

Table des matières

<u>Table des figures</u>	<u>p.3</u>
<u>Résumé – abstract – mots-clés</u>	<u>p.4</u>
<u>Glossaire</u>	<u>p.5</u>
<u>Introduction</u>	<u>p.6</u>
 <u>Cahier des charges</u>	 <u>p.7</u>
<i>Les rendez-vous client</i>	<i>p.7</i>
<i>Contraintes à respecter</i>	<i>p.7</i>
<i>Définition précise des besoins</i>	<i>p.8</i>
<u>Fonctionnement de l'application</u>	<u>p.8</u>
<u>Besoins en structure</u>	<u>p.8</u>
<u>Contraintes esthétiques et ergonomiques</u>	<u>p.8</u>
<u>Gestion de projet</u>	<u>p.9</u>
<i>Formation au langage Android</i>	<i>p.9</i>
<i>Méthodologie agile</i>	<i>p.9</i>
<u>Description théorique</u>	<u>p.9</u>
<u>Mise en pratique</u>	<u>p.10</u>
<i>Le planning prévisionnel</i>	<i>p.10</i>
<i>Les sprints hebdomadaires</i>	<i>p.11</i>
 <u>Le développement de l'application</u>	 <u>p.13</u>
<i>Choix techniques</i>	<i>p.13</i>
<u>Le format JSON</u>	<u>p.13</u>
<u>La pérennité et reprise en main du projet</u>	<u>p.14</u>
<i>Les classes annexes</i>	<i>p.14</i>
<u>Utils</u>	<u>p.15</u>
CameraUtil.java	p.15
JsonUtil.java	p.15
FilesUtil.java	p.16
<u>Gridview</u>	<u>p.17</u>
custom_gridview_menu.xml	p.17
custom_gridview_modif.xml	p.17
<i>Structure des activités</i>	<i>p.17</i>
<u>Architecture globale</u>	<u>p.18</u>
<u>Page de démarrage</u>	<u>p.18</u>
Main Activity	p.18
<u>Partie utilisateur</u>	<u>p.20</u>
User Menu Activity	p.20
User Seq Activity	p.21
<u>Partie administrateur</u>	<u>p.22</u>
Password Activity	p.22
Modif Menu Activity	p.23
Modif Seq Activity	p.24
<i>Le cahier de tests</i>	<i>p.28</i>
<u>Conclusion</u>	<u>p.30</u>
<u>Remerciements</u>	<u>p.31</u>
<u>Bibliographie</u>	<u>p.32</u>

Table des figures
(exemple : figure 1 – TITRE... page...)

Remarque : les différentes vues de l'application sont en perpétuel avancement, y compris à l'heure où la rédaction de ce rapport se termine. Elles ne sont donc pas toujours mises à jour et ne correspondent pas aux vues définitives, notamment sur le point esthétique (position des boutons et widgets, attractivité des couleurs). En effet, il a été décidé de rendre l'application fonctionnelle avant tout, et de régler les détails visuels ultérieurement.

<i>figure 1 – Gestion des tâches hebdomadaires sur la table Trello</i>	<i>p.10</i>
<i>figure 2 – Diagramme de Gantt annuel du projet</i>	<i>p.11</i>
<i>figure 3 – Exemple de la structure d'un séquentiel en format JSON</i>	<i>p.13</i>
<i>figure 4 – Architecture des activités de l'application</i>	<i>p.18</i>
<i>figure 5 – Activité de lancement</i>	<i>p.19</i>
<i>figure 6 – Activité présentant le menu des séquentiels disponibles à l'exécution</i>	<i>p.20</i>
<i>figure 7 – Activité d'exécution du séquentiel avant et après clic sur Suivant</i>	<i>p.21</i>
<i>figure 8 – Activité protégeant l'accès aux paramètres par un mot de passe</i>	<i>p.22</i>
<i>figure 9 – Activité permettant de créer ou de modifier un séquentiel</i>	<i>p.23</i>
<i>figure 10 – Activité permettant le paramétrage d'un séquentiel ou de ses étapes</i>	<i>p.25</i>
<i>figure 11 – Fonctionnement de la prise de photo</i>	<i>p.26</i>
<i>figure 12 – Enregistrement de la photo prise dans le dossier Photos de l'application</i>	<i>p.27</i>

Résumé

Ce document présente l'ensemble des objectifs initiaux et résultats obtenus, ainsi que les méthodes utilisées dans le cadre d'un sujet de projet M1 de l'ISEN-Brest.

Ce projet, soumis par l'équipe éducative de l'Institut Médico-Educatif de Plabennec, consiste à développer un agenda séquentiel sous forme d'une application sur tablette tactile, a pour but de guider les jeunes encadrés par l'Institut dans leurs activités quotidiennes.

En effet, ces jeunes souffrent de troubles autistiques ou mentaux divers, et par conséquent démontrent souvent certaines difficultés à se concentrer, à mémoriser l'ordonnancement d'une tâche du quotidien, ou à communiquer en général.

A cet effet, l'utilisation de la tablette tactile représente un support éducatif très ludique et apprécié des jeunes de l'Institut, et ce type d'application autorise une personnalisation individuelle difficile à reproduire sur d'autres supports.

Mots-clés : application sur tablette, support éducatif, autisme, Android, agenda séquentiel

Abstract

This document presents the set of initial goals and final results, as well as the methods used as part of a 4th year-student project at the ISEN-Brest.

This project, submitted by the educational team of the Medico-Educational Institute in Plabennec, consists in developing a sequential calendar in the form of a tablet application, and aims to guide the young people supervised by the Institute in their daily activities.

Indeed, these young people suffer from various autistic or mental disorders, and therefore often show some difficulties in focusing and concentrating, memorizing the scheduling of a daily task, or communicating in general.

For this purpose, the use of tablets is a very playful and attractive educational support to the young people at the Institute, and this type of application allows an individual customization that is difficult to duplicate on other educational media.

Keywords : tablet application, educational support, autism, Android, sequential calendar

Glossaire avec l'explication des abréviations

IME : Institut Médico Educatif de Plabennec, accueillant des jeunes souffrant de troubles autistiques ou autres troubles mentaux. L'IME est le client de l'application

Android : langage de programmation mobile combinant du Java et du XML

Séquentiel : nom donné aux tâches du quotidien représentées dans l'application. Ces tâches sont découpées en une séquence d'étapes ordonnées chronologiquement.

Activité : correspond en Android à une page ou vue de l'application. Une activité possède des éléments graphiques déterminés par le fichier .xml, et est implémentée par des méthodes dans un fichier .java.

Layout : signifie « disposition » en anglais. Nom donné au fichier .xml d'une activité puisque celui-ci va introduire les éléments graphiques et leur disposition au sein de l'activité

Sprint : terme utilisé en méthodologie agile, désignant une période élémentaire de travail, ritualisée par des réunions ou bilans de début et de fin de sprint. Ce projet a donc été découpé en 8 sprints d'environ une semaine.

Introduction

Le projet présenté consiste à développer une application sur tablette, sous Android, à destination de l'Institut Médico-Educatif de Plabennec. Cet institut (comme beaucoup d'autres établissements spécialisés) manque de supports éducatifs en général, et souhaite privilégier le support numérique pour diverses raisons :

- pratiques → les supports papier s'abîment ou sont perdus régulièrement, et nécessitent une réimpression fréquente
- ludiques → les jeunes montrent un intérêt accru pour les activités effectuées sur tablette tactile
- et surtout éducatives → les applications permettent une simplicité de personnalisation, délicate à répliquer sur d'autres types de supports

Ce n'est pas la première fois que l'ISEN collabore avec l'IME pour livrer une application sur mesure, et le client a donc l'habitude de traiter avec des étudiants dans cette optique.

Cette année, pour répondre aux besoins éducatifs de l'IME, l'application doit se présenter comme un agenda séquentiel, et vise à permettre aux jeunes atteints de troubles autistiques (ou d'autres troubles mentaux) d'effectuer des tâches simples.

Chacune de ces tâches est recensée dans un "séquentiel", qui est comme son nom l'indique, une séquence d'étapes ordonnée chronologiquement, ayant pour but d'assister les jeunes dans leurs activités quotidiennes.

Des exemples concrets seraient : se brosser les dents, s'habiller le matin, suivre une recette de cuisine...

Dans un premier temps, les rendez-vous avec le client ont aidé à déterminer les éléments et contraintes du cahier des charges avec plus de précision, puis à réviser les objectifs à mi-parcours.

Afin de démarrer le travail, une courte formation en ligne aux bases et spécificités du langage Android a été indispensable, et s'est poursuivie tout au long du projet à chaque fois que l'apprentissage d'une notion inconnue se révélait nécessaire.

Il a paru judicieux de fonctionner en méthodologie agile pour la durée du projet, facilement adaptable à un planning chronologique par ordre d'importance des fonctionnalités à ajouter dans l'application. Ce planning prévisionnel a été découpé par sprints hebdomadaires regroupant chacun une portion de fonctionnalités à développer en priorité.

Le travail de programmation a d'abord été de construire une structure minimale des activités et de la navigation dans l'application, pour ensuite étoffer chacune d'entre elles progressivement, toujours par l'ordre de priorité défini par le client, en passant par moments par des fonctions annexes supplémentaires comme la prise de photo par exemple. Il a également été utile de prendre le temps de la réflexion sur certains choix techniques concernant le fonctionnement général de l'application.

Enfin, ce travail a pu être ajusté au quotidien en relançant systématiquement la version la plus récente de l'application sur la tablette, et en testant chaque nouvelle fonction au fur et à mesure des ajouts ou modifications du projet.

Les résultats obtenus lors de ces tests ont ainsi été comparés au cahier des charges initial afin de faire un bilan et d'en déduire des pistes d'amélioration futures de l'application.

Cahier des charges

Les rendez-vous client

Deux rendez-vous avec le client ont ponctué le déroulement du projet, auxquels M. Napoléon a également assisté. Ceux-ci ont été pris auprès de M. Mickaël Quemener, éducateur spécialisé à l'IME de Plabennec, encadrant des adolescents ou jeunes adultes atteints de troubles mentaux (principalement autistiques). L'emploi du temps de M. Quemener étant assez chargé, il a fallu faire preuve d'anticipation et d'organisation afin d'obtenir des réponses à toutes les questions qui se poseraient durant le développement futur.

Au tout début du projet, la préparation au premier rendez-vous client a donc été cruciale afin de définir le cahier des charges et d'être en accord avec les besoins de l'IME.

Après une visite des locaux et un contact avec quelques-uns des jeunes de l'IME, accompagnés d'explications sur les dispositifs mis en place dans l'accueil quotidien des jeunes, les caractéristiques de l'application souhaitée sont abordées, et c'est le moment de poser toutes les questions préparées ou spontanées et de vérifier que les besoins identifiés sont bien compris.

De surcroît, l'application offre des possibilités d'améliorations assez vastes, et la durée du projet totalisant à peine 8 semaines, il a fallu être lucide et admettre que le temps était trop limité pour développer toutes les idées émises lors du rendez-vous. Il a donc été jugé préférable d'axer le développement sur les points impératifs et éventuellement quelques fonctionnalités supplémentaires de priorité élevée.

A l'aide des idées évoquées lors de l'échange avec M. Quemener, cet ordre de priorité du cahier des charges a été précisé et validé, et celui-ci a alors été mis sous forme de liste simple triée par importance décroissante.

Ce rendez-vous déclenche le lancement du développement, avec notamment la création du premier modèle général de l'arborescence des pages de l'application.

Pour le deuxième rendez-vous, le client s'est déplacé à l'ISEN et cela a été l'occasion de faire une démonstration de l'application développée à mi-parcours (au lendemain des vacances de printemps), de vérifier que le travail mené se place en adéquation avec les attentes réelles du client, et de redéfinir les priorités selon le temps restant imparti.

Contraintes à respecter

Deux contraintes majeures ont été identifiées dans le cahier des charges de ce projet :

- la première réside dans l'ergonomie, en effet l'application doit être très simple d'utilisation, très intuitive, avec peu d'options de navigation. Il faut garder à l'esprit que les utilisateurs sont des personnes qui ont des difficultés à lire ou à se concentrer. L'apprentissage du fonctionnement de l'application se veut quasi-inexistant, celle-ci doit être très facile à prendre en main.
- paradoxalement, afin de convenir à la majorité des utilisateurs potentiels, l'application a pour objectif de contenir une multitude de fonctionnalités différentes, la rendant ainsi adaptable à chaque individu. Ces fonctionnalités seront paramétrées par les éducateurs ou encadrants qui prépareront les séquentiels.

L'application a donc pour objectif d'être versatile sans gagner en complexité d'utilisation.

Définition précise des besoins

C'est lors du premier rendez-vous client que les éléments majeurs du cahier des charges ont pu être établi.

Fonctionnement de l'application

Sur le fonctionnement pur de l'application, c'est-à-dire la partie éducative demandée par l'IME, plusieurs requêtes sont à prendre en compte :

- la possibilité de créer un séquentiel à partir d'images importées
- la possibilité de sauvegarder ce séquentiel et le faire apparaître dans un menu
- la possibilité de supprimer un séquentiel
- la sélection d'un séquentiel parmi un menu doit s'exécuter au moins de manière manuelle (boutons Précédent et Suivant)
- l'objet séquentiel doit être extensible en nombre d'étapes : il est possible d'ajouter ou supprimer autant d'étapes que nécessaire
- chaque étape est accompagnée d'un petit texte explicatif (option éventuelle du choix entre image seule ou texte seul ou les deux)
- la possibilité de prendre une photo pour l'afficher en tant qu'image dans une étape
- la possibilité d'activer un mode automatique où les étapes sont minutées
- la possibilité d'activer la synthèse vocale sur un séquentiel
- la modification d'un séquentiel déjà existant (titre, images, nombre ou ordre des étapes, activation des différents modes, réglage de la vitesse des étapes...)
- éventuellement la possibilité de créer un séquentiel vidéo (en boucle?)

Besoins en structure

- accès à la partie paramétrage limité aux éducateurs (en particulier éviter qu'un jeune de l'institut puisse venir supprimer l'ensemble des séquentiels créés)
- faire des choix techniques intelligents assurant la longévité et maniabilité de l'application : garder en tête que ce projet pourrait idéalement être repris ultérieurement afin d'y apporter des améliorations ou autres fonctionnalités

Contraintes esthétiques et ergonomiques

Du côté visuel, les besoins sont assez simples : le mot d'ordre est le minimalisme et la lisibilité. La prise en main de l'application doit être intuitive. Afin d'adhérer au mieux à ces contraintes, voici ce qui a été mis en place :

- une navigation très minimale du côté utilisateur : un clic sur Démarrer puis deux activités à comprendre seulement
- des images et boutons d'assez grande taille, bien visibles, très peu de texte
- une police sans-serif (facilitant la lecture) et assez grande aussi
- des couleurs neutres sur un fond blanc suffisent
- un design aéré en général

De manière générale, les pages sont donc très épurées et leur design doit permettre d'aller droit au but et de trouver l'information recherchée immédiatement.

Gestion de projet

Formation au langage Android

Les notions élémentaires du langage Android sont enseignées en option Génie Logiciel à l'ISEN (majeure commune des deux membres du binôme en charge de l'application), mais il est évident que ces bases n'auraient pas suffi à mener à bien l'ensemble du projet. Cela s'explique par un manque d'expérience pratique et par la nécessité de coder des fonctions plus complexes lors du développement du projet (gestion et stockage de données notamment).

Afin de remédier à cela, le binôme s'est engagé à se former en autodidacte sur son temps personnel, grâce à des cours et exercices disponibles en ligne, ce qui a établi des fondations plus solides sur les subtilités des variables, les liens entre les différents fichiers, les méthodes ou fonctions couramment utilisées, etc.

Cet investissement a permis d'économiser un temps considérable lors de la reprise du projet en février. Ainsi, les différentes parties du développement ont été abordées avec plus de sérénité et de recul, et les recherches postérieures associées à des fonctions spécifiques ont été accomplies avec plus d'efficacité par la suite.

Méthodologie agile

Description théorique

La méthodologie agile est une méthode de gestion de projet itérative et incrémentale, mettant l'accent sur l'autonomie et l'auto-organisation au sein d'une équipe. Elle fonctionne par répétition de « sprints » d'une durée souvent fixe (ici, hebdomadaire), qui se déroulent de manière identique avec des bilans très réguliers concernant le travail effectué sur le sprint actuel et le travail à effectuer sur le sprint suivant. Ces bilans se font en présence du Scrum Master (ici, M. Napoléon) et/ou le Product Owner (en théorie, M. Quemener, mais en pratique, souvent représenté par M. Napoléon également), qui s'assurent alors de la réalisation des objectifs de chaque sprint par le binôme.

Cette méthode maximise le temps de productivité avec une répartition des tâches très aisée, et un avancement régulier certain. Elle est très adaptée à ce type de projet, c'est-à-dire comportant un objectif majeur (reproduction d'une tâche quotidienne à l'aide d'une application), mais dont les options sont quasi-infinies en terme d'adaptabilité et de design. Le cahier des charges est donc le document de référence, listant toutes les idées d'amélioration d'une telle application, afin de trier les fonctionnalités à développer par ordre de priorité demandé par le client.

Ainsi, durant les premiers sprints il a convenu de développer l'application minimale viable (MVP : Minimal Viable Product), puis à chaque nouveau sprint, il a suffi de sélectionner parmi la liste énoncée du cahier des charges (recopiée dans la table Trello) le lot de fonctionnalités à ajouter en priorité, le but étant d'avancer dans cette liste au maximum.

Mise en pratique

La mise en pratique de la méthodologie agile s'est effectuée à l'aide d'outils comme Trello et Github. Grâce à Trello, il est facile de trier les tâches à faire, à faire rapidement, et faites. Il est aussi possible d'attribuer les tâches à chaque membre du binôme, de leur donner une échéance... Sur la table Trello (*figure 1*), sont regroupées également les ressources nécessaires, telles que les coordonnées du client, le code source de l'application de l'année précédente, le guide fourni par l'IME concernant les supports visuels privilégiés au sein de l'institut, les liens web partagés (accès aux compte-rendus hebdomadaires, aux sources utilisées pendant le développement).

Pour travailler en parallèle sur le même projet, la gestion des versions en passant par Github s'est révélée indispensable, malgré quelques difficultés initiales d'installation et de prise en main.



figure 1 : Gestion des tâches hebdomadaires sur la table Trello

Le planning prévisionnel

A l'aide du logiciel GanttProject, ce diagramme de Gantt annuel prévisionnel (*figure 2*) a été établi dès la première semaine de projet. Il est volontairement peu détaillé et servait principalement à garder un œil sur les échéances majeures du projet au cours de l'année.

Les zones bleues correspondent aux sprints d'une semaine de travail pur sur le projet, les zones violettes représentent les rendez-vous avec le client, les zones orangées sont associées aux tâches administratives diverses (préparation du premier rendez-vous, aménagement du planning, installation des logiciels Android Studio, Github...), et les zones rouges en fin de projet marquent le travail lié à l'évaluation (rapport de projet, démonstration et soutenance).

La formation au langage Android est la seule tâche colorée en vert et sort des échéances de projet (sprints). En effet, ce langage étant assez dense en apprentissage, il a été jugé préférable de s'y former en autonomie petit à petit, grâce à des exercices quotidiens pendant les premières semaines suivant le début du projet, plutôt que d'y consacrer des journées entières en plein sprint.

Ainsi, les sprints du mois de février ont été abordés avec plus de sérénité par le binôme, et plus de recul envers les subtilités du langage.

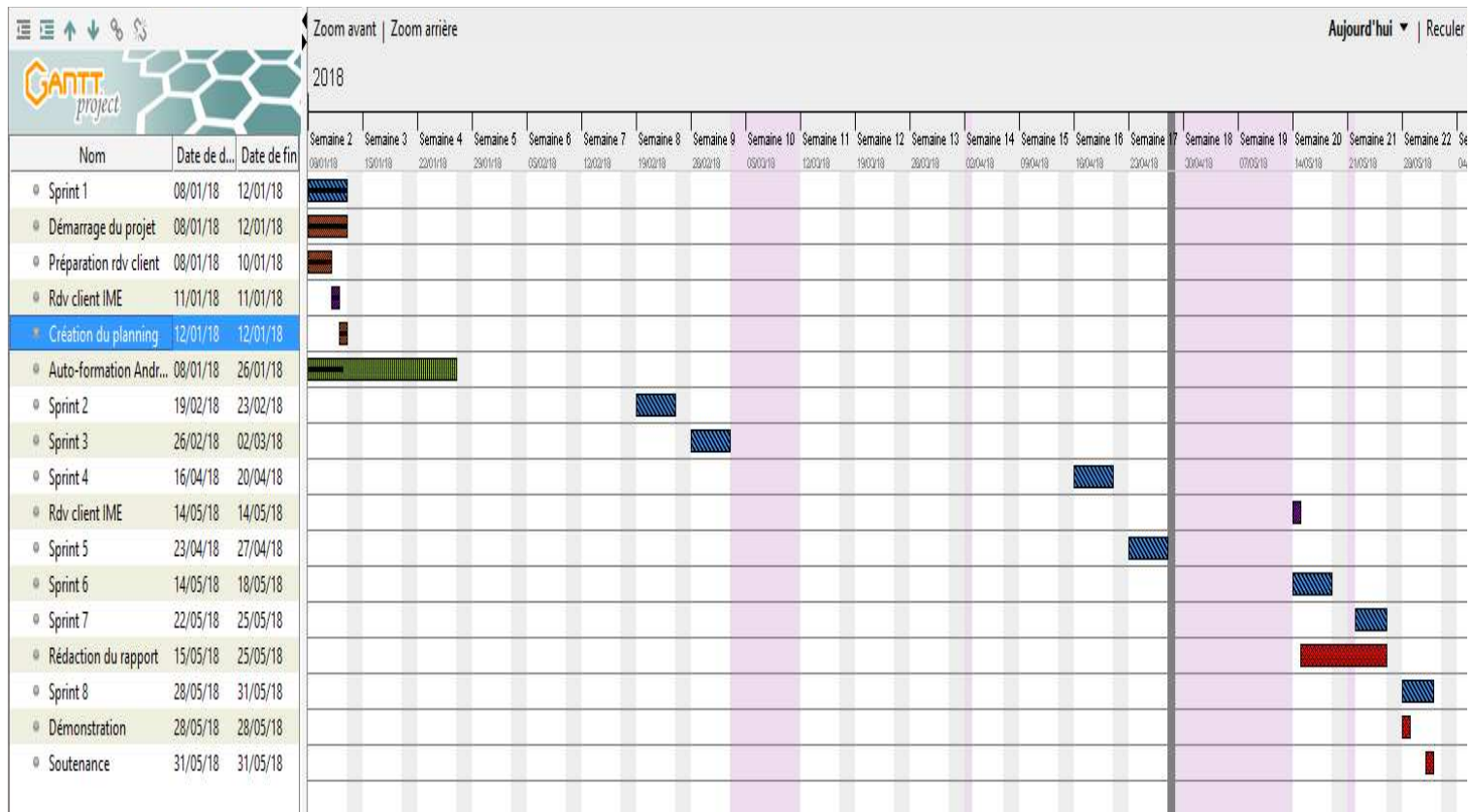


figure 2 : Diagramme de Gantt annuel du projet – découpage et principales échéances

Ce diagramme prévisionnel est très général et a donc été suivi sans problème en termes d'échéance des rendez-vous ou des évaluations.

Les sprints hebdomadaires

Au sein du binôme, la gestion de projet s'est effectuée en méthodologie agile, avec des sprints d'une durée d'une semaine, se découplant ainsi :

- le lundi matin, la définition des objectifs hebdomadaires par ordre de priorité, puis la répartition individuelle de l'exécution de ceux-ci
- tout au long de la semaine, chacun travaille presque de manière indépendante sur ses tâches de la semaine
- le vendredi après-midi, bilan avec l'encadrant (M. Napoléon) de l'avancement des objectifs, et préparation de la semaine suivante.

Cependant, une liaison avec le cahier des charges et la table Trello établie a permis de détailler le travail à fournir prioritairement lors de chaque sprint d'une semaine ([figure](#))

Sprints	Principales tâches effectuées
Sprint 1 – 08/01 au 12/01	<ul style="list-style-type: none"> - premier rendez-vous avec le client → identification des besoins - établissement du cahier des charges et de l'ordre de priorité <ul style="list-style-type: none"> - installation des logiciels - prévision des plannings - début de la formation en Android sur openclassrooms.com
Sprint 2 – 19/02 au 24/02	<ul style="list-style-type: none"> - création d'un répertoire sous Github - réalisation d'une maquette papier représentant l'architecture globale de l'application + validation par l'encadrant - choix du format de stockage des séquentiels (JSON) - démarrage de la programmation sous Android (vues des activités + gestion des séquentiels en JSON)
Sprint 3 – 26/02 au 02/03	<ul style="list-style-type: none"> - travail sur la conversion du format JSON - toutes les activités sont créées (même vides) - implémentation des liens entre chaque activité (navigation minimale terminée)
Sprint 4 – 16/04 au 20/04	<ul style="list-style-type: none"> - travail sur la prise de photo et son enregistrement dans le dossier souhaité (créé si inexistant) - intégration des fichiers JSON dans les activités - travail sur le chargement d'un séquentiel et l'exécution des étapes
Sprint 5 – 23/04 au 27/04	<ul style="list-style-type: none"> - ajout et implémentation des boutons ou widgets nécessaires à la page de paramétrage - création d'une boîte de dialogues permettant la navigation dans la galerie de photos ou d'images
Sprint 6 – 14/05 au 18/05	<ul style="list-style-type: none"> - 2ème rendez-vous avec le client : bilan et révision des objectifs - redéfinition des priorités et des tâches restantes + répartition <ul style="list-style-type: none"> - début de la rédaction du rapport de projet - travail sur le paramétrage du séquentiel et des étapes - amélioration de l'esthétique globale des activités
Sprint 7 – 21/05 au 25/05	<ul style="list-style-type: none"> - aboutissement de la partie paramétrage - création d'un séquentiel test - recherche sur la fonction de synthèse vocale - remise en forme du code source (commentaires, séparation des fonctions annexes puis appel dans les vues, épuration des parties trop lourdes...) - préparation de la démonstration - fin de la rédaction du rapport de projet
Sprint 8 – 28/05 au 31/05 (prévisionnel)	<ul style="list-style-type: none"> - rendu du rapport + démonstration (évaluation) - ajout de la fonctionnalité synthèse vocale (idéalement) - petites améliorations esthétiques ou fonctionnelles - préparation et passage de la soutenance (évaluation) - remise de l'application au client dans l'App store

Le développement de l'application

Le développement en Android possède la particularité de rassembler sur une vue (dite « activité ») la partie Back-end en Java et la partie Front-end en XML. Ainsi, pour créer une activité, il faut combiner au minimum les deux fichiers suivants :

- le fichier .xml ou « layout » (disposition, en anglais) dans lequel on déclare tous les éléments graphiques (boutons, champs de texte, saisies de texte pour citer les plus courants) ainsi que leurs caractéristiques (taille et type de police, position des boutons, marges...).
- le fichier .java dans lequel seront codées toutes les fonctions associées à cette activité (par exemple, faire correspondre une action au clic sur un bouton).

Il est donc possible de créer une vue uniquement avec un code XML, mais cette page sera statique, un texte saisi ne sera pas traité, un bouton ne sera pas cliquable...

Ces deux versants parallèles de l'application doivent donc communiquer et garder une cohérence parfaite lors de la programmation.

Choix techniques

Le format JSON

L'un des premiers problèmes techniques rencontré était de déterminer un format de stockage pour les séquentiels. Sur les conseils de l'encadrant, le fichier JSON, formant un tableau de clés-valeurs, s'est imposée comme une option intéressante pour la flexibilité qu'elle offre dans le stockage et la récupération de données. C'est une forme de stockage suffisamment légère et souvent utilisée dans le transfert de données sur le web, ce qui sera primordial dans l'éventualité d'une possibilité de synchronisation par Bluetooth entre différentes tablettes.

Ainsi, ce format peut contenir aussi bien les paramètres généraux du séquentiel que les détails liés à chacune des étapes de celui-ci (*figure 3*).

```
{
  "nom_seq": "Exemple de séquentiel",
  "icone_seq": null,
  "voix": true,
  "defilement": "minuteur",
  "etapes": [
    {
      "media": "/res/.../image.jpg",
      "nom_etape": "une image",
      "num_etape": 1,
      "minuteur": 15,
      "video_vitesse": null,
      "video_boucle": null
    },
    {
      "media": "/res/.../video.mp4",
      "nom_etape": "une vidéo",
      "num_etape": 2,
      "minuteur": null,
      "video_vitesse": 1,
      "video_boucle": true
    }
  ]
}
```

figure 3 : exemple de la structure d'un séquentiel en format JSON

Il est également doté d'une certaine souplesse dans l'optique d'une amélioration future du projet, car il sera alors très facile d'ajouter certaines fonctionnalités qui seront sauvegardées de la même manière, sous un format clé-valeur, au sein même des objets JSONObject et Sequentiel. La manipulation de ces objets n'en sera donc pas pour autant devenue ingérable, et cela assure donc les possibilités futures de modification du projet.

Enfin, il paraît évident que les séquentiels créés ne feront pas tous la même taille, d'où la nécessité d'avoir un format extensible, pouvant générer un nombre d'étapes variable selon les séquentiels. De plus, ce format permet de stocker tout type de variable, comme des images ou vidéos (à travers une chaîne de caractères exprimant le chemin auquel on peut trouver ces données), des nombres, des booléens... ce qui semble être un choix parfait pour une telle application.

La pérennité et reprise en main du projet

Dès le démarrage du projet, une problématique s'est soulevée très rapidement : s'il est admis que le temps imparti du projet est trop court pour aboutir toutes les idées possibles d'options dans cette application, comment faire en sorte de rendre le code source assez souple et clair pour une personne qui reprendrait le projet ultérieurement en vue de creuser les pistes d'amélioration ?

Les solutions trouvées afin de faciliter cette prise en main ont été de :

- structurer et commenter le code source au maximum
- détailler les méthodes utilisées dans les fonctions (notamment dans ce rapport qui pourra être considéré comme un support de documentation pour les personnes reprenant ce projet)
- créer des dossiers par types de fonctions (Utils, layouts particuliers, et activités à part) ce qui aère le code et permet de mieux retrouver les informations
- externaliser certaines fonctions afin de les rendre réutilisables à différents endroits
- séparer les ressources dans un autre fichier (séquentiels, photos, pictogrammes...) de manière à les rendre indépendantes de l'application
- porter une attention aux méthodes utilisées : vérifier qu'elles sont assez classiques et non spécifiques à une poignée de versions d'Android, afin qu'elles restent fonctionnelles longtemps

Les classes annexes

Comme mentionné précédemment, le nombre de fichiers associé à une vue doit être au minimum 2, mais peut très bien être supérieure à ce seuil. En effet, il est possible d'appeler (dans le fichier .java de l'activité) des fonctions regroupées ailleurs, ce qui permet de séparer les blocs de code trop volumineux, et de pouvoir appeler ces fonctions à plusieurs endroits de manière indépendante.

Souvent, celles-ci sont liées à une fonctionnalité particulière comme la prise de photo par exemple. On peut également créer un élément graphique plus complexe (exemple : le Gridview) puis l'appeler sur un autre fichier .xml afin de gagner en clarté ou si besoin est, de le réutiliser à différents endroits de l'application.

Utils

Le dossier Utils rassemble toutes les classes utilitaires qui sont appelées sur les différentes vues de l'application. Il permet de mieux s'y retrouver dans l'organisation des fichiers du code source, et de pouvoir mieux réutiliser ces fonctions à différents endroits, ainsi que de raccourcir la longueur du code présent pour chacune des classes associées aux activités. Ces classes représentent les plus laborieuses de tout le projet, et ont préalablement demandé un grand travail de recherche et de réflexion.

- CameraUtil.java

Cette classe a été créée dans le but de regrouper toutes les fonctions associées à l'accès à la caméra de la tablette. Cette classe contient la fonction prise de photo, et enregistrement de cette dernière directement dans le dossier « Media/Photos » de l'application (dossier créé si non-existant). Ainsi, toutes les photos prises sont faciles à retrouver pour les ajouter dans l'application.

Sur une amélioration future, la classe CameraUtil pourra également comporter la fonction prise de vidéo par exemple.

Elle contient actuellement trois méthodes :

- la fonction getPhotosDir (Context context), qui sert uniquement à créer le dossier Photos dans le dossier Media du stockage de fichiers de l'application, et appelle getExternalFilesDir()
- la fonction getRealPathFromURI (Context context, Uri uri) permettant de récupérer le chemin réel de la photo prise à partir de son URI.. Ce chemin est alors utilisé pour la copier dans le dossier créé précédemment
- et enfin la fonction copyToDir (Context context, Uri uri) qui utilise le chemin récupéré précédemment afin de copier la photo correspondante dans le dossier Photos. Ce chemin est stocké dans une chaîne de caractères input_path, qui est prise en fichier d'entrée afin de copier la photo en fichier de sortie.
Cette fonction possède également l'attribut timeStamp qui est un peu accessoire mais adapte le nom de la photo stockée à un format tout simple « IMG_ » + aaaaMMjj_HHmmSS + « .jpg » (par exemple : IMG_20180528_120000.jpg).

- JsonUtil.java

Cette classe contient les fonctions permettant de passer d'un objet Séquentiel à un JSON et vice-versa. Les fonctions qu'elle comporte sont utilisées en complément des fonctions « createFile » et « openJsonFile » de la classe FilesUtil.java.

Elle est scindée en deux parties complémentaires et indissociables pour le bon fonctionnement de la gestion de données :

- la méthode JSONObject toJson (Sequentiel sequentiel) permet de créer un objet JSONObject à partir de l'objet Sequentiel correspondant, et retourne un objet JsonObject. Elle constitue globalement une énumération de fonctions put() qui associent au JsonObject sequentielJson les paramètres clés-valeurs liées au séquentiel (paramètres du séquentiel ou de ses étapes, avec les étapes séparées dans un objet JsonArray, pour lequel on boucle les associations un nombre n de fois (n étant le nombre d'étapes).

- Et la méthode `Sequentiel toSequentiel (JSONObject jsonObject)` qui, à l'inverse, va créer l'objet `Sequentiel` de l'objet `JSONObject` correspondant. La procédure est très similaire, à la différence près que les fonctions utilisées sont une suite de `set()` et qu'on utilise la liste `List<Sequentiel.Etape>` pour boucler les étapes.

- FilesUtil.java

Cette classe contient les fonctions utilisées pour obtenir les séquentiels de l'application (récupérées ensuite dans la plupart des activités), ainsi que les médias associés (images ou photos) et pour créer les fichiers JSON correspondants. Les problématiques rencontrées pour réaliser ces fonctions ont demandé un travail assez conséquent :

- de compréhension du système de fichiers d'Android, du stockage interne et externe, de séparation des dossiers propres à l'application et les dossiers standards de l'appareil
- de respect de l'objectif de pérennité du projet (clair, lisible et réutilisable), ce qui a nécessité de décider du lieu de stockage des ressources de l'application
- de compréhension du système des permissions en Android (lecture et écriture de la carte SD ici), qui sont situées dans un fichier du projet séparé appelé le manifest. Notamment, c'est également dans ce même fichier que l'on autorise l'accès à la caméra, sans lequel la fonction prise de photo ne serait pas possible

Les fonctions développées sont multiples :

- la fonction `isExternalStorageWritable()` retourne un booléen (vrai ou faux) et vérifie simplement si le stockage externe est accessible en écriture
- la fonction `isExternalStorageReadable()` vérifie cette fois si le stockage externe est accessible en lecture
- la fonction `File getJsonDir (Context context)` retourne un objet `File` pointant sur l'emplacement du dossier `Json` de l'application, et va servir généralement à initialiser un `File` pointant sur un séquentiel particulier
- la fonction `File getMediaDir (Context context)` retourne un objet `File` pointant sur l'emplacement du dossier `Media` de l'application, et va servir généralement à initialiser un `File` pointant sur une image ou photo utilisée dans un séquentiel
- la fonction `createFile(byte[] data, File pathname, String filename)` doit créer un fichier à l'emplacement donné par *pathname* dont le titre est *filename* (extension incluse), et dont le contenu est fourni par *data*
- la fonction `deleteFile(File pathname, String filename)` doit supprimer le fichier nommé *filename* situé à l'emplacement donné par *pathname*. Cette fonction est utilisée par exemple pour supprimer un séquentiel dans la partie Paramètres.
- la fonction `JSONObject openJsonFile (File jsonFile)` va ouvrir le fichier `jsonFile`, enregistrer son contenu dans un tableau d'octets et utiliser ces derniers afin d'initialiser un objet `JSONObject`. Si ce fichier n'existe pas ou que le contenu ne permet pas l'initialisation d'un `JSONObject`, alors la fonction renvoie *null*
- la fonction `List<Sequentiel> listSequentiels (Context context)` retourne une liste des objets `Sequentiels` décrits par les fichiers du dossier `Json` de l'application
- la fonction `Bitmap mediaToBitmap(Context context, String media)` retourne un objet `Bitmap` situé au chemin relatif décrit par *media*, et retourne un `Bitmap` par défaut si le chemin n'existe pas ou ne correspond pas.

Gridview

Le Gridview est un élément graphique du langage Android qui permet de créer une grille modulable d'items répétés. L'utilisation de celui-ci s'est révélée particulièrement utile dans ce projet afin de faire un affichage propre et souple en taille du menu des séquentiels par exemple. Le choix de créer cet élément dans une page séparée s'explique par sa longueur en code, en effet il est plus aisé de se repérer dans le fichier .xml en référant un type de Gridview parmi d'autres éléments graphiques liées à la même vue, et de déclarer les caractéristiques propres au Gridview dans un fichier différent.

Ce choix est également pertinent car l'architecture de l'application possède une certaine symétrie, donc il a été possible de réutiliser ce même Gridview dans les activités qui comportaient des similitudes visuelles, ce qui a donc engendré un gain de temps considérable.

- custom_gridview_menu.xml

Ce type de Gridview a été créé pour les deux activités présentant un menu : User Menu Activity et Modif Menu Activity. Il va afficher une grille d'items constitués chacun d'une icône (ImageView) et d'une description ou titre (TextView), listant ainsi l'ensemble des séquentiels disponibles à la sélection. Ce layout définit ainsi les tailles d'images ou de police, la position des descriptions situées en-dessous des icônes, et le nombre de colonnes souhaité du Gridview est défini directement dans le layout de l'activité correspondante, lors de l'appel du Gridview et de son positionnement dans la vue associée.

- custom_gridview_modif.xml

Ce deuxième type de Gridview est utilisé dans Modif Seq Activity qui permet de paramétrer chaque étape du séquentiel. Il est basé sur le premier type mais possède quelques différences notables :

- les images affichées en ImageView ne sont pas des icônes de séquentiels, mais directement les images associées à chaque étape du séquentiel sélectionné
- les items de ce Gridview sont légèrement plus complexes car ils possèdent en plus d'une image et d'un texte, des boutons à droite et à gauche permettant de paramétrer la vitesse de l'étape et son ordre dans le séquentiel, boutons répétés à chaque étape afin de personnaliser au maximum chaque séquentiel
- enfin, lors de l'appel de ce Gridview, le nombre de colonnes a été fixé à 1, afin de mieux structurer l'affichage des étapes verticalement

Structure des activités

Après une formation initiale en ligne aux bases du langage Android, et le premier rendez-vous avec le client, l'architecture générale des vues de l'application a été imaginée (*figure 4*).

Architecture globale

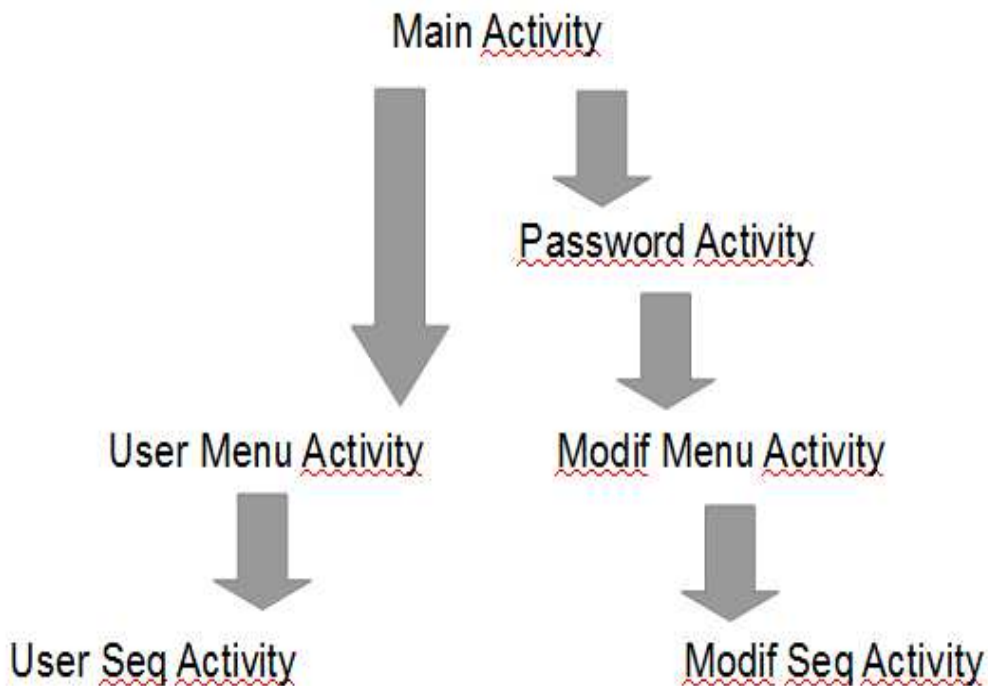


figure 4 : Architecture des activités de l'application

L'application est dotée d'une symétrie intentionnelle entre la partie utilisateur (User Menu Activity et User Seq Activity), et la partie administrateur (Modif Menu Activity et Modif Seq Activity), à la différence près que l'accès à cette dernière est intercalée par un mot de passe (Password Activity). Chacune de ces parties possède un menu de séquentiels à sélectionner, soit pour l'exécuter (User Menu Activity), soit pour le modifier (Modif Menu Activity). Une fois sélectionné, le séquentiel va soit s'exécuter (User Seq Activity) soit s'afficher pour le paramétrage (Modif Seq Activity).

La création de ce modèle a été directement suivie du développement de chaque activité (même vide) dès le début du projet, et surtout l'implémentation des liens entre ces activités. C'est-à-dire que, très rapidement, il était possible de naviguer sur l'ensemble des pages de l'application. Par la suite, cela a permis de se répartir le travail presque indépendamment, chacun travaillant sur des vues ou fonctions différentes.

Page de démarrage

- Main Activity (*figure 5*)

// La présentation

C'est la page qui est affichée au lancement de l'application.

Elle est très épurée, et possède simplement deux boutons : "Démarrer" et "Paramètres", qui permettent respectivement d'accéder à la partie utilisateur ou à la partie administrateur.

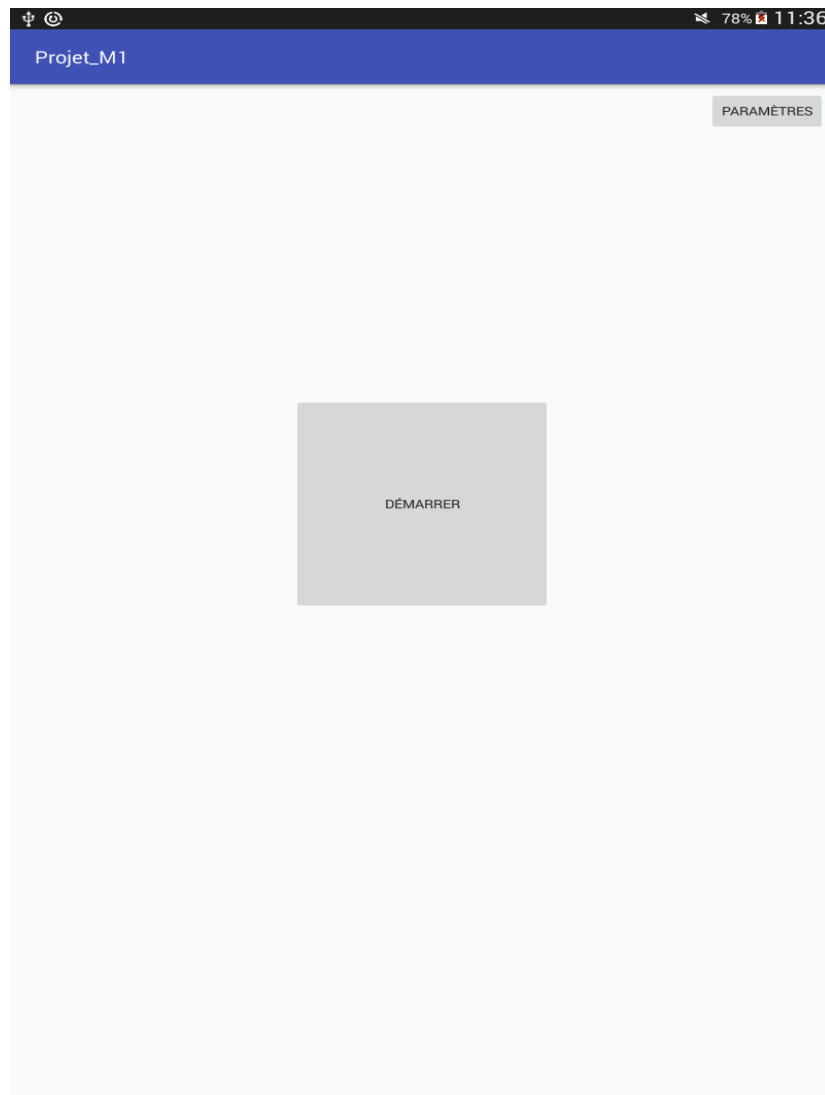


figure 5 : Activité de lancement

La pertinence de cette page centrale réside dans les deux points suivants :

- la séparation très simple d'une partie « administrateur » protégée par un mot de passe, qui sécurise la suppression de données éventuelle par un jeune de l'IME.
- La possibilité future (pourquoi pas un projet M1) de compléter facilement l'application avec des fonctionnalités supplémentaires, potentiellement indépendantes du développement actuel. En effet il suffirait d'ajouter un bouton sur l'activité Main et de le lier à la nouvelle partie souhaitée.

// Le développement

En terme de programmation, cette page était aisément la plus simple à mettre en œuvre, car elle contient uniquement deux méthodes correspondant à chaque bouton :

- la méthode startClick (View view) qui définit un Intent allant vers User Menu Activity, donc menant à la partie utilisateur de l'application
- la méthode paramClick (View view) qui définit un Intent allant vers Password Activity, menant cette fois à la partie administrateur de l'application

Ces méthodes sont simplement appelées dans le layout (activity_main.xml) dans la déclaration de leurs boutons respectifs (paramètre onClick = « nom de la méthode »).

Partie utilisateur

– User Menu Activity (*figure 6*)

// La présentation

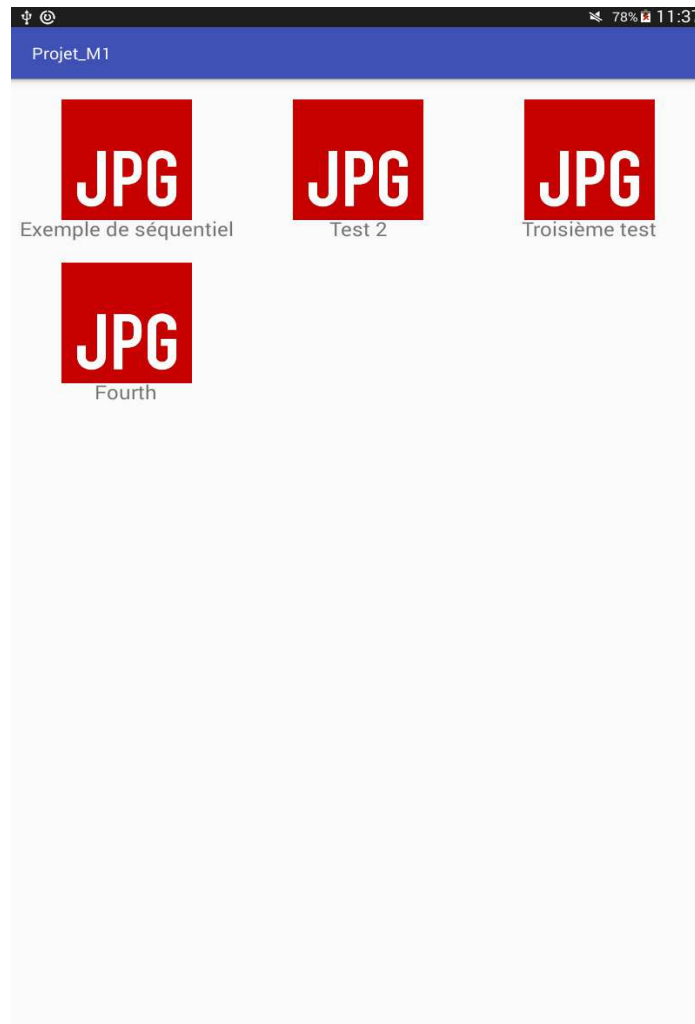


figure 6 : activité présentant le menu des séquentiels disponibles à l'exécution

C'est un menu de séquentiels qui permet à l'utilisateur de sélectionner la tâche qu'il souhaite reproduire. Elle se présente comme une grille (outil GridView sur Android) d'icônes accompagnées de leur titre ou description. L'icône correspond à la première image de chaque séquentiel. Le clic sur une icône va charger le séquentiel sélectionné et ouvrir la page d'exécution du séquentiel.

// Le développement

La classe User Menu Activity utilise le Gridview seqList importé du layout custom_gridview_menu.xml (comme défini précédemment, cf. la partie Fonctions annexes). Ce seqList va donc charger les items du Gridview : un tableau de bitmaps iconBitmap associées aux icônes des séquentiels, et un tableau de chaînes de caractères iconStr correspondant aux titres des séquentiels. Cette grille est incrémentée grâce à des boucles qui parcourent les tableaux cités précédemment.

Sur clic d'une icône, le séquentiel correspondant va être chargé sur la page suivante User Seq Activity qui permettra l'exécution de celui-ci. Cela se fait via une méthode simple d'Intent déclenché par la fonction `seqList.setOnItemClickListener(adapterView, view, position, id)` prenant en argument la position de l'icône sélectionnée.

– User Seq Activity (*figure 7*)

// La présentation

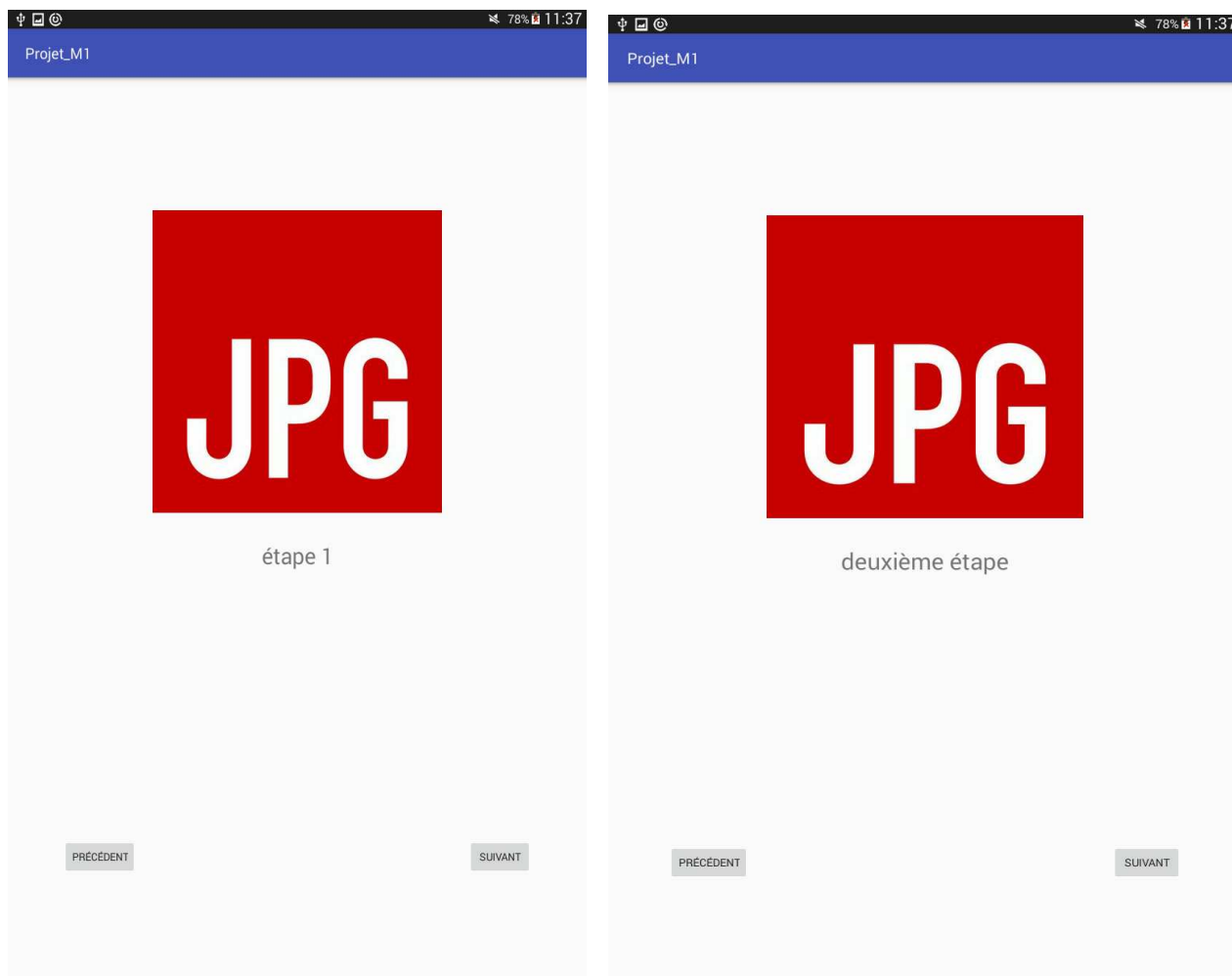


figure 7 : activité d'exécution du séquentiel avant et après clic sur Suivant

Cette page va exécuter le séquentiel sélectionné précédemment, selon les paramètres définis dans Modif Seq Activity, et enregistrés dans le format JSON. Le déroulement des étapes s'adapte donc aisément au type de tâche ou à l'utilisateur. Par exemple, pour l'activité "prendre une douche", le défilement peut se régler en automatique avec activation de la synthèse vocale. L'éducateur peut également diminuer la vitesse des séquentiels avant de prêter la tablette à un jeune plus lent en apprentissage que la moyenne... Elle peut être considérée comme le cœur de l'application, car elle constitue ainsi toute la partie éducative que le client souhaitait initialement.

// Le développement

Lors du chargement de cette page, la première étape du séquentiel sélectionné doit s'afficher automatiquement, ce qui est rendu possible grâce à l'objet `List<Sequentiel.Etape>` créé dans les fonctions annexes contenues dans `FilesUtil`. Le mode de défilement des étapes suivantes est défini dans les paramètres et peut donc se faire soit de manière manuelle à l'aide des boutons Précédent et Suivant (possédant chacun une méthode `onClick` simple permettant d'incrémenter ou de décrémenter le parcours de la liste des étapes), soit de manière automatique avec le minuteur lié à chaque étape en sauvegarde dans le fichier JSON.

La fonction `displayEtape()` permet quant à elle d'afficher le texte (description succincte) et l'image liés à chaque étape, et s'effectue à l'aide d'une fonction `setText()` qui va chercher le numéro de l'étape actuelle puis la description associée, puis d'une fonction `get()` qui récupère le chemin de l'image souhaitée.

Partie administrateur

– Password Activity (figure 8)

// La présentation

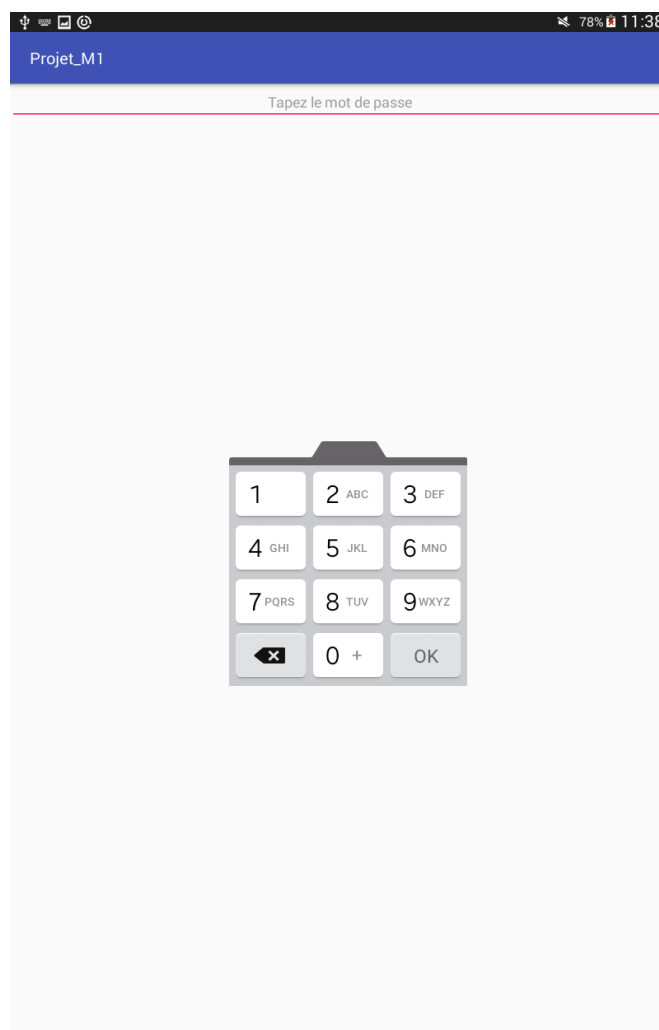


figure 8: activité protégeant l'accès aux paramètres par un mot de passe

Cette page sert uniquement à protéger les données paramétrées, et donc est accessible uniquement aux personnes détenant le mot de passe (les éducateurs). Cela permet d'éviter que les jeunes suppriment par erreur l'ensemble des séquentiels enregistrés par exemple. La modification du mot de passe se fait au niveau des paramètres de la page suivante.

// Le développement

Afin de développer cette page, il a suffi de créer un layout contenant un champ de saisie (ou EditText) password_field en paramétrant le type de saisie comme ceci : `inputType = « numberPassword »`, et cela fait apparaître le clavier numérique visible sur la figure.

Ensuite, dans la classe Password Activity, il a fallu implémenter la méthode password_click qui vérifie l'exactitude du mot de passe saisi lors de la validation sur OK. Ce mot de passe est une chaîne de caractères stockée dans *password*, modifiable par l'administrateur dans la partie Paramètres. Une simple comparaison de *password* avec le texte saisi, effectuée grâce à la fonction `password_field.getText().toString()` permet de :

- lancer la page Modif Menu Activity via un Intent, si la saisie est égale au mot de passe
- ou de rafraîchir la page Password Activity et d'afficher une notification « Mot de passe invalide ! » si la saisie est incorrecte.

- Modif Menu Activity (*figure 9*)

// La présentation

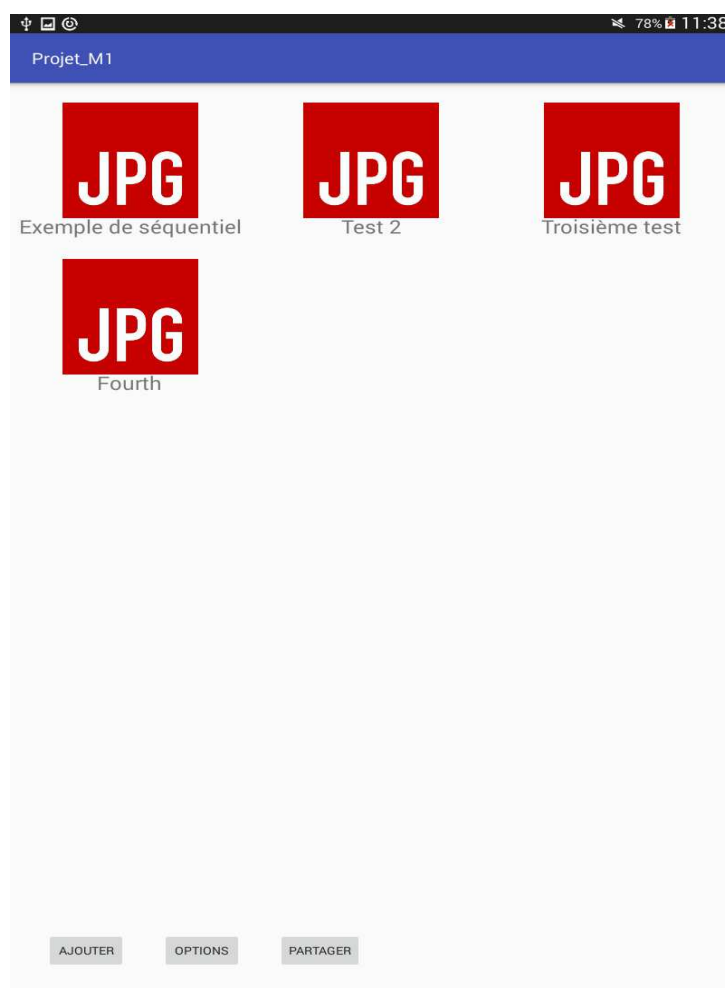


figure 9 : activité permettant de créer ou de modifier un séquentiel

Cette page charge les séquentiels enregistrés, et est quasi-identique à User Menu Activity. L'affichage se fait grâce à un GridView également. La différence réside dans le fait que le clic sur l'une des icônes va mener à une page de paramétrage des séquentiels, et non de leur exécution. De plus, elle contient les boutons qui permettent de créer un nouveau séquentiel, et de modifier le mot de passe par exemple. Un clic long sur l'une des icônes permet également de supprimer le séquentiel correspondant.

// Le développement

Le développement de cette page reprend les éléments principaux de User Menu Activity, avec le même type de grille d'affichage, à la différence près que la sélection d'un séquentiel va permettre de charger celui-ci dans la page Modif Seq Activity, et que les boutons du bas n'étaient pas présents précédemment.

Une différence majeure par rapport à User Menu Activity est qu'il est possible de supprimer un séquentiel sur cette activité : en effet il suffit de cliquer longuement sur l'icône représentant le séquentiel à supprimer. Cela passe par la méthode `setOnItemLongClickListener()` qui va appeler la fonction `deleteFile()` située dans `FilesUtil.java`.

Le bouton Ajouter permet de créer un nouveau séquentiel, en chargeant un Gridview vide lors de l'accès à la page suivante, à l'aide de la méthode `add_click`.

Le bouton Options va ouvrir une fenêtre de dialogues permettant plusieurs options (à développer prochainement) parmi lesquelles la modification du mot de passe d'accès aux Paramètres.

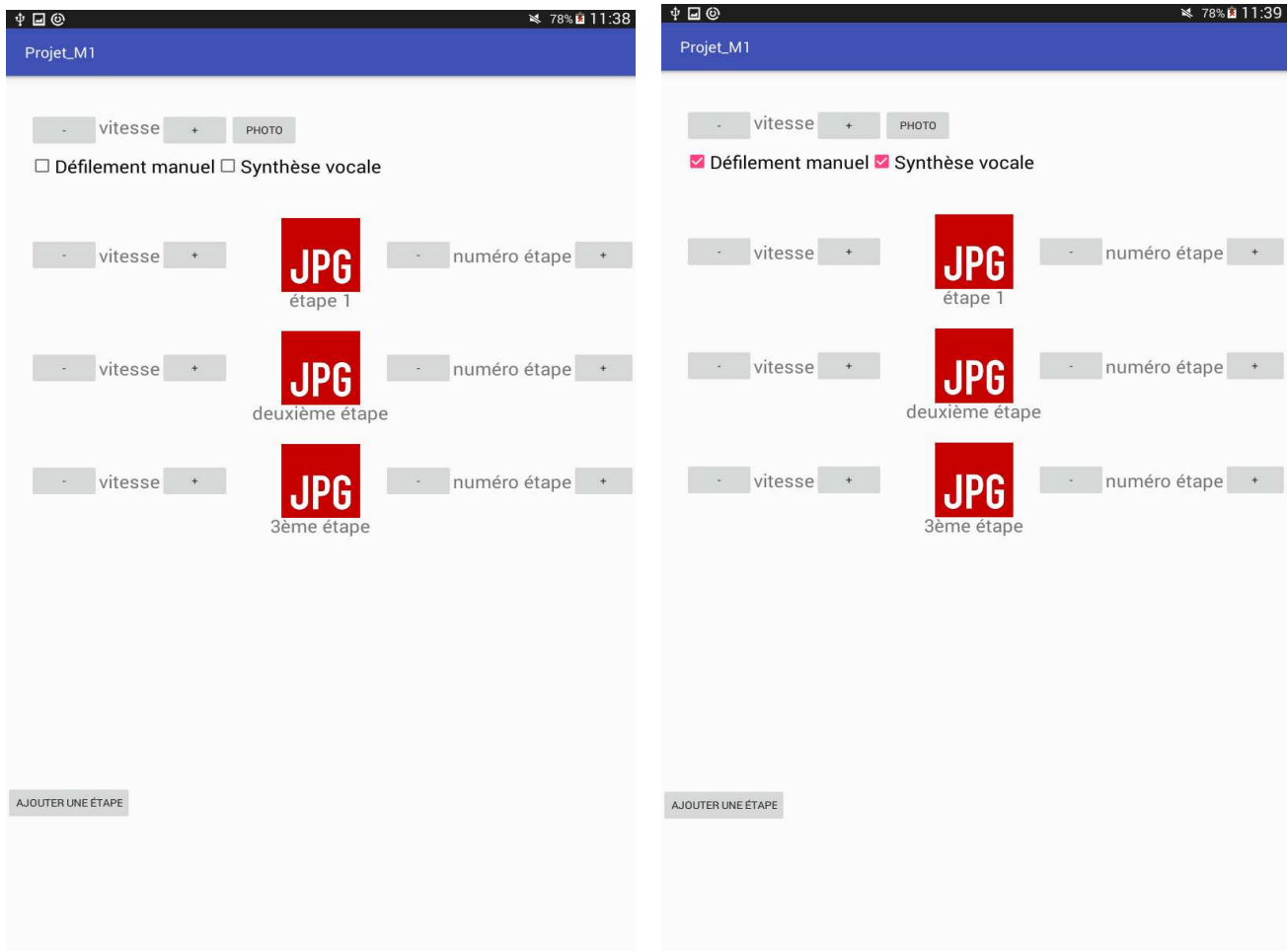
Le bouton Partager serait à rendre cliquable dès que les fonctions de synchronisation Bluetooth entre tablettes seront développées.

- Modif Seq Activity (figure 10)

// La présentation

Cette page définit tout le contexte d'utilisation de l'application : elle comporte toutes les options de paramètres développés, telles que le défilement et la vitesse des étapes, l'activation de la synthèse vocale, la définition du titre du séquentiel, la prise de photo.

Cette page s'ouvre sous deux modes : la création et la modification. Elle peut soit charger un séquentiel déjà créé (sélectionné précédemment), afficher l'ensemble de ses étapes avec les paramètres sauvegardés antérieurement, et permet alors la modification de ce séquentiel. Soit, sur clic du bouton "créer" (page précédente), elle peut charger un séquentiel vide et l'administrateur sera en mesure d'ajouter le nombre d'étapes voulues, d'y associer les images et descriptions, de régler tous les paramètres de ce nouveau séquentiel comme cité précédemment.



*figure 10 : activité permettant le paramétrage d'un séquentiel ou de ses étapes
Avant et après activation des options Défilement manuel et Synthèse vocale*

// Le développement

En terme de programmation, cette activité représente probablement la plus complexe car elle doit centraliser tous les paramètres qui seront sauvegardés dans le séquentiel sélectionné, déterminant ainsi tout le mode d'exécution de la partie utilisateur.

Cette activité a donc nécessité la création d'un grand nombre de widgets associés chacun à l'une des fonctionnalités implémentées.

Cette activité peut être divisée en deux parties. La première partie est liée aux options générales du séquentiel, comportant :

- deux boutons – et + permettant d'ajuster la vitesse globale du séquentiel, dans le cas d'un défilement automatique
- le bouton Photo qui appelle l'accès à la caméra et permet la prise d'une photo et enregistrement, dont les fonctions ont été détaillées précédemment (cf. CameraUtil.java)
- deux cases à cocher ou décocher, correspondant respectivement à l'activation du défilement manuel et l'activation de la synthèse vocale
- un champ de saisie permettant d'actualiser ou d'ajouter le titre du séquentiel
- un bouton Ajouter une étape, qui permet d'augmenter la taille du séquentiel
- et un bouton Valider qui permet de sauvegarder l'ensemble du séquentiel, et

l'enregistre dans les deux menus de l'application

La deuxième partie consiste à pouvoir modifier les étapes en elles-mêmes, avec :

- deux boutons – et + identiques aux boutons de la partie précédente, permettant cette fois de régler la vitesse d'une étape individuelle
- l'image et le texte de l'étape, qui sont modifiables sur un clic long (fonction `setOnItemLongClickListener()` avec ouverture d'une boîte de dialogues, permettant également la suppression de cette étape)
- et deux autres boutons – et + (fonctionnent encore de manière similaire) qui, quant à eux, permettent d'intervertir des étapes en changeant l'ordre des étapes

// *Fonctionnement de la prise de photo (figure 11)*

Afin d'ajouter une image, l'administrateur peut ajouter une image déjà enregistrée ou bien prendre une photo qui est enregistrée automatiquement dans le dossier "Media/Photos" de l'application.

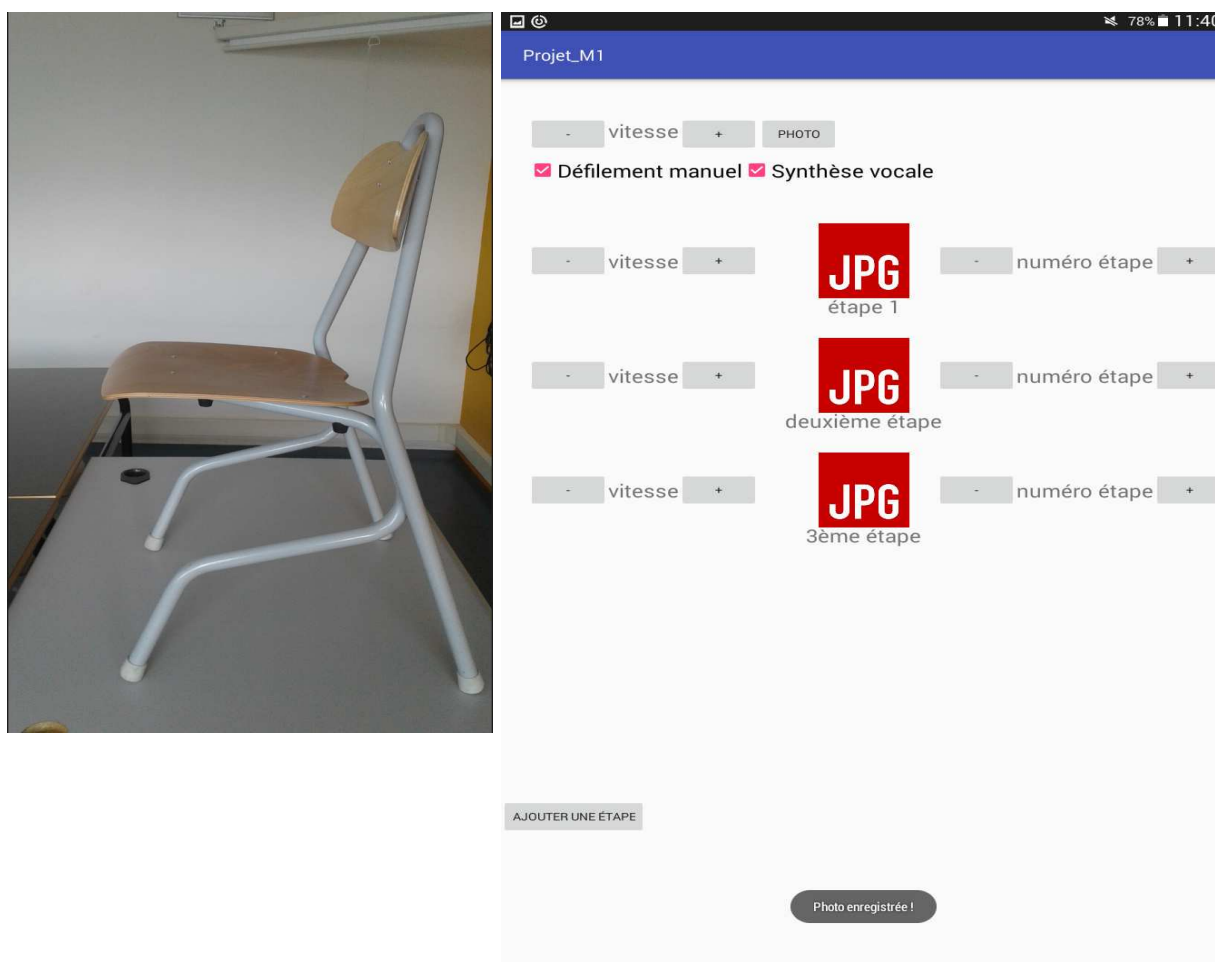


figure 11 : sur clic du bouton Photo, accès à la caméra, prise de la photo (chaise) validée puis retour à l'écran précédent avec notification « Photo enregistrée ! »

Dans le stockage interne de l'appareil, la photo enregistrée est bien présente à l'endroit souhaité, dans le dossier « Media/Photos » de l'application (*figure 12*)

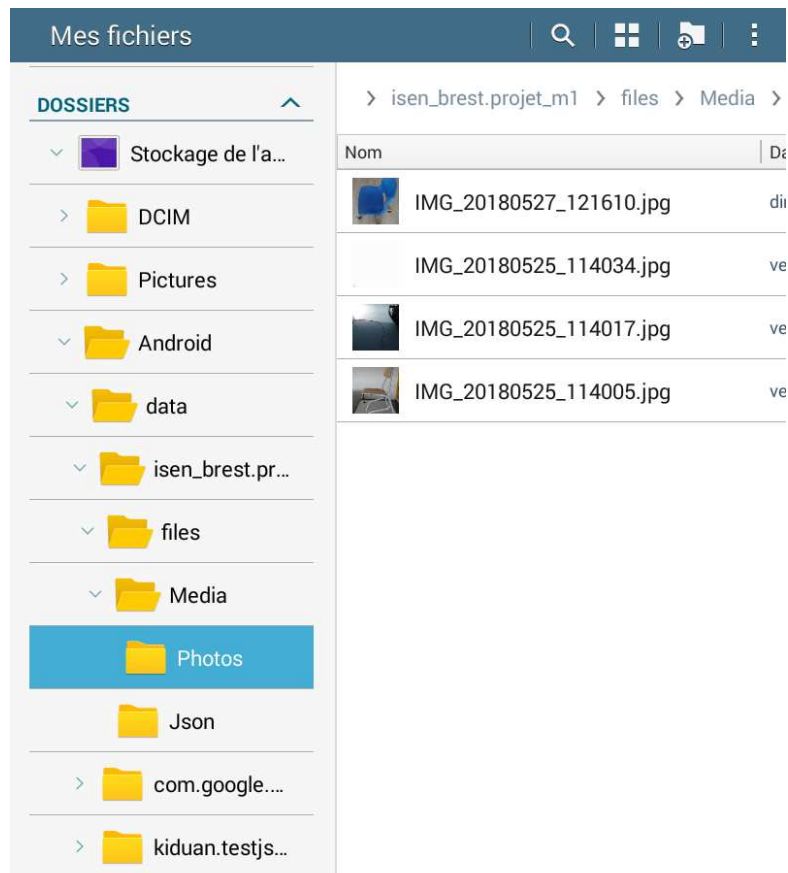


figure 12 : capture d'écran des fichiers de l'appareil dans le dossier avec présence de la photo de la chaise dans le dossier Photos de l'application

Cahier de tests

Le tableau ci-dessous regroupe l'ensemble des tests effectués pour vérifier le fonctionnement de chaque élément de l'application. Ce cahier de tests est séparé par activité, et les tests ont été effectués progressivement tout au long de l'avancement du projet.

<u>Fonction testée</u>	<u>Protocole de test</u>	<u>Résultat attendu</u>	<u>Validé</u>
Main Activity			
Lancement de l'application	Clic sur l'icône de l'application	Affichage de Main Activity	OK
Bouton Démarrer	Clic sur le bouton	Envoi à la page User Menu Activity	OK
Bouton Paramètres	Clic sur le bouton	Envoi à la page Password Activity	OK
User Menu Activity			
Sélection d'un séquentiel	Clic sur une icône	Envoi à la page User Seq Activity avec affichage de la première étape du séquentiel sélectionné	OK
Longueur du menu	Créer un Gridview suffisamment grand pour qu'il dépasse du cadre	Une barre verticale de défilement doit s'afficher automatiquement et permettre à l'utilisateur de naviguer sur l'ensemble du tableau sans problème	OK
User Seq Activity			
Bouton Suivant	Clic sur le bouton	Affichage de l'étape suivante	OK
Défilement manuel	Clic sur l'image	Affichage de l'étape suivante	OK
Bouton Précédent	Clic sur le bouton	Affichage de l'étape précédente	OK
Password Activity			
Validation du mot de passe	Taper le mot de passe correct dans le champ	Envoi à la page Modif Menu Activity	OK
Invalidation du mot de passe	Taper un mot de passe incorrect	Non-envoi vers la page suivante et notification « mot de passe invalide »	OK
Modif Menu Activity			
Sélection d'un séquentiel	Clic sur une icône	Envoi à la page Modif Seq Activity avec chargement du séquentiel correspondant	OK
Longueur du menu	Créer un Gridview suffisamment grand pour qu'il dépasse du cadre	Une barre verticale de défilement doit s'afficher automatiquement et permettre à l'utilisateur de naviguer sur l'ensemble du tableau sans problème	OK

Bouton Ajouter	Clic sur le bouton	Envoi à la page Modif Seq Activity avec chargement d'un séquentiel vide	OK
Fonction testée	Protocole de test	Résultat attendu	Validé
Modif Seq Activity			
Longueur du séquentiel	Créer un Gridview suffisamment grand pour qu'il dépasse du cadre	Une barre verticale de défilement doit s'afficher automatiquement et permettre à l'utilisateur de naviguer sur l'ensemble des étapes sans problème	OK
Bouton photo	Clic sur le bouton	Accès à la caméra de la tablette en mode photo	OK
Enregistrement photo	Prise d'une photo après clic sur « Photo »	Notification « Photo enregistrée ! » et vérification de sa présence dans le dossier « Media/Photos »	OK
Activation du défilement manuel	Cocher la case « Défilement manuel »	Le séquentiel ainsi paramétré va s'exécuter avec un défilement manuel	OK
Activation de la synthèse vocale	Cocher la case « Synthèse vocale »	Le séquentiel ainsi paramétré va s'exécuter avec synthèse vocale	A faire
Réglage vitesse du séquentiel	Clic sur les boutons – et + correspondants	Le séquentiel doit être enregistré avec une vitesse altérée	OK
Réglage vitesse des étapes	Clic sur les boutons – et + correspondants	Les étapes doivent être enregistrées avec une vitesse altérée	OK
Modification du titre du séquentiel	Clic sur le champ de saisie et saisie d'un nouveau titre	Le titre doit être enregistré et visible dans les menus	OK
Ajout/modification d'une image	Clic sur l'image actuelle et recherche dans la galerie de la nouvelle image	L'image doit être correctement modifiée et affichée dans les menus	OK
Ajout d'une étape	Clic sur Ajouter une étape	Une nouvelle étape vide s'affiche à la fin du séquentiel déjà créé	OK
Déplacement d'une étape	Clic sur les boutons – et + correspondants	Les étapes sont enregistrées avec un nouvel ordre et après rafraîchissement de la page, s'affichent correctement	OK
Sauvegarde du séquentiel	Clic sur Valider	Vérifier que les modifications sont prises en compte sur toutes les pages de menu et d'exécution	OK

Fonction testée	Protocole de test	Résultat attendu	Validé
JSON Util			
Conversion en JSON	Passer à la fonction un séquentiel dont les attributs sont connus	Le JSON doit retourner correctement toutes les caractéristiques du séquentiel et de ses étapes	OK
Conversion en Séquentiel	Passer à la fonction un JSON connu créé à la main	Le séquentiel doit s'afficher correctement (notamment le chemin des images doit afficher l'image)	OK
Files Util			
Fonctions getJsonDir et getMediaDir	Appeler un fichier File et afficher dans une chaîne de caractères son chemin	Comparer ce chemin avec celui de l'explorateur de fichiers de la tablette	OK
Fonction createFile	Passer à la fonction un chemin connu, un titre connu et une séquence d'octets connue	Lire le fichier résultat et vérifier que l'ensemble correspond à la création du fichier souhaitée	OK
Fonction deleteFile	Fournir le nom du fichier et son chemin dans la fonction	Vérifier que le fichier est bien supprimé au bon endroit	OK

Conclusion

Les objectifs initiaux du cahier des charges ont globalement été atteints afin de proposer un support éducatif numérique accompagnant les jeunes dans leurs tâches quotidiennes. En effet, toute la base des séquentiels, de leur structure, et leurs fonctions élémentaires a été construite, et de plus l'ensemble de cette construction s'est effectuée de manière à pouvoir continuer à concrétiser d'autres idées venant de l'IME.

Les choix techniques déterminés dans ce projet ont été judicieux et optimiseront la facilité de prolongation du travail effectué, par exemple avec la création de séquentiels en format JSON.

Néanmoins, quelques pistes d'amélioration ont été évoquées par l'éducateur de l'IME lors du second rendez-vous client :

- la mise en place de la synthèse vocale, lisant les étapes du séquentiel (si non abouti après le rendu de ce rapport)
- le traitement des séquentiels vidéos : avec idéalement le réglage de la vitesse, la possibilité de les lire en boucle ou en une fois
- la synchronisation entre tablettes (Bluetooth) ou le transfert des séquentiels en général
- la notion de récompense à la fin d'un séquentiel, ou d'une barre de progression permettant d'évaluer l'avancement de la tâche
- ajout éventuel de paramètres supplémentaires si demandés par l'IME

De manière générale, si ce projet est repris en main ultérieurement, il sera judicieux de demander le retour d'utilisation au bout de quelques mois directement auprès de l'IME, et de rester à l'écoute de leurs besoins. Quelque chose semblant prioritaire aujourd'hui pourrait perdre de l'importance face à une autre fonctionnalité plus intéressante à développer selon l'équipe éducative.

Il ne faut pas perdre de vue que cette application est avant tout créée sur mesure pour l'IME de Plabennec, et il est donc primordial de rendre cette application utilisable le plus souvent et le plus longtemps possible, en la faisant évoluer en conséquence de leurs remarques.

Malgré tout, et sur demande également du client, l'application sera déposée dans l'App Store afin d'élargir son public et d'augmenter sa visibilité. L'IME de Plabennec n'est pas le seul établissement de ce type à manquer de moyens éducatifs, et donc cette application peut être tout aussi utile à d'autres établissements voire aux familles des jeunes atteints de troubles autistiques, qui manquent également souvent d'outils qui pourraient faciliter leur apprentissage.

L'architecture de cette application est aussi faite de façon à pouvoir en élargir la fonction principale : l'idée de l'ajout d'un calendrier en lien avec les tâches ou d'un véritable agenda qui pourrait prévoir l'ensemble des tâches à effectuer le lendemain ou dans la journée serait peut-être à explorer. Les possibilités offertes sont multiples, et tout est construit de sorte que le projet soit facile à reprendre en main, toujours dans le but d'aider l'utilisateur à apprendre et d'aider l'éducateur à accompagner.

Remerciements

Merci à M. Thibault Napoléon, pour son suivi hebdomadaire et pour avoir effectué l'encadrement du projet dans la bonne humeur.

Merci à M. Mickaël Quemener (éducateur) et à l'Institut Médico Educatif de Plabennec, pour la confiance accordée et pour l'opportunité de développer un projet si valorisant.

Merci à Denis Le Gall, M1 à l'ISEN, pour sa générosité d'avoir prêté son PC personnel, et sans lequel il aurait été plus difficile de développer sous Android Studio.

Bibliographie

<https://openclassrooms.com/courses/developpez-votre-premiere-application-android>
Formation initiale au langage Android

<https://developer.android.com/>

<https://stackoverflow.com/>

<http://www.vogella.com/tutorials/android.html>

<https://developer.android.com/training/data-storage/files#WriteExternalStorage>

<https://developer.android.com/guide/topics/resources/drawable-resource>

<http://www.androidbegin.com/tutorial/android-display-images-from-sd-card-tutorial/>

<http://square.github.io/picasso/>

<http://www.appsdeveloperblog.com/java-into-json-json-into-java-all-possible-examples/>