

ECOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES



Avenue du Ciseau 15  
1348 Louvain-La-Neuve

# **Une application Android au service de l'apprentissage de la lecture chez l'enfant**

---

Travail de fin d'études présenté en vue de l'obtention du diplôme de bachelier en  
Informatique et Systèmes : finalité Technologie de l'Informatique



**Lucie HERRIER  
3TL1**

---

Rapporteur : Virginie VAN DEN SCHRIECK

**Année académique 2014 - 2015**

## Avant-propos

Je voudrais remercier plusieurs personnes pour leur aide à la réalisation de ce travail de fin d'études. Tout d'abord Mme Van den Schrieck, mon rapporteur, pour m'avoir mis en contact avec l'école des Bruyères et son pour feedback sur mon travail. Je remercie également Odile Paveau, institutrice à l'école primaire des Bruyères, ainsi que Laurence Henrion, pour m'avoir expliqué les méthodes d'apprentissage de la lecture chez les enfants et ainsi permis d'imaginer les exercices présents dans mon application.

Par ailleurs, je remercie l'équipe enseignante de la section Technologie de l'Informatique de l'EPHEC Louvain-La-Neuve, sans laquelle je n'aurais pas pu acquérir les compétences nécessaires à la réalisation de ce travail.

Enfin, merci à mes proches pour m'avoir encouragée tout au long de mon travail. Merci également à tous ceux que j'aurais oublié, et qui m'ont d'une manière ou d'une autre aidé et soutenu dans le cadre de ce TFE.

## Table des matières

<b>1</b>	<b>Introduction : énoncé de la thématique du TFE et motivations</b>	<b>1</b>
<b>2</b>	<b>Analyse du problème</b>	<b>2</b>
2.1	Rencontre avec des professionnelles de l'apprentissage de la lecture . . . . .	2
2.1.1	Rencontre avec une institutrice : Odile Paveau . . . . .	2
2.1.2	Rencontre avec une logopède : Laurence Henrion . . . . .	4
2.2	Conclusion des entrevues . . . . .	6
2.3	Étude des produits existants et critique sur la base des entrevues . . . . .	7
2.4	Cahier des charges . . . . .	7
<b>3</b>	<b>Présentation de l'application et des exercices</b>	<b>9</b>
<b>4</b>	<b>Outils et technologies utilisés</b>	<b>11</b>
<b>5</b>	<b>Développement de l'application</b>	<b>13</b>
5.1	Généralités . . . . .	13
5.2	Le menu principal . . . . .	15
5.3	L'exercice <i>Imagerie</i> . . . . .	15
5.4	L'exercice <i>Lecture flash</i> . . . . .	17
5.5	L'exercice <i>Anagrammes</i> . . . . .	19
5.6	L'exercice <i>Écouter le son</i> . . . . .	21
<b>6</b>	<b>Validation - test par des enfants</b>	<b>23</b>
6.1	Test 1 : l'exercice <i>Imagerie</i> . . . . .	23
6.2	Test 2 : l'exercice <i>Lecture flash</i> . . . . .	23
6.3	Test 3 : l'exercice <i>Anagrammes</i> . . . . .	24
6.4	Points relevés . . . . .	25
<b>7</b>	<b>Difficultés rencontrées</b>	<b>26</b>
<b>8</b>	<b>Pistes d'amélioration</b>	<b>28</b>
<b>9</b>	<b>Conclusion</b>	<b>30</b>
<b>10</b>	<b>Bibliographie</b>	<b>31</b>
<b>I</b>	<b>Compte-rendu des rencontres avec une institutrice et une logopède</b>	<b>33</b>
<b>II</b>	<b>Exemples de fiches Freinet</b>	<b>37</b>
<b>III</b>	<b>VOB du cycle inférieur</b>	<b>38</b>

## 1 Introduction : énoncé de la thématique du TFE et motivations

Le rôle des nouvelles technologies dans l'apprentissage chez les enfants est un sujet qui m'intéresse tout particulièrement. Je me suis souvent interrogée sur les bienfaits, ou méfaits selon certains, de l'utilisation d'appareils informatiques chez les plus jeunes. De ce fait, j'ai décidé de consacrer mon travail de fin d'études au développement d'une application Android afin d'aider et de compléter l'apprentissage de la lecture chez les enfants, en parallèle de l'école. La programmation Android est d'autant plus actuelle que le nombre de tablettes et de smartphones est en constante augmentation, et que la majorité de ces appareils tourne sous le célèbre système d'exploitation de Google. La pertinence de cette problématique a d'ailleurs été confirmée lors des travaux préparatoires de ce travail de fin d'études : parmi les applications disponibles, peu ont été réalisées en se basant sur les précieux conseils de professionnels du secteur de l'apprentissage.

Ce travail de développement d'application s'appuie essentiellement sur l'analyse des informations obtenues auprès d'une institutrice et d'une logopède. Il a également été tenu compte de l'avis des enfants. Intitulé *Une application Android au service de l'apprentissage de la lecture chez l'enfant*, ce travail de fin d'étude s'entend à démontrer qu'Android peut aider les enfants à apprendre, conjointement à la méthode traditionnelle employée par les instituteurs. En effet, une application permet aux enfants de s'entraîner à l'apprentissage de la lecture, tout en s'amusant.

La première partie est consacrée à l'analyse du problème, où j'expose ce que j'ai appris lors de mes entretiens avec les personnes travaillant dans le domaine de l'apprentissage de la lecture chez l'enfant, ainsi qu'une étude comparative du marché et un cahier des charges. Le point suivant consiste en une présentation de l'application ainsi que des exercices qui s'y retrouvent. Dans cette partie, j'explique les choix auxquels j'ai procédé afin de mettre en place des exercices reflétant au mieux les conseils obtenus précédemment. Par la suite, je présente les outils et technologies que j'ai utilisés afin de mener à bien ce projet, avant d'expliciter la manière dont s'est déroulé le développement Android à proprement parler. J'expose ensuite les résultats du test de l'application par des enfants. Enfin, je termine ce rapport par une analyse des difficultés que j'ai pu rencontrer tout au long de ce travail de fin d'études, ainsi que des améliorations qu'il serait possible d'apporter à l'application Android, avant de conclure.

## 2 Analyse du problème

Avant de me lancer dans la partie technique et la programmation de ce travail de fin d'études, il m'a fallu effectuer quelques recherches. En effet, ne s'improvise pas instituteur qui veut. Bien que nous ayons tous dans notre vie appris à lire, il m'était impossible d'imaginer concevoir sans aucune aide une application supposée aider les enfants dans leur apprentissage de la lecture. J'ai donc choisi de me documenter auprès de professionnelles travaillant avec des enfants. Afin de valider la pertinence de la problématique, je me suis également renseignée sur les applications existantes, tout en recoupant leur contenu avec les informations obtenues précédemment.

Ce point est consacré aux recherches que j'ai effectuées avant de commencer l'application en elle-même. Je présenterai tout d'abord les informations obtenues auprès de l'institutrice et de la logopède, ainsi que la conclusion de ces deux entrevues. J'analyserai ensuite les produits existants sur le marché sur la base des informations obtenues. Je terminerai avec le cahier des charges établi pour l'application suite à ces recherches.

### 2.1 Rencontre avec des professionnelles de l'apprentissage de la lecture

J'ai choisi de commencer mon travail par une étape de documentation auprès de professionnelles travaillant avec des enfants. Pour ce faire, j'ai eu recours à l'aide et l'expérience de Mme Odile Paveau, institutrice primaire à l'école des Bruyères de Louvain-La-Neuve, ainsi que de Mme Laurence Henrion, logopède exerçant à Louvain-La-Neuve également. Toutes deux m'ont dit être intéressées par le sujet de mon TFE. Elles ont accepté de me fournir des explications, des conseils, et des pistes pour le bon développement de mon application Android. Ci-dessous se trouvent les synthèses des entrevues ; le compte-rendu complet des rencontres figure quant à lui dans l'annexe I.

#### 2.1.1 Rencontre avec une institutrice : Odile Paveau

J'ai rencontré Odile Paveau le 16 décembre 2014 dans la classe de primaire où elle enseigne, à l'école des Bruyères de Louvain-La-Neuve. Lors de cette entrevue, trois grands points ont été abordés : les méthodes d'apprentissage les plus utilisées pour la lecture, les spécificités de l'apprentissage à l'école des Bruyères, et la présentation d'outils utilisés ainsi que des exemples d'exercices.

Il existe bon nombre de méthodes d'apprentissage, qui peuvent être très variées. Parmi celles-ci, Odile a fait mention de trois méthodes en particulier, et qui concernent l'apprentissage de la lecture :

- la méthode globale,
- la méthode syllabique,
- la méthode naturelle.

La méthode globale consiste à apprendre à lire à partir du mot en entier. Le but de cette méthode est "*de faire acquérir à l'élève une stratégie de déchiffrage des mots, voire des phrases, en tant qu'image visuelle indivisible*"<sup>1</sup>. En pratique, cela signifie que l'enfant, la première fois qu'il rencontre le mot, est invité à le deviner. Celui-ci mémorisera alors le mot en le rencontrant plusieurs fois dans des contextes différents (chansons, petites histoires, poèmes, ...). Le mot est

1. Citation tirée de l'article *Méthode globale*, [http://fr.wikipedia.org/wiki/M%C3%A9thode\\_globale](http://fr.wikipedia.org/wiki/M%C3%A9thode_globale)

dès lors associé à une idée, ce qui permet de qualifier cette méthode d'*idéovisuelle*.

La méthode syllabique, par opposition à la méthode globale, part des sons que forment les lettres et les syllabes afin de construire le mot. Celle-ci relie la phonétique des lettres avec l'alphabet afin de construire tout d'abord les syllabes, puis d'assembler ces dernières pour créer les mots. Cette méthode se base sur le décryptage progressif des phrases lues.

La méthode naturelle s'inspire quant à elle de la pédagogie Freinet. Cette dernière, mise au point par Célestin Freinet au siècle dernier, est fondée sur l'expression de la créativité des enfants. Il s'agit de donner à l'enfant un projet, qui lui sera utile dans son apprentissage, et qui prend en compte ses centres d'intérêts et le potentiel créatif et associatif de celui-ci. Du point de vue de l'apprentissage de la lecture, cette méthode implique de partir du sens des mots, afin de donner un sens à ce qui est appris. La collaboration de tout le groupe est nécessaire, et l'essai-erreur est appliqué. La pédagogie Freinet a d'abord été associée à la méthode globale. Cependant, son procédé différent dans la façon dont les mots sont appris, l'apprentissage de la lecture par cette pédagogie se nomme désormais méthode naturelle.

A l'école des Bruyères, les instituteurs ne se concentrent pas sur une méthode en particulier pour enseigner la lecture aux enfants. Ce sont les méthodes syllabique, pour son apprentissage fluide, et naturelle qui sont principalement utilisées, comparativement à "l'ancienne technique" des écoles primaire, dans laquelle la méthode globale primait. Néanmoins, la méthode globale n'est pas totalement remisée au placard, car elle est encore d'application dans certains exercices. Elle reste cependant peu utilisée, étant considérée comme lourde et peu optimale, à contrario des deux autres qui peuvent être plus ludiques. En effet, il est très important de faire sens pour l'enfant et de susciter son intérêt afin de faciliter l'apprentissage.

Pour apprendre à lire et mettre en pratique ces méthodes, les enfants utilisent divers outils. Tout d'abord, notons que la lecture s'apprend en premier lieu à l'aide de caractères imprimés en majuscules, puis en minuscules, et enfin avec la police de type écriture manuelle, appelée aussi *cursive*. Ceci permet aux enfants de se familiariser petit à petit avec les lettres, celles-ci présentant des motifs plus complexes au fil des étapes. Ensuite, l'enfant travaille beaucoup sur ses centres d'intérêts. Il faut le faire travailler sur des sujets qui le concernent ou l'amusent tels que son prénom, son âge, ce qu'il aime, des phrases rigolotes, etc.

Durant l'apprentissage, le rapport au son est très important pour l'enfant. Voici quelques exercices et outils qui sont proposés à l'école :

- Avec des lettres d'imprimerie disposées dans n'importe quel sens<sup>2</sup>, l'institutrice donne un son, et il faut retrouver la lettre qui produit ce son.
- Classer les prénoms des enfants de la classe en les rassemblant en fonction des sons par lesquels ils commencent. Ceci permet notamment de découvrir de nouveau phonèmes, comme dans le prénom *Hugo*, où le son entendu est *U* et le phonème associé est *Hu*.
- Dans un petit texte, qui peut être écrit sur la base d'une idée de l'enfant (par exemple "*Marie aime la danse et les poupées*"), retrouver les sons déjà connus et les souligner en couleur. La couleur aide ici et à mémoriser, et à différencier.
- Dans un texte, trouver le son qui revient le plus souvent. Cet exercice est une variante du précédent.

2. Le fait de mettre les lettres droites, à l'envers, ou de côté permet d'entraîner l'enfant à différencier les caractères.

- A partir d'un son, trouver des mots qui commencent par celui-ci, par exemple sous forme d'images pour ensuite voir comment s'écrit le mot. Au niveau supérieur, le son peut se trouver au milieu ou en fin de mot.
- Comme outil, créer un dictionnaire *référent*. Celui-ci reprend, pour chaque son (également les composés type *au*, *ou*, *en*, ...) un dessin représentatif du son (par exemple une chouette pour le son *ch*) ainsi qu'une liste des mots commençant par ce son.
- Retrouver un mot connu dans un texte (ici, la méthode globale est utilisée).
- Etc.

Lors des évaluations de l'apprentissage de la classe, la méthode naturelle est la plus souvent utilisée, car c'est celle qui fait le plus appel à l'imagination de l'enfant. Comme outil, les instituteurs utilisent notamment les fichiers Freinet. Crées sur base de la pédagogie Freinet, ces fichiers sont composés d'images associées à des mots de différents niveaux. L'idée est d'associer un (ou plusieurs) mot(s) lu(s) à une (ou plusieurs) image(s), et de vérifier les capacités de lecture de l'enfant en lui proposant soit le même mot à retrouver dans une liste sur base d'une image semblable à la précédente, soit un mot différent sur base d'une image différente des deux précédentes, soit encore l'association d'un nouveau mot en complément avec celui lu précédemment, sous forme d'une phrase. Des exemples de fichiers se trouvent à l'annexe II. Ces fichiers existent également pour d'autres matières, comme les mathématiques.

Enfin, à l'école des Bruyères, l'enfant à toujours accès à une boîte à outils pour s'aider en cas de difficultés. Celle-ci est composée du dictionnaire référent, mais aussi de panneaux et d'affiches se trouvant un peu partout dans la classe. Ceux-ci rappellent les couleurs identifiant les sons, les images associées, les dessins des enfants représentant des mots, des symboles associés aux lettres, etc.

### 2.1.2 Rencontre avec une logopède : Laurence Henrion

J'ai rencontré Laurence Henrion le 19 janvier 2015. Elle a accepté de me recevoir dans le cabinet où elle exerce son activité de logopède. Durant notre entrevue, elle a mentionné différents points que je considère comme importants, et que je détaillerai ci-dessous : l'importance de l'âge et les connaissances de bases pour la lecture, les exercices qu'elle utilise en tant que logopède et les applications spécialisées, et enfin les détails pratiques pouvant m'aider pour la programmation de mon application.

Comme expliqué précédemment, il existe plusieurs méthodes pour apprendre à l'enfant à lire. L'idéal, cependant, est de commencer à lui inculquer les bases de la lecture vers 3 ou 4 ans. En effet, ceci permet d'optimiser ses compétences par la suite. Ce principe vient notamment des méthodes pédagogiques proposées par Maria Montessori<sup>3</sup>. Cette pédagogie met l'accent sur l'importance des périodes sensibles de l'enfant<sup>4</sup>. Celle concernant le langage se déroule environ de 2 mois à 6 ans. La pédagogie Montessori part du principe qu'il existe une conscience phonologique fort présente chez l'enfant à ce moment. Celui-ci apprend beaucoup à l'aide de rimes, de comptines, etc. C'est-à-dire à l'aide de sons. Ceci rejoue ce qui m'avait déjà été dit par Odile à propos de la méthode syllabique : il est important de se baser sur le son que fait la lettre seule, ou le groupe de lettres, et non le nom qui lui est donné, car ce dernier est source de confusions.

3. Pour plus d'informations : *Pédagogie Montessori*, [http://fr.wikipedia.org/wiki/Pédagogie\\_Montessori](http://fr.wikipedia.org/wiki/Pédagogie_Montessori)

4. Une période sensible est un âge où l'enfant est plus prompt à développer certains aspects de son évolution naturellement.

Néanmoins, être dans la bonne période ne suffit pas pour maîtriser l'art de la lecture. En effet, pour apprendre à lire, l'enfant doit déjà maîtriser un certain vocabulaire à l'oral. Dès lors, si le vocabulaire de base n'est pas acquis, l'enfant ne sera pas capable de comprendre ce qu'il lit. Les premiers mots et textes seront donc composés de vocabulaire basique, pas de mots compliqués et peu fréquents, tels que *narval*, *okapi*, etc.

En tant que logopède, Laurence Henrion travaille avec différents exercices basés sur le son. Certains exercices rejoignent ce qui avait été dit aux Bruyères, notamment :

- La sélection d'un mot contenant un son entendu précédemment parmi plusieurs choix.
- Choisir une lettre qui produit le son prononcé. Ici, Laurence Henrion m'a spécifié que la lettre importait peu tant que le son était correct. Par exemple, pour le son *sss*, l'enfant peut choisir aussi bien le *s* que le *c* ou le *ç*.
- Relier un mot ou un son à une image.
- Associer les sons et les lettres à des couleurs qui leur seront spécifiques.

Un autre exercice qu'elle affectionne particulièrement est la lecture flash. Cet exercice consiste à mémoriser un mot (ou une phrase) affiché plus ou moins rapidement à l'écran, pour ensuite le réécrire correctement. Dans le cas présent, l'orthographe est travaillée en plus de la lecture. Ce sont deux matières très liées. Un jeu de lecture flash doit idéalement :

- avoir un temps réglable d'affichage selon le niveau de l'enfant,
- insister sur l'importance de la mémorisation,
- faire attention au sens de la lecture.

En plus de cela, l'exercice peut être amélioré grâce à :

- l'affichage de mots personnalisés en fonction des centres d'intérêt de l'enfant,
- la lecture effectuée avec un mot, et la réécriture sous la forme d'une image à choisir.

Laurence Henrion utilise beaucoup son iPad pour travailler avec les enfants. En effet, c'est plus fun pour l'enfant de jouer avec une tablette. La majorité des applications présentes sur l'iPad ont été créées par un spécialiste orthophoniste (appellation d'un logopède en France) : Emmanuel Crombez. Celui-ci réalise des applications pour iPad, iPhone et Mac sous le nom *ABC Applications*. Celles-ci sont destinées à aider les enfants dans leur apprentissage général et dans leurs difficultés : lecture, écriture, mathématiques, etc. Parmi ces jeux, on peut trouver<sup>5</sup> :

- *Anagrammes*. L'enfant doit retrouver des mots dont les lettres ont été mélangées. Au besoin, il peut s'aider en écoutant le mot qu'il doit reconstituer.
- *Nuages de mots*. Cet exercice propose à l'enfant de retrouver le genre et le nombre des mots affichés en "nuages". Quel mot est masculin, ou féminin ? Est-ce singulier ou pluriel ? Le jeu entraîne à la lecture mais aussi au classement. Ce jeu existe aussi sous la forme *Nuages de lettres*, adapté aux enfants de maternelle pour apprendre à reconnaître les lettres.
- *Mémo des mots*. Ce jeu se présente sous la forme du bien connu *Memory*. La différence étant qu'ici des mots sont utilisés à la place des images. *Etc.*

Pour terminer, j'ai reçu quelques conseils concernant ce qui était préférable d'éviter pour mon application, et ce qui pouvait m'être utile. Tout d'abord, il est déconseillé d'inclure trop d'éléments graphiques. Il vaut mieux opter pour un design simple et épuré. Trop d'objets sur l'écran, c'est prendre le risque que l'enfant soit distrait et ne se concentre pas sur ce qui est requis, donc perdre en efficacité. Ensuite, il existe une police open source conçue pour faciliter la lecture chez les

---

5. Liste complète sur le site d'Emmanuel Crombez : <http://abc-applications.com/ipad.html>.

personnes dyslexiques en accentuant certaines parties des lettres. Celle-ci se nomme *OpenDyslexic*. Dans le cas où celle-ci n'aurait pas été disponible, un police de type sans serif telle que *Verdana* est idéale. En dernier lieu, Laurence m'a conseillé d'utiliser le VOB (Vocabulaire Orthographique de Base) pour les exercices que je mettrai en place dans l'application. Il s'agit de la liste officielle en Belgique des mots qui doivent être connus et maîtrisés par les enfants à la fin de chaque cycle de primaire.

## 2.2 Conclusion des entrevues

Les interviews avec Odile Paveau et Laurence Henrion m'ont permis de comprendre les méthodes utilisées dans l'apprentissage de la lecture chez l'enfant. Elles m'ont expliqué ce qui était le plus efficace, ainsi que les erreurs que je devais éviter. Voici ce que j'en ai retiré.

Premièrement, je me suis renseignée un peu plus sur les différentes méthodes d'apprentissage afin de choisir ce que j'allais développer comme exercices. Je n'ai pas étudié le sujet en profondeur, l'essentiel pour moi étant de bien comprendre le fonctionnement des méthodes et ce que je pouvais en retirer. Voici en quelques mots une synthèse des différentes méthodes et pédagogies qui ont été abordées :

- la méthode globale : apprendre à lire à partir du mot en entier. C'est une méthode idéovisuelle qui veut que l'enfant mémorise les mots rencontrés sous forme d'image et d'idée, et puisse les reconnaître plus tard ;
- la méthode syllabique : opposée à la méthode globale, elle part du son des syllabes et des lettres pour former les mots. Il s'agit ici de déchiffrer ;
- la méthode naturelle inspirée de la pédagogie Freinet : elle part du sens des mots et des intérêts de l'enfant. L'idée est de stimuler l'enfant, sa créativité, ce qu'il aime, afin de mémoriser le mot ;
- la pédagogie Montessori : met l'accent sur les périodes sensibles de l'enfant et l'apprentissage spécifique lié à celles-ci. Pour la période liée au langage et à la lecture, il existe une conscience phonologique fort présente chez l'enfant : il apprend beaucoup grâce aux comptines, chansons, rimes, etc.

Tant Odile que Laurence ont insisté sur l'intérêt que doit porter l'enfant aux jeux et exercices, ainsi que sur l'efficacité de travailler avec les sons des lettres et les syllabes. Il est donc évident pour moi que les méthodes idéales pour une application sont la méthode syllabique et la méthode naturelle. Concernant la pédagogie Montessori, je n'ai pas de prise sur l'âge des utilisateurs.

À côté des méthodes d'apprentissage, différents éléments ressortent des entrevues. Tout d'abord, le point le plus important est de susciter l'intérêt de l'enfant pour l'apprentissage de la lecture. Éveiller sa curiosité en choisissant des sujets qui l'intéressent ou en utilisant des exercices ludiques lui permet d'apprendre plus efficacement.

D'autre part, certains exercices m'ont paru plus importants, ou plus intéressants comparés à d'autres. Le premier est celui où il faut retrouver les sons de lettres ou de syllabes dans les mots. Autant Odile que Laurence m'en ont parlé comme d'une "base" dont l'efficacité a été prouvée. Certains des autres exercices mentionnés me paraissent également adaptables pour mon application :

- les fichiers Freinet pour la lecture,
- retrouver des mots dans un texte,
- la lecture flash, sur laquelle Laurence Henrion a particulièrement insisté,

- les anagrammes,
- les nuages de mots.

Enfin, le dernier point, très important à mes yeux, est le conseil d'utilisation de la police spécialisée *OpenDyslexic* pour la lisibilité du texte de l'application, ainsi que l'emploi du VOB pour le vocabulaire proposé.

## 2.3 Étude des produits existants et critique sur la base des entrevues

À la suite des entrevues avec Odile Paveau et Laurence Henrion, j'ai étudié les produits existants sur le marché dans le domaine de l'apprentissage de lecture. J'ai exploré diverses plateformes afin de comparer ce qui existe déjà, et réfléchi à ce que j'avais la possibilité d'y apporter, tout en recoupant le contenu des applications et mes nouvelles connaissances.

J'ai concentré mes recherches sur les *stores* respectifs d'*Apple* et d'*Android*. Je ne me suis pas attardée sur le *Windows Store*, celui-ci étant selon moi encore trop peu fourni. J'ai constaté qu'il existe déjà des applications aidant les enfants dans leur apprentissage de la lecture, aussi bien pour les appareils Android que pour les iPads. Cependant, la plupart des applications disponibles le sont pour des langues autres que le Français. Il n'existe pas un grand choix d'applications en Français qui soient complètes tout en étant vraiment amusantes pour les enfants, surtout sous Android. Comparativement, il va sans dire que l'offre de l'*Apple Store* est plus variée, notamment grâce aux applications d'Emmanuel Crombez, non disponibles pour Android.

J'ai parcouru une série d'applications, notamment celles que Mme Van den Schrieck a installées pour ses enfants sur son iPad. La plupart ont un aspect très ludique et amusant. Malheureusement, avec ce que j'ai appris concernant les méthodes d'apprentissages de la lecture, certaines me semblent trop peu structurées et sans méthodologie concrètement définie.

Une application disponible sur les deux plateformes qui m'avait plu et que je considère comme la mieux conçue s'appelle *Lire avec Sami et Julie*. En effet, celle-ci favorise la méthode syllabique, ainsi que naturelle. Elle travaille sur le son, avec des images, et crée ainsi des associations chez l'enfant. Cependant, *Lire avec Sami et Julie* est une application au design et à l'ambiance sonore trop chargés. Cela n'aide pas l'enfant à se concentrer, comme me l'a expliqué Laurence Henrion.

Bien évidemment, les applications d'Emmanuel Crombez sont de très bonne facture. Le seul reproche que j'ai à leur propos est que chaque exercice représente une application séparée. Mon but est dès lors de réaliser une application qui soit plus complète que celles que j'ai pu rencontrer. J'entends par là qu'elle offre une plus grande variété d'exercices pour l'enfant, se basant sur les méthodes expliquées lors des entrevues, et appliquant les conseils reçus.

## 2.4 Cahier des charges

Sur la base des entrevues ainsi que de l'étude des produits existants sur le marché, j'ai établi un cahier des charges. Celui-ci reprend les différents points que j'aimerais mettre en place dans mon application, les spécificités de celles-ci, etc.

Tout d'abord, l'application présentera un design simple et épuré au maximum. Je ne veux pas pour autant que ce soit laid ou ennuyeux. La couleur de fond de l'application sera le bleu. En effet,

le bleu plaît à une majorité de la population et inspire le calme. Il s'agit d'une couleur établissant un bon compromis entre esthétique et ambiance pour se concentrer.

Toujours afin d'appliquer les conseils reçus, je ferai d'*OpenDyslexic* la police principale de l'application. Dans le cas où je n'y parviendrais pas, j'opterai pour du sans serif.

L'application sera composée de quatre exercices. Ceux-ci seront inspiré de la méthode syllabique ainsi que de la méthode naturelle, qui semblent être celles préconisées dans le domaine de l'apprentissage de la lecture. Ces exercices seront basés sur ce qui est utilisé par les instituteurs des Bruyères, ainsi que par la logopède. J'ai décidé de mettre en place :

- un exercice inspiré des fichiers Freinet utilisés pour les évaluations de la lecture. J'aime beaucoup l'association image et mot ;
- un exercice de lecture flash. Laurence Henrion a particulièrement insisté sur cet exercice lors de notre entrevue ;
- un exercice d'anagrammes ;
- un exercice de repérage du mot qui contient un son en particulier.

Tous ces exercices utiliseront comme base de vocabulaire le VOB.

Concernant les exercices, la première étape est de les implémenter avec chacun un niveau basique. Dans l'idéal, si le développement se passe bien, j'ajouterais des niveaux de difficulté. Dans ce cas, chaque exercice aurait donc 3 niveaux : facile, moyen et difficile.

Enfin, j'aimerais que l'application soit ludique, afin que les enfants aient envie de l'utiliser par eux-même. Ceci passe d'abord par les exercices qui doivent leur paraître "fun". Pour ce faire, j'utiliserais des bruitages, par exemple lors de la validation d'une réponse. J'aimerais également intégrer une mascotte à l'application, si possible.

### 3 Présentation de l'application et des exercices

L'application Android développée dans le cadre de ce TFE s'appelle *Manabu*. Il s'agit d'un jeu didactique ayant pour but d'aider les enfants dans leur apprentissage de la lecture. J'ai choisi de lui donner le nom de *Manabu*, qui signifie *apprendre* en Japonais. Je trouve que ce nom sonne bien, est facile à retenir, et correspond au thème de l'application. *Manabu* est conçue pour un usage principal sur tablette. Elle peut également fonctionner sur smartphone. Cependant, je trouve que la taille d'écran d'une tablette convient mieux à l'apprentissage chez l'enfant : plus grande surface de jeu, plus de "marge de manœuvre", plus convivial, plus facile d'un point de vue de la taille de la police des mots, etc.

Comme prévu dans le cahier des charges, l'application *Manabu* est composée de quatre exercices : *Imagerie*, *Lecture flash*, *Anagrammes*, *Ecouter le son*. Les noms des exercices rappellent le thème de ceux-ci.

Le premier exercice, *Imagerie*, est inspiré de la pédagogie Freinet<sup>6</sup>. Je me suis basée sur un outil utilisé dans les classes de primaire : les fichiers Freinet. Pour rappel, ces fichiers sont composés d'exercices d'évaluation dans différentes matières, telles que le français et les mathématiques. Dans le cas présent, je me suis inspirée de celles consacrées à l'évaluation de la lecture pour la première primaire.

Le principe de l'exercice, identique à celui des fichiers Freinet, est simple : une image représentative est associée à la lecture d'un mot ou d'une phrase. Une fois le mot lu, l'enfant retourne la fiche. Au dos de celle-ci se trouve la même image, où une image proche exprimant la même idée, ainsi que trois choix. L'enfant doit choisir parmi ces trois choix lequel correspond à ce qu'il a lu précédemment, en s'aidant de l'image. Il existe également une version plus compliquée, basée sur le principe de discrimination. Dans ce cas, deux mots/phrases sont présentées à l'enfant, accompagnés de deux images. Au dos se trouvera alors une troisième image, représentant le choix différent des deux lus précédemment. Il s'agit de celui à choisir.

Le deuxième exercice, *Lecture flash*, me vient de Laurence Henrion. Comme expliqué précédemment, le principe de ce type de lecture est assez simple. Comme son nom l'indique, il s'agit de lire un mot rapidement, et de le mémoriser. La durée de lecture du peut varier en fonction des facilités, ou difficultés, de l'enfant. Une fois le mot lu et mémorisé, il est demandé d'écrire celui-ci (ou taper dans le cas présent) à l'endroit prévu à cet effet. Pour un enfant plus expérimenté, le principe peut être appliqué à des phrases plus ou moins longues.

L'avantage de cet exercice est qu'il travaille non seulement les capacités de lecture de l'enfant, mais également celles de restitution ainsi que l'orthographe. Ceci permet notamment une meilleure mémorisation des mots et le travail de la rapidité de lecture. Le but du jeu étant évidemment de réussir à lire un maximum, en un minimum de temps.

Le troisième exercice est *Anagrammes*. Celui-ci fonctionne de la manière suivante : les lettres d'un mot sont mélangées de manière aléatoire. L'enfant, lors de l'affichage des lettres, entend le mot qu'il doit reconstituer. Il lui faut remettre les lettres dans l'ordre afin de compléter l'exercice. Bien entendu, l'enfant peut ré-écouter le mot autant de fois que cela lui est nécessaire.

---

6. Explications concernant cette pédagogie au point 2.1.1

Le défi de cet exercice réside en premier dans la reconnaissance des lettres, et ensuite dans la connaissance de l'orthographe des sons et des mots. En effet, certains sons sont composés de plusieurs lettres, tels que *ou*, *au*, *ai*, etc. Pour aider l'enfant dans la reconnaissance des lettres, les voyelles et les consonnes sont de couleurs différentes, ce qui permet déjà une première différentiation.

Enfin, le quatrième et dernier exercice, *Ecoute le son*, est une application de la méthode syllabique. Pour rappel, cette méthode d'apprentissage se base sur la reconnaissance des sons formés par les lettres et les syllabes pour enseigner la lecture.

L'exercice fonctionne de la manière suivante : plusieurs mots sont affichés à l'écran, et le son d'une syllabe ou d'une lettre est prononcé. L'enfant doit retrouver parmi les propositions le mot dans lequel se trouve le son entendu. Pour le niveau le plus facile, la syllabe se trouve au début du mot. Ensuite, la difficulté augmente, et il s'agit de trouver le son au milieu d'un mot, puis n'importe où dans le mot.

*Manabu* est donc une application idéale pour entraîner les enfants en parallèle de l'école, à la maison. Que ce soit pour améliorer les connaissances ou aider un enfant qui rencontre des difficultés, les différents exercices sont conçus dans le but d'apprendre en s'amusant. Ceux-ci ont été imaginés de manière à exploiter des méthodes similaires à celles utilisées durant la scolarité et dans le cadre de la logopédie. Il s'agit réellement d'un plus.

## 4 Outils et technologies utilisés

Ce point est consacré aux choix technologiques que j'ai effectués afin de mener à bien la réalisation de mon travail de fin d'études. Ceux-ci sont peu nombreux, compte tenu du fait qu'il est déjà possible d'effectuer beaucoup d'opérations avec les librairies de base d'Android. Par ailleurs, je n'avais pas besoin d'un grand nombre d'outils technologiques. Je commencerai par expliquer les spécifications du développement Android. Je détaillerai ensuite les outils utilisés tout au long du travail.

Le développement Android tel que je l'ai effectué est réalisé à l'aide de deux langages : XML et Java. Le XML est utilisé pour la description des interfaces graphiques de l'application. Ces fichiers sont appelés des layouts, ou encore parfois des vues. Le Java est quant à lui employé pour la programmation des actions à effectuer.

Lors du développement, la structure principale est l'application. Cette dernière inclut une ou plusieurs activités. Selon le site d'Android Developer, une activité se définit comme suit :

*"An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows."*<sup>7</sup>

Une activité est donc la composante principale d'une application (sans au moins une activité, ça ne fonctionne pas).

Bien entendu, à côté des activités existent encore d'autres concepts. Une activité peut être associée à une ou plusieurs vues (définies par les layouts), ce qui fait que l'interface peut changer. Lors de l'élaboration des layouts, divers éléments peuvent être mis en place pour construire l'interface graphique (boutons, champs texte, images, etc.). Je ne me lancerai pas ici dans l'explication de tout ce que l'on peut trouver en Android. Ceci sera détaillé par la suite au besoin.

En ce qui concerne les outils, j'avais en premier lieu choisi de développer mon application avec l'IDE Eclipse Juno, comprenant le SDK Android afin de pouvoir programmer pour Android. Cependant, j'ai rencontré quelques problèmes techniques qui m'ont poussée à changer d'IDE. En effet, voulant utiliser une librairie externe avec une dépendance Maven, j'ai donc installé les outils permettant d'utiliser Maven avec Eclipse. Malheureusement, après l'installation des outils, Eclipse a strictement refusé de fonctionner plus de quelques minutes après chaque démarrage, m'empêchant de convertir mon projet existant pour l'utilisation de Maven, et bloquant au final. En raison de ce problème, j'ai décidé de passer d'Eclipse à Android Studio, considéré maintenant comme stable et efficace pour la programmation Android. En effet, Android Studio est à présent l'IDE officiel pour Android. Toutefois, celui-ci ne fonctionne plus à l'aide de dépendances Maven pour les librairies externes, mais grâce à Gradle. Après avoir importé mon projet Eclipse dans Android Studio sans encombre, j'ai constaté de Gradle qu'il est vraiment facile d'utilisation. J'ai donc choisi de continuer mon développement à l'aide de ce nouvel IDE, sans regrets.

De plus, toujours concernant la programmation, j'ai décidé de faire régulièrement des tests sur smartphone et tablette afin de vérifier la compatibilité de l'application *Manabu* sur les différents appareils. J'avais à ma disposition : une tablette Samsung Galaxy 2 10.1, un smartphone Wiko Darkmoon, une tablette Samsung Galaxy Tab Pro et une tablette Nvidia Shield Tablet. La ver-

---

7. Android Developer, Activities, <http://developer.android.com/guide/components/activities.html>

sion du système d'exploitation Android est également différente entre les appareils : de la 4.0 Ice Cream Sandwich à la 5.1 Lollipop. Ceci m'assure de toucher un large public, mon application étant compatible à partir l'API 14, autrement dit 4.0, 4.0.1, 4.0.2 Ice Cream Sandwich. Par ailleurs, afin d'assurer la portabilité de mon travail entre les différents ordinateurs, j'ai créé des repositories sur GitHub, aussi bien pour le code que pour le rapport. Outre la portabilité, ceci me procure aussi un backup supplémentaire online.

Enfin, j'utilise divers outils, pas forcément technologiques, afin de compléter mon travail de fin d'études. Ceux-ci sont :

- Adobe Illustrator pour les graphismes, beaucoup plus adaptables lorsqu'ils sont réalisés en dessin vectoriel,
- Audacity, un logiciel d'enregistrement et d'édition audio,
- OpenDyslexic, une police de caractère adaptée pour faciliter la lecture,
- le VOB (Vocabulaire Orthographique de Base) du premier degré, pour les mots que les enfants devront lire ou reconstituer lors des jeux.

## 5 Développement de l'application

Ce point du rapport est dédié au déroulement du développement de l'application *Manabu*. J'expliquerai ce que j'ai mis en place point par point, en fonction des différentes parties de l'application. Je commencerai par énoncer quelques généralités concernant le programme, puis j'expliquerai le menu principal, et enfin les différents exercices au niveau facile, le 1. J'illustrerai au maximum mes propos par des screenshots<sup>8</sup> des différents exercices.

### 5.1 Généralités

Par généralités, il faut entendre les décisions que j'ai prises et qui affectent l'application dans son ensemble ou simplement des points nécessitant d'être explicités pour une meilleure compréhension du développement de l'application.

Tout d'abord, avant de me lancer dans la création de *Manabu*, j'ai suivi le premier tutoriel disponible sur le site officiel d'Android pour les développeurs, *Building Your First App*<sup>9</sup>. Ceci m'a permis de me familiariser avec le SDK Android et de connaître les bases de ce type de programmation.

Ensuite, il faut savoir que *Manabu* est composée de 5 activités. Une activité, comme expliqué au point *Outils et technologies utilisés* dans la définition d'Android Developer, représente un écran avec lequel l'utilisateur peut interagir. Dans la pratique, une activité est contenue dans une classe java qui lui est propre et associée avec un ou plusieurs layouts. Dans le cas présent, j'ai considéré que mon application est constituée de 5 parties différentes. J'ai donc transcrit chacune d'entre elles en une activité. Celles-ci sont :

- la *main activity*, autrement dit le menu principal. Il s'agit de l'écran d'ouverture de l'application ;
- l'activité de l'exercice *Imagerie* ;
- l'activité de l'exercice *Lecture flash* ;
- l'activité de l'exercice *Anagrammes* ;
- l'activité de l'exercice *Écouter le son*.

Toutes ces activités sont définies en plein écran et en orientation paysage par défaut. Il est impossible de jouer à *Manabu* en mode portrait. J'ai forcé cette orientation car il est possible d'y aligner plus d'éléments les uns à la suite des autres. C'est plus efficace quand les mots sont longs.

L'activité contenant le menu principal ne contient qu'un seul layout. Les autres activités, celles des exercices, en contiennent quant à elles plusieurs. Trois layouts sont communs aux quatre exercices. Il s'agit du layout du menu de démarrage (Fig. 1a), du layout d'affichage des règles du jeu (Fig. 1c) et du layout de fin d'exercice (Fig. 1b). Le layout de démarrage contient des boutons permettant de choisir le niveau, un bouton pour afficher les règles du jeu, et un bouton pour démarrer l'exercice. Le layout des règles contient en haut à droite un bouton permettant d'écouter les règles du jeu, au centre le texte des règles, et en bas un bouton pour revenir au layout des règles et un bouton pour démarrer le jeu. Le layout de fin d'exercice contient quant à lui un texte de félicitation du joueur, ainsi que deux boutons proposant le choix de soit revenir au menu principal, soit rejouer à l'exercice.

8. Ceux-ci ont été pris sur la Nvidia Shield Tablet, qui a une résolution de 1920\*1200 pixels.

9. Building Your First App, <http://developer.android.com/training/basics/firstapp/index.html>, consulté en janvier et février 2015

En outre, concernant les règles du jeu, j'ai rendu leur consultation également possible depuis les exercices déjà démarrés. Pour ce faire, j'ai placé en haut à gauche des layouts propres aux différents exercices un bouton " ? ". Un clic sur celui-ci ouvre un pop-up (Fig. 1d) ré-expliquant les règles du jeu.

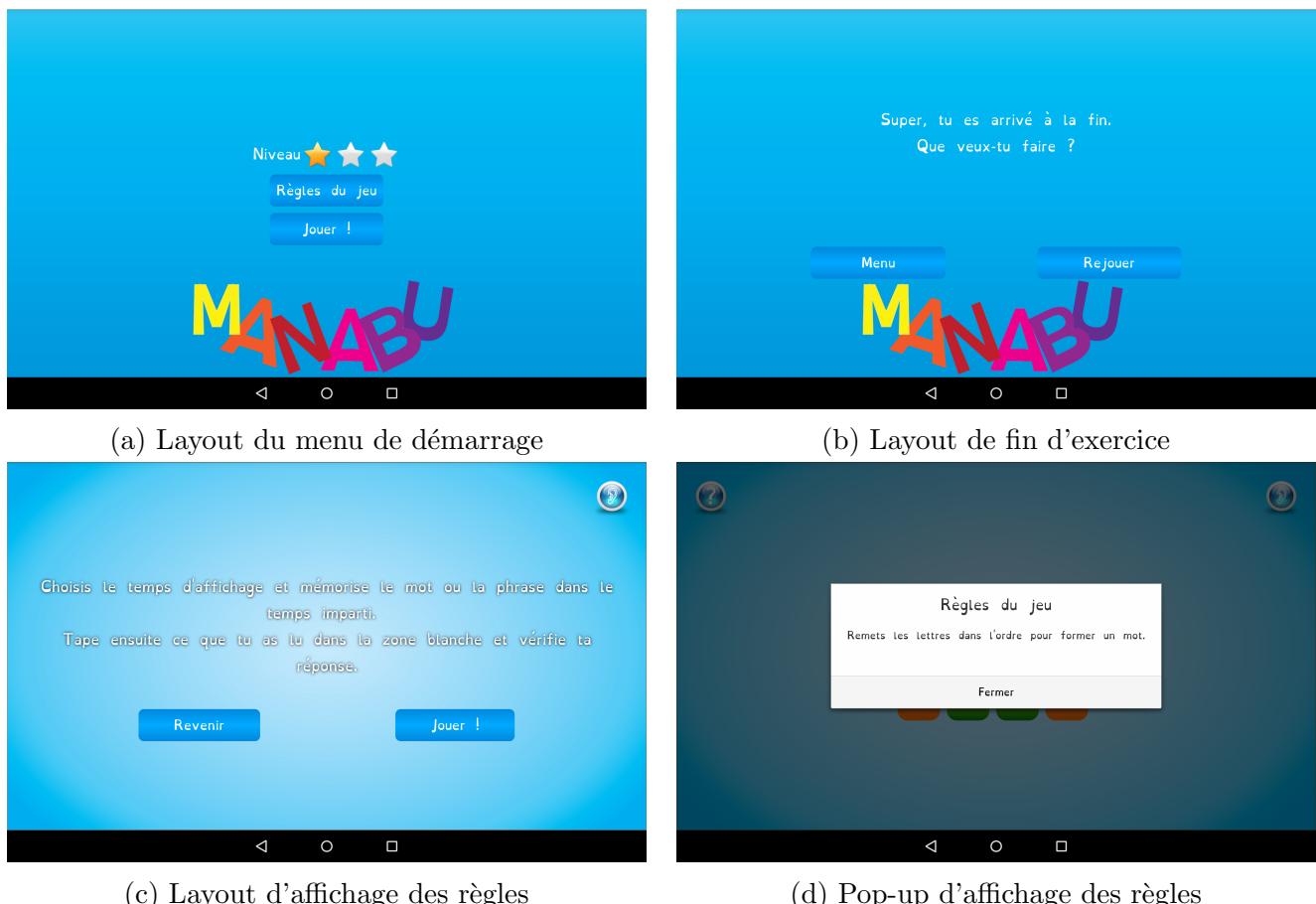


FIGURE 1 – Différents layouts communs et pop-up avec les règles

Par ailleurs, j'ai intégré *OpenDyslexic* en tant que police principale de *Manabu*. Ça n'a pas été la tâche la plus simple. J'ai tout d'abord essayé de modifier la police d'un élément (pensant par la suite pouvoir l'appliquer à toute l'application) via la méthode "classique". C'est-à-dire aller chercher la police dans un dossier et l'appliquer à chaque élément voulu. Cette méthode s'est révélée lourde à mettre en place. C'est en voulant intégrer par la suite la librairie *Calligraphy* de ChrisJenx<sup>10</sup> permettant d'utiliser une police pour toute l'application facilement que j'ai rencontré le problème avec Maven dans Eclipse. J'ai donc, comme dit précédemment, changé d'IDE pour Android Studio. Gradle a quant à lui ajouté sans problème la librairie à la liste des dépendances.

En dernier lieu, j'ai choisi d'exécuter plusieurs fois le même exercice lors du démarrage de celui-ci. Je trouve que devoir à chaque fois ré-appuyer sur le bouton pour recommencer est dérangeant. De ce fait, je lance les exercices par série de 10. Lorsque le premier est réussi, il passe automatiquement au suivant, et ainsi de suite.

10. Disponible gratuitement sur GitHub à l'adresse <https://github.com/chrisjenx/Calligraphy>.

Enfin, je ne présenterai ici que le premier niveau implémenté pour chaque exercice. L'algorithme des niveaux supérieurs étant identique (ou quasi) à celui du niveau facile, je ne juge pas nécessaire d'expliquer leurs mises en place.

## 5.2 Le menu principal

Le menu principal représente l'activité de base de l'application, celle depuis laquelle peuvent être démarrées les autres activités contenant les exercices. Le layout formant ce menu est relativement simple (Fig. 2). Il s'agit de quatre boutons disposés en un rectangle. Chacun des boutons lance un des exercices.

Pour chaque bouton, il m'a fallu implémenter une fonction spécialisée permettant de lancer la nouvelle activité correspondant à l'exercice désiré. Lancer une nouvelle activité se fait à l'aide d'un *intent*. Un intent est une sorte de lien de communication pour le système Android. Celui-ci signale qu'il se passe quelque chose. Dans le cas présent, l'intent signale que l'activité du menu principal va en démarrer une autre.

Enfin, je tiens à mentionner qu'il n'est pas possible de lancer les activités de plusieurs exercices à la fois. En effet, lorsqu'un exercice est lancé depuis le menu, l'activité en cours représentée passe du menu à l'activité de l'exercice, l'activité menu étant alors en attente. Il faut quitter l'exercice, et donc l'activité, pour revenir au menu et pouvoir en démarrer une autre.

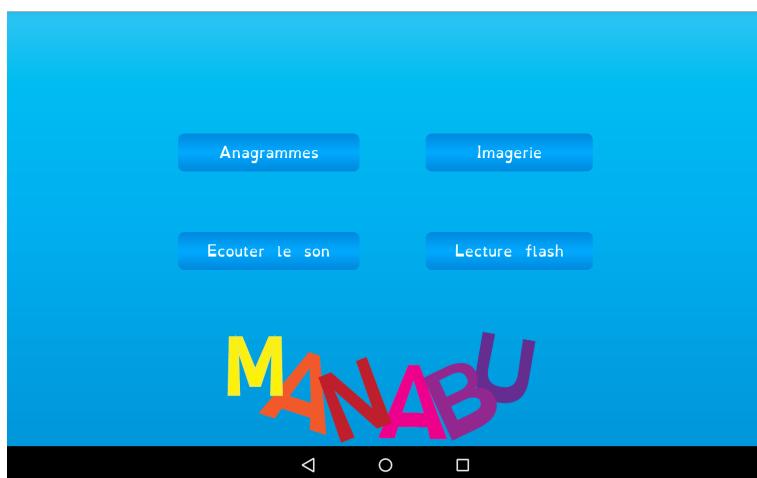


FIGURE 2 – Menu principal

## 5.3 L'exercice *Imagerie*

L'exercice *Imagerie* est le premier que j'ai choisi d'implémenter. Il s'agit de mon exercice favori, car il m'a permis d'exprimer ma créativité au travers des dessins réalisés. Pour le premier niveau, j'ai choisi de me baser sur la version la plus simple des fichiers Freinet : une image affichée avec un mot à mémoriser, et puis, sur base de la même image, retrouver ce mot parmi trois propositions.

Afin de réaliser ce niveau 1, il m'a fallu définir la liste de mots qui allaient être utilisés. J'ai déterminé une liste d'une vingtaine de mots à illustrer. Pour chacun de ceux-ci, j'ai également choisi deux autres mots ressemblant à la bonne réponse pour faire office de fausses réponses. J'ai réalisé les illustrations dans un style de dessin simple, à l'aide du logiciel de dessin vectoriel Adobe

Illustrator.

En ce qui concerne la partie programmation de l'exercice, celui-ci contient deux layouts XML qui lui sont spécifiques. Le premier layout présente l'affichage de l'image et du mot à retenir, ainsi qu'un bouton "Mémorisé" (Fig. 3a). Ce bouton permet de passer au layout suivant. Ce dernier est constitué de la même image que pour le premier layout, ainsi que trois boutons représentant le choix des mots (Fig. 3b).

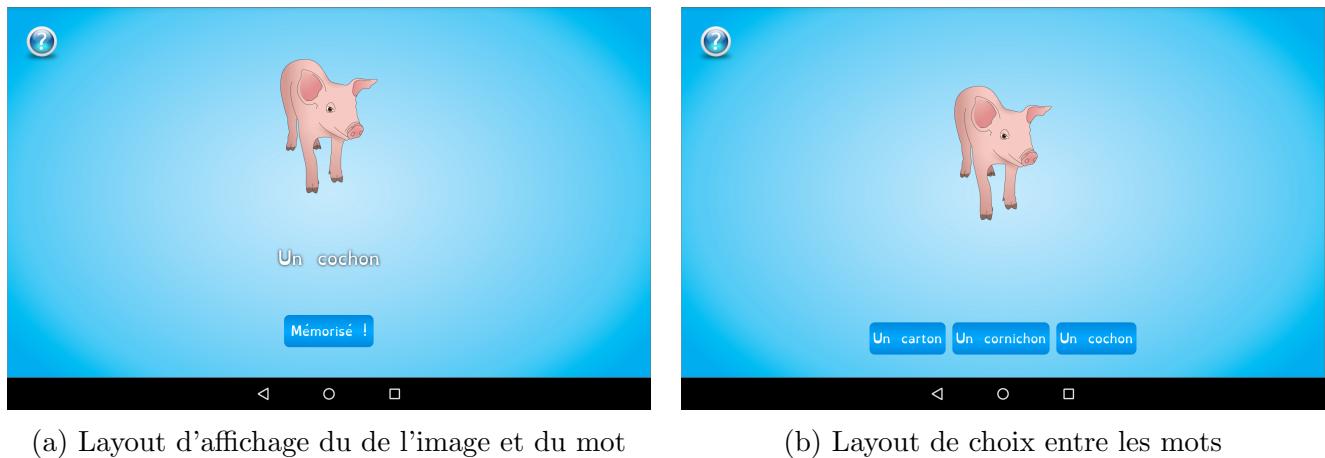


FIGURE 3 – Layouts de l'exercice *Imagerie*

Concernant le fonctionnement de l'exercice en lui-même, celui-ci a été défini dans la classe java correspondant à l'activité. Tout d'abord, il faut savoir qu'en Android, les chaînes de caractères sont stockées dans un fichier *strings.xml*. Les autres types de ressources sont également placées dans des dossiers spécialisés. Pour les images, ces dossiers s'appellent *drawable*. Il existe plusieurs dossiers *drawable*, stockant respectivement les mêmes images, mais correspondant toutefois à une résolution d'écran différente. Pour accéder à l'une des ressources, que ce soit chaîne de caractères, image, ou autre, le nom permettant de l'identifier est requis. Sachant cela, j'ai utilisé une nomenclature permettant de lier facilement les mots et les images leur correspondant. L'image et le mot correct qui lui est attribué portent le même nom. Celui-ci est de type *img\_XX*, où *XX* correspond au numéro identifiant le couple. Les deux réponses incorrectes sont quand à elles nommées dans le fichiers *strings.xml* *img\_XX\_1* et *img\_XX\_2*.

Le principe utilisé pour *Imagerie* est simple. L'image affichée est choisie au hasard parmi les possibilités. Pour ce faire, je génère un nombre random compris entre 0 et le nombre d'images - 1 (20 dans le cas présent). Ce nombre est alors concaténé à une chaîne de caractères afin de former le nom de la ressource que le système doit aller rechercher. Étant donné que l'exercice est constitué de 10 images à la suite, je vérifie également que la ressource n'a pas déjà été utilisée lors de la série (valable pour tous les exercices). À partir de cette chaîne de caractères, l'image et le mot peuvent être chargés et affichés sur le premier layout. Pour le second layout, celui proposant les choix, l'ordre d'affichage des réponses dans les boutons est également défini au hasard. J'applique alors un random entre 0 et 2. L'ordre de sortie des nombres me donne l'ordre d'affichage des réponses. Pour que le système puisse aller rechercher les mauvaises réponses, je concatène la chaîne précédente avec le numéro attribué à la réponse.

Concernant les réponses proposées pour l'exercice, il m'a fallu trouver un système permettant

à l'enfant qui joue de savoir clairement s'il s'est trompé où s'il a réussi. J'ai pour ce faire mis en place un *toast* qui apparaît lorsqu'on clique sur un des boutons de réponse. En Android, un *toast* est une sorte de notification qui se surimprime sur l'écran pendant un temps défini. Le plus souvent, il s'agit d'un message simple. Parfois, on y retrouve une image, pour ce faire, un layout spécifique est créé pour le *toast*. Je voulais mettre en place un toast avec une image et un texte : "V" et "Bien joué !" pour la bonne réponse (Fig. 4a), "X" et "Essaye encore !" pour la mauvaise (Fig. 4b). J'ai donc créé le layout nécessaire, utilisable dans les deux cas, car je passe l'identifiant de l'image et le message en paramètre à l'aide d'une fonction. En fonction du type de réponse, j'assigne au bouton correspondant l'action d'affichage du *toast* "réussi" ou "perdu". Par la suite, j'ai fait de ce *toast* une fonction utilisable dans les différentes *activités* qui composent mon application.

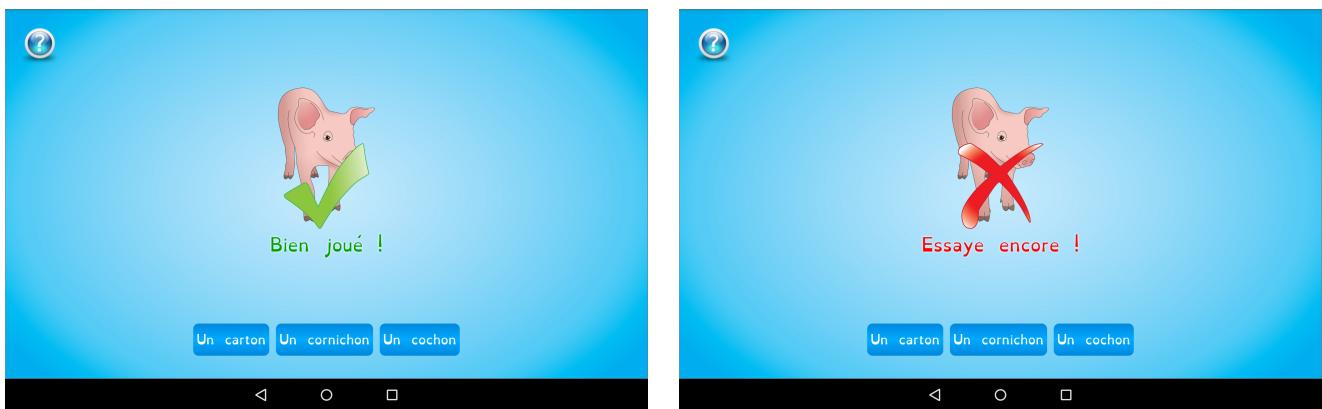


FIGURE 4 – Affichage des *toasts*

L'exercice *Imagerie* n'a pas été trop difficile à mettre en place. Une fois la logique définie, j'ai aisément pu implémenter les fonctions nécessaires. Toutefois, il s'agissait du premier exercice, j'ai donc du appréhender certaines notions. J'ai par exemple appris à lier une image ou une chaîne de caractères à un élément du layout xml à l'aide du code java afin de pouvoir modifier ce dernier.

## 5.4 L'exercice *Lecture flash*

L'exercice *Lecture flash* est le deuxième que j'ai mis en place pour l'application. Comme pour l'exercice précédent, j'ai d'abord commencé par le premier niveau de difficulté. Dans le cas présent, la difficulté entre les niveaux dépendra principalement de la vitesse de lecture. Pour ce premier niveau, j'ai choisi de laisser la possibilité d'afficher le mot pendant 20 secondes.

Les mots utilisés pour cet exercice n'ont pas été choisis au hasard. En effet, selon les conseils de Laurence Henrion et comme prévu dans le cahier des charges, j'utilise le VOB (Vocabulaire Orthographique de Base). Pour rappel, il s'agit d'une liste de vocabulaire que les enfants doivent maîtriser à la fin de chaque cycle. Dans le cadre de l'application, j'utilise le VOB du degré inférieur, qui correspond aux mots devant être connus en fin de deuxième primaire (cf. annexe III). J'ai donc recopié les 480 mots constituant le VOB du cycle inférieur dans le fichier *strings.xml*. Comme précédemment, afin de faciliter le choix des mots de manière aléatoire, les noms sont identiques et différenciés par un nombre. La nomenclature de ceux-ci est *str\_XXX*.

Pour l'interface graphique de cet exercice, il me fallait 3 layouts :

- un premier lors du démarrage, afin de choisir le nombre de secondes d'affichage des mots (Fig. 5a),
- un deuxième pour l'affichage du mot en lui-même. Très simple car il est constitué d'un seul élément (Fig. 5b),
- Un troisième et dernier avec un champ texte éditable et un bouton de vérification, qui est chargé une fois que le temps d'affichage du mot est écoulé (Fig. 5c).

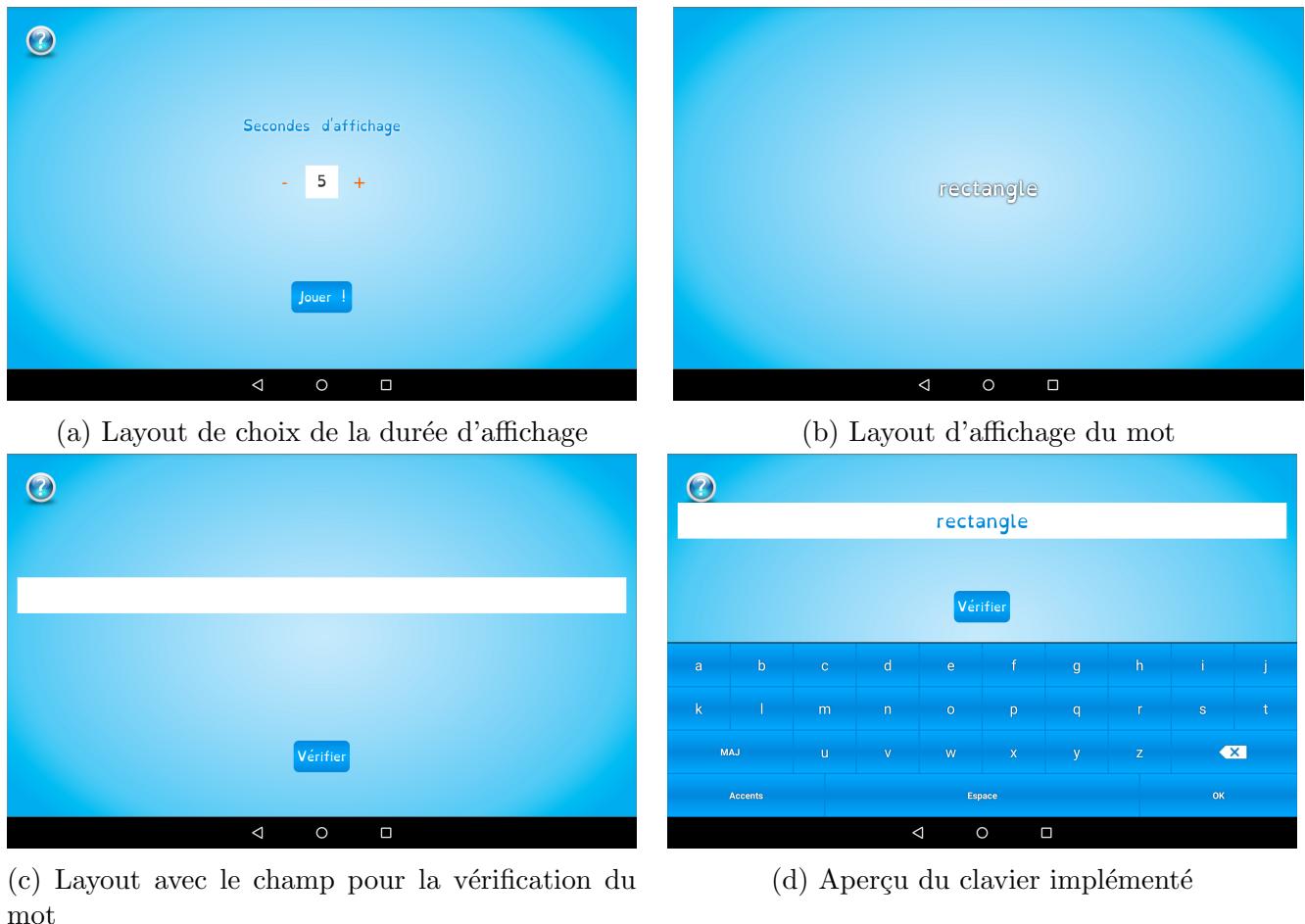


FIGURE 5 – Layouts de l'exercice *Lecture flash* et clavier implémenté

Du point de vue de la programmation java, j'ai créé un *NumberPicker* personnalisé pour définir le temps d'affichage du mot dans le premier layout. Un *NumberPicker* est un élément que l'on peut ajouter tel quel à un layout et qui permet de sélectionner un nombre dans un intervalle défini. Or, l'élément en tant que tel est très peu personnalisable. J'ai dès lors choisi de mettre en place le mien, ce qui est facile à faire. J'ai simplement aligné deux boutons avec un champ texte non éditable entre eux. J'ai assigné au bouton “-” la décrémentation du nombre dans le champ texte, et au “+” l'incrémantation.

La programmation concernant le fonctionnement de l'exercice en lui même est similaire à ce qui a été mis en place pour l'exercice *Imagerie*. J'ai repris le principe de génération d'un nombre au hasard à concaténer par la suite une la chaîne de caractères afin d'obtenir le nom de la ressource. Ainsi, le layout d'affichage récupère la ressource, et affiche le temps pendant le temps récupéré depuis le *NumberPicker* personnalisé. Le layout de vérification de l'écriture du mot à quand à lui

en mémoire le mot précédemment affiché, et le compare avec ce qui est entré dans le champ texte par l'enfant. Ce dernier layout dispose également d'un bouton caché. Celui-ci s'affiche après deux erreurs de la part de l'enfant et lui propose de relire le mot.

Avec le développement expliqué ci-dessus, l'exercice en lui-même est fonctionnel. J'ai implémenté en supplément un clavier propre à *Manabu*. Ceci m'a été demandé par les enfants avec lesquels j'ai eu l'occasion de tester l'application, et notamment cet exercice (cf. point 6.2). Ceux-ci préféraient avoir un clavier pour lequel il ne devaient pas réapprendre l'ordre des lettres, et donc un de type *alphabet* plutôt qu'un *azerty*. Afin de mettre en place mon propre *SoftKeyboard*, je me suis inspirée du tutoriel de Martin Pennings et j'ai téléchargé le code source disponible sur la page du celui-ci<sup>11</sup>. Après avoir essayé de compléter mon code sur la base du tutoriel seul, sans grand succès, j'ai décidé d'intégrer le code source précédemment téléchargé à mon projet.

Enfin, le code source de Martin Pennings étant destiné à mettre en place un *Softkeyboard* hexadécimal, je ne l'ai pas gardé tel quel. J'ai remplacé le seul layout fourni de base par quatre nouveau layouts composants mon clavier :

- un layout avec les 26 lettres de l'alphabet en minuscule (Fig. 5d),
- un layout avec les 26 lettres de l'alphabet en majuscule,
- un layout avec les lettres accentuées en minuscule et la ponctuation,
- un layout avec les lettres accentuées en majuscule et la ponctuation.

De ce fait, j'ai également modifié certaines parties du code précédemment intégré afin de l'adapter aux besoins de *Manabu*.

La mise en place du premier niveau de l'exercice *Lecture flash* en lui-même s'est donc déroulée sans encombre. Comme expliqué ci-dessus, la partie la plus ardue a été l'implémentation du clavier *alphabet* à partir du code de quelqu'un d'autre.

## 5.5 L'exercice *Anagrammes*

*Anagrammes* est le troisième exercice que j'ai implémenté. C'est également durant la mise en place de celui-ci que j'ai pu faire tester l'application par des enfants (cf. point 6).

Il s'agit d'un des deux exercices qui nécessitent le moins de layouts. En effet, je n'ai mis qu'un layout en place. Par défaut, celui-ci contient uniquement un bouton permettant de ré-écouter le mot dont il faut remettre les lettres dans l'ordre. Les autres éléments de ce layout sont ajoutés de manière programmée lors de la génération d'un nouvel anagramme.

Bien que mettre en place le layout ait été un jeu d'enfant, je ne peux pas dire qu'il en soit allé de même pour la programmation. Je m'y suis reprise plusieurs fois afin de parvenir à un résultat correct et le plus proche possible de ce que j'avais imaginé. Les difficultés rencontrées lors du développement de cet exercice seront expliquées par la suite, au point 7 *Difficultés rencontrées*. Voici donc l'explication des actions programmées pour cet exercice.

Tout d'abord, j'ai repris le bout de code de l'exercice *Lecture flash* servant à aller chercher une chaîne de caractères au hasard. En effet, j'utilise également le VOB pour les anagrammes. Pour le niveau 1, je limite le mot à minimum 3 et maximum 5 lettres. Les niveaux suivants augmentent

11. Maarten Pennings, *Android development : Custom keyboard*, <http://www.fampennings.nl/maarten/android/09keyboard/index.htm>, consulté le 18 mai 2015

bien entendu le nombre de lettres, ce qui les rend plus difficiles. Ensuite, je mélange les lettres du mot au hasard pour former l'anagramme<sup>12</sup>.

Ensuite, j'ajoute au layout de nouveaux boutons. Pour être exacte, j'ajoute deux fois plus de boutons qu'il y a de lettres dans le mot. La moitié de ces boutons sont ajoutés sous formes de *cases de validation*, blanches. Ceci signifie que les lettres mélangées viendront une par une se fixer sur l'un de ceux-ci pour reconstituer le mot original. La seconde moitié des boutons est évidemment réservée aux lettres dans le désordre. Une lettre correspond à un bouton. Les voyelles sont affichées sur un fond orange et les consonnes sont sur un fond vert. Ceci permet à l'enfant de différentier une première fois les lettres (Fig. 6a).

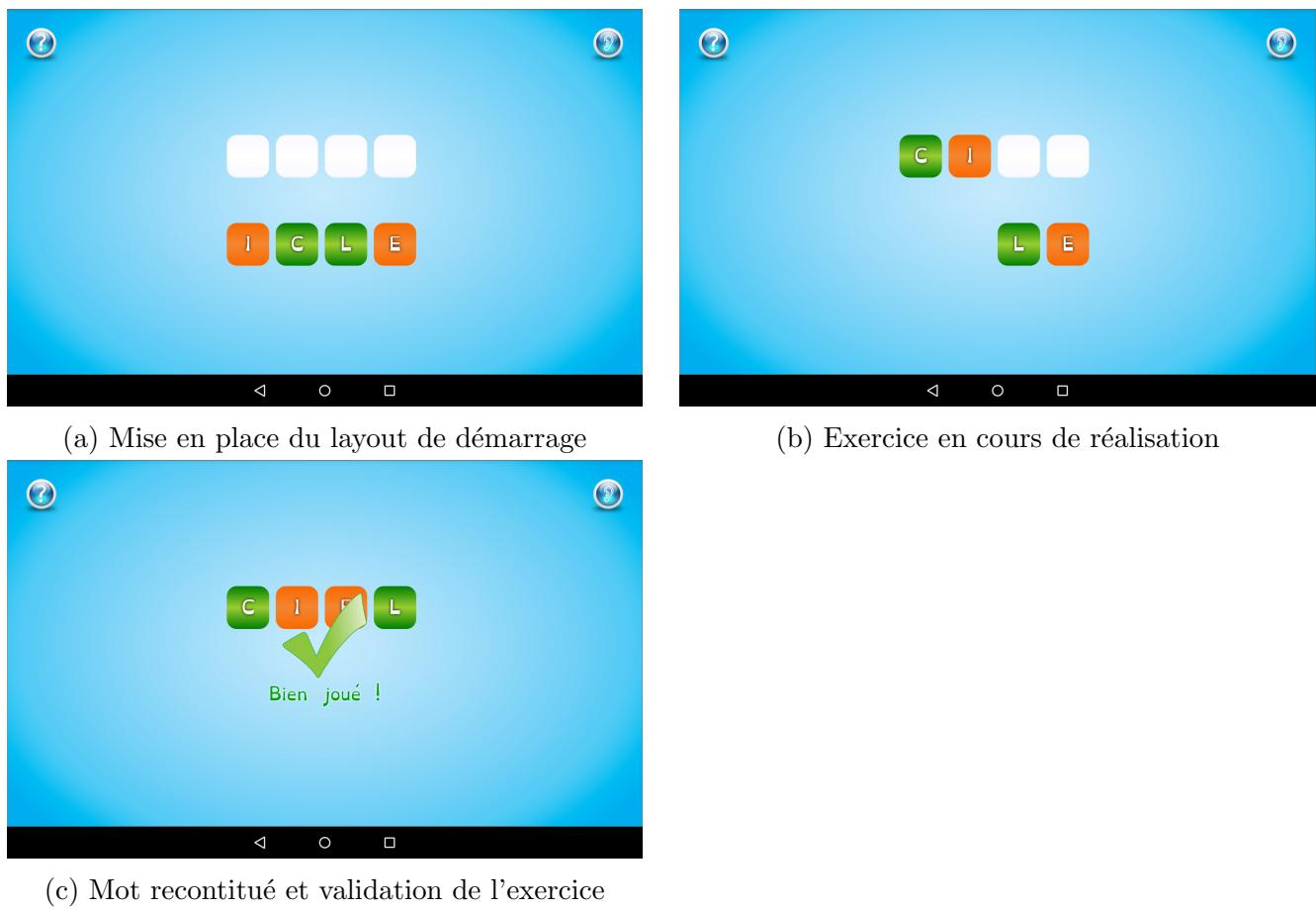


FIGURE 6 – Déroulement de l'exercice *Anagrammes*

La validation de la remise en ordre des lettres est l'étape cruciale de cet exercice. Les boutons contenant les lettres mélangées ont tous un *on touch listener*. Ce dernier leur permet de suivre le mouvement du doigt qui les guidera jusqu'à la bonne *case de validation* (Fig. 6b). Afin de valider la position d'une lettre sur une de ces cases, j'ai mis en place quelques petites astuces. En premier lieu, les *cases de validation* contiennent la lettre correcte qui doit s'y trouver, ainsi qu'un booléen signifiant si cette lettre a déjà été placée ou non. Les boutons des lettres à réorganiser contiennent aussi un booléen pour savoir si les enfants ont trouvé la bonne *case de validation*. Lors du mouvement d'une lettre, si celle-ci se trouve au-dessus de la case qui lui correspond, la

12. Un anagramme se définit un mot formé des mêmes lettres que le mot de base. Cependant, ici, je mélange toutes les lettres sans chercher à ce qu'elles forment un autre mot

position de la lettre deviendra identique à celle de la case et le mouvement ne sera alors plus possible. Le mouvement est empêché grâce aux deux booléens ayant alors pris la valeur *true*. Une fois l'anagramme complété (Fig. 6c), le suivant sera lancé, et ainsi de suite pour 10 anagrammes.

D'autre part, le son est une composante importante de l'exercice. Lors du lancement de chaque nouvel anagramme, l'enregistrement du mot est joué une fois afin d'indiquer à l'enfant le mot à reconstruire. Ceci m'a été demandé par des enfants lors du test (cf. point 6.3). Un bouton leur permet également de ré-écouter le mot à tout instant. De plus, j'ai ajouté un petit extra, qui selon moi donne un peu plus de consistance et de fun à l'exercice. Lorsqu'une lettre est placée correctement sur sa case, un son est fait entendre. Celui-ci valide l'action, et encourage l'enfant qui voit ainsi qu'il progresse.

Enfin, je peux confirmer que cet exercice m'a demandé plus de ressources que les deux précédents. Il a été plus chronophage également. Malgré cela, il a été très intéressant à développer, car il m'a forcé à imaginer des astuces et un algorithme plus complexe que je ne l'aurais pensé afin de parvenir à mes fins.

## 5.6 L'exercice *Écouter le son*

*Écouter le son* est le quatrième et dernier exercice de l'application *Manabu*. Celui-ci, tout comme *Anagrammes*, est composé d'un unique layout.

Ce layout est composé de cinq boutons. Au centre de celui-ci se trouvent trois boutons alignés verticalement. Ces boutons sont prévus pour contenir les différents mots qui serviront de choix pour trouver le son entendu. En haut à gauche un bouton permet d'ouvrir le pop-up des règles et en haut à droite du layout se trouve également le dernier bouton permettant de ré-écouter le son.

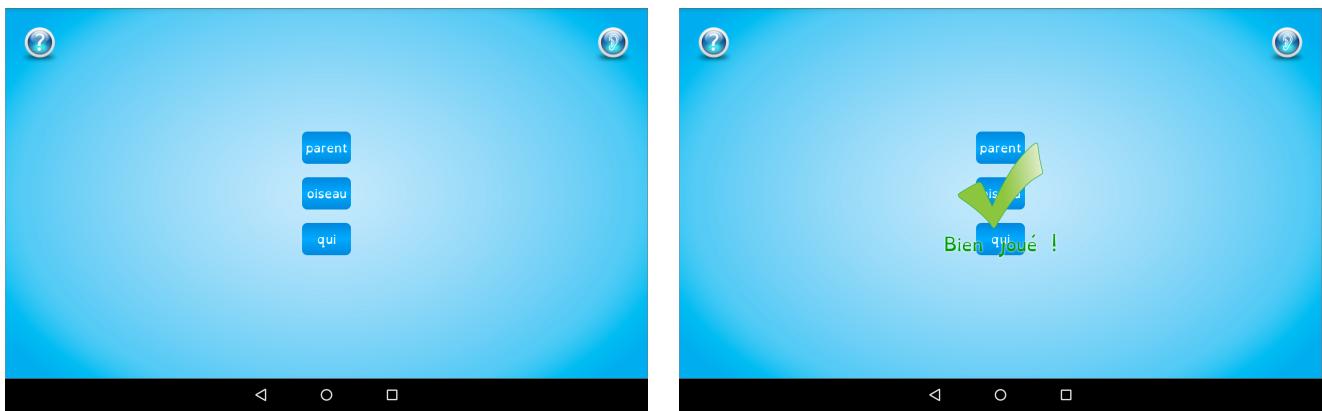
Pour la programmation, j'ai tout d'abord commencé par établir un algorithme sur papier avant de me lancer dans le développement proprement dit. J'ai décidé de repartir sur le principe du nombre aléatoire. Ici, le nombre aléatoire décide du son que l'enfant devra reconnaître. A partir de ce son, je retrouve les mots du VOB le contenant pour en sélectionner un. Cependant, chaque niveau demande à l'enfant de retrouver le mot contenant le son à des endroits différents. En effet, au niveau facile le son se trouve au début, au niveau moyen au milieu, et au niveau difficile n'importe où dans le mot.

Partant de cet énoncé, j'ai décidé d'éviter de me lancer dans des algorithmes compliqués, et de réaliser un fichier texte référentiel par niveau. Chacun des fichiers est une sorte de dictionnaire phonétique. Sur chaque ligne se trouve un code unique de type *XXYYY*. Le *XX* correspond au code que j'ai associé à un son<sup>13</sup>, et le *YYY* au numéro de la chaîne de caractères, utile pour la retrouver dans *strings.xml*. Dans le fichier, les codes sont classés par son (tous les *a*, puis tous les *au*, puis les *b*, et ainsi de suite). Selon le niveau sélectionné, au démarrage de l'exercice, le fichier correspondant à celui-ci est chargé en mémoire dans une *ArrayList* code par code. Un index de cette liste est créé afin de ne pas avoir à parcourir l'entièreté lors de la recherche d'un mot pour un son précis. Cet index est un tableau d'entiers de la taille du nombre de sons possibles. Chaque case de celui-ci contient la position du début d'un type de son dans l'*ArrayList*, la position dans le tableau elle-même équivalant à l'identifiant du son.

13. Par exemple, le son *a* de arbre est associé au code *AA*, le son *k* dans carré, kilo, ou quelque au code *KK*, etc.

Les événements programmés se déroulent dans un ordre défini. Tout d'abord, un nombre *random* est généré, avec pour maximum, le nombre de sons existants, puis stocké dans une variable. Celle-ci fait office d'identifiant et permet à l'application d'aller rechercher le son de la syllabe à jouer. A partir de cette même variable, l'application va rechercher dans l'index les positions minimale et maximale de l'*ArrayList* entre lesquelles sont compris les mots comprenant le son. Le mot qui sera associé au son est également choisi au hasard. Les positions précédemment établies servent de bornes pour générer un second nombre *random* qui sera utilisé en tant que position dans l'*ArrayList*. Ceci permet, grâce à la référence contenue dans la case de l'*ArrayList*, d'aller récupérer la chaîne de caractères qui sera définie comme bonne réponse dans *string.xml*.

Une fois les étapes précédentes réalisées, le programme va chercher deux autres chaînes de caractères au hasard parmi les mots du VOB, tout en s'assurant que celles-ci ne se trouvent pas parmi les mots considérés comme valides. Les trois chaînes différentes ainsi mémorisées, elles sont réparties entre les boutons, encore une fois de manière aléatoire (Fig. 7a). Cela permet de ne pas avoir le bon mot positionné toujours au même endroit. Le caractère correct ou non du clic sur le bouton définissant l'action à effectuer et le *toast* utilisé est également établi à ce moment (Fig. 7b).



(a) Layout principal de l'exercice avec le choix des mots

(b) Le mot correct a été sélectionné

FIGURE 7 – Déroulement de l'exercice *Écouter le son*

*Écouter le son* n'a pas été l'exercice le plus facile à implémenter, mais ce n'était pas le plus difficile non plus. Ce qui m'a pris le plus de temps était la création des dictionnaires phonétiques. En effet, je dois parcourir manuellement le VOB mot par mot et prononcer ceux-ci afin d'analyser les sons qui les composent et ainsi de définir les codes à ajouter au dictionnaire. La mise en place du code, une fois la logique définie, s'est déroulée sans trop d'encombres.

## 6 Validation - test par des enfants

Durant le développement de *Manabu*, j'ai eu l'occasion d'aller faire tester l'application par des enfants de première primaire. Cette occasion m'a été offerte grâce à Mme Van den Schrieck, qui m'a donné les coordonnées d'une dame, Mme Aliette Lochy, réalisant des tests de lecture et de reconnaissance des visages sur des enfants de première primaire durant les vacances de Pâques. J'ai donc pris contact avec Mme Lochy pour lui demander s'il était possible de participer à une des séances organisées avec les enfants afin d'avoir un feedback concernant mon application et pouvoir éventuellement rectifier le tir. Je me suis donc rendue à la faculté de psychologie de Louvain-La-Neuve le jeudi 9 avril, après accord de Mme Lochy.

Quatre enfants de première primaire (deux filles et deux garçons) étaient à ma disposition ce jour-là, afin que l'on parcoure ensemble les exercices déjà mis en place et qu'ils me donnent leur avis sur chacun d'entre eux. Le test a été réalisé sur la tablette Samsung Galaxy Tab 2 10.1, plus conviviale qu'un smartphone. Les exercices déjà développés dans l'application pour être suffisamment fonctionnels ce jour là étaient :

- Imagerie : niveau 1,
- Lecture flash : niveau 1,
- Anagrammes : niveau 1, mais incomplet.

Je ne compte pas détailler ici ce que m'a dit chaque enfant, les commentaires étant très similaires pour chacun d'entre eux. Je vais expliquer ma procédure de test, un exercice à la fois, et détailler par exercice les questions posées aux enfants ainsi que les réponses obtenues.

### 6.1 Test 1 : l'exercice *Imagerie*

Pour cet exercice, j'ai tout d'abord expliqué aux enfants les règles du jeu. Il s'agissait, pour une série de 10 images, de lire le mot associé à chacune d'entre elle. Par image, ils appuyaient ensuite sur le bouton *mémorisé*, et devaient enfin choisir parmi les 3 choix proposés lequel correspondait au mot lu précédemment, l'image étant toujours affichée.

Pour rappel, lors de ce test, le seul niveau implémenté était le niveau 1 : un mot = une image.

Les commentaires que j'ai obtenu de la part des enfants étaient similaires. Ils m'ont tous mentionné que le niveau de cet exercice était trop facile pour leur niveau d'apprentissage (quasi fin de première primaire). Lorsque je leur ai posé la question de ce qui pourrait être d'avantage à leur niveau en demandant si une phrase complète plutôt qu'un mot serait plus difficile, ils m'ont répondu positivement. J'en ai déduit qu'il faut soit des exercices plus compliqués, soit une discrimination. Par discrimination, j'entends : montrer deux images associées chacune à un mot ou une phrase, et pour le choix, montrer une troisième image avec le mot correspondant à sélectionner plutôt que les deux montrés précédemment.

### 6.2 Test 2 : l'exercice *Lecture flash*

Comme précédemment, j'ai commencé par expliquer aux enfants les règles du jeu. Ils devaient commencer par me dire environ de combien de secondes ils avaient besoin pour lire un mot. Pour ce faire, je leur permettais de tester différents timings afin de choisir celui qui leur convenait le mieux. Une fois le timing choisi, je leur expliquais le principe. Pour une série de 10 mots, ils devaient lire le mot dans le temps imparti, et ensuite ré-écrire celui-ci à l'aide du clavier de la

tablette, accent et caractères spéciaux compris.

Le temps de lecture était variable selon les enfants. Les filles étaient plus rapides que les garçons pour lire un mot de taille moyenne (plus de 4 ou 5 lettres). J'avais donc, pour le niveau 1, une durée de lecture variant entre 10 et 20 secondes selon les enfants.

Cet exercice de lecture flash était considéré comme plus difficile par les enfants. En effet, ils avaient besoin de plus de concentration, car il leur fallait retenir le mot pour pouvoir le ré-écrire. Un des garçons m'a cependant dit qu'il trouvait cet exercice facile. Pourtant, au vu de ses résultats, j'ai constaté qu'il préférait écrire le mot plutôt que de le lire, ce qui lui posait quelques problèmes.

Enfin, la ré-écriture du mot posait problème. Non pas que les enfants n'avaient pas retenu le mot, mais le clavier par défaut en *azerty* les perturbait. Ils rencontraient des difficultés à situer les lettres sur le clavier, et confondaient certaines d'entre elles (*b* et *d*, *q* et *p*). Aussi m'ont-ils tous mentionné qu'il préféreraient disposer d'un clavier de type *alphabet*, avec les accents et caractères type -, ', ... à disposition.

### 6.3 Test 3 : l'exercice *Anagrammes*

Comme mentionné précédemment, cet exercice n'était pas complet lors de son évaluation auprès des enfants. J'avais implémenté l'algorithme de création de l'anagramme, mais pas sa validation une fois les lettres remises dans l'ordre. Par ailleurs, l'affichage à ce moment présentait une zone où le mot était écrit, à côté des lettres mélangées.

J'ai tout d'abord demandé aux enfants de remettre les lettres dans l'ordre, sur base du mot qu'ils lisaienr juste à côté. Ceci était bien entendu trop facile pour eux. Ils n'avaient qu'à regarder les lettres du mot, et à copier.

Ensuite, j'ai caché le mot affiché en entier pour ne laisser que les lettres mélangées, et j'ai demandé de remettre les lettres dans l'ordre pour former un mot. Les mots étaient constitués au maximum de 5 lettres. Sans aucune aide extérieure, il était quasi impossible pour les enfants de remettre les lettres en ordre. J'ai alors essayé le même type d'exercice, mais en prononçant le mot. Ceci à permis aux enfants de réussir l'exercice, à l'exception de quelques mots.

Les mots posant problème aux enfants étaient ceux qui comprenaient plusieurs phonèmes similaires, mais ne s'écrivent pas de la même manière. Ainsi, pour le mot "aimer" par exemple, un des enfants cherchait deux fois la lettre *e* pour écrire le son "é" entendu au début et à la fin du mot. J'ai dû lui donner un indice en lui expliquant l'association des voyelles pour former un son afin qu'il trouve l'orthographe correcte.

Ces essais concernant les anagrammes m'ont permis de clarifier la méthode à employer pour mettre en place l'exercice. J'en ai déduit qu'il était plus simple pour l'enfant d'associer le mot et l'ordre des lettres à partir du son du mot prononcé. J'ai donc choisi de faire entendre le mot sous forme de son lors de la génération de l'anagramme et de laisser la possibilité à l'enfant de rejouer celui-ci afin de l'aider.

## 6.4 Points relevés

Il ressort des questions posées et des avis obtenus de la part des enfants que quelques points principaux peuvent être relevés concernant les exercices :

- Fin de première primaire, les enfants ont un niveau suffisant pour savoir lire plus ou moins aisément un mot seul. A ce stade, il est plus intéressant de s'orienter vers les phrases ou la discrimination des mots. Néanmoins, les mots seuls restent essentiels au début de l'apprentissage.
- L'implémentation d'un clavier spécifique à l'application est nécessaire. Il doit être sous forme *alphabet* et non *azerty*, et contenir les accents et caractères fréquemment rencontrés tels que -, ', etc.
- Les problèmes rencontrés par certains dans la discrimination des lettres me confortent dans l'idée que l'utilisation de la police spécifique, *OpenDyslexic*, s'impose.
- Le son, dans le cas de l'exercice avec les anagrammes, est une composante nécessaire à réussite de celui-ci.

D'autre part, j'ai posé quelques questions aux enfants concernant l'application dans sa globalité. Je leur ai demandé si celle-ci leur plaisait ou non. Ils ont tous répondu par l'affirmative, une des filles précisant qu'elle trouvait plus chouette d'apprendre sur une tablette.  
Enfin, je leur ai demandé s'ils aimeraient avoir de la musique en plus, comme dans un jeu, ou s'il pensaient que ça les distrairait. Tous m'ont dit qu'il préféraient ne pas entendre de musique, car il leur est plus facile de se concentrer dans le silence.

## 7 Difficultés rencontrées

Créer une application Android de A à Z pour la première fois ne se déroule bien évidemment pas sans rencontrer quelques difficultés. Que ce soit pour la programmation ou pour le reste du travail, rien ne se réalise jamais sans rencontrer quelques obstacles. Ce point du rapport est dédié aux petits problèmes et autres difficultés auxquels j'ai été confrontés tout au long de mon travail.

La difficulté principale, lorsqu'un projet de l'ampleur de ce travail de fin d'études est réalisé, c'est de trouver la motivation. En effet, bien que le sujet choisi m'intéresse, ce n'était pas facile tous les jours. Durant les premiers mois de sa réalisation, avec le stage en parallèle, je rentrais souvent fatiguée. Dans ces cas là, il faut trouver l'énergie de se mettre à travailler en soirée et réussir à se motiver. Les autres jours, weekends ou après le stage, il m'arrivait d'avoir des jours "sans", comme tout le monde. Enfin, parfois ma démotivation était due à un obstacle rencontré dans mon travail, comme je vais en décrire plusieurs par la suite. Néanmoins, lorsqu'une baisse de motivation se faisait sentir, je pensais au diplôme à la clé, et je me disais que cela vaut donc vraiment la peine de travailler.

Le premier problème technique auquel j'ai dû faire face a été le bug d'Eclipse. Comme je l'ai déjà expliqué précédemment, ce bug est apparu lors de l'installation du plugin Maven pour Eclipse. Alors que celle-ci s'était déroulée correctement, j'ai dû redémarrer Eclipse. À partir de là, je n'ai plus su travailler avec cet IDE, car il refusait de fonctionner après seulement quelques minutes. J'ai essayé de résoudre ce problème, sans succès. Je me suis donc résolue à transférer mon projet sous Android Studio et à abandonner Eclipse.

Une autre difficulté rencontrée concerne l'enregistrement des sons utilisés dans l'application. Ce n'est pas l'enregistrement en lui même qui m'a posé problème, bien qu'il m'ait pris pas mal de temps. La difficulté était dans la prononciation des mots. Lors de l'enregistrement du VOB, il fallait que je fasse attention à prononcer les mots le mieux possible, sans pour autant tomber dans l'excès. J'ai du m'y reprendre plusieurs fois pour certains d'entre eux.

Ensuite, j'ai rencontré quelques soucis lorsque j'ai implémenté le clavier *alphabet* pour l'exercice de lecture flash. Comme expliqué au point 5.4, j'ai utilisé le code source disponible avec le tutoriel de Martin Pennings. Cependant, avant de remarquer que le code était disponible et ainsi me simplifier la vie, j'ai essayé d'implémenter le clavier sur base du tutoriel uniquement. Pendant quelques heures, je me suis appliquée, pour au final me retrouver avec du code qui fonctionnait aléatoirement. Effectivement, une fois que j'ai vu le code original, les points que j'avais eu du mal à comprendre se sont éclaircis. Dès lors, j'ai opté pour la solution de facilité : supprimer mon code défectueux, ajouter le code original à l'application, et modifier ce dernier pour répondre à mes besoins.

L'exercice des anagrammes m'a aussi posé des problèmes. Celui qui m'a accaparée le plus longtemps est la mise en place des boutons de validation et des lettres mélangées. Il faut savoir que lorsque le *listener* permettant le mouvement des lettres est implémenté, celui-ci ne fonctionne que dans l'élément layout dans lequel il est défini<sup>14</sup>. Par exemple, il est possible de mettre en place un *LinearLayout* (type structure de layout, un peu comme une grosse boîte) global vertical avec encore deux *LinearLayout* horizontaux (petites boîtes à l'intérieur de la grosse), tout ça pour obtenir un seul et même layout (vue) à afficher sur l'écran. Si les lettres se trouvent dans un des

14. Un layout (= une vue) peut être composé de plusieurs éléments layout s'imbriquant les uns dans les autres

*LinearLayout* interne, elles ne pourront pas se déplacer au dessus du second. Pour éviter cette contrainte, j'ai donc fait le choix d'utiliser une seule "grosse boîte". Ceci rendait impossible l'utilisation du *LinearLayout*, compte tenu de la disposition des éléments de l'interface. J'ai tout d'abord opté pour un *GridLayout*, qui permet d'organiser les éléments sous forme de grille. Cependant, celui-ci ne me permettait pas de centrer le contenu de l'interface comme je le désirais, je suis donc passée au *RelativeLayout*. Celui-ci permet de décrire la position des éléments de manière relative par rapport au parent et aux autres éléments de la vue. Ce type d'élément layout m'a pris un peu de temps à prendre en main, mais répond finalement à mes attentes.

L'autre difficulté des anagrammes a été la mise en place de l'algorithme de validation de la position des lettres. En premier lieu, il m'a fallu comprendre comment le *listener* permettant de bouger les lettres fonctionnait. A partir de cela, j'ai rusé pour réussir à valider les positions des boutons, le fait qu'ils soient au dessus de la bonne *case de validation*, etc. Malgré mes idées d'algorithme réfléchies auparavant, et inscrites sur papier à côté de moi, j'ai dû faire plusieurs essais avant d'y parvenir.

Enfin, je voudrais mentionner que respecter le planning n'a pas été aussi aisément que je l'aurais imaginé. Je n'ai jamais été une grande fanatique des plannings. Je trouve qu'il est d'autant plus dur de se donner des objectifs à atteindre et de les quantifier dans le temps que l'échéance est éloignée. Ajoutons à cela des baisses de motivation, des imprévus, des soucis, etc, et le planning en a quelque peu souffert. Malgré cela, j'ai rattrapé autant que possible le retard pris, et j'ai fait de mon mieux afin d'atteindre les objectifs que je me suis fixés.

## 8 Pistes d'amélioration

Avant d'arriver à la fin de ce rapport, j'aimerais parler des améliorations qu'il serait possible d'ajouter à *Manabu*. Ce rapport étant rendu deux semaines avant la défense, je commencerai par expliquer ce que j'espère apporter comme complément d'ici la date fatidique du 19 juin. J'explorerai ensuite diverses pistes d'amélioration qui pourraient rendre mon application encore plus attrayante et efficace.

J'aimerais présenter une application complète telle que celle que j'avais imaginée au départ. Quatre exercices, et pour chacun d'entre eux, trois niveaux de difficulté. A l'heure où j'écris ces lignes, les quatre exercices sont tous fonctionnels pour le niveau 1 : facile. J'aimerais donc implémenter les niveaux moyen et difficile pour tous. C'est déjà le cas pour *Anagrammes*, et je suis sûre de pouvoir y arriver au moins pour deux des autres exercices : *Lecture Flash* et *Écouter le son*.

Par ailleurs, je ne suis pas entièrement satisfaite de l'aspect graphique de l'application. Je compte retravailler certains layouts afin qu'ils soient plus agréables visuellement. J'aimerais, si j'ai le temps, ajouter la mascotte dont j'avais parlé dans le cahier des charges. Je trouve que celle-ci apporterait un intérêt ludique supplémentaire. Personnellement, je pense que les enfants peuvent s'attacher à une mascotte.

Concernant les pistes d'amélioration supplémentaires, je pense que rien n'est jamais parfait. Je pourrais trouver énormément de choses à rajouter, de détails à améliorer, mais ce n'est pas le but ici, et ce ne sera pas le cas. Je vais simplement proposer quelques idées qui apporteraient véritablement quelque chose à *Manabu*.

La premier ajout serait d'enregistrer les scores des enfants pour les différents exercices. L'enregistrement des scores est selon moi source de controverses. Certains pensent que cela permet à l'enfant de mesurer sa progression et qu'ils s'agit de données intéressantes. Pour ma part, j'ai choisi de ne pas implémenter de scores car j'estime qu'il s'agit de quelque chose qui peut stresser l'enfant. En effet, si deux enfants comparent leurs scores respectifs, si l'un est nettement moins avancé, il peut se sentir défavorisé. Or, chaque enfant avance à son rythme. Je conçois qu'enregistrer les scores puisse être intéressant, cependant, je ne vois pas l'utilité de mettre une pression supplémentaire sur les épaules de l'enfant. Je proposerais plutôt un système d'*achievements*, par exemple :

- la série d'anagrammes résolue en moins de X minutes,
- achèvement du niveau facile, ce pour chaque exercice et chaque niveau,
- niveau facile de la lecture flash réussi avec le temps de lecture minimal,
- etc.

Une deuxième amélioration possible serait la personnalisation des différents exercices en fonction des centres d'intérêt de l'enfant. Par exemple, si un enfant adore le sujet de l'espace et que l'exercice *Imagerie* est son favori, on peut imaginer acheter une extension pour cette exercice. Celle-ci contiendrait des images et de nouveaux mots en rapport avec le thème de l'espace à apprendre à l'enfant. Ce dernier serait alors plus prompt à utiliser l'application, celle-ci étant en rapport avec ses centres d'intérêt. On peut bien entendu imaginer cette personnalisation pour chacun des exercices présents dans *Manabu*.

Enfin, la troisième amélioration possible découle d'une idée énoncée de la rencontre avec Lau-

rence Henrion et concerne l'exercice *Lecture flash*. Il s'agit simplement de pouvoir ajouter manuellement un mot parmi ceux déjà existants dans l'exercice. Ceci permettrait à un adulte responsable (parent ou autre) d'ajouter du nouveau vocabulaire. Ce dernier serait bien évidemment mémorisé pour les futures utilisations de *Manabu*. Il s'agit d'une alternative au téléchargement d'extensions mentionné dans le paragraphe précédent, et qui augmente les possibilités de personnalisation.

## 9 Conclusion

A la fin de ce rapport, je peux affirmer que je suis parvenue à réaliser une application Android permettant d'aider les enfants dans leur apprentissage de la lecture en dehors de l'école. Celle-ci est destinée aux enfants débutant dans la lecture, avec un niveau de première primaire ou même inférieur. *Manabu* est une application conçue pour fonctionner principalement sur tablette, mais pouvant également être exécutée sur smartphone. Elle est à la fois conviviale, amusante et éducative. Les quatre exercices qui la composent sont inspirés de conseils venant de professionnelles de l'apprentissage de la lecture. Ils mettent en pratique deux méthodes d'apprentissage qui ont prouvé leur efficacité : la méthode syllabique et la méthode naturelle.

J'ai démontré tout au long de ce rapport le travail effectué. J'ai commencé par expliquer l'analyse que j'ai faite de la problématique, et les recherches en découlant. J'ai établi un cahier des charges et présenté l'application en conséquence. J'ai ensuite énuméré les outils que j'ai utilisés tout au long de ce travail de fin d'études. Par après, j'ai explicité le déroulement du développement de l'application pour chacun des exercices et menus, ainsi que les résultats obtenus après un test de l'application par des enfants. Enfin, j'ai établi la liste des difficultés que j'ai rencontrées durant ce projet, et j'ai énuméré quelques améliorations qu'il est possible d'apporter à l'application.

Enfin, je suis fière d'avoir réalisé *Manabu*. C'est toujours un sentiment positif de parvenir à réaliser un projet de cette ampleur. J'ai rencontré des difficultés, tout ne s'est pas réalisé comme je l'avais prévu d'un claquement de doigt, mais ce sont les aléas de la programmation. J'ai également découvert le développement Android qui, bien qu'il utilise le langage Java, fonctionne par certains aspects différemment. J'ai appris à utiliser différents éléments composant une application et à inclure une librairie externe. En outre, le test de *Manabu* par des enfants m'a confirmé que j'ai fait le bon choix de sujet de TFE. Ils ont apprécié les exercices et se sont amusés, ce qui pour moi est une grande réussite.

## 10 Bibliographie

### Sites web

1. Android : button pressed and to follow gesture, <http://stackoverflow.com/questions/20444211/android-button-pressed-and-to-follow-gesture>, consulté en avril 2015.
2. Android color xml resource file, <http://stackoverflow.com/questions/3769762/android-color-xml-resource-file>, consulté en mars 2015.
3. Android Developer, *Dialog*, <http://developer.android.com/guide/topics/ui/dialogs.html>, consulté en mai 2015.
4. Android Developer, *Toast*, <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>, consulté en février 2015.
5. Android : How do I get string from resources using its name ?, <http://stackoverflow.com/questions/7493287/android-how-do-i-get-string-from-resources-using-its-name>, consulté en février 2015.
6. API Android, *Class Index*<sup>15</sup>, <http://developer.android.com/reference/classes.html>, consulté en mai 2015.
7. Benbourahla Nazim, *Introduction aux styles sous Android*, <http://www.tutos-android.com/introduction-styles-android>, consulté en mars 2015.
8. Emmanuel Crombez, *ABC Applications*, <http://abc-applications.com/>, consulté en janvier 2015.
9. Fichiers de lecture, <http://www.icem-pedagogie-freinet.org/node/3013>, consulté en janvier 2015.
10. Force an android activity to always use landscape mode, <http://stackoverflow.com/questions/2150287/force-an-android-activity-to-always-use-landscape-mode>, consulté en février 2015.
11. How to Make an Activity Fullscreen, <http://www.androidsnippets.com/how-to-make-an-activity-fullscreen>, consulté en février 2015.
12. How to set delay in android ?, <http://stackoverflow.com/questions/15874117/how-to-set-delay-in-android>, consulté en février 2015.
13. How to use custom fonts in Android apps (and not get fat), <http://blog.goyello.com/2014/08/01/how-to-use-custom-fonts-in-android-apps-and-not-get-fat-3/>, consulté en février 2015.
14. Martin Pennings, *Android development : Custom keyboard*, <http://www.fampennings.nl/maarten/android/09keyboard/index.htm>, consulté en mai 2015.
15. OpenDyslexic, <https://gumroad.com/l/OpenDyslexic>, consulté en mai 2015.
16. Universal Soundbank, *Bruitages, sons et loops gratuits*, <http://www.universal-soundbank.com/>, consulté en mai 2015.
17. Varun Arora, *How to customize different buttons in Android*, <http://mrbool.com/how-to-customize-different-buttons-in-android/27747>, consulté en mars 2015.
18. Wikipédia, *Apprentissage de la lecture*, [http://fr.wikipedia.org/wiki/Apprentissage\\_de\\_la\\_lecture](http://fr.wikipedia.org/wiki/Apprentissage_de_la_lecture), consulté en janvier 2015.

---

15. Lieu de départ pour mes recherches sur divers éléments dans l'API

19. Wikipédia, *Méthode globale*, [http://fr.wikipedia.org/wiki/Mthode\\_globale](http://fr.wikipedia.org/wiki/Mthode_globale), consulté en janvier 2015.
20. Wikipédia, *Méthode syllabique*, [http://fr.wikipedia.org/wiki/Mthode\\_syllabique](http://fr.wikipedia.org/wiki/Mthode_syllabique), consulté en janvier 2015.
21. Wikipédia, *Pédagogie Freinet*, [http://fr.wikipedia.org/wiki/Pdagogie\\_Freinet](http://fr.wikipedia.org/wiki/Pdagogie_Freinet), consulté en janvier 2015.
22. Wikipédia, *Pédagogie Montessori*, [http://fr.wikipedia.org/wiki/Pdagogie\\_Montessori](http://fr.wikipedia.org/wiki/Pdagogie_Montessori), consulté en janvier 2015.

## Autres

1. Notes personnelles prises pendant les rencontres avec Odile Paveau et Laurence Henrion.
2. Vocabulaire orthographique de base : degré inférieur, téléchargé depuis <http://www.enseignons.be/upload/fondamental/francais/190607071618vob-degre-inferieur.pdf> le 26 janvier 2015.

## Annexe I : Compte-rendu des rencontres avec une institutrice et une logopède

### Rencontre avec une institutrice : Odile Paveau

J'ai rencontré Odile Paveau le 16 décembre 2014 dans la classe de primaire où elle enseigne, à l'école des Bruyères de Louvain-La-Neuve.

Elle m'a expliqué ce qu'elle, en temps qu'institutrice primaire, utilise comme méthodes pour enseigner la lecture. Elle m'a expliqué que la lecture est une discrimination à la fois visuelle et auditive. Elle m'a de même montré les outils utilisés à l'école afin de faciliter l'enseignement et l'apprentissage chez les enfants, ainsi que "l'ordre" dans lequel ceux-ci sont utilisés. Par exemple, un des premiers acquis de l'enfant sera toujours de savoir lire et écrire son prénom, puis celui de ses camarades de classe.

Elle m'a tout d'abord parlé des différentes méthodes d'apprentissage qui peuvent être utilisées. Parmi celles-ci, on retrouve :

- la méthode globale,
- la méthode syllabique,
- la méthode naturelle.

La méthode globale consiste à apprendre à lire à partir du mot en entier. Le but de cette méthode est "*de faire acquérir à l'élève une stratégie de déchiffrage des mots, voire des phrases, en tant qu'image visuelle indivisible*"<sup>16</sup>. En pratique, cela signifie que l'enfant, la première fois qu'il rencontre le mot, est invité à le deviner. Celui-ci mémorisera alors le mot en le rencontrant plusieurs fois dans des contextes différents (chansons, petites histoires, poèmes, ...). Le mot est dès lors associé à une idée, ce qui permet de qualifier cette méthode d'*idéovisuelle*.

La méthode syllabique, par opposition à la méthode globale, part des sons que forment les lettres et les syllabes afin de construire le mot. Celle-ci relie la phonétique des lettres avec l'alphabet afin de construire tout d'abord les syllabes, puis d'assembler ces dernières pour créer les mots. Cette méthode se base sur le décryptage progressif des phrases lues.

La méthode naturelle s'inspire quant à elle de la pédagogie Freinet. Cette dernière, mise au point par Célestin Freinet au siècle dernier, est fondée sur l'expression de la créativité des enfants. Il s'agit de donner à l'enfant un projet, qui lui sera utile dans son apprentissage, et qui prend en compte ses centres d'intérêts et le potentiel créatif et associatif de celui-ci. Du point de vue de l'apprentissage de la lecture, cette méthode implique de partir du sens des mots, afin de donner un sens à ce qui est appris. La collaboration de tout le groupe est nécessaire, et l'essai-erreur est appliqué. La pédagogie Freinet a d'abord été associée à la méthode globale. Cependant, son procédé différent dans la façon dont les mots sont appris, l'apprentissage de la lecture par cette pédagogie se nomme désormais méthode naturelle.

Odile m'a ensuite fait part des techniques utilisées par elle-même et ses collègues aux Bruyères. La lecture s'apprend tout d'abord sur des caractères imprimés en majuscules, puis en minuscules, et enfin avec la police de type écriture manuelle, appelée aussi *cursive*. Ils utilisent principalement la méthode syllabique et la méthode naturelle. Comme mentionné, ils discriminent à la fois les

16. Citation tirée de l'article *Méthode globale*, [http://fr.wikipedia.org/wiki/M%C3%A9thode\\_globale](http://fr.wikipedia.org/wiki/M%C3%A9thode_globale)

mots de manière visuelle et auditive. Toujours dans l'exemple du premier acquis que représente le prénom, en début de première primaire, les prénoms des enfants sont classés sur un tableau selon le son par lequel ils commencent. Ensuite, ils observent comment s'écrit le son du début du prénom, ce qui leur apprend déjà de premières syllabes. Par exemple : Amélie commence par le son "A", qui s'écrit de la même manière, tandis qu'Hugo commence par le son "U" mais s'écrit "HU".

Durant la suite de l'apprentissage, le rapport au son reste très important pour l'enfant. Voici quelques exercices et outils qui sont proposés à l'école :

- Avec des lettres d'imprimerie disposées dans n'importe quel sens<sup>17</sup>, l'institutrice donne un son, et il faut retrouver la lettre qui produit ce son.
- Dans un petit texte, qui peut être écrit sur base d'une idée de l'enfant (par exemple "*Marie aime la danse et les poupées*"), retrouver les sons déjà connus et les souligner en couleur. La couleur aide ici et à mémoriser, et à différencier ;
- Dans un texte, trouver le son qui revient le plus souvent. Cet exercice est une variante du précédent ;
- A partir d'un son, trouver des mots qui commencent par celui-ci, par exemple sous forme d'images pour ensuite voir comment s'écrit le mot. Au niveau supérieur, le son peut se trouver au milieu ou en fin de mot ;
- Comme outil, créer un dictionnaire *référent*. Celui-ci reprend, pour chaque son (également les composés type *au, ou, en, ...*) un dessin représentatif du son (par exemple une chouette pour le son *ch*) ainsi qu'une liste des mots commençant par ce son.

Parmi les exemples ci-dessus, on constate que la méthode syllabique n'est pas la seule appliquée. En effet, on retrouve la méthode naturelle dans l'association avec les couleurs et les images. En effet, il est très important de faire sens pour l'enfant et de susciter son intérêt pour faciliter l'apprentissage. Ainsi, le faire travailler sur des sujets qui le concernent tels que son âge, ce qu'il aime, faire des phrases rigolotes, etc. l'aidera à apprendre.

La méthode globale est elle aussi utilisée aux Bruyères, mais peu comparativement aux deux autres. Celle-ci est considérée comme plus lourde et moins efficace. Néanmoins, on retrouve son application dans des exercices de repérage d'un mot en particulier dans un texte, de chasse aux mots dans la classe, et quelques autres.

Lors des évaluations concernant l'apprentissage de la classe, la méthode naturelle est la plus souvent utilisée, car c'est celle qui fait le plus appel à l'imagination de l'enfant. Comme outil, les instituteurs utilisent notamment les fichiers Freinet. Crées à partir de la pédagogie Freinet, ces fichiers sont composés d'images associées à des mots de différents niveaux. L'idée est d'associer un (ou plusieurs) mot(s) lu(s) à une (ou plusieurs) image(s), et de vérifier les capacités de lecture de l'enfant en lui proposant soit le même mot à retrouver dans une liste sur base d'une image semblable à la précédente, soit un mot différent sur base d'une image différente des deux précédentes, soit encore l'association d'un nouveau mot en complément avec celui lu précédemment, sous forme d'une phrase. Ces fichiers existent également pour d'autres matières, comme les mathématiques.

Enfin, à l'école des Bruyères, l'enfant à toujours accès à une boîte à outils pour s'aider en cas de difficultés. Celle-ci est composée du dictionnaire référent, mais aussi de panneaux et d'affiches

17. Le fait de mettre les lettres droites, à l'envers, ou de côté permet d'entraîner l'enfant à différencier les caractères.

se trouvant un peu partout dans la classe. Ceux-ci rappellent les couleurs identifiant les sons, les images associées, les dessins des enfants représentant des mots, des symboles associés aux lettres (par exemple une montagne double pour le M), etc. Mis à part ces outils, il est important de faire attention à l'ancre gauche - droite correspondant au sens de la lecture. Certains enfants ont du mal avec cette notion.

## Rencontre avec une logopède : Laurence Henrion

J'ai rencontré Laurence Henrion le 19 janvier 2015. Elle a accepté de me recevoir dans le cabinet où elle exerce son activité de logopède, pour répondre à quelques questions.

Elle a commencé par m'expliquer qu'il existe plusieurs méthodes pour apprendre à l'enfant à lire. L'idéal, cependant, est de commencer à lui inculquer les bases de la lecture vers 3 ou 4 ans. En effet, cela permet d'optimiser ses compétences par la suite. Ce principe vient notamment des méthodes pédagogiques proposées par Maria Montessori<sup>18</sup>. Cette pédagogie met l'accent sur l'importance des périodes sensibles de l'enfant<sup>19</sup>. Celle concernant le langage se déroule environ de 2 mois à 6 ans. La pédagogie Montessori part du principe qu'il existe une conscience phonologique forte présente chez l'enfant à ce moment. Celui-ci apprend beaucoup à l'aide de rimes, de comptines, etc. C'est-à-dire à l'aide de sons. Ceci rejoint ce qui m'avait déjà été dit par Odile à propos de la méthode syllabique : il est important de se baser sur le son que fait la lettre seule, ou le groupe de lettres, et non le nom qui lui est donné, car ce dernier apporte des confusions.

De plus, elle a insisté sur le point suivant : pour apprendre à lire, l'enfant doit déjà maîtriser un certain vocabulaire à l'oral. Dès lors, si le vocabulaire de base n'est pas acquis, l'enfant ne sera pas capable de comprendre ce qu'il lit. Les premiers mots et textes seront donc composés de vocabulaire basique, pas de mots compliqués et peu fréquents, tels que *narval*, *okapi*, etc.

En tant que logopède, elle travaille essentiellement sur base des sons pour aider l'enfant à différencier les lettres qui posent problème. Elle ne s'est pas étendue sur son travail, ceci n'étant pas le but de la rencontre, mais m'a expliqué ce qu'elle considérait comme la technique la plus porteuse de résultats. Cette technique se base sur la méthode syllabique et la pédagogie Montessori, et rejoint en beaucoup de points celle appliquée par les instituteurs de l'école de Bruyères expliquée dans le point précédent. Parmi les points communs avec ce qui m'avait été expliqué précédemment, on trouve :

- La sélection d'un mot contenant un son entendu précédemment parmi plusieurs choix.
- Choisir une lettre qui produit le son prononcé. Ici, Laurence Henrion m'a spécifié que la lettre importait peu, tant que le son était correct. Par exemple, pour le son *sss*, l'enfant peut choisir aussi bien le *s* que le *c* ou le *ç*.
- Relier un mot ou un son à une image.
- Associer les sons et les lettres à des couleurs qui leur seront spécifiques. Au départ, il peut simplement s'agir de donner par exemple les voyelles en rouge et les consonnes en bleu.

Laurence Henrion travaille avec les enfants essentiellement avec des applications sur son iPad. Elle était donc de bon conseil pour me renseigner sur ce qu'elle considère qu'il manque à ces applications, et sur les erreurs à éviter. Elle m'a dit que sur iPad, elle trouve suffisamment d'ap-

18. Pour plus d'informations : *Pédagogie Montessori*, [http://fr.wikipedia.org/wiki/Pédagogie\\_Montessori](http://fr.wikipedia.org/wiki/Pédagogie_Montessori)

19. Une période sensible est un âge où l'enfant est plus prompt à développer certains aspects de son évolution naturellement.

plications spécialisées (peu toutefois reprenant plusieurs exercices), mais qu'il lui semble qu'il y a un manque de ce côté sur Android (ce pourquoi elle travaille sur iPad). Ceci m'a bien évidemment conforté dans mon choix.

La majorité applications présentes sur l'iPad ont été créées par un spécialiste orthophoniste (appellation d'un logopède en France) : Emmanuel Crombez. Celui-ci réalise des applications pour iPad, iPhone et Mac sous le nom *ABC Applications*. Celles-ci sont destinées à aider les enfants dans leur apprentissage général et dans leurs difficultés : lecture, écriture, mathématiques, etc. Parmi ces jeux, on peut trouver<sup>20</sup> :

- *Anagrammes*. L'enfant doit retrouver des mots dont les lettres ont été mélangées. Au besoin, il peut s'aider en écoutant le mot qu'il doit reconstituer.
- *Nuages de mots*. Cet exercice propose à l'enfant de retrouver le genre et le nombre des mots affichés en "nuages". Quel mot est masculin, ou féminin ? Est-ce singulier ou pluriel ? Le jeu entraîne à la lecture mais aussi au classement. Ce jeu existe aussi sous la forme *Nuages de lettres*, adapté aux enfants de maternelle pour apprendre à reconnaître les lettres.
- *Mémo des mots*. Ce jeu se présente sous la forme du bien connu *Memory*. La différence étant qu'ici des mots sont utilisés à la place des images. *Etc.*

Le gros avantage des applications d'Emmanuel Crombez est qu'elles sont créées par une personne maîtrisant les processus d'apprentissage de l'enfant. De ce fait, il a pu optimiser l'interface afin d'éviter trop de distraction ou de difficultés. Sur cette base, Laurence Henrion m'a conseillé de ne pas inclure trop d'éléments graphiques et de garder un design simple et épuré. Trop d'objets sur l'écran, c'est prendre le risque que l'enfant soit distrait et ne se concentre pas sur ce qui est requis, donc perdre en efficacité. Par ailleurs, elle m'a conseillé d'utiliser la police *OpenDyslexic* pour mon application. Elle a pu tester cette police via les jeux de *ABC Applications*. *OpenDyslexic* est une police open source conçue pour faciliter la lecture chez les personnes dyslexiques en accentuant certaines parties des lettres. Dans le cas où cette police n'aurait pas été disponible, une police sans serif type *Verdana* aurait convenu. Les polices sans serif sont plus faciles à déchiffrer, les pattes des lettres avec serif encombrant la vue.

Un autre exercice, indépendant de ceux précités et pour lequel Laurence Henrion portait grand intérêt est la *lecture flash*. Cet exercice consiste à mémoriser un mot (ou une phrase) affiché plus ou moins rapidement à l'écran, pour ensuite le réécrire correctement. Dans le cas présent, l'orthographe est travaillée en plus de la lecture. Ce sont deux matières très liées. Un jeu de lecture flash doit idéalement :

- avoir un temps réglable d'affichage selon le niveau de l'enfant
- insister sur l'importance de la mémorisation
- faire attention au sens de la lecture

En plus de cela, l'exercice peut être amélioré grâce à :

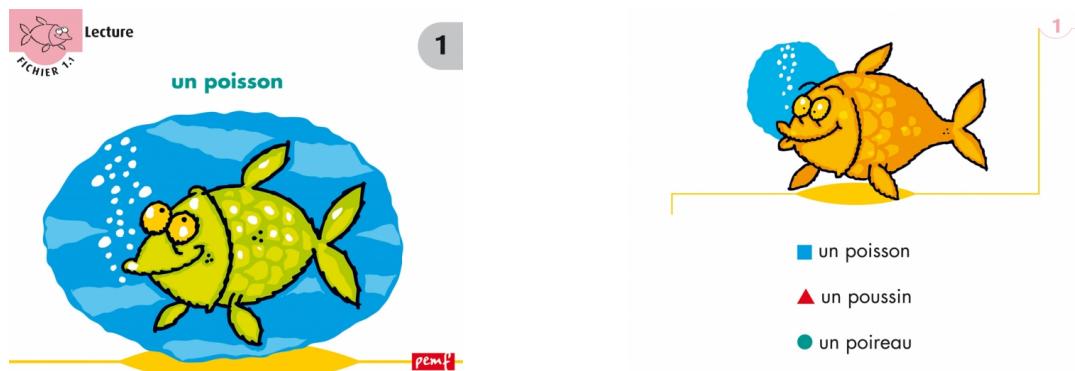
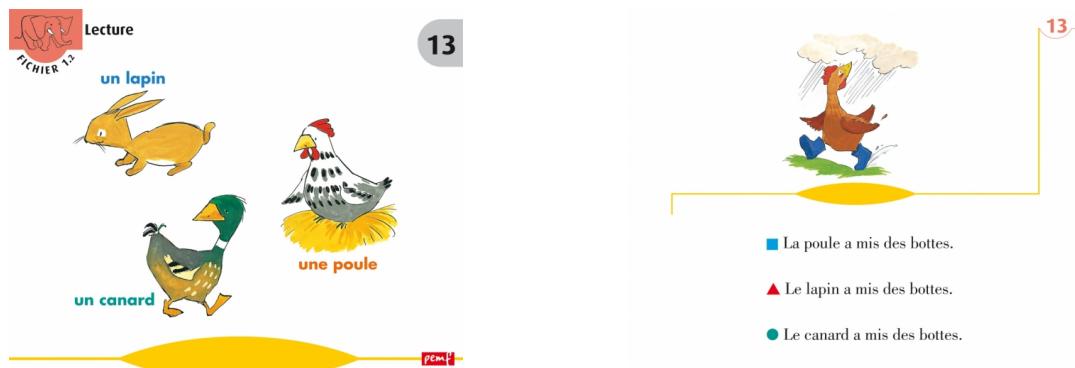
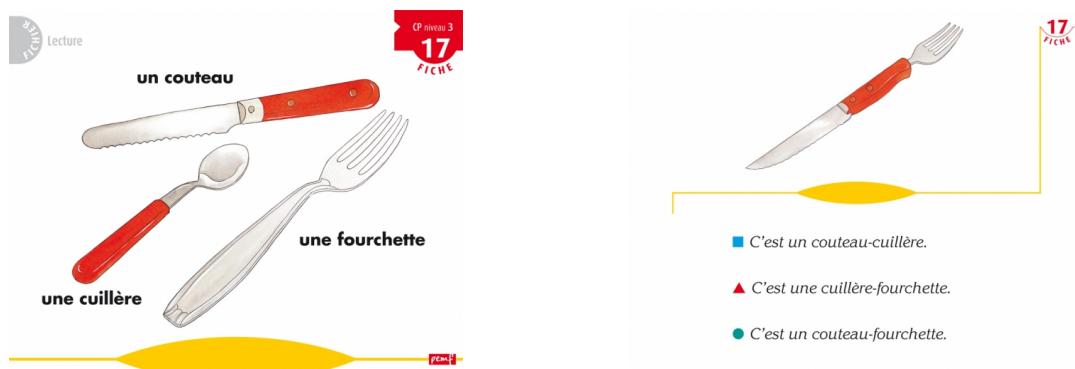
- l'affichage de mots personnalisés en fonction des centres d'intérêt de l'enfant
- la lecture effectuée avec un mot, et la réécriture sous la forme d'une image à choisir

Enfin, j'ai également reçu comme conseil d'utiliser le VOB (Vocabulaire Orthographique de Base) pour les exercices que je mettrai en place dans l'application. Il s'agit de la liste officielle en Belgique des mots qui doivent être connus et maîtrisés par les enfants à la fin de chaque cycle de primaire.

---

20. Liste complète sur le site d'Emmanuel Crombez : <http://abc-applications.com/ipad.html>.

## Annexe II : Exemples de fiches Freinet

FIGURE 8 – Source : <http://www.icem-pedagogie-freinet.org/node/3921>FIGURE 9 – Source : <http://www.icem-pedagogie-freinet.org/node/3979>FIGURE 10 – Source : <http://www.icem-pedagogie-freinet.org/node/3944>

### **Annexe III : VOB du cycle inférieur**

Voici la liste des mots devant être connus et maîtrisés par les enfants, téléchargée depuis <http://www.enseignons.be/upload/fondamental/francais/190607071618vob-degre-inferieur.pdf> le 26 janvier 2015.

## **Vocabulaire orthographique de base : degré inférieur**

<b>A</b>	<b>B</b>	<b>C</b>	ciel	déjà
à	balle	cadeau	cinéma	déjeuner
acheter	ballon	café	cinq	demain
aider	banc	cahier	classe	demander
aimer	bas	calcul	coeur	demi
aller	bateau	camarade	coin	dent
alors	beau	camion	combien	depuis
ami	beaucoup	car	comme	dernier
amusant	bébé	carnaval	comment	des
an	bête	carré	comprendre	dessin
animal	bien	carte	connaître	deux
année	bientôt	casser	content	devant
apprendre	blanc	cave	côté	devenir
après-midi	bleu	chambre	couleur	dictée
arbre	boire	chanter	coup	dimanche
armoire	bois	chapeau	courir	dîner
arriver	bon	chaque	coûter	dire
assez	bonbon	chasser	crayon	disque
attendre	bonhomme	chat	cri	dix
au	bonjour	chaud	crier	docteur
aussi	bonsoir	chemin	croire	donner
auto	bouche	chemise	cuisine	dormir
autre	boule	cher		dos
avant	bouquet	chercher	<b>D</b>	double
avec	bout	cheval	dame	doux
avion	branche	cheveu	dans	droit
avoir	bras	chez	danse	droite
avrill	bruit	chien	d'après	du
	bureau	chocolat	de	dur
		chose	décembre	

<b>E</b>	fermer	grimper	journal	<b>M</b>
eau	fête	gros	journée	ma
école	feu		juillet	machine
écouter	feuille	<b>H</b>	juin	madame
écrire	février	haut		magasin
élève	figure	heure	<b>K</b>	mais
elle	fille	heureux	kilo	main
en	fin	hier		maintenant
encore	finir	histoire	<b>L</b>	mais
encre	fleur	hiver	la , le	maison
enfant	fois	homme	laid	mal
enfin	forme	huit	laine	malade
ensemble	former		lait	maman
entendre	fort	I	langue	manger
entre	franc	ici	lapin	marchand
entrer	français	il	large	marcher
et	frère	il y a	laver	mardi
été	froid	image	leçon	marron
étoile	fruit		lecture	mars
être	fumée	<b>J</b>	légume	matin
	fumer	jamais	les vacances	mauvais
<b>F</b>		jambe	les yeux	me , m'
facile	<b>G</b>	janvier	lettre	méchant
faim	garage	jardin	lire	même
faire	garçon	jaune	lit	mer
famille	garder	je, ,j'	livre	mercredi
farine	gare	jeter	loin	mère
faute	gauche	jeu	long	mesurer
femme	gentil	jeudi	lourd	mètre
fenêtre	grand	joli	lui	mettre
fer	grand-mère	jouer	lundi	minute
ferme	grand-père	jour	lune	moi

moins	nuage	personne	promenade	rue
mois	nuit	petit	propre	
moitié		peu	puis	<b>S</b>
monde	<b>O</b>	peur		sa
monsieur	octobre	photo	<b>Q</b>	sac
montagne	oeil	piano	quand	sale
monter	oeuf	pièce	quatre	salle
montre	oiseau	pied	que	samedi
montrer	on	pierre	quelque(s)	s'amuser
mot	oncle	place	qui	sans
moto	orange	placer		sapin
mouche	ou	plaisir	<b>R</b>	s'arrêter
mouchoir	oui	pleurer	radio	sauter
mur	ouvrir	pluie	rectangle	se lever
musique		plus	regarder	se promener
	<b>P</b>	plusieurs	reine	semaine
<b>N</b>	page	poids	rendre	sept
nager	pain	poire	rentrer	septembre
ne , n'	papa	poisson	rester	seul
neige	papier	pomme	retourner	si
neiger	par	porter	revenir	six
neuf	parent	poupée	riche	soeur
nez	parfois	pour	rien	soi
ni	parler	pousser	rire	soif
nid	partie	pouvoir	rivière	soir
nom	partir	premier	robe	soleil
nombre	pas	prénom	roi	sous
non	passer	prendre	rond	souvent
nous	pauvre	préparer	rose	sucré
nouveau	pays	prêt	roux	sur
novembre	pendant	printemps	rouge	
nu	père	prix	route	

<b>T</b>	<b>V</b>			
ta,ton,tes	vélo			
table	vendre			
tante	vendredi			
te	venir			
télévision	vent			
temps	vie			
tenir	vieux			
terre	village			
tête	ville			
toi	vin			
tout	vite			
tomber	voici			
toucher	voilà			
toujours	voir			
tour	voiture			
tourner	voler			
tout	vouloir			
train	vous			
travail	voyage			
travailler				
très	<b>Z</b>			
triangle	zéro			
trois				
trou				
tu				
<b>U</b>				
un				
une				
usine				