



MASTER HANDI

Domaine : Sciences Technologie, Santé (STS)

Mention : Ingénierie et Cognition

Spécialité : Technologie et Handicap

Rapport de stage M2

Développement d'applications « Android » pour l'entreprise *Dystri*

Auteur : Pierre-Charles HANSE

Directeur de stage : D.GIRARD
Lieu du stage : Boulevard Sébastopol, Paris

Coordonnateur :
J. LOPEZ KRAHE

Paris, Septembre 2014

[Logo entreprise]



Résumé :

Abstract :

Remerciements :

Avant toute présentation de cette expérience professionnelle, il me semble important de commencer ce rapport de stage par des remerciements, à ceux qui m'ont beaucoup appris au cours de ce stage, et tous ceux qui ont eu la bienveillance de faire de ce stage un moment très profitable.

Aussi, je remercie Monsieur Daniel Girard, mon maître de stage qui m'a formé et accompagné tout au long de cette expérience professionnelle, me donnant sa confiance et m'impliquant grandement au sein du projet qu'il a initié. Je souhaite à tout étudiant d'avoir un pareil maître de stage. Je remercie de même l'ensemble des employés de DYSTRI et de ATALAN pour les conseils qu'ils ont pu me prodiguer au cours de ces six mois de stage.

Je tiens aussi à remercier l'intégralité de l'équipe pédagogique du master Sciences de l'Ingénierie et Cognition mention « Technologie & Handicap », qui a su me former aux nouvelles technologies, tout en me sensibilisant aux thématiques liées aux handicaps durant ces deux dernières années grandement enrichissantes et réjouissantes.

Enfin j'exprime toute ma gratitude à ma famille et à mes proches pour leur soutien pendant cette période six mois qui n'était pas de tout repos.

Sommaire

I. Introduction	
1. Contextes du stage	
❖ Contexte Universitaire	
❖ Contexte Entreprise	
2. Présentation de l'entreprise	
❖ Histoire de l'entreprise	
❖ Les activités de DYSTRI	
3. Cadre du stage	
❖ Environnement de l'entreprise	
❖ Objectif du stage	
❖ Réalité du stage	
II. Mes projets de stage	
1. Contexte des projets	
❖ Particularité d'une petite entreprise	
2. Présentation des outils	
❖ Ressources matérielles	
❖ Ressources logicielles	
3. Les projets	
Projet – EazyText	
1 - Présentation	
2 – Structure du code	
3 – Interface graphiques	
4 – Les fonctionnalités	
A – Les fonctionnalités des boutons latéraux	
B – Les fonctionnalités propres au clavier	
C - Les fonctionnalités du composant éditeur de texte	
D - Les fonctionnalités de copier-coller	
5 – Les notions avancées de programmation utilisés	
6 - Bilan	
Projet – EazySchool	
1 - Présentation	
2 – Structure du code	
3 – Interface graphiques	
4 – Les fonctionnalités	
5 – Les notions avancées de programmation utilisés	
6 - Bilan	
III. Conclusion	
❖ Bilan global	
BIBLIOGRAPHIE	
GLOSSAIRE	

I. Introduction

1. Contexte du stage

❖ Contexte Universitaire

Dans le cadre du master HANDI (Ingénierie et Cognition, spécialité « Technologie & Handicap » à l'université Saint Denis - Paris 8, un stage en entreprise est prévu. Il doit durer de 3 à 6 mois et être en rapport avec la formation bien entendu.



Les étudiants sont chargés de rechercher l'entreprise par eux-même. De ce fait, nous nous familiarisons avec l'exercice délicat des entretiens, des préparations de CV et autres lettres de motivation. C'est en quelque sorte les prémices de la recherche d'emploi à venir.

Pour ma part, l'opportunité de trouver mon stage s'est faite au sein de l'université Paris 8. Le co-directeur de la société DYSTRI, Monsieur Daniel Girard, était venu directement présenter son offre de stage à notre promotion, nous avons discuté ultérieurement. Par la suite une fois la convention de stage signée, un premier mois d'essai a eu lieu comme le veut le règlement. Dans le cas présent, il a été réussi.

❖ Contexte Entreprise

Durant toute la période du stage, l'étudiant doit s'adapter à l'entreprise, de par les règles vestimentaires, horaires et sociales aussi, qui structurent le quotidien dans l'entreprise. La nécessité de cette adaptation rejoint là convenablement les conditions d'intégration dans un poste titulaire réellement. Concernant ces règles, elles varient selon les entreprises ; chez DYSTRI les règles sont plutôt souples, néanmoins elles sont bien présentes, et les interactions avec les autres entreprises augmentent cette nécessité d'adaptation et d'intégration.

2. Présentation de l'entreprise

❖ Histoire de l'entreprise

L'entreprise DYSTRI a été créé en 2010, par Daniel Girard et Romuald Guéguan. Ses deux

fondateurs étaient auparavant dirigeants de deux entreprises qui proposaient des solutions d'équipement et de compensation autour de la thématique du handicap moteur. Ils ont unis leur forces en 2010 pour fonder DYSTRI.

Cette entreprise est spécialisée dans la commercialisation de logiciels de compensation pour la dyslexie. Assez rapidement, DYSTRI devient leader dans la distribution des logiciels à destination des inspections académiques, devenues par ailleurs les DFDEM.

DYSTRI se caractérise aussi par un chiffre d'affaire multiplié par 250 % ces dernières années, une rentabilité et une pérennité certaine.

❖ **Les activités de DYSTRI**

DYSTRI commercialise tout un ensemble de logiciels visant à compenser la dyslexie au quotidien, au niveau de l'écriture et de la lecture. Ils ont aussi développé la marque KARDI, destinée aux orthophonistes, et dont la vocation est de proposer des solutions pour les paramédicaux et les particuliers.

DYSTRI, bien qu'une des entreprises leader dans le domaine de la dyslexie, est composé de seulement cinq membres : direction, commercial, communication et logistique, conseil et expertise.

Basé d'une part à Rennes, où se situe le pôle direction et logistique, et d'autre part en région Parisienne, où se trouve le pôle opérationnel, DYSTRI a une structure légère et souple. L'entreprise fonctionne beaucoup avec la prise de contrôle à distance pour le service après-vente des clients, et diffuse beaucoup d'informations sur le cloud. Les représentants de l'entreprise ont une très bonne expertise des logiciels vendus, il sont bien au courant des besoins réels des clients, grâce à l'investissement des membres de DYSTRI sur le terrain.

Travailler à leur côté m'a permis d'apprécier grandement l'importance de la pratique des logiciels existants, de se tenir au courant des nouveautés, des mises-à-jour des produits, de dialoguer beaucoup avec les clients et les corps de métiers qui les entourent, afin de reconnaître au mieux leurs attentes.

Depuis peu, DYSTRI s'intéresse au portage d'applications Windows vers Android, dans le domaine de la dyslexie bien entendu.

3. Cadre du stage

❖ Environnement de l'entreprise

L'environnement du stage était particulièrement intéressant, car j'ai eu à alterné entre différents contextes de travail.

D'une part il y avait les bureaux de DYSTRI à Cognières, dans lesquels j'avais à travailler avec le co-directeur de l'entreprise M.Girard, ainsi que différents intervenants tels que son assistante commerciale Mlle G.Fath El Khir et un stagiaire étudiant en école de publicité Monsieur Clément Leniau.

D'autre part, j'ai eu à travailler régulièrement dans les locaux de l'entreprise Atalan, boulevard Sébastopol. Je travaille dans le cadre agréable d'un open space avec 3 autres informaticiens. La société Atalan est spécialisée dans le conseil en accessibilité du Web et, ce qui est commun au master HANDI, à la sensibilisation au handicap.

Enfin, j'ai eu aussi à travailler à distance avec Monsieur Romuald Gueguan, co-directeur de DYSTRI et associé de Monsieur Girard, qui travaille au siège de DYSTRI à Rennes. J'ai par ailleurs eu l'occasion de me déplacer à Rennes pour le rencontrer.

Ces environnements multiples m'ont permis de m'exercer à m'adapter à des contextes différents, à savoir gérer mes déplacements et à m'intégrer d'autant plus.

Cette pluralité a engendré une communication importante par le biais des technologies classiques, telles que les e-mails, le téléphone, Skype (pour la communication vocale en temps réel via l'ordinateur), Google Drive (pour le partage de contenus numérisés tels que les documents d'avancement de projet). Toutes ces technologies ont permis un échange de flux d'informations aisé.

De plus cet environnement multiple de travail confère un dynamisme certain et un rythme de travail constant. Le dynamisme de Monsieur Girard, mais aussi l'implication des intervenants et la motivation de l'équipe d'ATALAN, particulièrement investi dans leurs missions, tous ces éléments ont formé un contexte particulièrement riche pour ma formation, et je leur en suis très reconnaissant.

Mes tâches sont dictés par mes directeurs, lors de réunions de travail ou par entretien à distance (lorsque les disponibilités géographiques ne coïncident pas). Au cours de ces réunions je suis la plupart du temps participant, sinon animateur lorsqu'il s'agit de présenter l'état d'avancement des projets. Il est à noter que Monsieur Girard m'a fortement impliqué dans toutes les étapes des projets que nous avons eu à faire, tout autant de la phase de création, de conception et de faisabilité jusqu'à celle de la réalisation et du développement. Si des retouches étaient à faire, cela était notifié, et c'était parfois le cas, ce qui conditionnait mes travaux à venir.

Mes responsables m'ont gratifié d'une grande confiance, et ils me présentaient avant tout comme collaborateur et développeur, plutôt que stagiaire. Une telle confiance responsabilise amplement et est d'autant plus motivante.

Cela m'a permis d'exercer mes apprentissages récents du master HANDI dans un contexte professionnel, allant jusqu'à me charger de m'entretenir par téléphone avec le directeur de Prolexis par exemple. Ou bien même de participer à des réunions, Monsieur Girard et moi, avec l'équipe de la société IMANENS, composé alors d'un développeur android, un développeur iOS, un responsable développement et un chef de projet.

Les rapports avec les membres de DYSTRI, mais aussi d'ATALAN, furent bons, voir même conviviales, mon intégration est plutôt réussie.

Des règles fondamentales, telles que le code vestimentaire et les horaires à respecter étaient de mise, ce qui est commun à toute entreprise. Par exemple chez Atalan, le bureau ouvrait à 9h30 et fermait vers 18H00, bien que cela pouvait varier, selon que le travail était achevé ou non, que des rendez-vous étaient prévus, ce qui rendait les journées de travail peu monotones, et apportait une souplesse nécessaire au bon fonctionnement des entreprises.

Pour me rendre chez DYSTRI, à Cognières, j'avais à prendre le métro puis le train. Pour aller aux bureaux d'ATALAN, j'avais à prendre le métro uniquement. Enfin pour me rendre à Rennes et rencontrer Monsieur Guegan, j'ai eu à prendre le TGV Paris-Rennes.

Concernant le style vestimentaire, s'il n'y avait pas de client attendu aux bureaux ou aucun déplacement prévu dans d'autres entreprises, une tenue décontractée était suffisante (jean et chemise, polo ou t-shirt même), sinon une tenue plus classique était requise (chaussures de ville, pantalon coupe droite et chemise) bien que cela ne m'a pas été mentionné, mais cela me semblait évident.

En somme l'environnement de travail était sérieux et dynamique, mais aussi par moment très convivial et tout le temps communicatif. C'est un ensemble d'éléments positifs non négligeables lors d'un stage en entreprise.

❖ Objectif du stage

Du 1^{er} mars au 1^{er} septembre 2014, j'ai donc effectué un stage de six mois au sein de l'entreprise DYSTRI. Au cours de ce stage dans la section Recherche & Développement, j'ai pu m'intéresser au développement d'applications android, prévus pour smartphones et aux tablettes tactiles. L'objectif initial de DYSTRI était de développer des outils adaptés aux dyslexiques sur la plate-forme android, afin de créer un panel d'outils similaires à ceux distribués par DYSTRI sur Windows.

Développements d'outils adaptés aux dyslexiques sur la plate-forme android

Etat des lieux

- Identifier les outils existants sur Windows
- Analyser les possibilités d'Android
- Identifier les nouveautés envisageables dans le cadre de la dyslexie
- Proposer des solutions et des maquettes

Etude de la faisabilité

- Lecture approfondie de la documentation de développement Android
- Réalisation de mini-application pour chaque élément technique envisagé

Développement des outils

- Développer les applications
- Distribuer les outils

❖ **Réalité du stage**

Bien que l'objectif initial était assez précis, mes tâches furent plus nombreuses que prévu, ce qui est commun à tout stage à vrai-dire. J'ai été amené à travailler sur d'autres projets, pas forcément centrés sur la dyslexie. Le modèle économique de distribution des applications android étant ce qu'il est (il s'agit de vente d'applications facturés quelques euros à l'utilisateur, contrairement à Windows où les prix sont plus conséquents), j'ai travaillé donc sur d'autres projets plus globaux, en rapport avec la diffusion de contenu éducatif et pédagogique, destinés à rendre pérenne le projet global qui, objectivement, peut difficilement être suffisant sans la production applications moins spécialisées.

D'un point de vue technique, j'ai été amené à aider de temps en temps sur des domaines totalement différents de la programmation android, comme par exemple en développement web ou à l'utilisation avancé de logiciels internes à l'entreprise.

Ces tâches parallèles m'ont permis d'élargir mes compétences, ce qui est une bonne chose, bien que le temps consacré aux tâches annexes décrites ci-dessus correspond à du temps moins passé sur le projet initial.

D'un point de vue humain, j'ai eu l'opportunité de rencontrer des partenaires de DYSTRI à plusieurs reprises, de travailler avec eux, et donc d'avoir des responsabilités plus importantes puisque je représentais DYSTRI et que je devais être à la hauteur de la confiance qui m'a été donné. En somme, j'ai été bien plus impliqué que je ne l'aurai imaginé. L'expérience professionnel et humaine est d'autant plus intéressante.

Cette expérience m'a permis d'être plus à l'aise dans les entretiens et les réunions d'une part, et d'autre part d'un point de vue technique j'ai beaucoup appris sur Android. Nous y reviendrons par la suite dans ce rapport.

II. Mes projets de stage

1. Contexte des projets

❖ Particularité d'une petite entreprise

Pour chacun des projets abordés, il n'y a pas eu de planification formelle classique comme l'on pourrait retrouver dans la gestion de projet des grandes entreprises faisant appel à des chefs de projet et de nombreux développeurs. Etant le seul développeur à proprement parlé pour chacun des projets, et travaillant en étroite collaboration avec Monsieur D.Girard, nous avons choisi d'adopter l'approche Agile, aux vues des besoins et des moyens en notre possession typique d'une petite structure.

L'approche Agile repose sur les techniques itératives, incrémentales et adaptatives de gestion de projet. Par conséquent, ce choix implique d'adopter une méthode de programmation *Extrem Programming*, basé sur une grande interaction entre le concepteur/développeur et le client. L'avantage principal est d'avoir des résultats très proches de la demande du client, mais comme inconvénient une grande consommation de temps.

La planification des tâches varie constamment, tous les deux ou trois jours on doit réajuster les priorités. J'ai donc choisi de ne pas utiliser de planifications formelles telles que celles présentées au cours de Mme Pousset (Gestion de projet). J'ai utilisé plutôt un mode personnel pour prioriser les tâches.

C'est-à-dire que seuls macros tâches sont définis, décomposés en tâches unitaires dont les priorités varient selon les imprévus et les changements dictés par les réunions fréquentes. Parfois il arrive qu'il y est des suppression de tâches ou l'ajout de tâches dans l'urgence. Une fois une urgence traitée, on reprend la séquence prévue initialement. C'est la solution qui m'a semblé la plus adaptée au contexte, bien qu'ayant des défauts.

Certes la gestion du temps n'était pas optimale, néanmoins cela a permis d'avancer et répondre aux besoins du client.

Dans le cas présent, et cela rejoint les projets que nous avons réaliser au cours l'année en master HANDI, le client et l' équipe de conception/développement était confondu. Les projets étaient à réaliser pour DYSTRI, d'où la méthode Agile qui nécessite une interaction

fréquente entre concepteur et client.

Je pense que cela est commun à de nombreuses entreprises TPE/TME, composés de quelques personnes dont la quantité de temps de travail est mince. J'ai donc du apprendre à utiliser au mieux mon temps de travail et être efficace. Il était nécessaire de savoir passer rapidement d'une tâche à une autre, de savoir juger rapidement de la priorité au moment présent. Cela responsabilise d'autant plus et oblige à bien ordonner ses idées.

2. Présentation des outils

❖ Ressources matérielles

Ordinateur

J'ai utilisé mon propre ordinateur portable pour travailler en local et sur le web en me connectant au réseau wifi des bureaux dans lesquels j'étais amené à travailler. J'utilisais aussi mon ordinateur lors des déplacements et des démonstrations dans les entreprises partenaires de DYSTRI.

Il s'agit d'un ordinateur portable relativement puissant doté un iCore i7. Cela permet de pouvoir travailler avec fluidité et même d'utiliser des émulateurs android, assez coûteux en charge processeur.

L'autonomie est de 5 heures environ. C'est suffisant pour les réunions à l'extérieur ou les tâches traitements de texte lors des déplacements typiquement.

Stockage de données - Echange

J'ai utilisé un disque dur externe de 500 Go pour mes sauvegardes. De plus les fichiers des projets étaient dans un dossier personnel sur DropBox (stockage sur internet).

Les documents de conception étaient sur le Google Drive, un outil web pour partager du contenu textuel en mode lecture et en mode écriture.

Connexion Internet

Le bureaux où je travaillais disposaient du Wifi comme connexion internet pour utiliser l'internet.

L'accès n'est aucunement restreint à ma connaissance. Nous avons tous les même droits d'accès au web, je pouvais accéder sans problème à l'ensemble des données nécessaires.

Smartphone

Pour tester les applications android développées, j'ai un smartphone Samsung Galaxy SII prêté par le master HANDI de l'université Paris 8. La version logicielle de l'appareil est 4.1.2

Le bureaux où je travaillais disposaient du Wifi comme connexion internet pour utiliser l'internet.

L'accès n'est aucunement restreint à ma connaissance. Nous avons tous les même droits d'accès au web, je pouvais accéder sans problème à l'ensemble des données nécessaires.

Tablette tactiles

J'ai utilisé une tablette tactile fournie par DYSTRI pour les tests sur ce type de support.

Il s'agit d'une tablette tactile ACER de 10,1 pouces, 16Go de stockage interne, doté des technologies WIFI, BlueTooth et d'Android 4.0 Icre Cream Sandwich.

❖ Ressources logicielles

Remarque : La liste n'est pas exhaustive, néanmoins les ressources logicielles essentielles sont énumérer ci-dessous.

Eclipse

Il s'agit d'un des principales IDE (Integrated Development Environment) très connu du monde informatique. Via un système de plug-in, la plupart des langages de programmation sont disponibles.



Il contient toutes les fonctionnalités nécessaires et il est relativement simple à utiliser bien qu'un temps d'adaptation est nécessaire. Toutefois nous avons l'habitude de l'utiliser en master HANDI, et je l'ai beaucoup utilisé aussi pendant cours de mes études antérieures. Il est à noter qu'il subsiste des bugs qui font perdre du temps parfois. J'ai dû réinstaller le logiciel à plusieurs reprises, notamment par rapport à la reconnaissance des périphériques android. J'utilisais Eclipse sur Linux à la base, mais je suis passé sur Windows par la suite pour le gain de temps au final. Eclipse est gratuit.

PHP

PHP est un langage Web de script situé du côté serveur, . PHP permet de rendre les pages dynamiques, son utilisation est souvent couplée avec une base de données, MySQL typiquement. PHP est gratuit.



HTML

HTML est un langage Web qui fournit du contenu statique contrairement au PHP. Cependant il peut être couplé à PHP, pour la production de contenu dynamique. HTML est gratuit aussi.



CSS (Cascading StyleSheet)

CSS est un langage de mise en forme, associé à une page HTML/PHP. Il s'agit d'une feuille de style qui sert à mettre en forme le visuel des pages web que l'on crée. Ses capacités sont grandes et peu coûteuses en terme de temps processeur.

FileZilla client

C'est un client multiplateforme FTP/FTPS/SFTP, très connu.

Grâce à ce logiciel, le transfert de données du local vers un serveur web, ou du serveur web vers le local, est simple et efficace.



MySQL

MySQL est un système de gestion de base de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels.



PhotoShop / GIMP

Ces 2 outils sont particulièrement connus du monde Informatique orienté graphisme, dont les web designer.

Ils permettent la retouche d'images de manière très poussée, avec des fonctionnalités de dessins et de traçages classiques, ainsi que des fonctionnalités mathématiques permettant de faire des effets ou des filtres sur les images à traiter.



Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Utile pour les retouches d'icônes notamment.

JAVA

Le langage JAVA est un langage de programmation informatique orienté objet créé en 1995. Il est très répandu depuis de nombreuses années et est utilisé pour la programmation android.



ADT Plug-In

Android Development Tools est un plug-in pour l'IDE Eclipse, qui fournit tous les outils nécessaires pour le développement d'applications android, notamment de quoi travailler l'interface intuitivement, d'interagir avec les périphériques android et même d'utiliser un émulateur de smartphones et de tablettes android.



3. Les projets

❖ **Projet *EazyText***

1. Présentation

EazyText est l'application android qui a été le coeur du stage.

Le but premier d'*EazyText* est de fournir un éditeur de texte doté d'outils avancés de synthèse vocale et de suivi de lecture. Aussi, un clavier android particulier est intégré et pré-requis pour le bon fonctionnement de l'application.

Outre les outils de synthèse vocale, de capture vocale et suivi de lecture, l'innovation première de *EazyText* réside dans le clavier que l'on retrouve, une fois défini comme clavier par défaut, dans tous les applications du téléphone.

Ce clavier est doté de boutons spéciaux qui offrent la possibilité d'utiliser sur du texte la synthèse vocale, cela dans n'importe quel application.

Le grand point fort du clavier est de pouvoir envoyer du texte en un seul clic depuis n'importe quel application contenant du texte, vers l'application *EazyText*, sans avoir à passer par le copier-coller habituel.

EazyText requiert un appareil android doté d'une version logicielle 4.0 au minimum.

2. Structure du code

Au niveau du code, un projet java android est composé de nombreux éléments. Je vais vous présenter ici les éléments principaux dans le cas présent.

++ Dossier SRC ++

Nous avons un dossier « src » contenant 3 packages, dans lesquels se trouvent les fichiers .java représentant les classes créés.

++ Dossier RES ++

Il y a aussi un dossier « res », contenant plusieurs dossiers :

- drawables => fichiers images et fichiers .xml de transition d'images
- layout => fichiers .xml d'interfaces classiques
- menu => fichiers .xml d'interfaces des menus
- values => fichiers .xml centralisant les informations utilitaires pour les couleurs, les chaînes de caractères, les dimensions des éléments graphiques et les paramètres graphiques
- xml => fichier .xml définissant les éléments des menus de préférences

++ Fichier AndroidManifest.xml ++

Ce fichier indispensable représente les informations essentielles de l'application .

Il contient :

- le nom du package de l'application
- les composants de l'application (les activités, les services, les broadcast receivers) et leurs paramètres
- les processus supportés par l'application
- les permissions systèmes de l'application
- les permissions nécessaires pour les applications externes qui veulent interagir avec notre application
- le niveau minimum de l'API Android que l'application requiert pour fonctionner.
- les bibliothèques nécessaires au bon fonctionnement de l'application

3. Interface graphique

L'interface est prévu pour des tablettes tactiles, avec une utilisation en mode paysage.

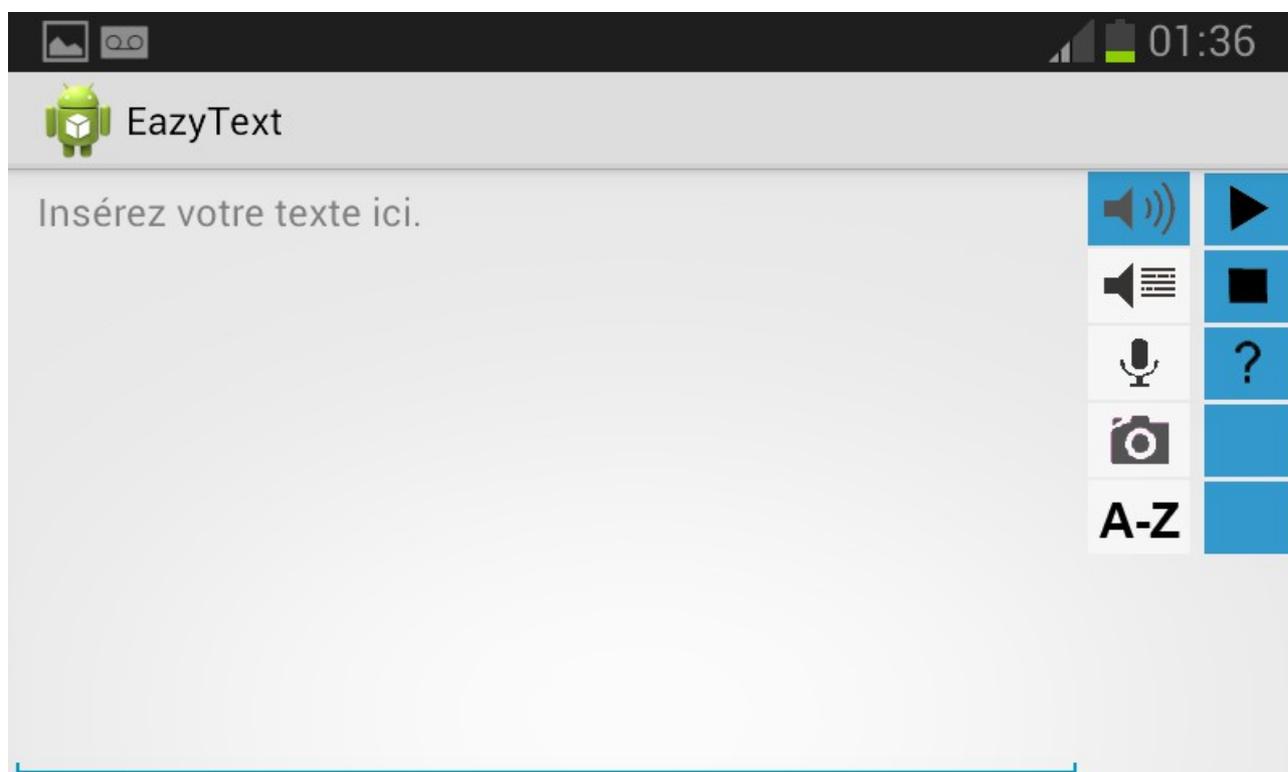
L'éditeur de texte

Sur la partie centrale, il y a l'éditeur de texte.

Au lancement de l'application, si le clavier *EazyText* n'est pas défini par défaut, le composant éditeur de texte est remplacé par un bouton qui envoie vers les préférences de l'appareil et qui propose de définir le clavier par défaut.

PHOTO BOUTON

Tant que le clavier n'est pas défini par défaut, l'éditeur de texte n'est donc pas disponible. En revanche si le clavier *EazyText* est défini par défaut, le bouton disparaît et fait place à l'éditeur de texte.



Les boutons

Sur le côté gauche ou droit (cela dépend de la préférence utilisateur « gaucher » ou « droitier »), il y a un ensemble de 10 boutons contenus dans 2 colonnes de 5 rangées.

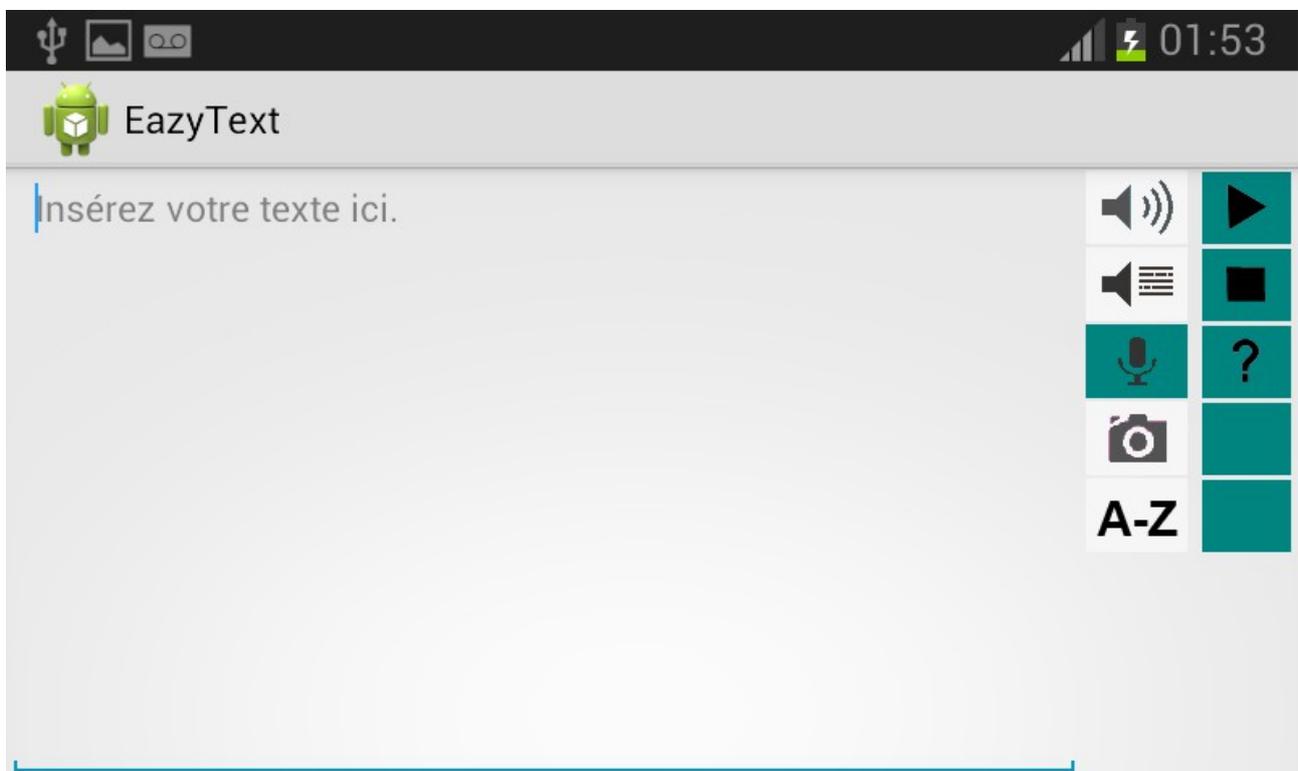
Dans la première colonne, on trouve les icônes qui définissent le mode d'utilisation de l'application , Il y a donc 5 principaux modes d'utilisations différents.

Les boutons de la colonne de droite ont des fonctionnalités unitaires qui varient selon le mode d'utilisation.

A noter que lorsqu'on appuie sur un bouton de la colonne de gauche, la couleur des boutons de droite prennent tous la couleur du bouton sur lequel on vient de cliquer.

Afin que le changement de couleur ne soit pas trop brusque, j'ai implémenté un système de transition de couleurs assez avancé.

Nous y reviendrons par la suite dans la description des fonctionnalités.



L'interface graphique est donc assez épuré, nous sommes allé à l'essentiel.

Le menu des préférences

De plus niveau interface graphique, il y a aussi l'interface du menu de préférences. En cliquant sur le bouton de préférences commun à tous les smartphones et les tablettes graphiques, l'écran de préférences classique s'affiche.

Le menu est composé de plusieurs éléments, regroupé en 3 catégories :

☒ *Catégorie « Synthèse vocale »*

- une seekbar pour choisir la vitesse d'élocution de la synthèse vocale, entre 0 et 40.

(une seekbar correspond à une jauge doté d'un curseur qui permet de choisir une valeur entre un intervalle)

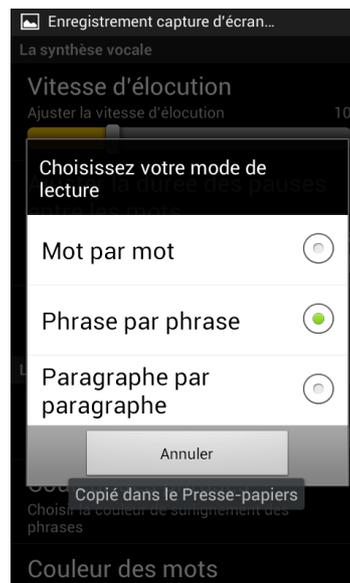
- une seekbar pour choisir la durée des pauses entre les mots, entre 0 et 30ms

- une liste qui permet de choisir le mode de lecture (mot par mot, phrase par phrase ou paragraphe par paragraphe)



☒ *Catégorie « Editeur de texte »*

- une seekbar pour choisir la taille de la police, entre 0 et 30ms
- un composant spécial permettant de choisir la couleur de colorisation des phrases
- un composant similaire au précédent, qui permet cette fois de choisir la couleur de colorisation des mots



☒ *Catégorie « Ergonomie »*

- une case à cocher pour activer l'interface spéciale pour gaucher
- une case à cocher pour activer la colorisation des phrases uniquement



Le clavier

Le clavier intégré avec *EazyText* adopte une interface classique.

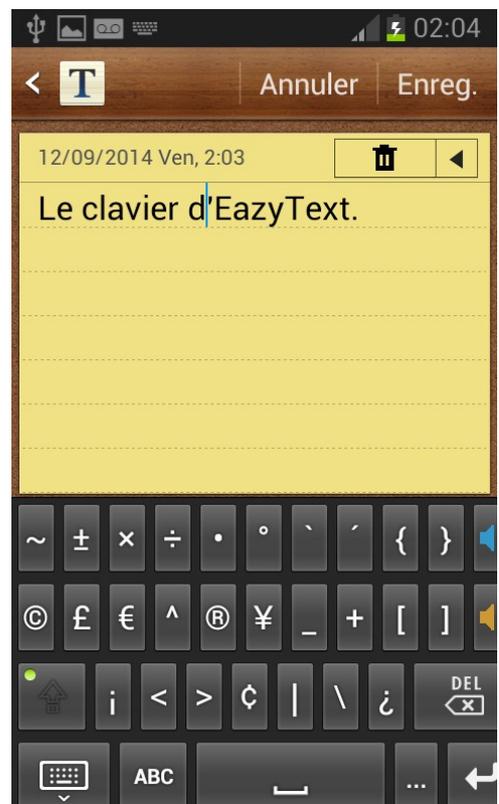
L'ensemble des lettres de l'alphabet français sont présentés. Il y a aussi un bouton pour les majuscules, un bouton pour effacer, un bouton pour basculer vers des caractères spéciaux.

Deux boutons particuliers ont été ajoutés.

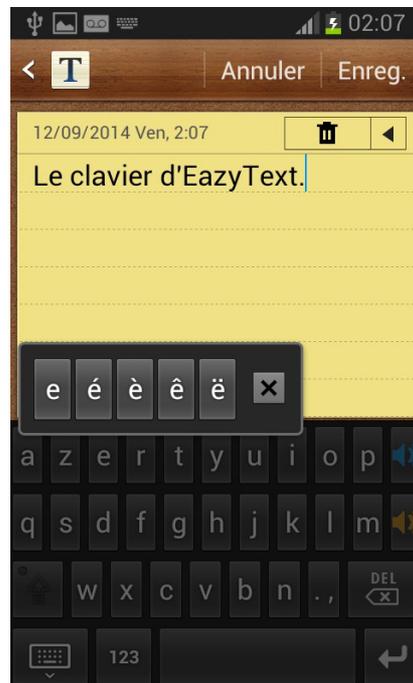
Tout à droite, un bouton avec le logo d'un haut parleur bleu, qui permet de jouer la synthèse vocale instantanément.

Juste en dessous, un bouton au logo similaire, mais de couleur orange cette fois, permet d'envoyer le texte vers *EazyText* directement.

La description des fonctionnalités seront décrites en détail dans la partie suivante.



Un appui prolongé sur une lettre du clavier peut faire apparaître un menu contextuel qui permet de choisir les différents accents possible pour une touche.



4. Les fonctionnalités

L'application *EazyText* a de nombreuses fonctionnalités, qui sont activés par trois biais :

- soit par les boutons se trouvant sur le côté
- soit par les boutons du clavier, que l'on retrouve aussi bien dans *EazyText* que dans les autres applications qui font appels au clavier
- soit via le composant éditeur de texte de *EazyText*
- soit via la fonctionnalité de copier-coller que l'on retrouve dans toutes les applications android qui permettent de sélectionner du texte

A – Les fonctionnalités des boutons latéraux

Comme nous avons vu précédemment, les boutons latéraux sont au nombre de dix, cinq boutons réparties dans 2 colonnes différentes.

Les boutons de la première colonne conditionnent les fonctionnalités des boutons de la seconde colonne.

Fonctions des boutons de la première colonne



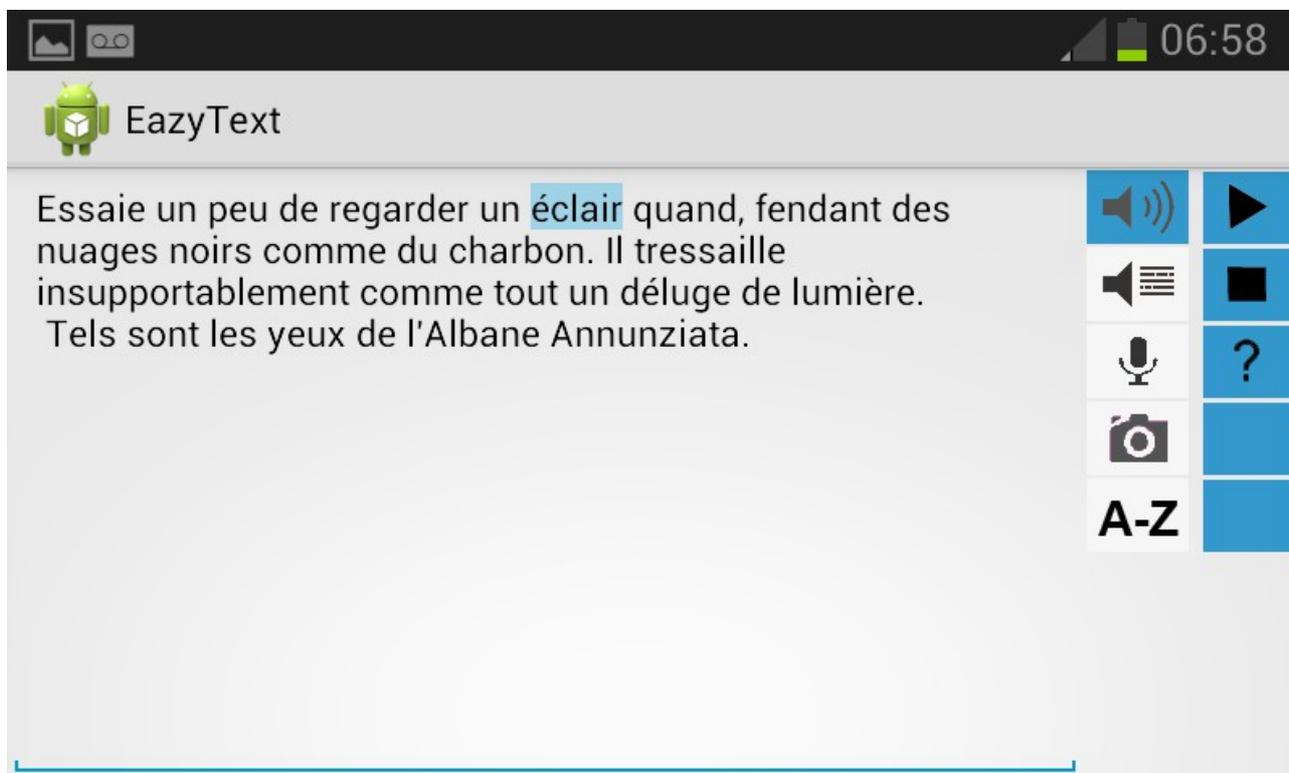
Le bouton de « Lecture simple » permet d'activer le mode « Lecture simple ».

Dans ce mode, il est possible de lire le texte contenu dans l'éditeur via la synthèse vocale, de mettre sur pause, de stopper la lecture ou d'afficher l'aide.

Ces fonctionnalités sont activables par les boutons - lecture, pause, stop et aide – qui se trouvent de la deuxième colonne.

En mode « lecture simple », au fur et à mesure que les mots sont lu par la synthèse vocale, ils sont mis en sélection, un par un.

Il n'y donc aucune colorisation de mot, seulement des mots mis en sélection.



De base dans Android, la sélection de mot entoure le texte d'une couleur bleue translucide comme on peut le constater dans cette capture d'écran.

Dans les captures d'écran ci-dessous, il s'agit d'exemples où la colorisation des mots est défini en rouge, et la colorisation des phrases/paragraphes est en jaune.



Le bouton de « Lecture avancé » permet d'activer le mode « Lecture avancé ».

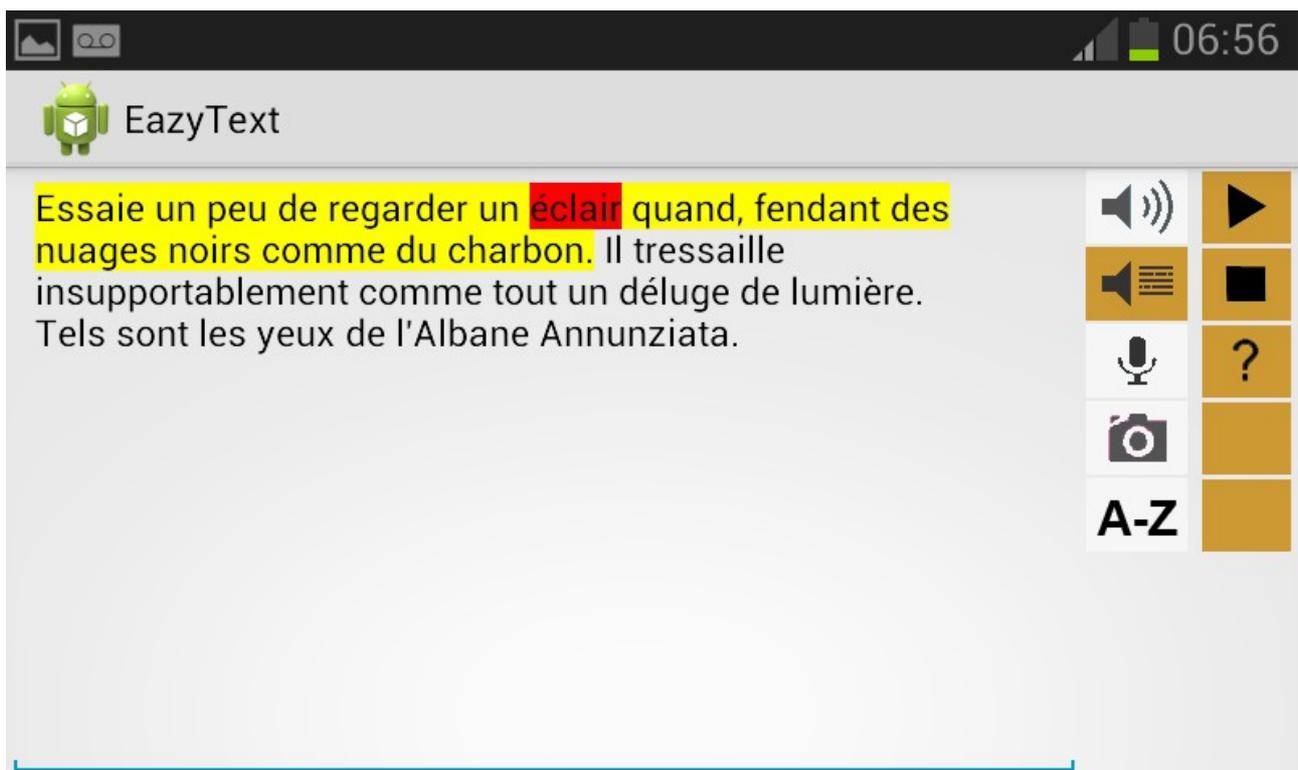
Dans ce mode, il est possible de lire le texte contenu dans l'éditeur via la synthèse vocale, de mettre sur pause, de stopper la lecture ou d'afficher l'aide.

Ces fonctionnalités sont activables par les boutons - lecture, pause, stop et aide – qui se trouvent de la deuxième colonne.

En mode « lecture avancé», au fur et à mesure que les mots sont lu par la synthèse vocale, ils sont colorisé (contrairement au mode lecture simple, où les mots sont juste mis en sélection)

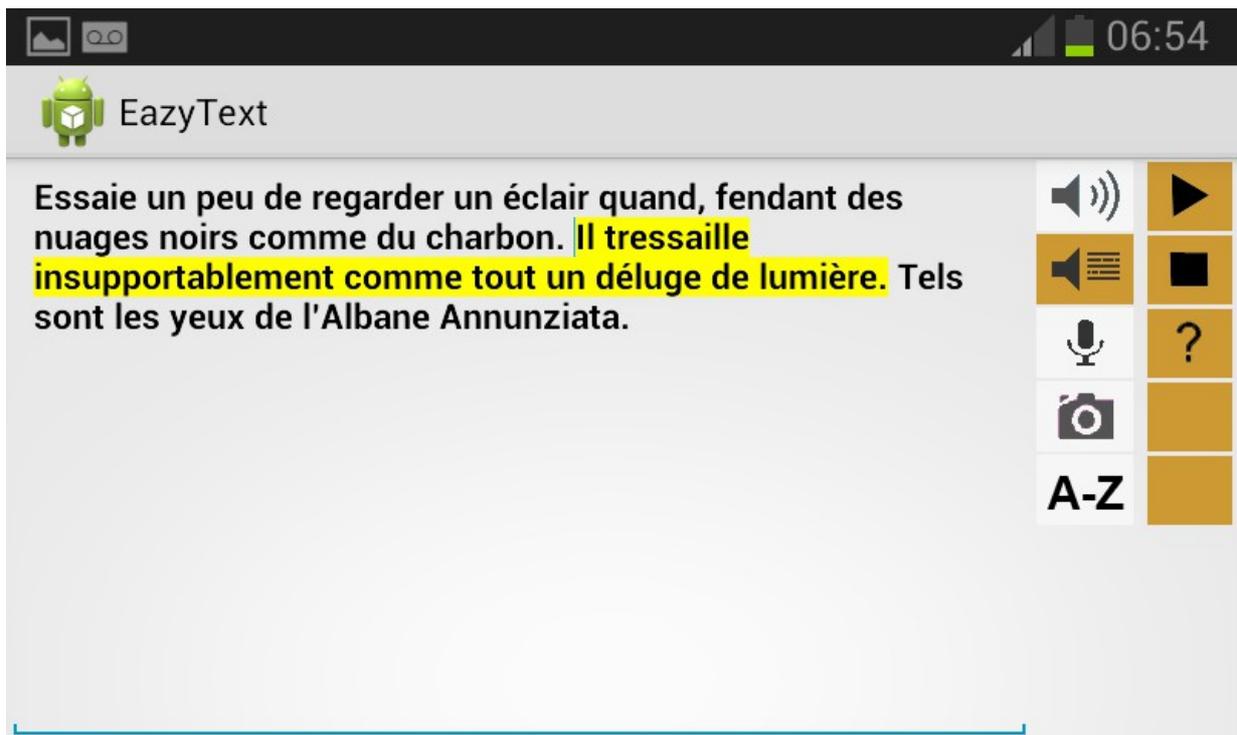
Selon les préférences, il y a plusieurs cas de figure :

- En mode « Mot à mot », le mot lu est colorisé d'une certaine couleur.

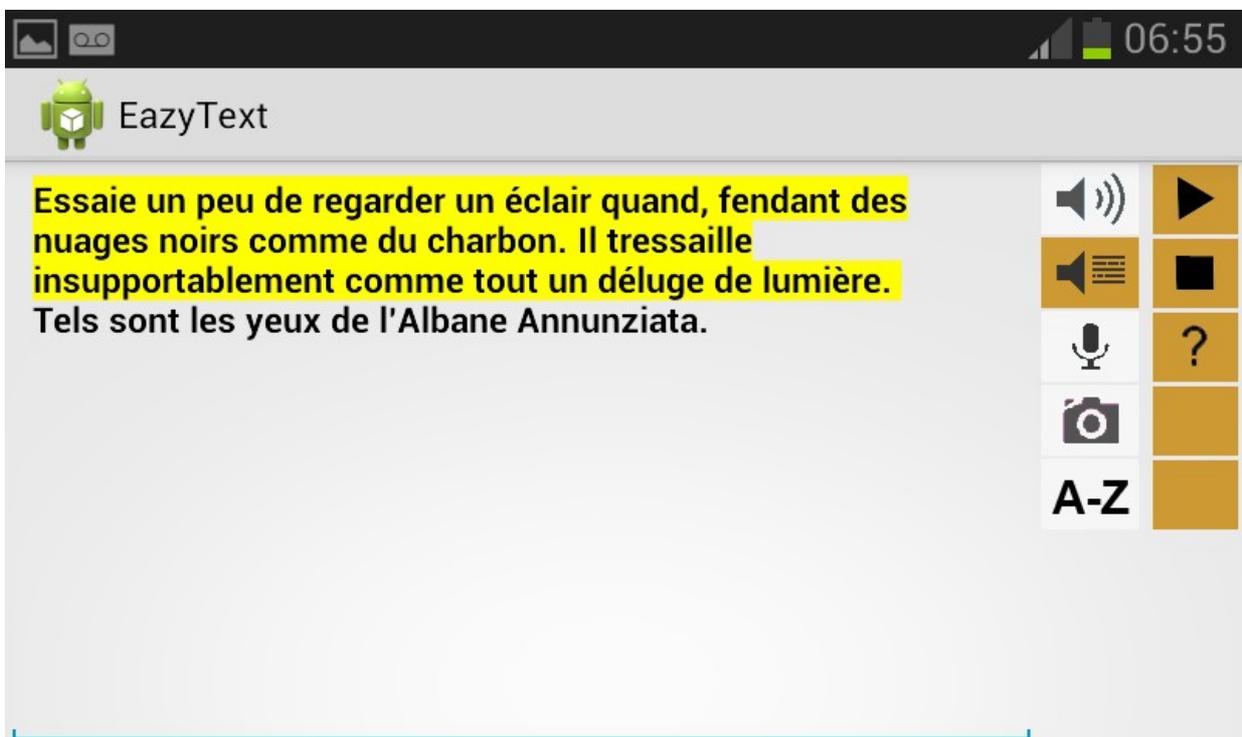


- En mode «Phrase par phrase », le mot est colorisé d'une certaine couleur, et la phrase

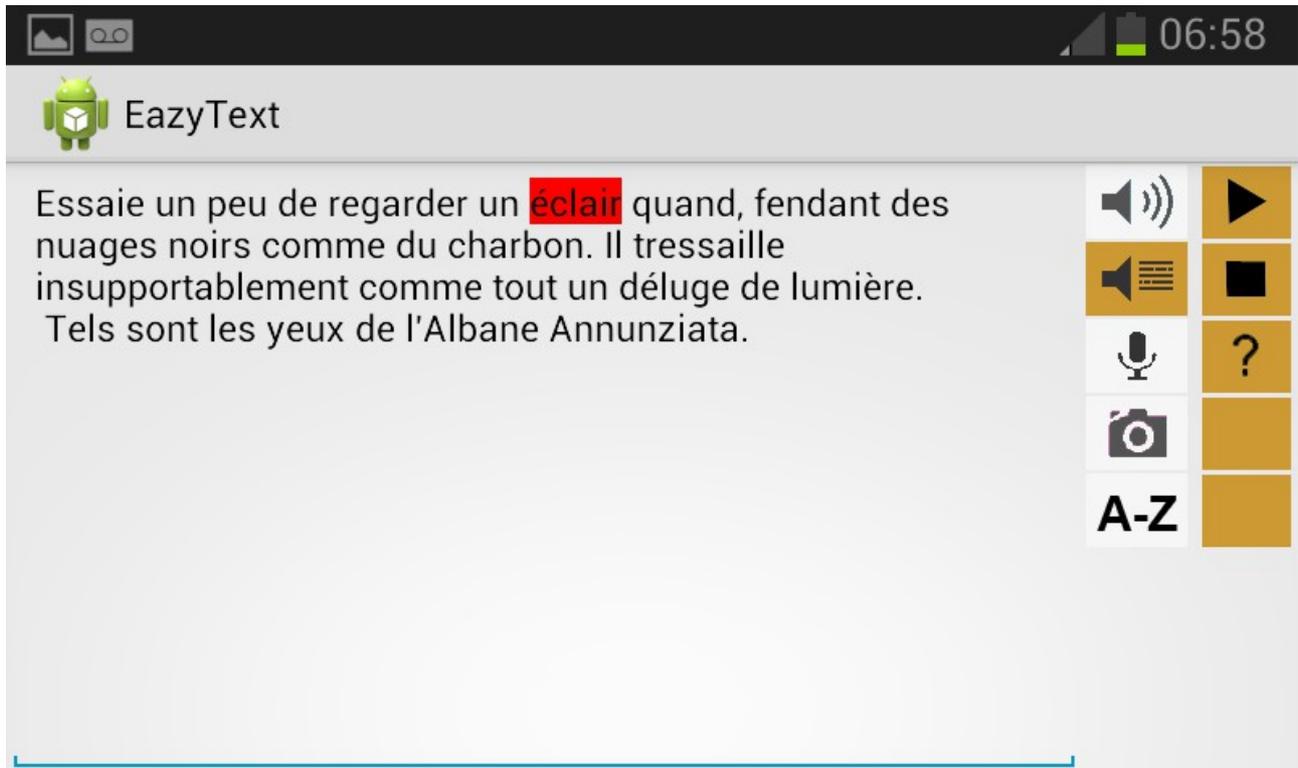
,contenant le mot en cours de lecture, est colorisé d'une autre couleur.



- En mode « Paragraphe par paragraphe », le mot est colorisé d'une certaine couleur, et le paragraphe, contenant le mot en cours de lecture, est colorisé d'une autre couleur.

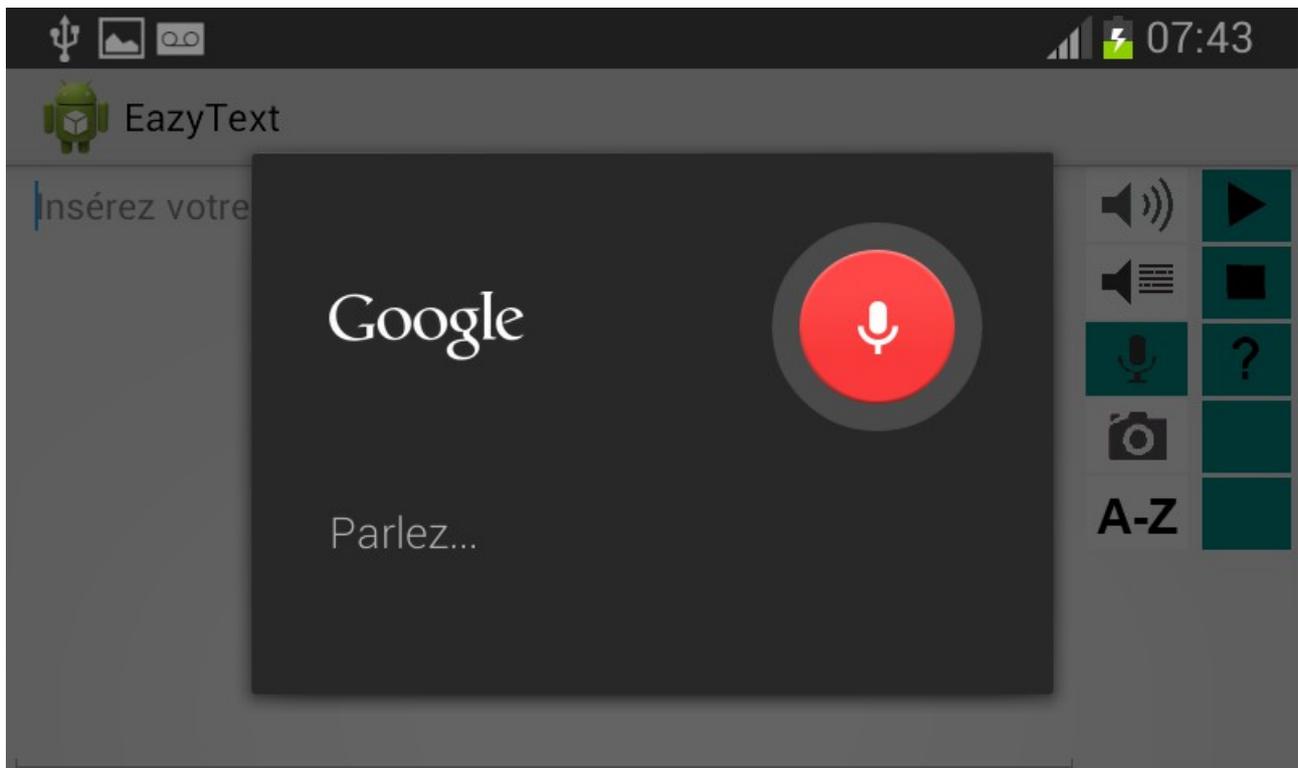


Enfin lorsque l'option « coloriser les mots seulement » est cochée dans les préférences, seul le mot lu sera colorisé. Ni les phrases ni les paragraphes seront colorisés.



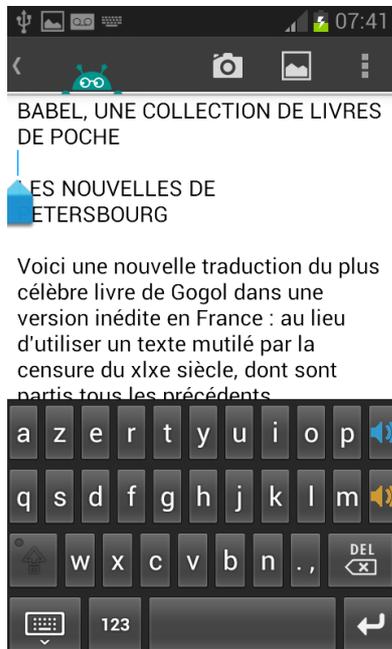


Le bouton « mode reconnaissance vocale » permet tout simplement, lorsqu'on appuie sur le bouton de lecture situé à droite (représenté par un triangle noir sur fond vert dans le cas présent), de lancer la reconnaissance vocale android afin de pouvoir écrire son texte. La reconnaissance vocale fonctionne en mode embarqué, c'est-à-dire qu'il n'est pas nécessaire d'être connecté à internet pour qu'elle fonctionne.





Le bouton « mode capture d'image » permet, lorsqu'on appuie sur le bouton de lecture situé à droite (représenté par un triangle noir sur fond rose dans le cas présent), de lancer l'OCR Text Fairy, un OCR (*Optical Character Recognition*) open source développé par une équipe de développeurs allemande. Via cet outil, on peut en quelques étapes numériser du texte que l'on prend en photo. Une fois ce texte numérisé, on a accès à un éditeur de texte dans Text Fairy.



Vous l'aurez compris alors, via cet éditeur de texte on retrouve notre clavier EazyText qui permet d'envoyer le texte numérisé vers notre application EazyText, en une seule étape (l'appui sur le bouton haut parleur de couleur orange).

FairyText est particulièrement efficace avec des appareils dotés de 5MegaPixels.

A-Z

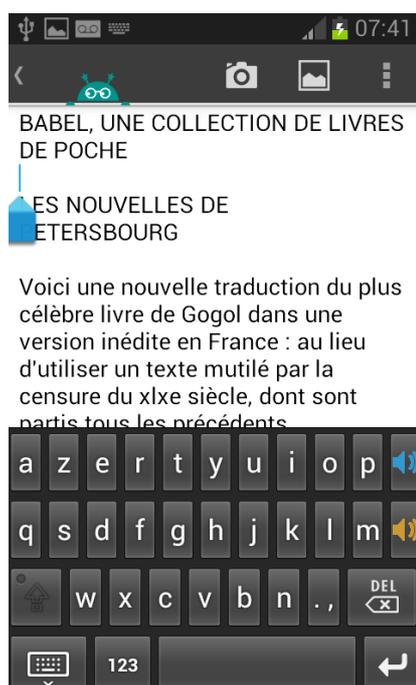
Le dernier bouton de la liste représente le mode « Dico » qui n'est pas encore implémenté. Ce mode permettra de faire appel à un dictionnaire très évolué , du type de ceux produits par ProLexis. Un partenariat avec cet entreprise a été envisagé et est en suspens pour le moment. J'avais été amené d'ailleurs à m'entretenir directement avec le président de Prolexis , Mme Wanda RZEWUSKI, pour discuter des faisabilités techniques à disposition sur Android.

B – Les fonctionnalités propres au clavier

Le clavier EazyText comporte les fonctionnalités classiques d'édition que l'on retrouve sur tous les claviers tout d'abord.

En plus de cela, le clavier EazyText comporte deux boutons spéciaux aux fonctionnalités innovantes.

Le bouton haut-parleur bleu situé tout en haut à droite permet lors d'un clic simple de lire via synthèse vocale le texte. Cela lit tout le texte sélectionné. Si il n'y a pas de sélection de texte, tout le texte contenu dans l'éditeur de texte est lu directement.



Etant donné que l'utilisateur peut cliquer sur ce bouton dans n'importe quel application qui fait appel à un clavier, la colorisation du texte lu est impossible, nous ne pouvons modifier le code des autres applications. Seul la synthèse vocale est effectuée.

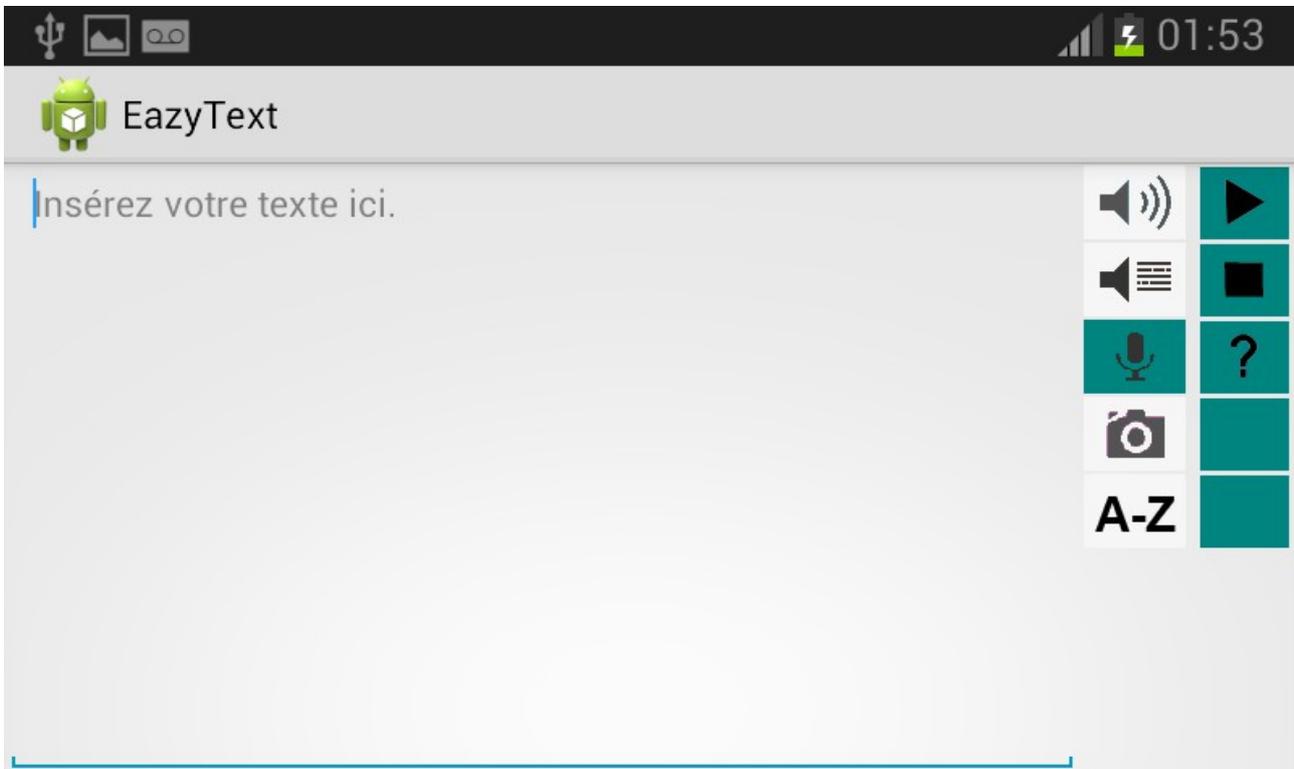
Il s'agit néanmoins de ce qu'on appelle une « augmentation » du système. En cela cette fonctionnalité est remarquable et enrichi indirectement l'API android contenu dans le smartphone ou la tablette tactile de l'utilisateur.

Le bouton haut-parleur orange, situé en dessous du bouton haut-parleur bleu dont la fonctionnalité a été décrite ci-dessus, a une fonctionnalité d'exportation du texte.

Lorsqu'on effectue un appui simple dessus, l'application EazyText se lance en mode « Lecture simple ». Il lit automatiquement le texte qui avait été sélectionné au préalable. Si aucun texte n'avait été sélectionné dans l'application antérieure lambda, c'est tout le texte qu'il y avait de contenu dans l'éditeur qui est alors envoyé vers EazyText.

C - Les fonctionnalités du composant éditeur de texte

L'éditeur de texte de l'application EazyText est simple. On peut y saisir du texte.



Il suffit d'un appui simple sur la zone de l'éditeur de texte pour afficher le clavier EazyText.

L'éditeur de texte comprend les fonctionnalités de copier-coller et de sélectionner d'une partie ou de tout le texte.

D - Les fonctionnalités de copier-coller

La dernière fonctionnalité notable de *EazyText* réside dans la gestion du copier-coller.

En effet, quelque soit l'application dans laquelle l'utilisateur se trouve, s'il utilise la fonctionnalité de copie de texte, l'application *EazyText* se lance en mode « Lecture simple » avec le texte copié qui est lu directement.

Par la suite, libre à l'utilisateur d'utiliser le mode de lecture avancé d'EazyText par exemple ou de revenir à son application antérieure en appuyant sur le bouton de retour commun à tous les smartphones et tablettes tactiles .

5 – Les notions avancées de programmation utilisés

Décrire l'ensemble des notions utilisées dans le développement de *EazyText* et du clavier *EazyText* serait bien trop long et peu pertinent.

Je vais donc vous présenter une liste non-exhaustive des notions assez avancées que j'ai utilisé dans ce projet.

Les transitions

Cette notion prend place au niveau de l'interface graphique, lorsqu'on un bouton passe d'une image à une autre. Il existe une fonctionnalité avancée dans Android qui permet de faire une animation d'une image à une autre, afin que le changement ne soit pas brutale.

Dans *EazyText*, lors du passage d'un mode à un autre, les boutons de droite voient leur fond changer de couleur. J'utilise ici la notion de `TransitionDrawable` pour que le changement se fasse progressivement. Cela donne un rendu agréable, sans quoi ce serait brutal.

Il faut pour cela créer un fichier xml représentant la transition d'un bouton à un autre. Ici on voit 2 items drawable énumérer dans une balise transition. Les deux items représentent l'image de départ (`@drawable/bpausebleu`) et l'image d'arrivée(`@drawable/bpausevert`)

```
<?xml version="1.0" encoding="UTF-8"?>
<transition xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bpausebleu" />
    <item android:drawable="@drawable/bpausevert" />
</transition>
```

Ensuite dans l'activité principale (langage java), on définit le fichier xml créé (décrit ci-dessus) comme image pour le bouton `bR1`. Puis on initialise l'objet `TransitionDrawable` via l'appel de la méthode `getDrawable()` sur l'objet `bR1` (objet qui correspond à notre bouton). Et enfin, on lance la transition via la méthode `startTransition` qui prend en paramètre le temps que durera la transition (en millisecondes) . Dans le cas présent, les transitions durent 500ms.

```
bR1.setImageResource(R.drawable.tplaybleuvert);
transition1 = (TransitionDrawable) bR1.getDrawable();
transition1.startTransition(durationMillis);
```

Le projet EazyText contient plus de 80 transitions différentes, qui sont activées lors d'un passage d'un mode à un autre (en appuyant sur les boutons de la première colonne donc).

Les interactions entre applications

L'application EazyText et le clavier EazyText ont la particularité de pouvoir lancer d'autres applications.

EazyText peut lancer FairyText (l'applicaiton OCR) par exemple. Et ce clavier EazyText peut lancer l'application EazyText (c'est un de ses deux grandes fonctionnalités , comme nous l'avons vu précédemment).

Ces interactions entre applications nécessitent qu'elles puissent « dialoguer » entre elles et qu'elles en aient la permission surtout.

Pour cela, il faut le spécifier à plusieurs endroits.

Prenons l'exemple du clavier EazyText (qui est une application en soit, à différencier de l'application EazyText) et qui fait appel à l'application EazyText.

Tout d'abord, il faut notifier dans le fichier AndroidManifest du clavier EazyText que le clavier peut lancer une application externe.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hanse.eazytextclavier">
    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="19" />
    <application android:label="@string/ime_name">
        <service android:name="EazyTextClavier"
            android:permission="android.permission.BIND_INPUT_METHOD">
            <intent-filter>
                <action android:name="android.view.InputMethod" />
            </intent-filter>
            <meta-data android:name="android.view.im" android:resource="@xml/method" />
        </service>

        <activity android:name="com.hanse.eazytextclavier.ImePreferences"
            android:label="@string/settings_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

    <activity
        android:name="com.hanse.eazytext.MainActivity"
        android:configChanges="keyboardHidden/orientation/screenSize"
        android:excludeFromRecents="true"
        android:screenOrientation="portrait" >
    </activity>

</application>
</manifest>

```

J'ai surligné ici la partie qui nous intéresse. On voit que l'activité (c'est ainsi qu'on nomme une application dans la programmation android) EazyText est répertorié et que son utilisation est paramétré.

C'est une erreur classique d'oublier de mentionner cela dans le fichier AndroidManifest, ce qui cause des bugs difficiles à interpréter lors de la compilation du projet.

Ensuite il faut faire deux choses.

Dans l'activité de notre clavier, créer la fonctionnalité qui ouvre l'application externe (EazyText dans le cas présent).

```

Intent i = new Intent(Intent.ACTION_MAIN);
i.setComponent(new ComponentName("com.hanse.eazytext",
    "com.hanse.eazytext.MainActivity"));
i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
i.putExtra("MODE", "okay");
i.putExtra("EXTRACTED_TEXT", extractText());
startActivity(i);

```

Pour se faire, comme on voit ci-dessus, on créé un objet de type Intent. Un objet Intent permet de lancer une activité externe (une autre application) typiquement. On définit ensuite son paramétrage grâce à l'appel de divers méthodes, qui permettent notamment de savoir le nom exact de l'activité à lancer, et aussi de partager des informations entre les deux applications. Ici on partage du texte entre les deux applications. Le texte à échanger est le résultat de l'appel à la méthode extractText() ici.

Puis dans l'activité externe (EazyText donc), spécifier comment réagir lorsque l'activité externe est lancé à partir d'une autre activité (l'activité de notre clavier EazyText ici). Dans la méthode qui est déclenché lors du lancement de l'application (la méthode onCreate()), méthode que l'on a surchargé (ca veut dire qu'on la redéfinit pour qu'elle est un comportement différent), on regarde si on créé un objet Intent qui permet de récupérer les informations partagés entre les deux applications :

```

Intent intent1 = getIntent();
String okay = intent1.getStringExtra("MODE");
if (okay != null) {
    if (intent1 != null) {
        handleSendText(intent1);
    }
}

```

Pour faire simple, ce bout de code signifie qu'on fait appel à la méthode `handleSendText` si et seulement si l'application a été lancée par le clavier EasyText. Pour rappel une méthode en java est semblable à une fonction en langage C par exemple.

Voici le code de cette fonction :

```

private void handleSendText(Intent intent) {
    String okay = intent.getStringExtra("MODE");
    if (okay != null) {
        sharedText = intent.getStringExtra("EXTRACTED_TEXT");
        if ((sharedText != null)
            && !sharedText.contains("android.widget.EditText")) {
            mEditText.setText(sharedText);
            scanText();
            new TrackTTS().execute();
        }
    }
}

```

Cette méthode a pour fonctionnalité de stocker dans la variable `String sharedText` le texte à partager entre les deux applications.

En somme, pour lancer une application à partir d'une autre application, il faut au préalable faire trois choses :

- modifier l'androidManifest de l'application émettrice pour accorder les permission et paramétrer le lancement de l'application
- modifier l'activité de l'application émettrice pour lancer l'application réceptrice
- modifier l'activité de l'application réceptrice pour définir son comportement quand elle est lancée par l'application émettrice uniquement

Le copier coller

Afin de modifier le comportement de l'appareil android lorsque l'utilisateur copie du texte (pour rappel lorsque l'utilisateur copie du texte dans n'importe quel application, ca lance EazyText), on procède à une « augmentation » du système android.

Pour se faire, on agit par l'intermédiaire du clavier EazyText qui est une activité particulière, c'est ce qu'on appelle un service.

C'est à dire que c'est une activité qui tourne en tâche de fond, il n'y a pas besoin de lancer quoique ce soit pour le clavier marche, il suffit qu'il soit activé par défaut.

J'ai créé la classe *ClipboardListener* qui implémente la classe *ClipboardManager.OnPrimaryClipChangedListener* :

```
class ClipboardListener implements ClipboardManager.OnPrimaryClipChangedListener {  
    public void onPrimaryClipChanged() {  
        ClipboardManager clipBoard = (ClipboardManager)  
            getSystemService(CLIPBOARD_SERVICE);  
        Intent i = new Intent(Intent.ACTION_MAIN);  
        i.setComponent(new ComponentName("com.hanse.eazytext",  
            "com.hanse.eazytext.MainActivity"));  
        i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
        i.putExtra("MODE", "okay");  
        i.putExtra("EXTRACTED_TEXT", clipBoard.getText());  
        startActivity(i);  
    }  
}
```

Puis dans la méthode *OnCreate()* du service représentant le clavier EazyText , on instancie un objet de la classe présenté ci-dessus :

```
clip = new ClipboardListener();  
clipBoard.addPrimaryClipChangedListener(clip);
```

L'objet *clip* est ainsi paramétré de manière à ce qu'il déclenche la méthode *onPrimaryClipChanged()* lorsque l'utilisateur fait appel à la fonctionnalité de copie de texte. Cette méthode, une fois déclenché, créé un intent dans lequel est stocké le texte copié, puis cet intent lance EazyText, de la manière que l'on a pu voir précédemment dans la section « Les interactions entre applications ».

En somme, on utilise le fait que le clavier EazyText est activé par défaut et que c'est un service toujours opérationnel . Ainsi on a pas besoin de lancer l'application EazyText pour que la fonctionnalité avancée de détection de copie de texte opère.

Le clavier

Comme on a pu le voir précédemment, le clavier EazyText est une application en soit. Plus exactement, il s'agit d'un service, au sens de la programmation android.

Cette application comporte huit classes différentes. La classe principale implémentant le service se nomme EazyTextClavier. Elle hérite de InputMethodService, qui est la classe mère nécessaire lorsqu'on veut implémenter son propre clavier android.

Aussi, le service EazyTextClavier implémente KeyboardView.OnKeyboardActionListener .

Autrement dit EazyTextClavier surcharge les méthodes de KeyboardView.OnKeyboardActionListener.

Voici l'en-tête de cette classe pour être plus explicite :

```
public class EazyTextClavier extends InputMethodService implements
    KeyboardView.OnKeyboardActionListener{
        .....
        .....
        .....
}
```

Pour avoir un clavier au comportement classique, je me suis basé sur le projet exemple fournit par le SDK Android. Ce projet se nomme SoftKeyboard et est la base conseillé pour implémenter son propre clavier. Il fournit un clavier américain mais il contient quelques bugs.

Un des bugs que j'ai dû corriger par exemple, c'est le fait que le texte en cours d'édition s'efface totalement lorsqu'on y insère des caractères spéciaux.

Une fois les bugs corrigés, j'ai modifié le clavier de type qwerty pour mettre une interface azerty, j'ai ajouté de nombreux caractères pour les accents notamment.

Bien entendu, j'ai aussi ajouter les deux fameux boutons situé à droite du clavier : haut-parleur bleu et haut-parleur orange.

Pour se faire, j'ai du agir à deux endroits différents :

- dans le xml (azerty.xml) représentant l'interface du clavier

```
<Key
    android:codes="9998"
    android:keyEdgeFlags="right"
    android:keyIcon="@drawable/hpbleu"
    android:keyWidth="10%p" />

<Key
    android:codes="9999"
    android:keyEdgeFlags="right"
    android:keyIcon="@drawable/hpbeige"
    android:keyWidth="10%p" />
```

Ce bout de code présente les boutons, on voit qu'on y définit l'image des boutons, l'espace occupé par les boutons l'identifiant du bouton. Cet identifiant sert dans le service *EazyTextClavier* au niveau de l'algorithme qui définit le comportement lors de l'appui sur une touche du clavier :

```
private void handleCharacter(int primaryCode, int[] keyCodes) {
    if (isInputViewShown()) {
        if (mInputView.isShifted()) {
            primaryCode = Character.toUpperCase(primaryCode);
        }
    }
    if (isAlphabet(primaryCode)/* && mPredictionOn*/) {
        mComposing.append((char) primaryCode);
        getCurrentInputConnection().setComposingText(mComposing, 1);
        updateShiftKeyState(getCurrentInputEditorInfo());
        updateCandidates();
    } else if (primaryCode == 9998) {
        if (!isPlay) {
            currentUtterance = extractText();
            map.put(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID,
                    currentUtterance);
            speakOut(extractText());
        } else {
            tts.stop();
        }
    }

    } else if (primaryCode == 9999) {
        Intent i = new Intent(Intent.ACTION_MAIN);
        i.setComponent(new ComponentName("com.hanse.eazytext",
            "com.hanse.eazytext.MainActivity"));
        i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        i.putExtra("MODE", "okay");
        i.putExtra("EXTRACTED_TEXT", extractText());
        startActivity(i);
    } else {
        getCurrentInputConnection().commitText(
            String.valueOf((char) primaryCode), 1);
    }
}
```

Les parties surlignées correspondent au comportement du clavier lors de l'appui sur les touches avec l'icone de haut parleur bleu et orange respectivement .

L'appui sur le haut parleur bleu (primaryCode 9998) lance la synthèse vocale après avoir récupéré le texte, si une synthèse vocale n'est pas déjà en cours.

L'appui sur le haut parleur orange (primaryCode 9999) lance EazyText - de manière analogue à ce qu'on a vu dans la section « Interactions entre applications » du rapport - si et seulement si une synthèse vocale n'est pas déjà en cours.

En somme l'implémentation d'un clavier spécial a été un projet en soit-même, qui mériterait une présentation détaillée qui n'a pas vraiment sa place dans un rapport de stage à mon avis, je vous invite plutôt à lire les annexes contenant l'essentiel du code.

A noter que tout le contenu informatique des projets présentés dans le rapport est stocké sur un CD-ROM fourni au Master Handi.

Dans le cas de l'application *EazyText*, j'ai surchargé `onCreate()` bien entendu, qui se déclenche au lancement de l'application. C'est le point d'entrée de l'application en quelque sorte. C'est ici qu'on initialise les principaux composants de l'application :

```
_____@Override
protected void onCreate(Bundle savedInstanceState) {
    Log.d("onCreate", "Debut");
    super.onCreate(savedInstanceState);
        getPref();
    if (!dexter)
        setContentView(R.layout.activitymaind);
    else
        setContentView(R.layout.activitymaing);
    initDone = false;
    isPlay = false;
    si = savedInstanceState;
    imeManager = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
    enableIME = new Intent(Settings.ACTION_INPUT_METHOD_SETTINGS);
    ttsIntent = new Intent(Intent.ACTION_MAIN);
    mState = NONE;
    currentUtteranceStart = "";
    initButtons();
    init_tts();
    intent1 = getIntent();
    String okay = intent1.getStringExtra("MODE");
    if (okay != null) {
        if (intent1 != null) {
            Log.d("oncreate", "Un intent a été récupéré");
            handleSendText(intent1);
        }
    }

    Log.d("onCreate", "Fin");
}
}
```

J'ai surchargé la méthode `onResume()` aussi. Cette méthode se déclenche lorsque l'application *EazyText* était en arrière-plan et que l'utilisateur la réactive.

```
_____@Override
protected void onResume() {
    if (mode == 1 || mode == 0) {
        bR1.setImageResource(R.drawable.bplaybleu);
        bR2.setImageResource(R.drawable.bstopbleu);
        bR3.setImageResource(R.drawable.baidebleu);
        bR4.setImageResource(R.drawable.bbleu);
        bR5.setImageResource(R.drawable.bbleu);
    } else if (mode == 2) {
        bR1.setImageResource(R.drawable.bplaybeige);
        bR2.setImageResource(R.drawable.bstopbeige);
        bR3.setImageResource(R.drawable.baidebeige);
        bR4.setImageResource(R.drawable.bbeige);
    }
}
```

```

        bR5.setImageResource(R.drawable.bbeige);
    } else if (mode == 3) {
        bR1.setImageResource(R.drawable.bplayvert);
        bR2.setImageResource(R.drawable.bstopvert);
        bR3.setImageResource(R.drawable.baidevert);
        bR4.setImageResource(R.drawable.bvert);
        bR5.setImageResource(R.drawable.bvert);
    } else if (mode == 4) {
        bR1.setImageResource(R.drawable.bplayrose);
        bR2.setImageResource(R.drawable.bstoprose);
        bR3.setImageResource(R.drawable.baiderose);
        bR4.setImageResource(R.drawable.brose);
        bR5.setImageResource(R.drawable.brose);
    } else if (mode == 5) {
        bR1.setImageResource(R.drawable.bplayorange);
        bR2.setImageResource(R.drawable.bstoporange);
        bR3.setImageResource(R.drawable.baideorange);
        bR4.setImageResource(R.drawable.borange);
        bR5.setImageResource(R.drawable.borange);
    }
    super.onResume();
}
}

```

Ici comme on peut voir, je fais en sorte que les icônes des boutons de droite soient les bonnes, si jamais le mode de lecture à changer entre temps.

Algorithme de colorisation en temps réel

J'ai dû créer différents algorithmes pour coloriser le texte selon le mode de lecture et les options prédéfinis (colorisation mot par mot, phrase par phrase, paragraphe par paragraphe etc.)

Ces méthodes se nomment SpeakByWord(), SpeakBySentences, colorByParagraph(), colorBySentence(), colorByWord() . Je vous invite à consulter les annexes pour voir en détail le code de ces méthodes. Elles se trouvent dans la classe MainActivity du projet EazyText.

Voici de manière très simplifié, sous forme de macro séquences, l'algorithme général de colorisation du texte lors du lancement de la synthèse vocale , dans le cas du mode « Lecture avancé » avec une colorisation mot à mot (cas le plus complet).

Remarque 1:

Au niveau programmation, lorsqu'on demande qu'un mot soit lu en synthèse vocale, on ne sait pas à quel moment il va être lu. Par conséquent on peut pas faire tout simplement la séquence : lire un mot A, colorier le mot A, lire le mot B, colorier le mot B etc.

J'ai du procéder autrement.

Remarque 2:

Android dispose d'une classe `TextToSpeech.setOnUtteranceProgressListener`, qui contient des méthodes qui se déclenchent lorsqu'un mot commence à être lu, est entrain d'être lu et lorsque le mot a fini d'être lu. J'utilise cette classe pour que la colorisation soit en temps réel, en redéfinissant principalement la méthode `onStart(String utteranceId)` qui se déclenche à chaque fois qu'un mot commence à être lu, et la méthode `onDone(String utteranceId)` qui se déclenche à chaque fois qu'un mot vient d'être lu.

Si le texte à lire n'est pas vide,

on met dans un tableau chacun des mots avec des identifiants

on fournit à la synthèse vocale ce tableau en paramètre

on lance la synthèse vocale de tout le tableau de mots

Fin Si

Lorsqu'un mot commence à être lu,

si le mot est le premier d'une phrase

on enlève les colorisations précédentes

on colorise le mot

on colorise la phrase

sinon

on colorise le mot

fin si

fin lorsque

6 – Bilan du projet EazyText

Le projet *EazyText* est fini à 95 %. Il subsiste quelques bugs de compatibilité avec des logiciels et avec certaines tablettes. Néanmoins en l'état EazyText est opérationnel. Il existe des applications présentant des fonctionnalités de colorisation similaire en mot à mot, en revanche seule cette application propose de coloriser les phrases en même temps que les mots.

De plus, aucun autre logiciel destiné aux dyslexiques propose un clavier intégré avec de telles fonctionnalités. Des entreprises partenaires de DYSTRI sont très intéressé par le projet, de part son côté innovant mais aussi de par le support android qui n'est pas courant dans ces

entreprises qui ont l'habitude de développer sur Windows et sur iOS.

Il est à noter que dans le domaine du handicap, encore beaucoup d'outils fonctionnent uniquement sur des supports qui paraissent maintenant désuets, tels que Windows. Le développement sur Android s'inscrit dans une perspective d'avenir de toute évidence, et en cela *EazyText* a toute sa raison d'être.

Aussi plusieurs aspects techniques sont assez poussés à mon sens. Alors qu'il est courant de trouver sur internet des solutions à tout problème de programmation, il s'avère qu'il y a quasiment personne qui s'est intéressé aux fonctionnalités avancées de synthèse vocale et d'échanges de contenu textuel entre applications android par l'intermédiaire de boutons présents sur le clavier.

Pour ces raisons, je suis à titre personnel très satisfait de ce projet qui s'inscrit à la fois dans le master « Technologie et Handicaps » et dans le département « Recherche et Développement » de l'Entreprise DYSTRI. Je crois pas me tromper en affirmant qu'il répond aux attentes des deux entités.

❖ **Projet EazySchool**

1 – Présentation

EazySchool est la première application *android* que j'ai créé pour DYSTRI au cours de ce stage. Il a fallu trois semaines en tout et pour tout pour développer cette application et le site internet qui va avec.

EazySchool est une application android destinée aux collégiens. L'utilisateur peut télécharger des solutions d'exercices de mathématiques en effectuant une recherche via le formulaire proposé.

Les solutions d'exercices de mathématiques sont au format PDF. Ils sont stocké sur un serveur web. Le professeur de mathématique et/ou les membres de chez DYSTRI peuvent mettre autant de solutions qu'ils le souhaitent via un formulaire sur un site internet développé spécialement pour *EazySchool*. Ce formulaire permet d'upload les PDF sur le serveur.

Il est à noter que cette application a un but pédagogique : il fournit à l'élève des solutions d'exercices et non pas des corrigés. L'élève peut donc vérifier si il a trouvé la bonne réponse à l'exercice. Si ce n'est pas le cas, alors logiquement il cherchera à trouver la bonne méthode pour arriver au résultat.

EazySchool fonctionne pour Android 4.0 minimum et requiert une connexion internet pour télécharger les fichiers PDF.

Remarque :

EazySchool fournit un lecteur de PDF si jamais l'utilisateur n'a pas déjà un lecteur de PDF sur sa tablette tactile ou son smartphone.

2 - Structure du code

++ Dossier SRC ++

Nous avons un dossier « src » contenant 3 packages, dans lesquels se trouvent les fichiers .java représentant les classes créés.

++ Dossier RES ++

Il y a aussi un dossier « res », contenant plusieurs dossiers :

- layout => fichiers .xml d'interfaces classiques
- menu => fichiers .xml d'interfaces des menus et du lecteur de PDF fournit
- values => fichiers .xml centralisant les informations utilitaires pour les couleurs, les chaînes de caractères, les dimensions des éléments graphiques et les paramètres graphiques

++ Fichier AndroidManifest.xml ++

Ce fichier indispensable représente les informations essentielles de l'application .

Il contient :

- le nom du package de l'application
- les composants de l'application (les activités, les services, les broadcast receivers) et leurs paramètres
- les processus supportés par l'application
- les permissions systèmes de l'application
- les permissions nécessaires pour les applications externes qui veulent interagir avec notre application
- le niveau minimum de l'API Android que l'application requiert pour fonctionner.
- les bibliothèques nécessaires au bon fonctionnement de l'application

3 – Interface graphique

EazySchool est doté d'une interface sous forme d'onglets.

Il y a deux onglets principaux, auxquels correspondent deux activités : l'onglet de recherche et l'onglet listant les pdf téléchargés.

Onglet « Rechercher une solution » recherche

Il s'agit d'un formulaire contenant cinq composants pour préciser la recherche de solutions d'exercices de mathématiques, et d'un bouton rechercher.

Les composants sont situés les uns en dessous des autres de manière linéaire.

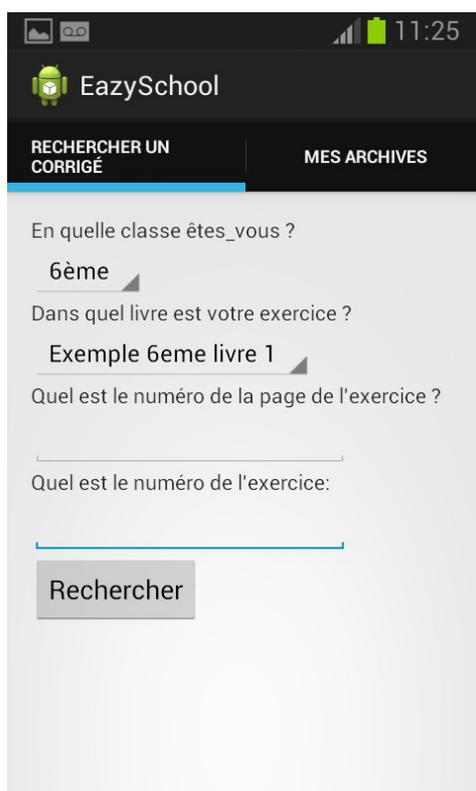
Le premier composant est un menu déroulant pour choisir la classe de l'élève – 6ème à 3ème.

Le deuxième composant est un autre menu déroulant pour choisir le livre dans lequel se trouve l'exercice.

Le troisième composant est un champ numérique pour indiquer le numéro de la page de l'exercice,

Le quatrième composant est un champ numérique pour indiquer le numéro de l'exercice,

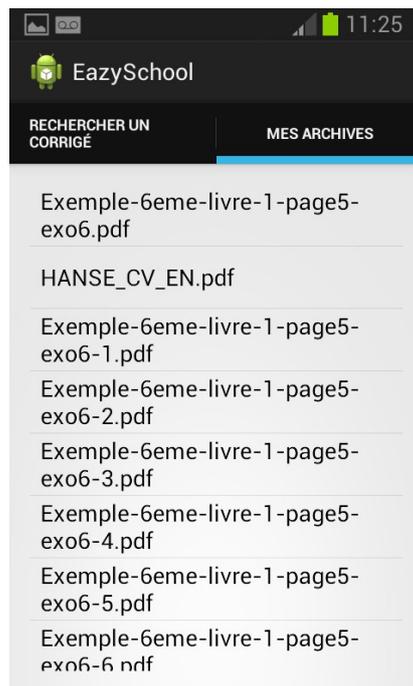
Le dernier composant est un bouton Rechercher.



The screenshot displays the EazySchool application interface on a mobile device. At the top, the status bar shows the time as 11:25. Below the status bar, the app's logo and name 'EazySchool' are visible. The main interface features two tabs: 'RECHERCHER UN CORRIGÉ' (selected) and 'MES ARCHIVES'. The search form consists of five components arranged vertically: 1. A dropdown menu for 'En quelle classe êtes-vous ?' with '6ème' selected. 2. A dropdown menu for 'Dans quel livre est votre exercice ?' with 'Exemple 6eme livre 1' selected. 3. A text input field for 'Quel est le numéro de la page de l'exercice ?'. 4. A text input field for 'Quel est le numéro de l'exercice:'. 5. A 'Rechercher' button at the bottom.

Onglet «Archives »

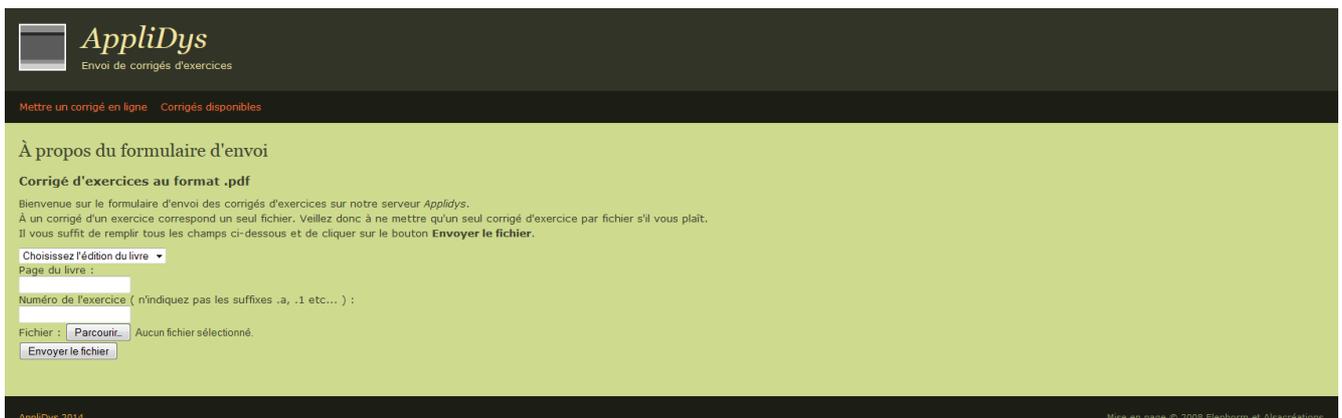
Dans cette activité, il y a simplement une liste représentant tous les documents PDF qui ont été téléchargés par l'application *EazySchool*.



Site Web

Le site a pour adresse applydys.com/eazyschoolphp/uploadF.php

Il est composé, en analogie avec l'interface android, d'une page formulaire permettant d'upload un fichier pdf en indiquant ses caractéristiques, et d'une page listant les fichiers PDF présents sur le serveur.



AppliDys
Envoi de corrigés d'exercices

Mettre un corrigé en ligne Corrigés disponibles

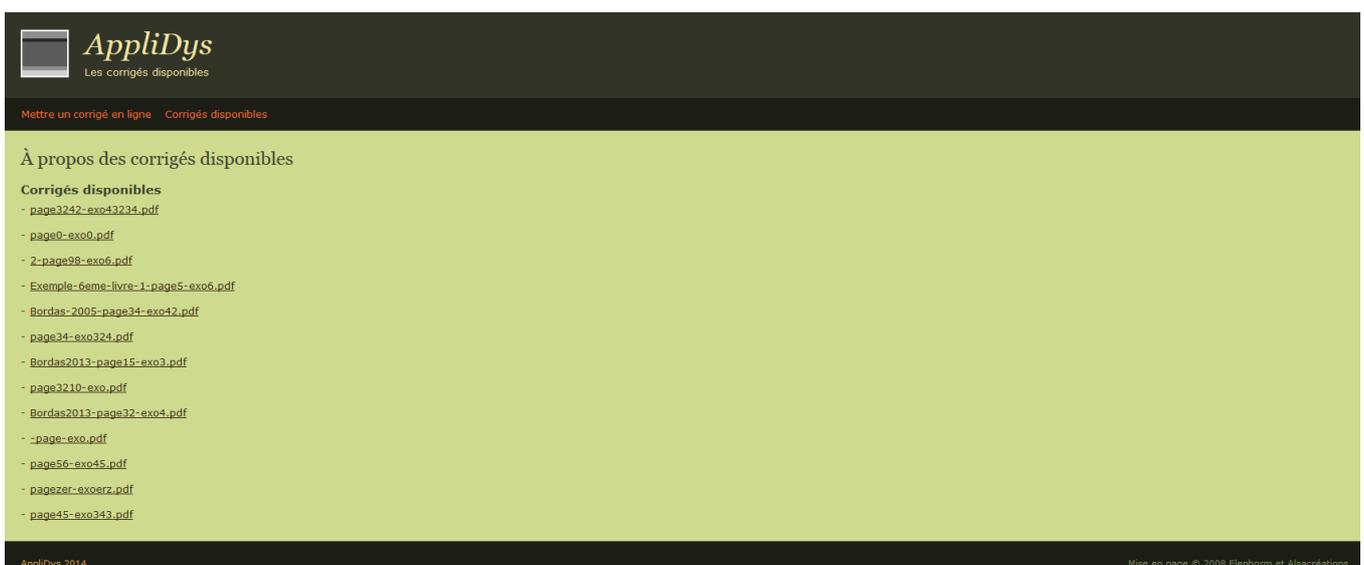
À propos du formulaire d'envoi

Corrigé d'exercices au format .pdf

Bienvenue sur le formulaire d'envoi des corrigés d'exercices sur notre serveur *AppliDys*.
À un corrigé d'un exercice correspond un seul fichier. Veuillez donc à ne mettre qu'un seul corrigé d'exercice par fichier s'il vous plaît.
Il vous suffit de remplir tous les champs ci-dessous et de cliquer sur le bouton **Envoyer le fichier**.

Choisissez l'édition du livre ▼
Page du livre :
Numéro de l'exercice (n'indiquez pas les suffixes .a, .1 etc...) :
Fichier : Aucun fichier sélectionné.

AppliDys 2014 Mise en page © 2008 Elephorm et Alsacrérations



AppliDys
Les corrigés disponibles

Mettre un corrigé en ligne Corrigés disponibles

À propos des corrigés disponibles

Corrigés disponibles

- page3242-exo43234.pdf
- page0-exo0.pdf
- 2-page98-exo6.pdf
- Exemple-6eme-livre-1-page5-exo6.pdf
- Bordas-2005-page34-exo42.pdf
- page34-exo324.pdf
- Bordas2013-page15-exo3.pdf
- page3210-exo.pdf
- Bordas2013-page32-exo4.pdf
- -page-exo.pdf
- page56-exo45.pdf
- pagezer-exoerz.pdf
- page45-exo343.pdf

AppliDys 2014 Mise en page © 2008 Elephorm et Alsacrérations

4 – Fonctionnalités

Les fonctionnalités de l'application sont les suivantes : un formulaire de recherche, une liste de fichiers PDF à ouvrir, une connexion à un serveur web, un téléchargement de fichiers sur serveurs web, envois de SMS.

Le site web a pour fonctionnalités : l'upload de fichiers pdfs via un formulaire.

Formulaire de recherche

Le formulaire est composé de deux listes déroulantes, deux champs de saisie numérique et d'un bouton de validation.

Le premier formulaire permet de choisir la classe de l'élève, parmi un choix restreint de quatre classes – 6ème à 3ème.

Le deuxième formulaire permet de spécifier le livre d'exercices, parmi un choix prédéfini.

Le premier champ de saisie numérique permet de spécifier le numéro de la page de l'exercice.

Le deuxième champ de saisie numérique permet de spécifier le numéro de l'exercice.

Le bouton « Rechercher » permet de lancer la recherche sur le site web applydis.com/eazyschoolphp/. En fonction des choix faits préalablement via les composants cités ci-dessus, une chaîne de caractère est créé avec le formatage suivant : *nomdulivre-pageX-exoY.pdf*

, où X est le numéro de la page et Y le numéro de l'exercice.

Si la chaîne de caractère correspond au nom d'un fichier PDF stocké sur le serveur, alors le bouton « Rechercher » se transforme en un bouton « Télécharger », sinon un champ texte apparaît en dessous du bouton « Rechercher » indiquant à l'utilisateur que la

solution n'est pas disponible pour l'instant.

En cliquant sur le bouton « Télécharger », le fichier PDF se télécharge dans un dossier *EazySchool* créé automatiquement sur l'appareil *android* de l'utilisateur . Un message texte indique à l'utilisateur l'état d'avancement du téléchargement .

Liste des PDF

Dans le second onglet, il y a la liste des PDFs téléchargés jusqu'à présent par l'utilisateur. En cliquant sur un des éléments de la liste, le fichier PDF s'ouvre automatiquement avec le logiciel favori de lecture de PDF de l'utilisateur. S'il n'a pas de lecteur de PDF, le fichier s'ouvre avec un lecteur PDF basique que j'ai intégré à EazySchool.

Envoi de sms

Dans le cas où l'appareil android de l'utilisateur est un smartphone, l'application envoie un sms à l'administration de DYSTRI que un fichier PDF a été téléchargé par un utilisateur. Cette fonctionnalité est expérimental et fonctionnelle. Elle a pour objectif de tester la faisabilité d'envoi de sms, qui pourraient être payants par la suite. L'utilisateur serait prévenu à plusieurs reprises avant l'envoi du sms bien entendu. Cette option de SMS payant est encore à l'étude, ouvrant le champ à la problématique de la monétisation des applications android, qui soient gratuites, soient coûteuses deux ou trois euros maximum lors de leur téléchargement.

Upload des fichiers

Le site internet permet d'uploader des PDF sur le serveur. Ces fichiers sont censé correspondre à des solutions d'exercices de mathématiques du programme de 6ème à la 3ème. Les fichiers au format PDF doivent suivre le formatage suivant : *nomdulivre-pageX-exoY.pdf*

, où X est le numéro de la page et Y le numéro de l'exercice.

Si un fichier du même nom existe déjà, l'upload ne s'effectue, de même si le formatage n'est pas correcte.

5 - Notions avancées

Pour ce projet je ne vois pas de notion avancée significative à mettre en avant, si ce n'est l'utilisation du pattern de Fragments couplé aux ActionBarActivitys.

La plupart des autres notions utilisées sont assez classiques, notamment le téléchargement d'un fichier sur un serveur FTP qui se fait en AsyncTask via un objet

DownloadManager.Request.

Les Fragments

Un fragment est un composant indépendant utilisable dans une activité. Un fragment est semblable à une activité classique, à ceci près qu'il est possible de réutiliser ce fragment aisément au sein d'autres activités ou lors d'une mise en page particulière.

Un fragment s'exécute dans le contexte d'une activité. Il possède son propre cycle de vie tout comme une activité et possède aussi sa propre interface utilisateur.

Les fragments peuvent être ajoutés à une activité de manière statique ou dynamique.

Dans EazySchool, j'ai utilisé un SectionsPagerAdapter qui permet d'allouer dynamiquement les fragments.

Le code est disponible de manière détaillée dans les annexes.

6 - Bilan

EazySchool est le premier projet que j'ai réalisé pour DYSTRI, au cours du mois d'avril. Ce projet est fini à 60 %. 80% du travail restant consiste à produire les exercices de mathématiques et les uploader sur le serveur web applydis.com

Concernant le développement, il reste encore à peaufiner certains aspects graphiques, à ajouter une ou deux fonctionnalités, comme le nettoyage du dossier de PDF lorsqu'il atteint une taille trop importante par exemple.

Il reste encore à choisir le moyen de monétisation de l'application.

Ce premier projet a permis de concrétiser un premier objectif donné par mon employeur, ce qui m'a mis ensuite sur une bonne voie pour m'atteler au cœur du stage : l'application *EazyText*. Néanmoins c'est avec fierté aussi que je reviens sur cette application qui utilise les dernières préconisations d'android en terme d'interface via sa gestion dynamique des fragments et sa présentation en forme d'onglets.

III. Conclusion

Après six mois passés chez DYSTRI, mes objectifs de stage ont été atteints, à savoir le développement d'applications android pour DYSTRI et mon intégration dans le milieu professionnel. Les principaux défis du stage étaient de pouvoir s'adapter aux situations de la vie en entreprise, approfondir mes compétences en programmation *android* et de produire du contenu technologie autour de la thématique des handicaps.

▣ Bilan global

Ce stage m'a offert l'opportunité de m'immerger de manière très concrète dans le monde de l'entreprise, et de constater les différences, non-négligeables, entre le monde Universitaire et le monde de l'entreprise. Mes compétences en programmation android ont considérablement évolué. Mes années d'études passées en informatique m'ont permis de trouver rapidement les informations nécessaires à l'avancée des projets en toute autonomie, les méthodes de programmation furent une bonne base aussi. Ce stage m'a été grandement bénéfique aussi du point de vue de l'informatique. J'ai acquis de solides compétences en développement android, que ce soit au niveau interface ou fonctionnel (synthèse vocale avancée, augmentation du système via l'implémentation d'un clavier particulier, transitions de couleur, interaction avec des serveurs web etc.)

La gestion de projet vue en cours cette année avec Mme Pousset m'a permis d'anticiper les problèmes dû au manquement d'organisation que je pouvais avoir par le passé.

Les relations au sein des entreprises que j'ai côtoyé au cours de ce stage m'ont aidé à mieux appréhender le milieu professionnel sous différents aspects : aussi bien dans les rendez-vous d'entreprises, qu'au quotidien avec les employés à tout niveau hiérarchique, ou encore avec mon maître de stage avec qui j'étais en contact continuellement. A ce propos, le dynamisme de Monsieur Girard, son savoir-faire en relations humaines et commerciales, son expertise concernant les besoins des personnes présentant des handicaps, me sera, j'en suis certain, d'un grand atout pour l'avenir en terme d'exemple à suivre.

Au-delà d'enrichir mes connaissances techniques, ce stage m'a permis de comprendre l'importance de la

communication entre collaborateurs au sein d'un projet, notamment lorsque les collaborateurs ont des spécialités différentes (dirigeant d'entreprise, développeurs, ergonomes, commerciaux, designers, publicistes).

D'un point de vue purement personnel, ce stage m'a aidé à prendre confiance en mes capacités au niveau de l'informatique. Toutefois, le stage m'a confirmé mon idée que j'avais de l'informatique comme métier. Je me vois encore moins faire devenir développeur. Je pense que c'est une bonne base pour débiter, mais je vais essayer par la suite de me pencher vers des métiers que je ne connais pas encore vraiment mais dont les domaines ont été abordés et présentés au cours du Master Handi, à savoir la formation, la gestion de projet ou bien encore le conseil/expertise autour toujours de la thématique des handicaps .

En somme ce stage a aiguisé encore plus ma curiosité, et m'a insufflé un souffle dynamisant.