Google Glass en logistique

Rapport final de PFE

Tuteur école : Laurent Grisoni
Tuteur entreprise: Laurent Borel

Vincent Meunier
Jérémy Gondry
IMA5 promo 2015

POLYTECH LILLE

Université Lille1 Sciences et Technologies

Sommaire

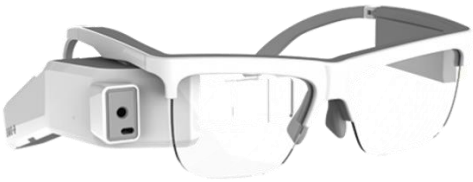
1 Introduction

1.1 Présentation Entrepôt/Problématique

Notre projet vise à l'optimisation du stockage et de l'adressage des palettes dans un entrepôt. En effet, lors du déchargement d'un camion, chercher un emplacement libre pour les palettes dans un entrepôt occupant une grande superficie fait perdre du temps et donc de la productivité dans le stockage. Sachant qu'un cariste perd un temps non négligeable à la recherche d'emplacements disponibles, optimiser cette tâche de recherche permettrait de gagner du temps sur cette opération.
Notre projet consiste donc à fluidifier le déchargement en indiquant au cariste l'emplacement libre le plus proche.

Développer un prototype permettrait de rendre compte des moyens existants qui pourrait être mis en place dans une solution métier.

Google Glass



ORA d'Optivent

Pour ne pas gêner le travail des employés, un dispositif main-libre comme une paire de Google Glass semble adapté. Ce projet soulève plusieurs problématiques : développer une interface homme-machine apportant une information pertinente pour l'utilisateur sans pour autant être perturbante.
L'idée serait de réaliser dans les grandes lignes un système similaire à ce scénario d'utilisation : http://n.mynews.ly/!lB.Bs4fw

Une évolution proposée du sujet est de récupérer certaines informations sur les magasiniers (nombre de colis délivrés, classement…) afin de les afficher sur un grand écran. L'idée étant d'améliorer la productivité en se basant sur un principe de compétition saine.

1.2 Cahier des Charges

En partant de l'analyse du sujet et des discussions avec notre tuteur, nous avons défini un cahier des charges plus précis des tâches à accomplir:

- **Développer une application embarquée sur une paire de lunette Smart Glass**
    - Comparer les différentes offres en matière de Smart Glass notamment Google Glass & ORA de Optinvent
      Les Google Glass ne sont pas les seules smart glasses sur le marché. Il nous faudra choisir les appareils les plus adaptés.
    - Concevoir une interface homme-machine non perturbante pour l'utilisateur.
    - Proposer un système de reconnaissance vocale en français pour l'interaction main libre avec l'utilisateur.
- **Etudier la nécessité d'utiliser une géolocalisation en entrepôt et l'implémenter si besoin**
      L'information pertinente pourrait être une direction à suivre dans l'entrepôt ou simplement un numéro d'emplacement (l'adressage de l'entrepôt est intuitif).
    - Via RSSI (Wifi, Bluetooh Low Energy ...)
    - Via traitement d'images (reconnaissance de la signalétique en entrepôt)
- **Etudier les possibilités d'optimisation du système d'adressage des palettes**
      En ce qui concerne l'adressage des palettes en entrepôt, la solution existante repose sur un flashage des codes barres associés à l'emplacement venant d'être occupé. Une étude de faisabilité/utilité sera aussi faite sur une solution de remplacement grâce aux possibilités qu'offrent les smart glasses.
    - Implémenter la reconnaissance de code barre
    - La tester en réalité métier (utilité ajoutée, habitude des magasiniers, vitesse de traitement...)
- **Réfléchir sur un système pour afficher des informations personnelles**
    - Proposer plusieurs architectures possibles d'affichage.
    - Faire une connexion avec une base de donnée pour récupérer les informations nécessaires.
    - Etudier les informations utiles et possibles à afficher sur l'écran.
    - Implémenter une méthode de détection pour repérer l'utilisateur le plus proche afin d'afficher ses données directement.

2 Etude du Besoin

2.1 Réalité métier

Tout au long de notre projet il nous aura été demandé de réfléchir sur nos réalisations et de prendre du recul par rapport à notre travail. En effet, le concept de réalité métier a souvent été abordé pour savoir si un magasinier pourrait réellement se servir de notre application, et si cela lui apporte une vraie plus valu dans son travail. Dans cette optique de réflexion plusieurs sujets on été abordés:

- Une interface graphique peu intrusive
- Une information pertinente et simple à récupérer
- Un système main libre qui ne gène pas l'utilisateur

Ce sont ces trois axes qui nous ont guidés lors de la création de l'interface des Google Glass.

En ce qui concerne l'affichage sur la télévision, il faut bien réfléchir quant aux informations à afficher pour ne pas entraîner une approche méfiante des utilisateurs. De plus le système devra être robuste et facilement déployable sur plusieurs sites.

2.2 Possibilités et limites des SmartGlasses

	Google Glass	**ORA**
Connectique	Compatible BLE (à partir de la XE 16 KitKat Update), Wifi	Compatible BLE, Wifi
Autonomie en utilisation normale	de 1h à 2h	de 4h à 8h
Son	Conductivité Cranienne	Prise jack
Interface	Ecran occupant une faible superficie dans le champ de vision	Ecran 2 positions (moyennes et basses), plus étendu permettant une réalité augmentée
Système d'exploitation	Android 4.4	Android 4.2.2
Calculs	CPU & GPU	CPU & GPU
Prix	1500$	949$

Les Google Glass sont bien plus chères que les ORA. De plus les ORA permettent d'avoir une réalité augmentée et sont dotées d'une bien plus grande autonomie. Cependant après avoir testé le prototype des lunettes ORA et assisté à une conférence à l'Imaginarium de Tourcoing sur les Google Glass, il en est ressorti que les Google Glass seraient finalement plus adaptée pour notre projet, l'affichage des ORA est trop présent, et il faut la mettre en position basse pour ne pas être perturbé et le produit n'est pas encore fini, de plus le poids est mal réparti, ce déséquilibre dérangerait forcément un magasinier sur son chariot. A l'inverse, les Google Glass sont plus légères, plus équilibrées. Et permettent d'obtenir des informations dans un coin du champ de vision non gênant.

Des applications sur les Google Glass pour la médecine ont été développées (leur efficacité d'utilisation main libre peut se faire une place en milieu stérile), également pour les pompiers (une application qui affiche pour un modèle de voiture accidentée donné, la procédure de désossage pour en extirper un corps, les moyens actuels reposant sur des gros manuels).

3 Travail réalisé

3.1 Prise en main des SmartGlass

3.1.1 Prise en main des GoogleGlass

L'utilisation de la Mirror API permet à des développeurs de coder des applications dans le langage que l'on souhaite, qui seraient exécutées sur les serveurs de Google et seraient téléchargeables par d'autres utilisateurs. Les serveurs s'occuperaient des échanges d'informations, beaucoup d'outils sur les lunettes fonctionnent sur ce principe.
Le GDK, une surcouche du SDK, permet d'avoir une emprise sur le matériel des lunettes (communication Bluetooth, accéléromètre, etc...) via des méthodes et objets spécifiques à celui-ci. Le développement se basant sur le GDK permet une instantanéité de l'application.
C'est la raison pour laquelle nous avons choisi de travailler avec le GDK.

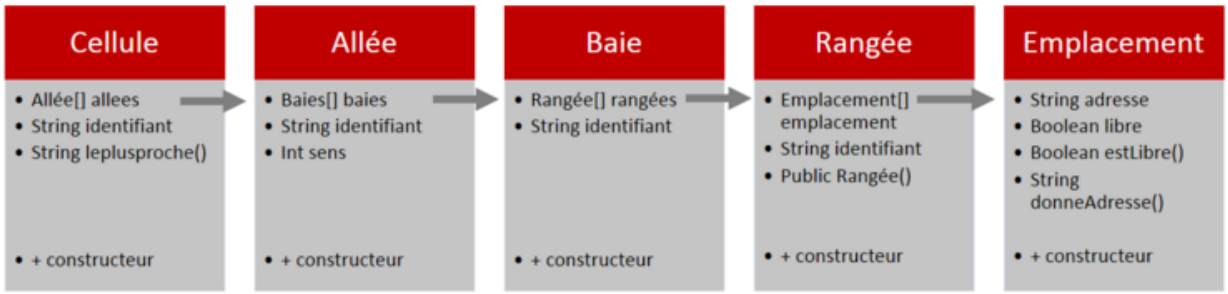
Nous avons rencontré soucis de drivers, le problème venant de l'association entre les ID des lunettes et ceux qui sont renseignés dans le fichier "AndroidWinusb" après modification du fichier, il a fallu autoriser notre système d'exploitation sur lequel nous développons pour autoriser l'utilisation de pilote dont la signature numérique est erronée.
Après ce problème réglé nous avons pris en main les fonctions du GDK en programmant des interactions simple (vocal, touchpad) pour mieux s'imprégner du développement sur cette appareil.

3.1.2 Prise en main des ORA de Optinvent

Ces lunettes possèdent un système Android 4.2.2 Avec une interface graphique classique. Le touchpad permet de déplacer le curseur sur l'écran et de sélectionner/lancer une application. Nous avons rencontré un problème de drivers également, il nous a fallu demander au support technique d'Optinvent, les drivers pour notre système d'exploitation.
Une fois le problème de driver résolu nous n'avons eu aucun souci à charger un programme et à le lancer sur les lunettes.

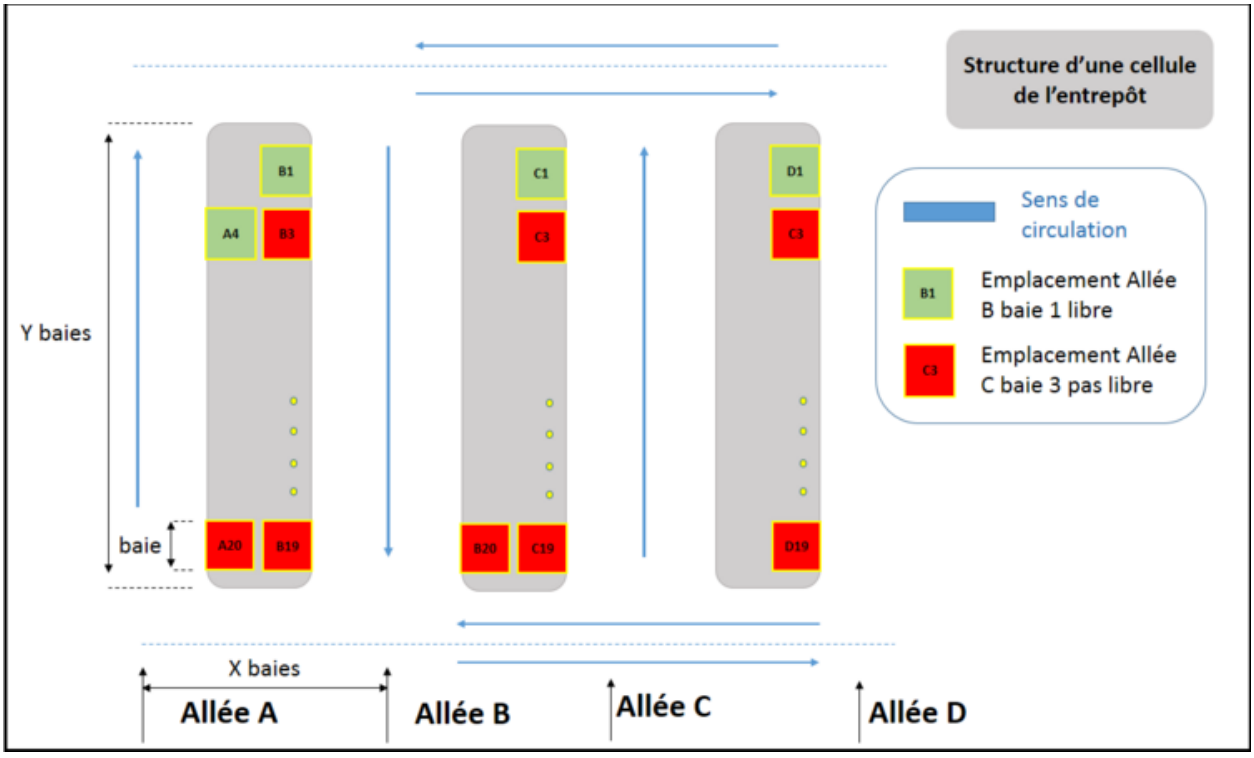
3.2 Algorithme de recherche

Choix technique : Dans un premier temps la recherche se base sur l'analyse d'une structure de donnée en dur dans l'application. La communication avec la base de données n'étant pour le moment pas une priorité pour l'entreprise, et pour tester notre algorithme de recherche, nous avons opté pour cette structure de donnée récursive.



Nous avons écri une première méthode de détermination. On choisit l'emplacement, parmi tous ceux qui sont libres, pour lequel la distance de parcours est minimale entre le départ et celui-ci. Ici l'unité de distance est la baie. La simplicité de l'algorithme réside dans le fait que le magasinier ne transporte qu'une seule caisse à la fois et repars toujours au même point de départ. C'est une sélection intelligente des adresses et non un parcours intelligent des adresses (l'emplacement libre trouvé le plus tôt n'est pas forcément le plus proche).

Ci dessous un plan de l'entrepôt.



On fixe une distance entre les allées (X baies) et une longueur d'allée (Y baies). L'entrepôt étant récent et assez homogène dans sa topologie.
Si on se place à l'entrée de l'Allée A :

Le déplacement jusqu'à A4 (sens de circulation direct) est de :

Y baies - (indice de la baie/2 +indice de la baie % 2)

Le déplacement jusqu'à B1 (sens de circulation indirect) est de :

Y baies + différence entre l'allée A et B donc X baies + indice de la baie/2 +indice de la baie % 2

Le déplacement jusqu'à C1 (sens de circulation direct) est de :

Y baies - (indice de la baie/2 +indice de la baie % 2)+diff entre l'allée A et C donc 2*X baies.

Si on se place à l'entrée de l'Allée B :

Le déplacement jusqu'à B1 (sens de circulation indirect avec contournement) est de :

2*X baies + Y baies + 2*X baies + (indice de la baie/2 +indice de la baie %

On peut généraliser l'ensemble de ces situations de déplacement avec l'algorithme suivant

```
Leplusproche( Allee depart )
  distance_la_plus_petite = valeurMAX
    le_plus_proche=""
      Pour chaque Allee de Cellule
        sens=Allee.sens
        Pour chaque Baie de Allee
          Pour chaque Rang de Allee
                Pour chaque Emplacement de Rang
            Si Emplacement estlibre

            indice_baie = selection_indice_baie(Emplacement.adresse)
                    allee_cible = selection_indice_allee(Emplacement.adresse)
                    difference_allee = valeur_absolue(depart -cible)
               Si difference_allee==0 et sens.estindirect
                 //sens indirect +contournement
                distance_actuelle=(indice_baie/2)+indice_baie%2+(2*(difference_allee))+4*X+Y

               Sinon
                  Si sens estdirect
                    //sens direct
                    distance_actuelle=(Y-((indice_baie/2)+indice_baie%2))+2*difference_allee
                  Sinon
                    //sens indirect mais pas de contournement
                    distance_actuelle=((indice_baie/2)+indice_baie%2)+2*difference_allee+Y
                  Finsi
```

```
            Finsi
            Si distance_actuelle < distance_la_plus_petite
              le_plus_proche=Emplacement.adresse
              distance_la_plus_petite=distance_actuelle
            Finsi
          Finsi
        Finpour
      Finpour
    Finpour
  Finpour
 retourne le_plus_proche
Finleplusproche
```

3.3 Interaction et Interface

Les Google Glass possèdent plusieurs moyens différents d'afficher l'interface graphique d'une application:

- Les live Cards : La Cards s'affiche sur la timeline dans les évènements passés. Elle est mise à jour périodiquement mais ne permet pas d'interactions poussées avec l'utilisateur.
- Static Cards (correspond plus au développement d'applications asynchrones, la réception de mails par exemple)
- L'immersion : Ce mode ressemble plus à la programmation classique sur Android. Une fois l'application lancée, les lunettes n'affiche plus que celle-ci. Les actions comme le glissement sur le touchpad par exemple entraînent une action dans l'application.
- Ongoing task: Un service en fond déclenchant l'affichage de vue et démarrant par événement (par exemple reconnaissance d'un réseau wifi pour la communication avec une base de données)

Pour notre programme nous avons choisi le mode immersion. En effet, le programme à but professionnel ne nécessite pas d'être quitté pour n'être regarder que ponctuellement.

3.3.1 Reconnaissance vocale

La reconnaissance vocale de base sur Google Glass est fortement simplifiée avec le GDK. Nous avons assez facilement mis en oeuvre un menu vocal. Cependant beaucoup de mots sont difficiles à identifier, et pour les digits et les mots en une syllabe la reconnaissance est encore plus laborieuse.
Nous avons donc commencé à nous intéresser à un autre système de reconnaissance vocale pour palier à ces problèmes et pour permettre une reconnaissance vocale en français (offline).
La librairie PocketSphnix est l'une des plus utilisée en open-source, c'est donc celle que nous avons choisi d'implémenter. Pour supporter une langue, elle nécessite:

- Un modèle acoustique (ensemble d'informations pour le traitement sonore)

- Un dictionnaire associant les **mots** et leur *phonèmes* dont voici un extrait propre à notre application

```
i ii
j jj yy
j(2) jj ii
porte pp oo rr tt
porte(2) pp oo rr tt ee
```

- Une grammaire, c'est un fichier définissant comment seraient articulés les mots à reconnaître et limitant les possibilités de langage). Par exemple pour la reconnaissance vocale de la porte de départ, nous utilisons celle-ci.

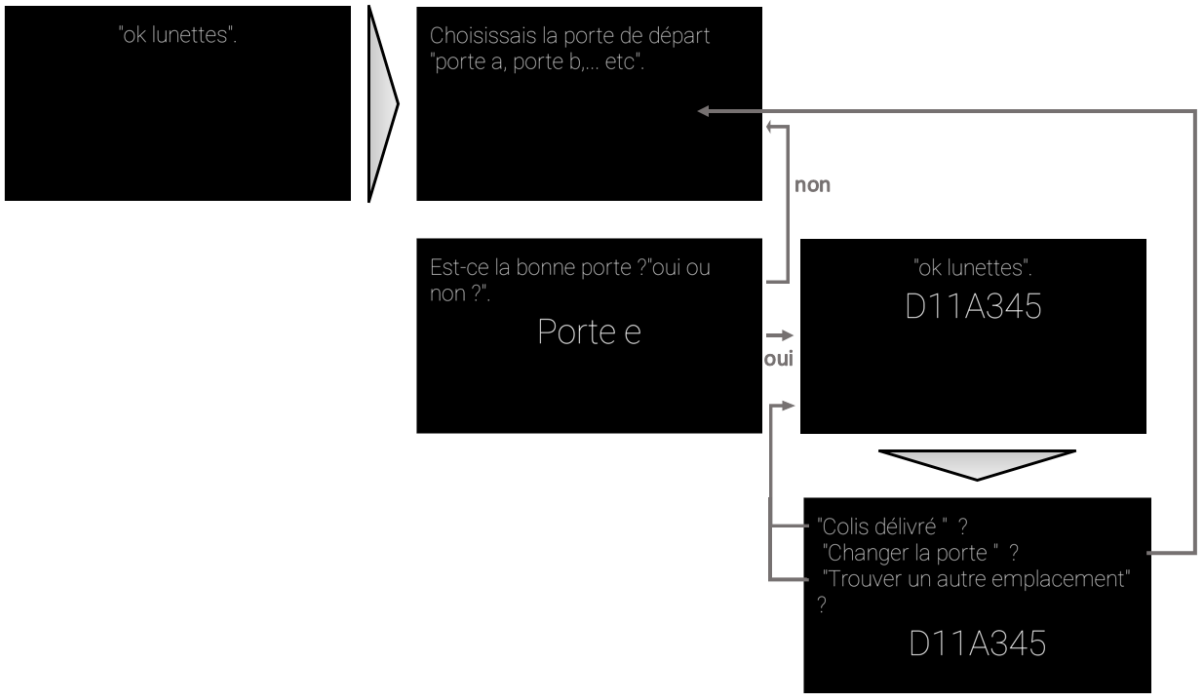
```
#JSGF V1.0;
 grammar porte;
 <porte> = a |
           b |
           c |
           d |
           e |
           f  |
           g |
           h |
           i  |
           j ;
 public <porte> = <porte>+;
```

3.3.2 Scénario d'utilisation.

Nous avons donc programmé le logiciel prototype avec une reconnaissance vocale pour se déplacer dans les différents menus et afficher le résultat.
Dans la version anglaise (initiale) pour commencer une interaction avec l'application, on utilise le mot clef « ok glass ». Nous avons étoffé cette reconnaissance vocale, sur la version française de l'application. A la manière du menu vocal des google glass, nous avons utilisé un délimiteur "ok lunettes" pour démarrer l'interaction avec l'application, afin que les discussions et bruits ambiants n'interfèrent pas sur l'utilisation de celle-ci.
Sous pocketsphinx, le délimiteur n'apparait pas dans une grammaire. La grammaire permet de prédestiner l'application à choisir entre deux mots. Dans notre application, le moteur de reconnaissance vocale choisi la distance la plus courte, séparant le mot prononcé par l'utilisateur et ceux qu'il est possible d'entendre. Ainsi entre "oui" et "non", l'application choisira forcément l'un des deux, même si l'utilisateur dit "bonjour"

L'avantage du délimiteur est que sa détection repose sur un seuillage entre ce qui est prononcé et ce qui doit être reconnu, nous pouvons régler ce seuil.

Et voici les screenshots relatant d'une recherche d'emplacement:

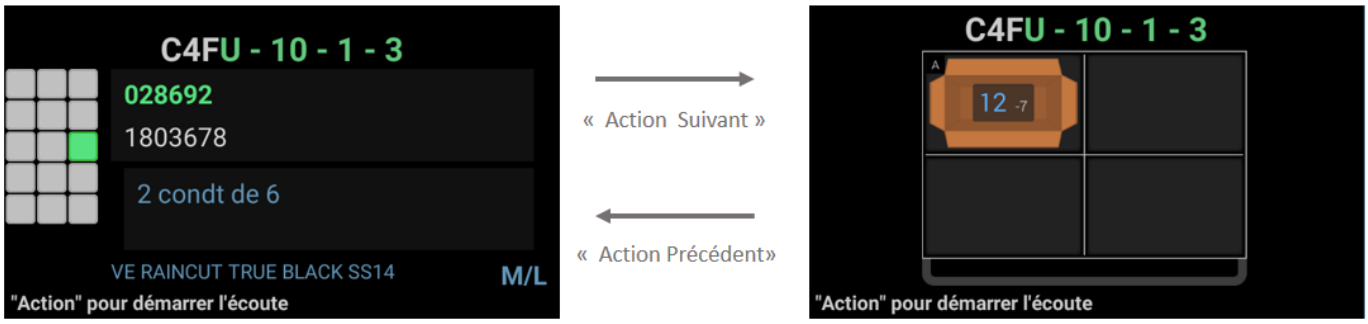


En ce qui concerne la version anglaise de l'application, nous avons utilisé le système propre aux Google Glass, celui-ci fonctionne de manière assez différente et à nécessité un temps d'apprentissage, toutefois plus court. Mais le système proposé par PocketSphinx est beaucoup plus générique, son utilisation s'étend à tous les systèmes android et à toutes les langues. La version française est la plus susceptible d'être utilisée.

3.4 Adaptation ORA vers Google Glass (programme Open).

Comme nous avions pris en main la programmation sur le GDK propre aux Google Glass, notre tuteur nous a demandé de réaliser un portage entre une application des ORA (SDK) vers les Google Glass.
Cette application android mains-libres pour les ORA one de Optinvent a été développée pour permettre à l'entreprise, d'accélerer les opérations de picking (prélévement à l'article) en entrepôt. Le but étant de comparer l'usage de cette application sur deux supports différents entre les lunettes.
Nous avons procédé à des ajustements au niveau de la gestion des layouts de l'interface, à équilibrer la place qu'occupe certaines informations données par l'application.
Cette application communique par bluetooth avec une autre application lancée sur une tablette android, cette dernière permet de réaliser l'appairage avec la smartglass. La tablette étant la seule à être connectée au serveur de l'entreprise (l'AS400) via un réseau sécurisé. Nous avons du appairer manuellement les glass avec la tablette, et retirer dans le code le lancement d'un intent permettant aux

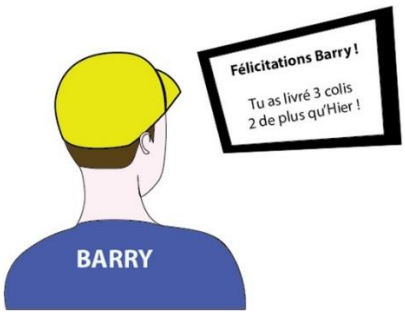
lunettes de se rendre visible sur le réseau. Lorsque cet intent est lancé il génère une exception non attendue qui ne concerne que les google glass. Aprés recherches sur le net, il semble que c'est un problème récurrent qui a été signalé mais pas encore résolu.
Ci dessous des screenshot de l'application en question adaptée par nos soins:



L'écran de gauche, correspond à la phase de prélévement d'un article ici "VE RAINCUT TRUE BLACK SS14" une veste à l'adresse C4FU-10-1-3, il faut en prélever 12, et le tableau à gauche correspond à la façade de l'étagère concernée, il indique grossièrement (case verte) le carton à déballer, en partant du principe que l'utilisateur respecte le sens de circulation de l'entrepôt. L'écran de droite, correspond à la phase de dépôt des articles dans le chariot de l'utilisateur, il lui indique dans quel compartiment le placer.
Nous avons participé à une demi-journée de test en production avec le responsable du picking, le responsable innovation, notre tuteur et d'autres parties prenantes du projet. La version sur les google glass a bien rempli son rôle à la manière de la version originale sur les ORA one, les préférences des parties prenantes se sont portés essentiellement sur les avantages qu'offrent les supports.

4 TV connectée

Comme cela se fait déjà dans plusieurs entrepôts, notre tuteur est aussi intéressé pour afficher des informations personnalisées sur un écran. Certaines données serait donc affichées sur une grande télévision à l'utilisateur le plus proche (détection grâce à un module BLE) installée en salle de vie des magasinier. Le but étant de d'informer les employés sur leur résultats (sans pression managériale) tout en entraînant un esprit de compétition afin d'améliorer sainement la productivité.



4.1 Étude du système

4.1.1 Proposition d'architecture

Nous avons fait une analyse de plusieurs architectures possible afin de réaliser cette fonction, nous en avons ressorti deux système qui correspondent à la fois à nos compétence et qui donnerais de bons résultats:
Utiliser une Raspberry Pie avec un OS embarqué qui afficherais les données sur grand écran de télévision. Couplée à un dongle BLE pour détecter les beacons et une connexion filaire ou wifi avec la base de donnée.
Programmer une application sur une tablette tactile géante (de type ViewSonic VSD231 compatible BLE). Faire la gestion de la base de donnée grâce à la connexion wifi de la tablette.

La première architecture nous permettrait une meilleur souplesse dans la programmation mais serait en même temps plus compliquée à mettre en oeuvre.
La deuxième quant à elle est plus restreinte mais correspond mieux au désir de l'entreprise car ce système serait plus simple à déployer et donnerais moins l'impression de n'être juste qu'un prototype. C'est pour cela que nous avons développé l'interface sur cette architecture.

4.1.2 Choix des infos

Nous avons du analyser et choisir des informations pertinentes et intéressante à connaître pour l'utilisateur.

Pour cela nous avons profité d'un exercice à Polytech' pour faire un brainstorming sur ce sujet. Voici le résultat de toute les informations que nous comptons afficher après une deuxième réflexion avec notre tuteur entreprise.

Personnel	Professionnel
● Photo/Avatar ● Nom/prénom ● Agenda ● Rappel des réunions ● Heure ● Météo ● Actualités de l'entreprise	● Classement par rapport aux autres ● Courbe/Graphique d'évolution ● Statistique (meilleur jour/mois de l'année) ● Nombre de paquet restant ● Nombre de colis posés ● Colis ou temps restant à faire ● Distance parcouru ● Ration d'erreur ● Poids transporté ● Objectifs de la journée

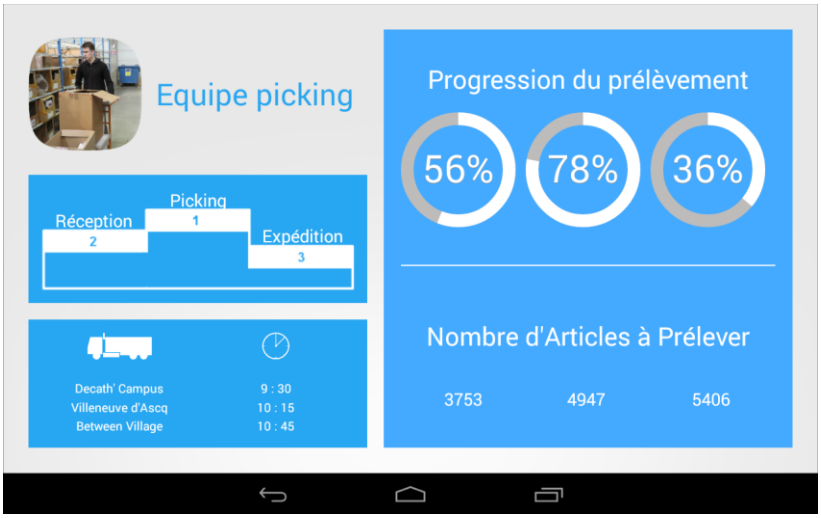
D'abord centrée sur les statistiques individuelles, il a été décidé que l'interface affichera les statistiques propres à l'équipe (équipe picking, équipe réception, équipe expédition).

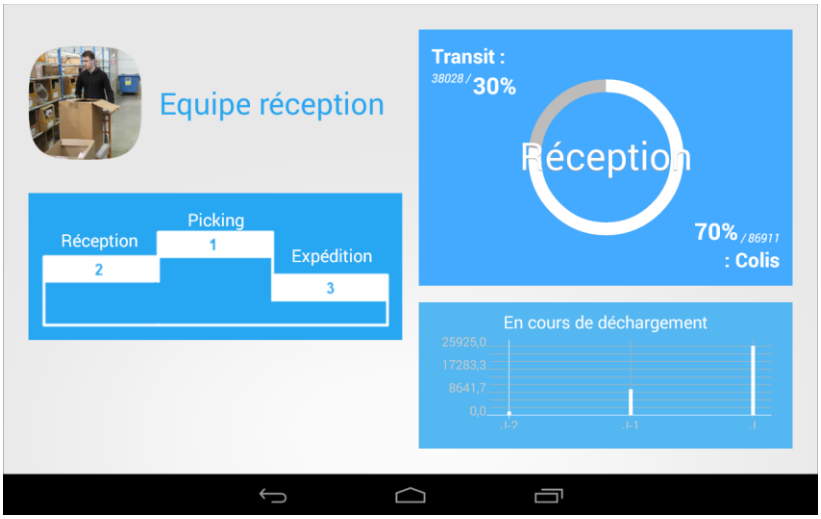
4.2 Interface utilisateur

Après avoir choisit les données à afficher et le support nous avons dut nous pencher sur la conception de l'interface graphique. Après une réunions à l'entreprise nous avons définit trois points important pour développer cette interface:

- Simple et intuitive
- En accord avec les couleurs de l'entreprise
- Moderne (système de tuile ou autre)

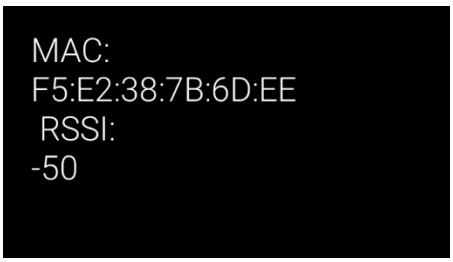
Nous avons donc beaucoup travaillé avec les système d'affichage android et les dessins xml afin de travailler l'interface et de la peaufiner grâce aux retour de l'entreprise. Voici le résultat





4.3 Détection BLE

Le but de l'écran est d'afficher des informations en fonction de la personne qui se présente devant lui. Pour cela la technologie retenue à été le BLE. Chaque magazinier serait équipé d'un module BLE qui serait reconnue à l'aide de l'adresse MAC de celui-ci. On considérera que le module dont le signal est le plus fort (RSSI en dB) est celui le plus proche.
Pour cela nous avons programmé une fonction de recherche de périphériques Bluetooth sur les Google Glass dans un premier temps puis sur la tablette.



Ci contre une capture d'écran du programme prototype sur les google glass, cela nous donne également l'ordre de grandeur de la force du signal alors que le beacons est assez proche -50 dbm.

4.4 Connexion Base de Donnée

Pour tester la connectivité de l'application sur la tablette géante, dans le cadre de la fonctionnalité de TV connectée, nous voulions une base de donnée accessible depuis n'importe où. L'école met à disposition un site avec nom de domaine, nous avons donc profité de cet outil pour créer une BDD SQL. Nous interrogeons cette BDD via un HttpRequest lancé dans l'application android sur une page php qui contient la requête SQL. En GET nous envoyons l'adresse mac du beacons le plus proche, qui permet de ne sélectionner que les informations nécessaires. L'idée est de pousser l'utilisation du matériel, et de prendre en main cet aspect de l'application, même si le système final pourrait reposer plutôt sur un web service réalisé par l'équipe de développement de l'entreprise.
Nous réceptionnons correctement les informations à partir de la base de donnée cependant, nous n'avons pas approfondi l'exploitation des informations accessible par cette base de donnée, par manque de temps et pour des raisons de sécurité.

5 Conclusion

5.1 Tests et Validation

Google Glass
L'interface utilisateur sur les Google Glass a été bien accueillie par les parties prenantes, nous n'avons pas développé la partie communication avec le serveur de l'entreprise. Dans certains cas, connaître la porte de départ n'est pas la seule information dont à besoin un magasinier pour adresser un colis, et l'entreprise commence à envisager d'utiliser une autre technologie pour le faire.

Portage de l'application
Le portage de l'application a permis à l'entreprise de réaliser un comparatif de support pour une même application en production, qui n'était pas prévu initialement. L'entreprise étudiera la possibilité d'utiliser des Google Glass, plus confortable que les ORA one, pour leur secteur de Smart Picking, qui est beaucoup plus susceptible dans le futur proche d'utiliser des Smart Glass.

Télévision Connectée
Après plusieurs remaniements, nous avons établi avec notre tuteur l'interface finale d'affichage d'information. Il était vraiment intéressant de modifier notre application en fonction des retours de l'entreprise afin que notre interface corresponde au mieux aux besoins de l'utilisateur.
De plus nous avons pu acquérir une bonne connaissance des interfaces Android et comment les manipuler.

5.2 Bilan personnel

L'aspect réalité métier a occupé une place importante dans le projet, et a constitué l'une des dimensions enrichissantes du projet, développer des systèmes pour les utilisateurs en pensant au confort d'utilisation. Le retour des utilisateurs sur notre projet est aussi un aspect intéressant afin de nous initier à la satisfaction client lors du développement d'un projet.
Ce projet a été l'occasion de prendre en main le développement sous Android, chose que nous n'avions jamais faite auparavant, et de travailler sur les technologies à la mode, que sont les SmartGlass, les tablettes Android ainsi que le BLE.
Nous sommes enjoué d'avoir pu prendre part à la démarche d'avancée de l'entreprise dans ces processus d'innovation et nous espérons que notre travail aura pu les aider à s'orienter vers des solutions futures d'innovation.