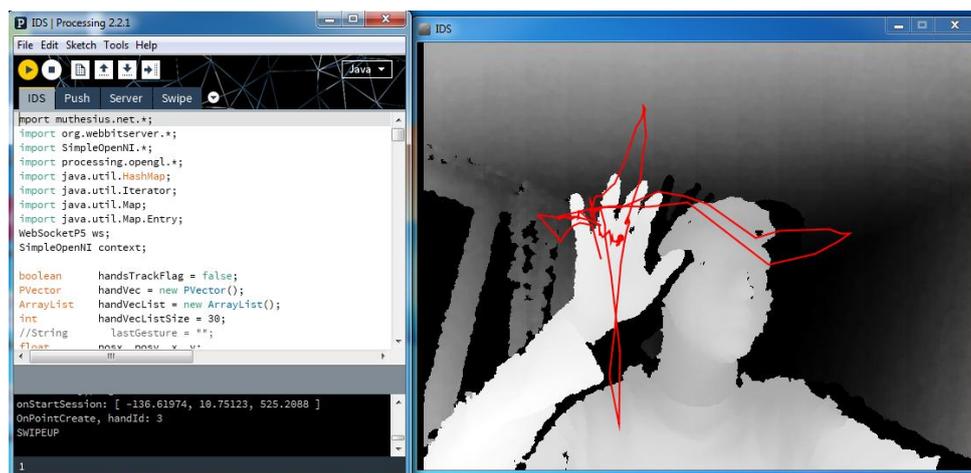
- **nite-win64-1.5.2.21-dev**
- **openni-win64-1.5.4.0-dev**
- **SensorKinect092-Bin-Win64-v5.1.2.1**

Il faut faire attention lors de l'installation de la kinect sous Windows car il y a un conflit de pilote entre SensorKinect et Windows Kinect. Il faut bien utiliser SensorKinect.

Nous créons un serveur avec Webbit et nous gérons la communication entre Processing et la page web en utilisant WebSocket.

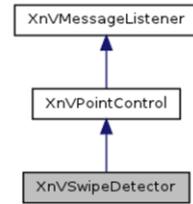
Pour détecter les gestes nous utilisons l'API OpenNI.

Pour utiliser la détection des gestes dans reveal.js nous utilisons les fonctions de NITE API qui sont dans OpenNI.

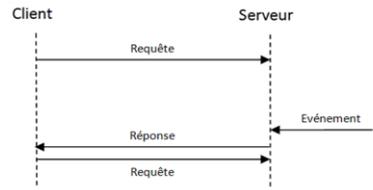


Processing

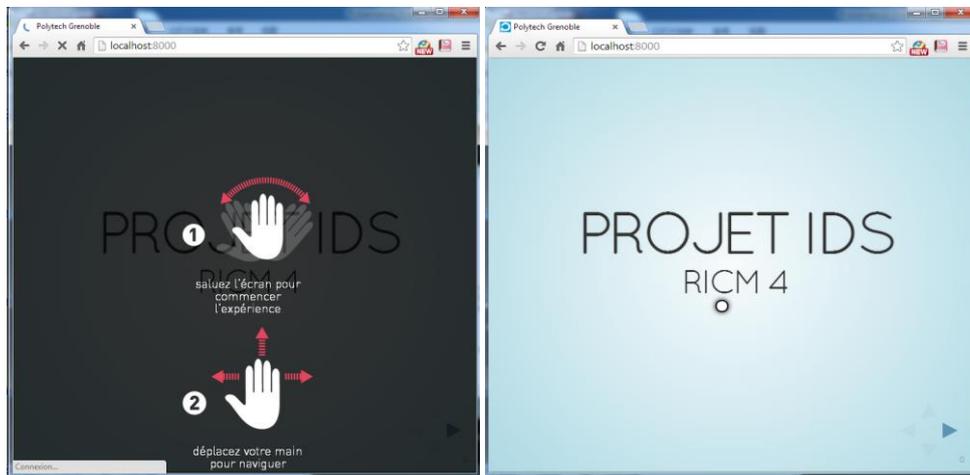
\*NITE API: XnVSwipeDetector



\* webbit server  
\* websocket



## WebSocket, Webbit Server, NITE API et OpenNI



Page web

Au niveau du fonctionnement, c'est assez intuitif. Il suffit de saluer la borne pour que notre main soit détectée puis de balayer avec notre main devant l'écran de la borne pour faire défiler les slides.

## Conclusion :

- Tout fonctionne correctement
- Le cahier des charges a été respecté
- Nous avons appris plein de nouvelles choses