

# Rapport de MOOC

## I – Introduction

Le choix de ce mooc a été difficile à faire car plusieurs traitaient du sujet qui m'intéressait. En effet, le développement d'applis mobiles est une activité nécessaire, et de plus en plus importante dans le monde informatique actuel. C'est donc vers ce sujet que je me suis dirigé, espérant réaliser le mooc validé l'an dernier par un élève désormais en 3A. C'est finalement M Brunel qui m'a affecté au cours Edx : Professional Android App Développement de l'université Galileo.

## II – Première session

La première session du mooc est consacrée à un ensemble de vidéos qui servent de base au cours.

### **Videos de structure**

Certaines vidéos avaient pour objectif de présenter la structure du cours : comment celui-ci s'étalerait dans la durée, comment on pouvait (en payant) contacter et obtenir l'aide de professeurs et de professionnels, comment on pouvait aussi (en payant encore) accéder au contenu étendu du cours, et enfin rappeler une dizaine de fois comment, toujours en payant, on pouvait obtenir un diplôme.

### **Videos de contextualisation**

D'autres vidéos avaient pour objectif de mettre en place le cadre de travail associé au cours. Bien naturellement le professeur rappelait ce que l'on allait étudier, comment s'imbriquaient les leçons et les exercices. En outre, on y découvrait les outils que l'on allait employer pour travailler.

## **Test**

A la fin de cette session d'introduction se trouvait un sondage à destination des étudiants ainsi qu'un test visant à évaluer leur connaissance des informations qui leur avaient été données dans l'introduction au cours.

## **Prérequis**

Pour réussir ce mooc, il est bien sûr indispensable de maîtriser l'objet. Malheureusement le niveau attendu est relativement élevé, ce qui a été pour moi une difficulté qu'il a fallu difficilement surmonter pour parvenir à avancer. Par ailleurs, il est aussi attendu d'être un bon connaisseur d'android, système sur lequel je n'ai jamais travaillé.

Ces prérequis exigeants, et pour lesquels la formation scolaire ne couvre pas tout ce qui est attendu, ont composé la grande difficulté de cet enseignement que je devais suivre tout en m'exposant à l'inconnu.

## **III – Outils**

Plusieurs outils informatiques ont été utilisés dans le cadre de ce mooc, certains pour le codage, d'autres pour la gestion des différents projets.

### **Gitlab**

Savoir utiliser un git est d'une grande importance pour la gestion efficace de projet. Ainsi, les connaissances acquises par rapport à un git possédant plusieurs branches sont utiles pour réussir à s'y repérer et à y récupérer le maximum d'informations efficaces. En l'occurrence, c'étaient les différentes ressources des projets proposés par le mooc qui se trouvaient dans un gitlab arborescent.

### **Oracle VM Virtualbox**

Il s'agit d'une machine virtuelle permettant de tester directement sur son ordinateur l'application en cours de développement. Elle affiche à l'écran une représentation de téléphone dont on peut choisir le modèle, et dont le comportement reproduit à l'identique celui d'une telle machine soumise à l'application testée.

### **Android Studio IDE**

Il s'agit cette fois-ci d'un environnement de développement pour android, permettant de coder de manière efficace au sein d'une interface ergonomique, et très bien aménagée.

Ces quelques outils ont été particulièrement utiles pour l'enseignement. Si Android Studio a été aisé à prendre en main et à employer, ça n'a malheureusement pas été le cas de la machine virtuelle, puisque je n'en avais jamais géré auparavant. Le Git aussi a pu poser quelques difficultés, dues à une utilisation trop disparate de ma part de cet outil de gestion.

## IV – Deuxième session

Lors de la deuxième session de travail, je me suis attaqué à la première leçon. Celle-ci vise à expliquer les différentes façons d'implémenter une application, et à lancer le codage d'une application de chat en ligne.

### **MVC-MVP**

Il s'agit d'un sujet assez aisé à comprendre : on nous explique la différence entre la traditionnelle architecture modèle-vue-contôleur (MVC) du web et les architectures modèle-vue-présentateur (MVP) que l'on peut employer dans le développement d'applis mobiles.

Tandis que d'un côté le MVC place le travail logique de l'application dans le contrôleur, laissant la vue gérer le graphisme et le modèle les données, l'architecture MVP propose comme alternative de séparer la logique de représentation, rattachée à la vue, et la logique d'arrière plan, réservée au présentateur qui a pour but de formater les données du modèle.

### **MVVM**

Dans cette architecture, les trois composantes sont le modèle, la vue et la vue-modèle. Tout le travail de validation et d'analyse se fait dans le modèle, qui est aussi chargé des données. La vue sert à mettre en place les couches et l'environnement graphique ; enfin la vue-modèle remplit la vue et structure l'affichage à l'écran.

### **Videos de codage**

Après cette approche théorique, le cours enchaîne avec une session de codage et d'implémentation. En me procurant sur le git les ressources nécessaires, j'ai pu programmer en même temps que le professeur les bases d'un système de connexion. Malheureusement cet enseignement comportait quelques difficultés.

- Premièrement, l'absence totale de contexte. Ni les sources, ni le framework, ni les parties déjà implémentées du code ne sont présentées, ce qui demande beaucoup d'adaptation et des capacités à deviner. Par un simple mimétisme, on peut cependant réussir à réaliser le travail demandé.

- Ensuite, l'absence de sens global. Concrètement, faire des copiers-collers de code, remplir des trous et supprimer certaines parties sans explication m'a surpris et m'a laissé dans le flou. Comment comprendre le sens global de l'implémentation pour android en ne faisant que de la sculpture sur code ?

Néanmoins, un accent audible, la qualité de la transcription, et enfin le fait d'avoir une vidéo de bonne résolution permettant de voir et de comprendre le code du professeur ont été d'excellents atouts de compréhension, qui m'ont permis de progresser de manière plus sûre.

## **Eventbus**

Cette séquence de vidéos est suivie par une paire de vidéos traitant de suites d'évènements déclenchables par l'utilisateur. Cette approche par le codage sans préambule théorique m'a complètement échappé.

## V – Troisième session

La session suivante s'est intéressée à Firebase et à des outils de graphisme. Comme toujours, une approche théorique a précédé une approche pratique en vidéo. Cette partie a particulièrement éveillé mon attention ainsi que mon intérêt.

### **Firestore : présentation**

A l'origine, Firestore était un outil de synchronisation de données entre applications. Grâce aux travaux de Google, ses fonctionnalités se sont étendues et permettent aujourd'hui, notamment grâce à Firestore analytics, d'obtenir des données considérables sur sa propre application.

Pour ma part cette fonctionnalité particulière m'a semblé pour le moins capitale, puisque n'importe quel projet que je pourrais construire, équipé des jeux de données de Firestore Analytics, serait assuré de pouvoir bénéficier d'un développement particulièrement agile, flexible et bien guidé.

### **Coordinator layout**

Le Coordinator Layout est une fonctionnalité de Google Materials Design qui permet de combiner des vues ou d'en travailler pour en faire soit des éléments graphiques de haut niveau, soit des conteneurs d'éléments plus petits. L'instructeur a donc expliqué le fonctionnement de cette fonctionnalité avant qu'on ne fasse l'expérience de son implémentation.

Bien réalisé, un tutoriel simple et illustré m'a permis de bien comprendre comment fonctionnait l'implémentation du Coordinator Layout.

## **Constraint Layout**

Il s'agit d'un dérivé du précédent particulièrement puissant, car il permet de choisir les relations et dépendances entre différentes vues et plusieurs objets visuels. Typiquement, il peut être utilisé pour réaliser des aspects graphiques de Parallaxe.

La légèreté et l'efficacité de cet outil m'ont immédiatement convaincu, et ça a été un vrai plaisir que de travailler dessus pendant la session.

## **Démonstrations**

Il a donc fallu implémenter le constraint layout dans l'application de chat en cours de développement depuis la session précédente. Pour ce faire j'ai utilisé pour la première fois Oracle VirtualBox, ce qui ne s'est pas avéré sans difficultés. Il m'a bien fallu quatre ou cinq heures pour comprendre comment me servir de cet outil.

## **VI – Quatrième session**

La suite de la leçon m'a confronté aux librairies : Butterknife, Glide et ImageCircleView. C'est la partie du cours qui est pour moi restée la plus obscure.

### **Présentation**

Les vidéos de présentation de ces librairies, bien plus longues que les autres, étaient aussi bien plus vagues : l'absence d'image de graphique, de représentation ont rendu pour moi les choses très peu concrètes et m'ont poussé à essayer d'aller voir par moi-même à quoi pouvaient ressembler ces librairies.

### **Mise en place**

Par bonheur, le tutoriel de mise en place était encore une fois bien illustré et bien expliqué, et j'ai donc pu le réaliser sans difficulté. Ces tutoriels ont été accompagnés d'une vidéo expliquant comment s'enregistrer sur Firebase.

## **VII – Cinquième session**

La session de travail suivante fut consacrée au développement de l'application de chat en ligne. Ce développement est fait sur quatre axes : une page d'enregistrement, un annuaire de contacts, une page d'ajout de nouveau

contact et enfin, les échanges de messages.

### **Codage des premières parties**

Les premières parties n'ont pas été difficiles à implémenter, puisqu'il s'agissait encore une fois de modifier et compléter un code préexistant fourni par les cours du mooc.

### **Codage de l'échange de messages**

Cette partie a été plus difficile à réaliser : tout d'abord j'ai rencontré de nombreux problèmes techniques avec android studio ; ensuite parce que pour chatter, il faut deux téléphones, et je n'en simulais qu'un seul avec ma machine virtuelle, ce qui m'empêchait de réaliser des tests complets sur l'application, bien que les login et ajouts de contacts fussent fonctionnels.

### **Test**

Cette leçon se terminait sur un test portant sur les parties étudiées. Ce mooc m'étant été affecté en octobre, et se terminant le même mois, les sessions d'examens ont été closes bien trop vite pour que je puisse passer les évaluations. Néanmoins les questionnaires et leurs réponses sont restés disponibles en ligne, ce qui m'a permis d'évaluer mes connaissances, bien que ce soit hors d'un cadre officiel.

## **VIII – Bilan et conclusion**

J'ai donc pu réaliser grâce à ce mooc une première application mobile, basique. Bien que mon avancement ait été difficile et précaire, la matière abordée ne manquait pas d'intérêt. Si les exercices ont pu m'aider, c'est surtout de la partie théorique que j'ai appris, notamment sur l'univers du développement, et des sessions de codage qui m'ont permis de progresser en code objet.

J'ai tout de même le regret que ce mooc, si lourd et avancé, n'ait pas un format plus léger et plus abordable pour les néophytes tels que moi. C'est en effet difficile de s'immerger dans le monde du codage d'applis mobiles, surtout lorsque ce sujet n'a jamais été abordé auparavant.

Par dessus-tout, ce mooc m'a apporté beaucoup, tant du point de vue technique que du point de vue théorique. Bien que pas aussi pédagogique qu'une UE classique, ce fut une expérience qui valait vraiment la peine d'être vécue, que je suis heureux d'avoir pu faire, et que je serai heureux de poursuivre pendant les vacances de Noël.