

The background features a decorative graphic consisting of three concentric blue circles of varying sizes, positioned in the upper right and lower right areas. Two thin, light blue diagonal lines cross the page, one from the top left to the middle right, and another from the top right to the bottom left.

Jeu de l'oie pédagogique

Rapport Projet Tutoré

Rapport technique décrivant le travail réalisé durant les deux semaines consacrées aux projets tutorés. Il contient les objectifs, les étapes de développement et les difficultés rencontrées. Il y aura un mode d'emploi d'utilisation de l'application sur les nouvelles fonctionnalités intégrées.

Adam Mebarki – Florian Van Den Berghe
30/01/2018

Table des matières

Introduction	3
I- Objectifs à réaliser	3
Refonte visuelle	3
Refonte du code source	4
Gestion des questions	4
II- Organisation et répartition des tâches	5
II- Améliorations et difficultés	5
Architecture de l'application.....	5
Interface de l'application	7
Difficultés	9
IV- Avenir du projet	9
Conclusion	10

Introduction

Dans le cadre de notre formation et dans le but de développer nos capacités, il est demandé aux étudiants de travailler sur des projets tutorés. Ces projets se réalisent sur deux semaines en groupes composés de 2 à 4 étudiants. Le but de cet exercice est de pouvoir évaluer notre capacité à travailler dans un temps limité et de répondre aux attentes d'un client tout en planifiant notre travail.

Ce rapport décrira les objectifs sur le projet qui nous a été confié, les améliorations qui lui ont été apporté ainsi que les difficultés rencontrées. Il se terminera par les possibilités d'avenir pour l'application.

I Objectifs à réaliser

Le but du projet tutoré qui nous a été confié était de reprendre un ancien projet et d'être capable d'apporter des améliorations. Elles ont pour but de rendre le projet plus accessible et plus abordable pour des utilisations futures.

Sous la supervision de notre client Mr Lionel Buathier, il a été convenu de travailler sur trois points essentiels :

- ✓ Effectuer une refonte visuelle de l'application pour plus de clarté,
- ✓ Réaliser une refonte du code source pour le rendre plus souple,
- ✓ Intégrer une gestion des questions pédagogiques pour les enseignants.

Refonte visuelle

Les couleurs et la taille de la police d'écriture n'étaient également pas adaptées à tous les élèves et pouvaient être gênantes dans certains cas.

Certains éléments graphiques ne s'adaptaient pas bien à toutes les tailles d'écrans et n'étaient pas forcément simples à comprendre pour tout le monde. Il était donc nécessaire de simplifier l'interface afin d'améliorer l'accessibilité des éléments nécessaires à la mise en route du jeu.

Refonte du code source

Tout comme pour la refonte graphique, celle du code source était nécessaire pour plusieurs raisons. La grande problématique du précédent projet était qu'il était difficile d'apporter des modifications.

La réalisation des classes et des méthodes ne permettaient pas d'inclure ou d'ajouter de nouvelles fonctionnalités sans causer des erreurs. De plus la partie logique du projet, celle non visible par l'utilisateur en générale, était étroitement liée avec la partie graphique. Certaines interactions propres à Android n'étaient pas respectées. Par conséquent des erreurs graphiques internes ou visuelles pouvaient se déclencher.

Par conséquent chaque classe était surchargée et difficile à exploiter. Il était primordial de détruire les anciens fondements pour parvenir à un code source structuré. A la fin, il fallait que cette refonte propose un code source possédant une architecture souple, modifiable et facile à prendre en main.

Gestion des questions

L'une des fonctionnalités qui a été pensée par les précédents développeurs mais qui n'a pas pu être mise en place était la gestion des questions. C'était une partie conséquente à réaliser et qui devait faciliter les enseignants sur :

- ✓ L'importation des questions pour le jeu,
- ✓ La suppression d'une liste ou d'une question particulière,
- ✓ Avoir un aperçu des questions importées,
- ✓ Sauvegarder ces mêmes questions dans une base de données.

L'ancien projet possédait un début d'implémentation d'une base de données locale. Cependant comme pour le code source, il était préférable de partir de zéro dans le but de proposer un modèle adaptable.

II Organisation et répartition des tâches

Il a été décidé dans un premier temps d'appliquer la nouvelle architecture. Une fois cette architecture mise en place, Il a été convenu que l'un s'occuperait de la refonte visuelle pendant que l'autre aurait la tâche de faire la refonte du code source.

Pour éviter d'avoir des différences et des retards sur l'avancement des deux parties, nous avons décidé de scinder le travail en plusieurs parties. Chaque partie comportait un travail graphique et un travail du code source :

- ✓ La page principale et la page gérant les questions,
- ✓ La page des paramètres du jeu,
- ✓ La page contenant le jeu de l'oie,
- ✓ La page gérant les paramètres de l'application.

Les derniers jours, le temps a été essentiellement consacré sur des retouches, de la gestion des erreurs et des messages.

Il était plus facile de faire les modifications pas à pas et sur la même partie, cela permettait d'échanger, de modifier rapidement s'il y avait des différences.

III Améliorations et difficultés

1- Architecture de l'application

La refonte du code source et la création d'une architecture pour le projet s'imposaient. L'ancien projet possédait une structure non exploitable et non compréhensible pour une lecture ou une utilisation future.

La nouvelle architecture est basée sur le design pattern MVVM (Model View ViewModel). C'est une architecture qui a été vu durant notre formation professionnelle. Tout comme le MVP (Model View Presenter), ce design pattern demande beaucoup de connaissances sur Android et sur les outils à utiliser en relation avec ce langage (DataBinding, Injection de dépendances, ...). Il n'était pas intéressant d'appliquer un vrai MVVM, qui aurait pris beaucoup trop de temps à mettre en place, sur ce projet.

Notre architecture se base donc sur les débuts d'une architecture MVVM, elle comprend trois principaux piliers que sont le Model, le ViewModel et l'Activity(View).

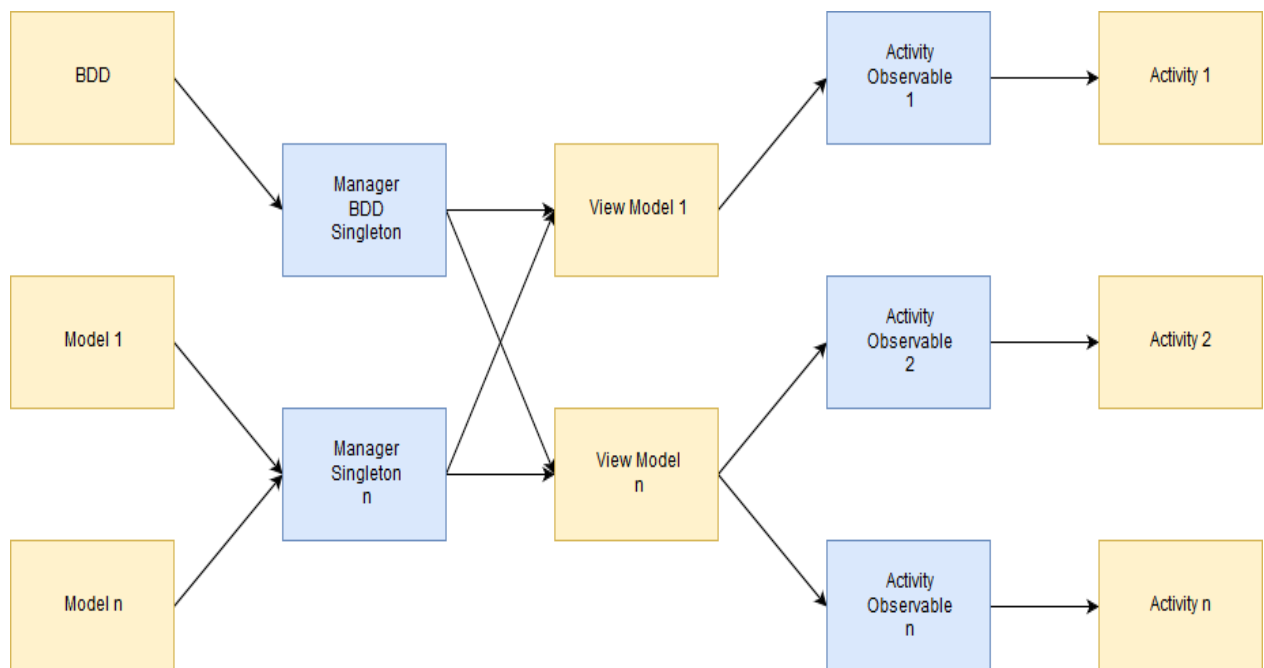


Schéma représentant l'architecture de l'application Jeu de l'oie

Le but de ce design pattern est de faire en sorte qu'aucun des trois piliers puissent « dialoguer », modifier directement les deux autres. Ces interactions sont faites grâce à la création de Managers Singleton et d'interfaces Observable.

Le principe est que chaque Model possède son propre manager qui gère, vérifie et retourne le résultat au ViewModel concerné. L'intérêt du Singleton est d'éviter une duplication ou des créations multiples de Models. Elles pourraient engendrer des erreurs et des conflits durant l'utilisation de l'application.

Par exemple, le GameManager possède un seul Model du jeu. C'est lui et seulement lui qui fait appel aux fonctions créant, modifiant le Model du jeu. Car il possède la seule instance du jeu de toute l'application.

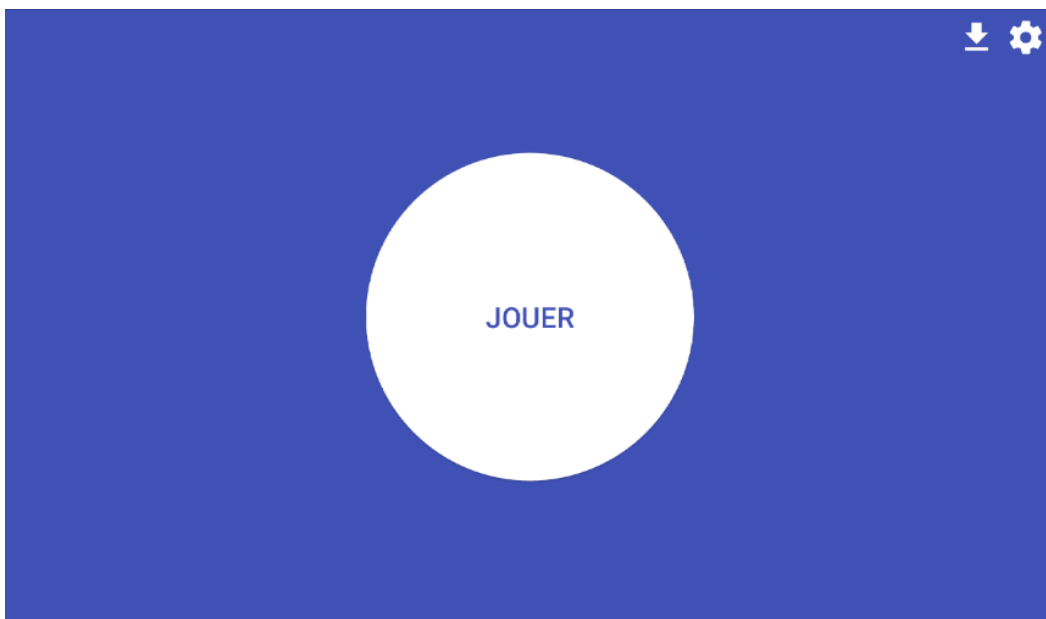
Le ViewModel par l'intermédiaire du Manager possède le nécessaire pour traiter la partie logique de l'application. Ce pilier s'occupe de traiter les données qu'il reçoit, de les vérifier et de les envoyer à la vue. Pour faire cette dernière action, il passe par un Observable. Chaque vue implémente son propre Observable et par conséquent elles sont à l'écoute du moindre changement fait par le ViewModel associé. La View se charge uniquement d'afficher le contenu envoyé par le ViewModel.

Dans ce projet, il y a un cas particulier ; c'est celui concernant la gestion des questions. En effet, la conservation des questions sur l'application était nécessaire et demandée par notre client. Il a fallu créer une base de données(BDD) en interne. Cette réalisation a été effectuée grâce à la librairie d'Android ROOM qui facilite grandement la gestion des bases de données. Il y a donc un Manager Singleton qui relie l'application à la base de données. Ce manager peut être appelé par les autres managers qui ont besoin d'un accès à la BDD.

Pour l'instant seul le QuestionsManager est concerné, mais on peut rajouter un nouveau Manager se chargeant des scores et des joueurs et qui s'occuperait de les stocker dans la BDD grâce au manager relié à la BDD.

L'application est vouée à être utilisée par les enseignants et par les élèves. Il était donc nécessaire de créer une sécurité sur la gestion des questions et sur les paramètres pour éviter d'éventuelles mauvaises manipulations. Une demande de mot de passe de quatre chiffres a donc été rajoutée. C'est une sécurité rudimentaire qui peut être améliorée.

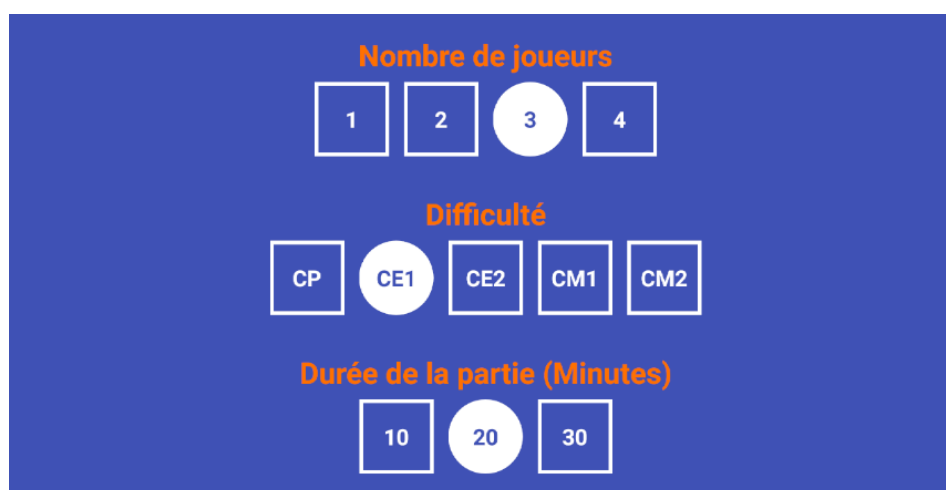
2- Interface de l'application



Lorsque l'utilisateur ouvrira l'application, il tombera sur cette page principale. Le design a été pensé afin de séparer les paramètres et le bouton principal destiné pour le jeu de l'oie et son paramétrage. Les deux boutons situés en haut à droite de l'écran sont destinés à ouvrir, de gauche à droite, la fenêtre pour la gestion des questions et les paramètres globaux de l'application. L'interface a été réalisée de cette manière pour guider plus facilement les utilisateurs dans l'application.



L'une des principales nouveautés de ce projet est la gestion des questions de l'application ou importées via l'option se trouvant en haut à droite de l'écran. Pour ouvrir cette page il est demandé d'entrer un mot de passe. L'accès sera uniquement réservé aux enseignants qui pourront voir la liste des questions présentes, en plus des réponses, et ils auront la possibilité de supprimer une ou plusieurs questions. Pour l'instant l'interface est simple et sobre, et affiche uniquement les informations importantes.



Pour le paramétrage du jeu de l'oie, la précédente version était constituée principalement de pickers défilant les choix possibles. L'ergonomie et la lisibilité n'étaient pas forcément claires et les chances de modifier une valeur, avant de valider, étaient élevées.

La mise en place de boutons au lieu de pickers a donné une meilleure lisibilité et une sûreté au niveau du choix qui sera effectué par l'utilisateur.

3- Difficultés

Une des difficultés rencontrées au cours de ces améliorations s'est présentée avec les « NumberPicker ». Ils permettaient de choisir le thème de la partie, ainsi que le temps de réponse pour chaque joueur. En effet, si aucune valeur n'était sélectionnée, l'application considérait les deux options concernées comme inexistantes. Et elles étaient nécessaires pour le bon déroulement du jeu. Sans ces données, le jeu se fermait subitement sans explications.

Pour contrer ce problème, une valeur par défaut a été ajoutée si l'utilisateur décide de ne pas touché aux « NumberPicker ». Cette valeur correspond bien entendu à la première valeur affichée par chaque « NumberPicker ».

Lors de l'importation d'une liste de questions, il apparaissait que certains caractères ne s'affichaient pas correctement, rendant illisibles le contenu chargé. Le problème était dû à un mauvais encodage des caractères au moment de leur importation dans l'application. La seule solution pour la résoudre était de « forcer » manuellement le système à prendre le bon encodage. Par défaut, le système lisait avec un encodage ne possédant pas les caractères spéciaux ou spécifiques à certaines langues. Il fallait lui indiquer le bon encodage.

Pour finir sur les principales difficultés rencontrées, la séparation des attributs graphiques et des algorithmes a été assez complexe et lente à réaliser. Le précédent projet mixait ces deux parties, et elles formaient donc un ensemble homogène, fonctionnel et complexe. Cependant, sa structure n'était pas adaptée et comportait des parties très instables. Il était important de faire cette séparation pour avoir deux parties bien distinctes. Le temps et la compréhension du code ont été essentielles pour effectuer cette séparation.

IV Avenir du projet

Le projet est fonctionnel et peut être utilisé immédiatement. Il est dorénavant bien plus facile de lui apporter des modifications mineures ou majeures. Cependant certaines fonctionnalités prioritaires n'ont pas pu être incorporées durant ces deux semaines :

- **Couleurs, taille et police des caractères** : l'application doit prendre en compte les personnes ayant des difficultés visuelles. Elle se doit d'être lisible et adaptable selon les personnes. Il est donc essentiel qu'elle offre une

possibilité de modifier la couleur des pages ainsi que la taille et la police des caractères.

- **Modifier la couleur et les textes des cases du jeu** : les cases du jeu de l'oie ne sont pas concernées par la fonctionnalité précédente. Cependant, les couleurs et les textes se doivent d'être modifiables pour plus de lisibilité.
- **Afficher le temps de réponse des joueurs** : Chaque joueur a un temps pour répondre à une question. Ce temps n'est pas affiché et il est plutôt intéressant pour un joueur de savoir le temps qu'il lui reste pour répondre.
- **Animer le déplacement des pions** : les animations ont été altérées lors de la refonte. Pour plus de compréhension, il faudrait corriger les animations.

Outre ces fonctionnalités qui sont prioritaires, plusieurs améliorations sont possibles. Elles offriront plus de profondeur à l'application, ce qui l'a rendra plus efficace en plus d'être attrayante :

- **Améliorer la gestion des questions** : Il serait intéressant d'offrir une option pour trier les questions par thématique. Avoir plus de choix sur les types de fichier à importer. Ne pas uniquement se contenter de fichier CSV, et ouvrir l'importation sur d'autres extensions.
- **Gestion des scores et des parties** : ROOM étant en place, il est bien plus facile de sauvegarder le score des joueurs ainsi que l'état des parties non finies. Il serait intéressant de stocker les données afin de permettre aux joueurs de reprendre des parties et d'avoir un système de progression.
- **Effectuer des études qualitatives et quantitatives** : l'intérêt de ces études est de posséder un retour sur le design et les fonctionnalités proposées. Le but étant de toujours améliorer l'application et de répondre aux attentes des élèves et des enseignants.

CONCLUSION

Ces deux semaines consacrées à ce projet ont été intenses car il y avait un travail conséquent à réaliser. Les trois principaux objectifs ont été atteints. Désormais, l'application est bien plus abordable qu'auparavant que ce soit sur le plan de son utilisation ou sur le plan du développement.

Ce travail nous a offert la chance d'appliquer des connaissances acquises récemment, comme par exemple l'architecture MVVM, les Managers et les Observables, en les utilisant à notre façon.

Non seulement on a pu mettre en pratique ce que l'on savait mais nous avons aussi pu tester de nouveaux outils comme la librairie ROOM, les « NumberPicker », ... afin d'améliorer nos capacités en tant que développeur.

La gestion du temps et la mise en place d'un planning ont été importantes afin de respecter les consignes et les souhaits de notre client. Nous avons su nous adapter aux demandes et les appliquer rapidement et efficacement sur le projet.

Nous sommes satisfaits du travail réalisé sur le jeu de l'oie pédagogique. Avoir atteint les trois objectifs est bien entendu très satisfaisant. Ces deux semaines ont été riches et nous ont gratifié de nouvelles connaissances et aptitudes pour développer efficacement et durablement sur le long terme.