

# Projet Android : NougatQuizz

2017 - 2018

## Présentation du projet

### Réalisé par :

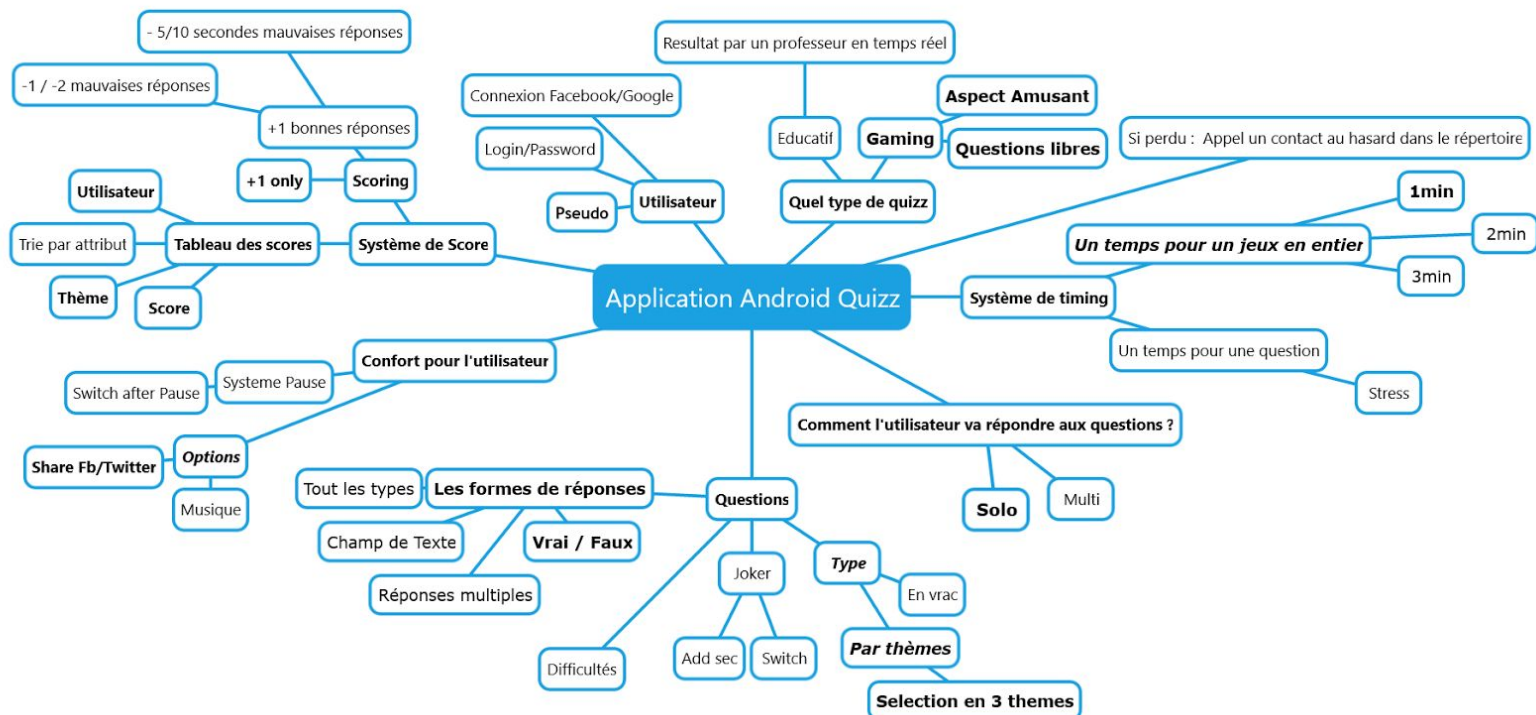
BRICENO-AGUILERA David  
CHOAIN Pierre-Alexandre  
COURMONT Guillaume  
MROCZKOWSKI Hugo



# Introduction



Notre consigne était simple et large : réaliser une application de quizz. Il nous est paru évident de faire un brainstorming pour définir plus précisément ce sujet et choisir une orientation à notre application. Nous avons alors envisagé plusieurs scénarios d'usage. De multiples types d'application tels que des outils pédagogiques, des jeux, ... Nous avons envisagé de multiples système de score/notation. Voici les idées qui en sont ressorties :



Nous avons alors choisi de nous orienter vers une gamification de l'application. En d'autres termes, nous avons décidé de faire un petit jeu mobile. Pourquoi ? C'est très simple, aujourd'hui la plupart des applications que nous, étudiants, avons sur nos smartphones sont des mini-jeux. Il nous semblait pertinent de nous orienter de la sorte et de cibler des utilisateurs de notre tranche d'âge car c'est une cible que nous connaissons bien.

Nous avons ensuite voulu un jeu très rapide, idéal pour les trajets en bus, métro ou les pauses entre les cours.

Pour cette raison, nous avons choisi de faire des parties chronométrées, avec un temps relativement court. Nous avons décidé que les parties dureront 1 minute. Nous voulions également que les questions soient rapides pour donner un sentiment de stress à l'utilisateur. C'est quelque chose de commun aux jeux à succès, par exemple Flappy Bird. Pour cela, nous avons choisi des questions "vrai ou faux" selon un thème choisi par l'utilisateur parmi 3 thèmes aléatoires. Ceci va également permettre d'apporter aux utilisateurs un côté ludique, ils vont apprendre des choses en ayant face à eux des questions qui paraissent compliquées mais dotées d'une réponse simple.

Pour motiver les joueurs à donner le meilleur d'eux même, il fallait un système de score. Ce dernier correspondra au nombre de questions répondues correctement durant le temps imparti.

Pour se comparer aux autres, nous avons pensé à intégrer un tableau général des scores sur lequel figurera le pseudo des joueurs, le thème joué et le score obtenu.

Nous avons décidé de mettre un simple pseudo pour les informations de l'utilisateur. Il n'y a pas besoin d'un système de compte (login/mot de passe) pour notre jeu mais, uniquement un pseudo. C'est d'ailleurs beaucoup plus utilisé pour l'aspect gaming, qui fait d'ailleurs un peu pensé au système de score utilisé sur le Pinball 3D de windows XP.

Enfin, nous envisageons d'ajouter un partage du score sur les réseaux sociaux si le temps nous le permet.

En globalité, notre application a été réfléchi et travaillé. Nos choix d'apporter une dimension ludique, de la simplicité, de l'intuitivité et de la rapidité n'ont pas été laissés au hasard.

## I) Fonctionnalités

Nous allons maintenant présenter plus en détail l'ensemble des fonctionnalités de NougatQuizz.

### Le Quizz

Une fois le joueur prêt à réaliser le quizz, il lui suffit de lancer une partie via le bouton principal « Touchez pour jouer » qui est directement sur l'écran d'ouverture. Nous avons voulu une bonne ergonomie et une expérience utilisateur très simple.

Le quizz est une série de questions "vrai ou faux" auxquelles on répond en appuyant sur un bouton.

Le but du jeu est de répondre à un maximum de questions pendant le temps imparti.

### Le temps

Nous avons mis en place un compte à rebours permettant aux joueurs de terminer le Quizz. Une barre de progression va indiquer le temps restant aux questions. L'intérêt de ce système est d'avoir des parties assez rapides. Nous allons faire défiler les différentes questions en essayant de minimiser les latences pour éviter l'impatience de chargement. Ce chronomètre visible va permettre à l'utilisateur de se concentrer en augmentant son stress. Le but du jeu est donc de répondre à un maximum de questions pendant le temps imparti.

### Un système de score

Les bonnes réponses sont évidemment des points qui sont attribués. Voici probablement un des sujets le plus délicat sur lequel nous avons travaillé. Il fallait un moyen de motiver le joueur, l'inciter à se surpasser et donc à continuer à utiliser l'application. Ce score va permettre à l'utilisateur de se comparer à ses amis, et donc d'inciter d'autres utilisateurs à télécharger NougatQuizz.

### Le tableau des scores + le partage

A la fin de la partie, l'utilisateur est invité à saisir son pseudonyme pour envoyer son score sur le serveur de jeu et ainsi le partager avec toute la communauté. C'est un peu le principe des applications telles que Waze où les meilleurs utilisateurs sont affichés chez tout le monde.

Sur l'écran principal de l'application, il y a un bouton « Scores » représenté par une icône de trophée permettant d'afficher le tableau des scores. Sur ce dernier, nous allons trouver le pseudo, le thème joué et le score obtenu par les 5 meilleurs joueurs de notre application.

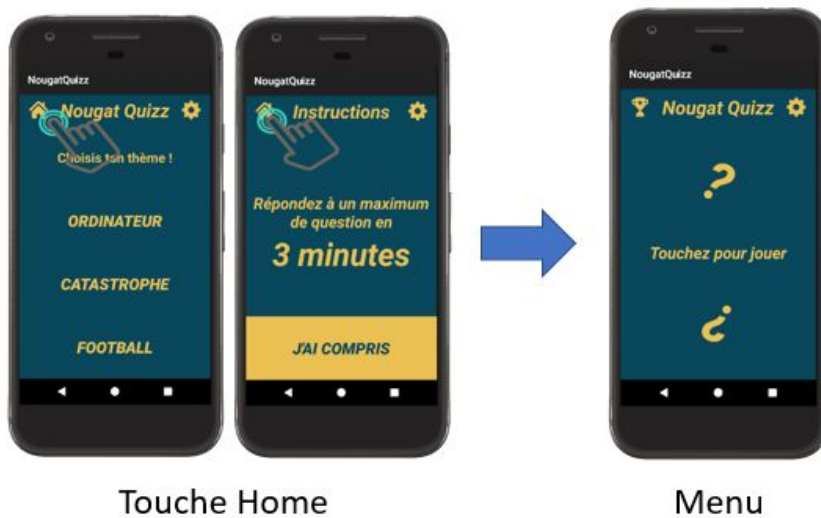
Si le temps nous le permet, nous ajouterons des boutons de partage du score sur les réseaux sociaux (Twitter/Facebook).

## STORY BOARD

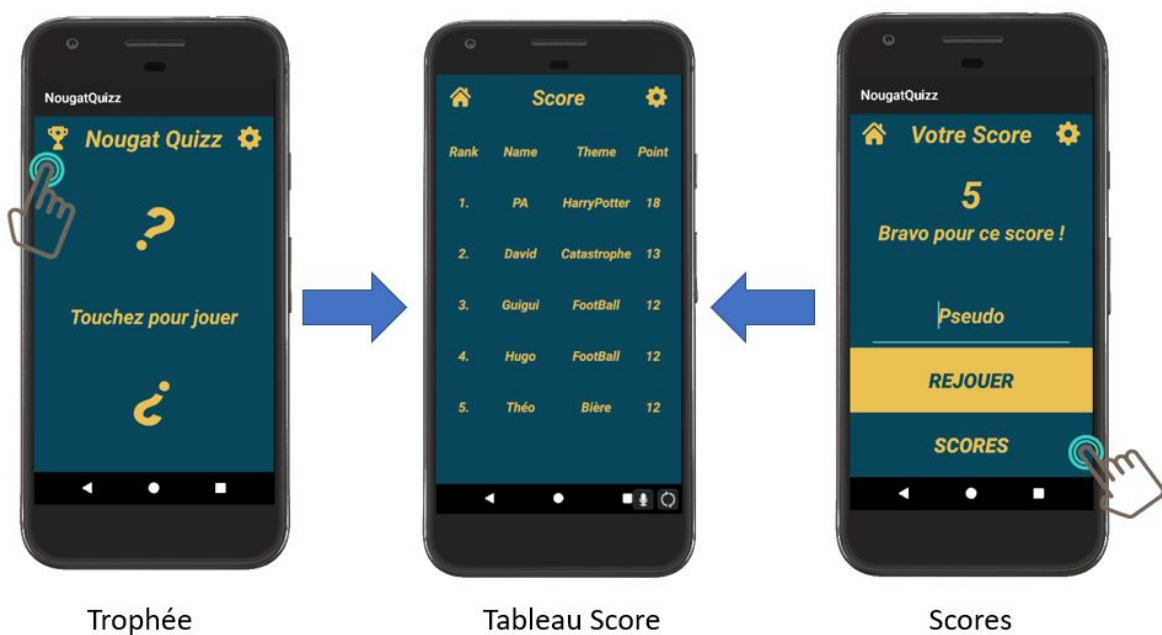
Scénario principal d'utilisation



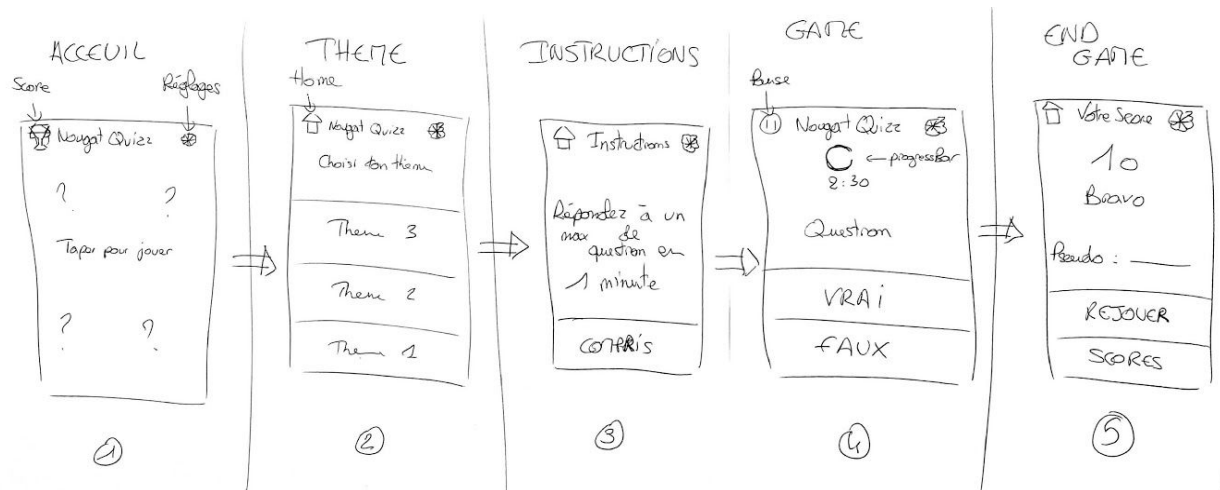
Retour au menu



Aller voir le tableau des scores :



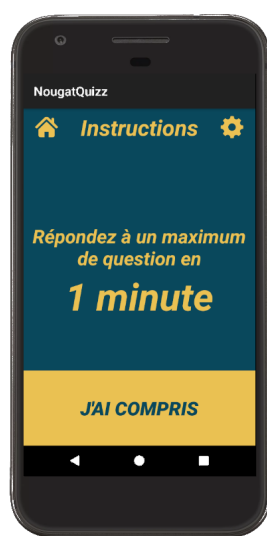
## II) Maquette de l'application



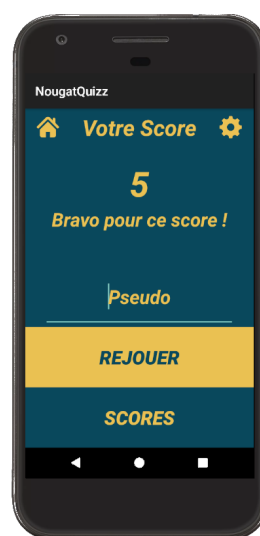
A l'ouverture de l'application nous arrivons sur la page d'accueil(1), il suffit de cliquer pour démarrer le jeu, lors du premier démarrage de l'application nous avons les instructions qui s'affichent(2).



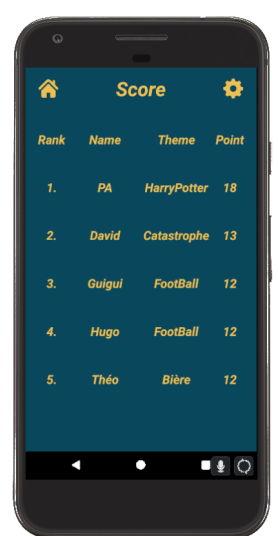
(1)



(2)



(3)



(4)

L'icône Maison permet de retourner à l'écran d'accueil (1).

L'icône Score (3) ouvre la page du tableau de scores

On aura aussi une page de fin de partie(4), avec visualisation du score et ajout d'un nom pour sauvegarder le score dans le tableau.

Puis l'écran de jeu avec les différentes questions au fur et à mesure des réponses.

L'icône en haut à gauche permet de faire une pause durant le quizz. C'est une option qui apporte un soulagement pour l'utilisateur.



### III) Spécifications techniques

Lors de notre brainstorming, nous avons eu de nombreuses idées quant à l'expérience utilisateur. Nous avons retenu l'idée d'avoir une liste de questions qui puisse être mise à jour.

De plus, nous avons fait le choix de stocker un tableau des scores pour l'ensemble de nos joueurs. Il nous fallait donc un serveur. Ce serveur devra s'occuper de stocker et retourner les questions et les scores. Pour y connecter l'application, il embarquera une API RESTful.

D'un point de vue technologique, nous avons décidé d'utiliser NodeJS car c'est un outil que nous connaissons bien, avec lequel nous sommes à l'aise et qui dispose d'un très grand écosystème de packages.

#### Gestion des questions

Pour gérer les questions, nous allons utiliser un fichier JSON qui sera stocké sur le serveur.

Nous avons fait le choix de considérer les questions comme des objets (pour pouvoir y ajouter facilement des champs dans le futur, par exemple des indices ou des difficultés de question variables). Ces objets questions sont regroupés dans des tableaux triés par thème.

Ainsi, nous pouvons très facilement parcourir nos thèmes pour en obtenir des questions.

Voici à quoi va ressembler notre fichier JSON :

```

{
  "theme1": [
    {
      "question": "ici une question",
      "answer": false
    },
    {
      "question": "ici une autre question",
      "answer": false
    }
  ],
  "theme2": [
    {...}
  ]
}

```

## Gestion des scores

Comme expliqué plus haut, nous allons également stocker des scores sur le serveur. Pour cela, à la fin d'une partie, nous allons demander à notre joueur de saisir un pseudo. Son résultat sera alors ajouté à un fichier JSON trié comme celui-ci :

```

{
  "100": [
    {
      "name": "Joueur1",
      "theme": "themejoué"
    },
    {
      "name": "Joueur2",
      "theme": "themejoué"
    }
  ],
  "80": [ ... ]
}

```

## Connexion au serveur

Pour récupérer les données de notre serveur, il va falloir utiliser une API REST. Pour effectuer les requêtes, nous avons utilisé la librairie Retrofit.

## Requêtes disponibles

Pour la réalisation de notre API, nous allons utiliser le paquet Express pour NodeJS. Celui-ci va nous permettre d'écouter et de répondre sur les URL suivants :



- **GET** .../scores/:count

Retourne les :count premiers joueurs du classement

Format de la réponse:

```
▼ scores:
  ▼ 0:
    name: "Test1"
    theme: "Theme1"
    score: "10"
  ▼ 1:
    name: "Test2"
    theme: "Theme1"
    score: "10"
  ▼ 2:
    name: "Test3"
    theme: "Theme2"
    score: "10"
```

- **POST** .../scores/:name/:thème/:score/:count

Permet d'ajouter un score donné pour un utilisateur donné.

Retourne les :count meilleurs scores enregistrés

Format de la réponse:

```
▼ scores:
  ▼ 0:
    name: "Test1"
    theme: "Theme1"
    score: "10"
  ▼ 1:
    name: "Test2"
    theme: "Theme1"
    score: "10"
  ▼ 2:
    name: "Test3"
    theme: "Theme2"
    score: "10"
```

- **GET** .../randomthemes/:count

Retourne une liste de :count thèmes aléatoires

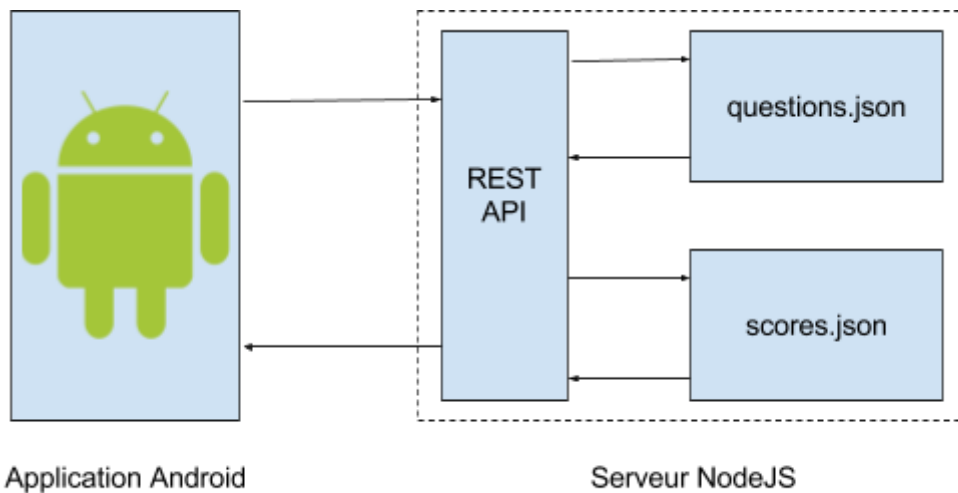
Format de la réponse:

```
▼ themes:
  0: "Ordinateur"
  1: "USA"
  2: "Vin"
```

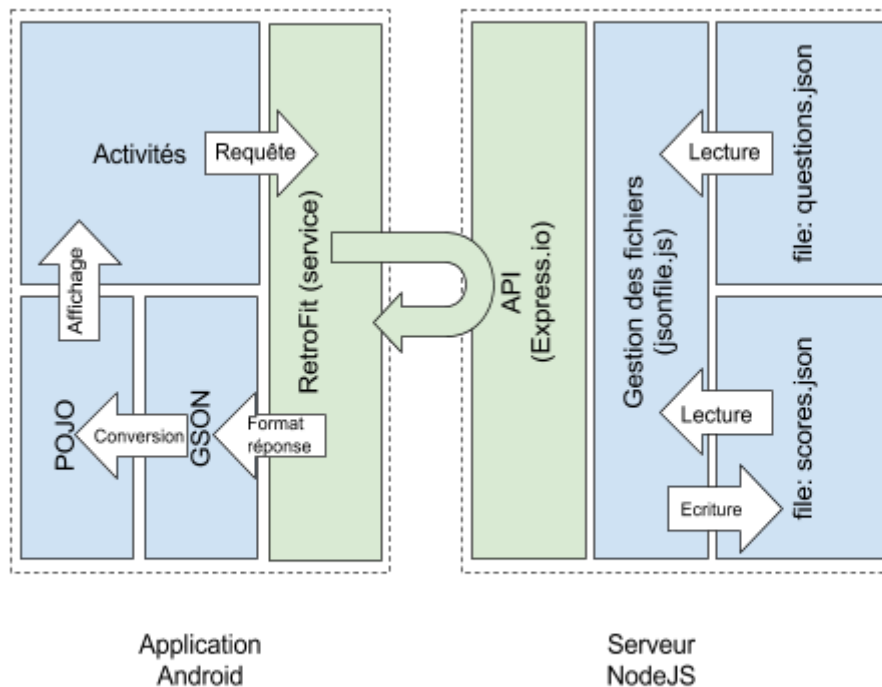
- **GET** .../randomquestions/:thème/:count  
Retourne une liste de *:count* questions aléatoires dans le thème *:thème*  
format de la réponse:

```
▼ questions:
  ▼ 0:
    question: "Tom Elvis Jedusor n'est autre que le fameux Voldemort."
    answer: true
  ▼ 1:
    question: "Dolores Ombrage interdit à ses élèves de pratiquer la magie."
    answer: true
```

### Schéma général de l'application



## Schéma technique application



## IV) Organisation du projet

### Outils utilisés

- **Facebook Messenger** : Pour la messagerie instantanée entre les différents membres du groupe.



- **Git** : Comme logiciel de gestion de versions décentralisés des fichiers de l'application.



- **Google Drive** et **GSuite** : Pour le partage de documents texte et de la présentation orale pour la soutenance du projet.



## Roadmap

[illegible]