

FindX

- quizz de mathématique pour le 5eme -

Ana Dima, 1241F

Cristina Dumitrescu, 1241F

Ana Pirvulescu, 1241F

Prof.: Dragos Stoica

Contenu:

- 1. C`est quoi FindX?**
- 2. Implementation**
- 3. Conclusions**
- 4. Le futur**

C'est quoi FindX?

FindX est une application pour les étudiants de 5eme qui teste leurs connaissances de mathématique. L'application a des questions avec 3 variantes de réponse et au final, l'utilisateur reçoit un nombre de étoiles correspondant au nombre de réponses correctes.

Le but d'application est d'aider les élèves à apprendre la mathématique dans une manière amusante pour eux.

C'est quoi FindX?

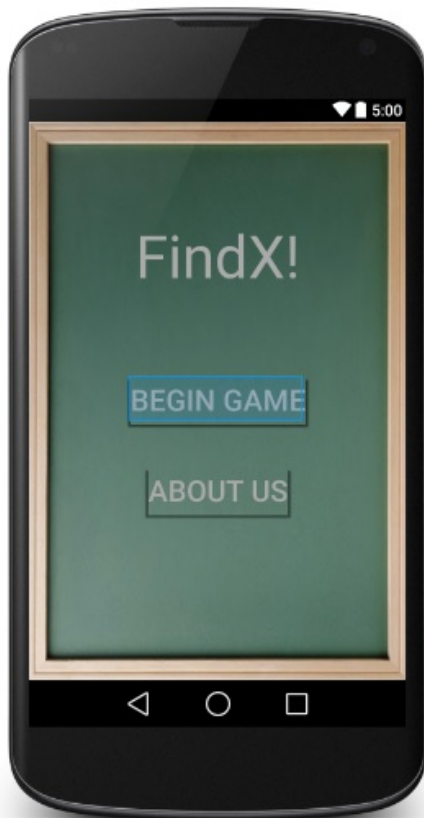
Notre projet consiste en 4 etapes.

1. La Classe StartActivity

C'est le premier activity de l'application. Elle a 3 attributs:

- Button butStart - le commencement du jeu
- Button butAbout - des informations sur le jeu et les createurs
- TextView text1 - afficher le nom de l'application

Les deux methodes qu'elle utilise, onCreate() et onView() definissent les fonctionalites du buttons: le passage de l'activite en cours de deroulement, a l'activite suivante, le quiz.



2. La classe Question.

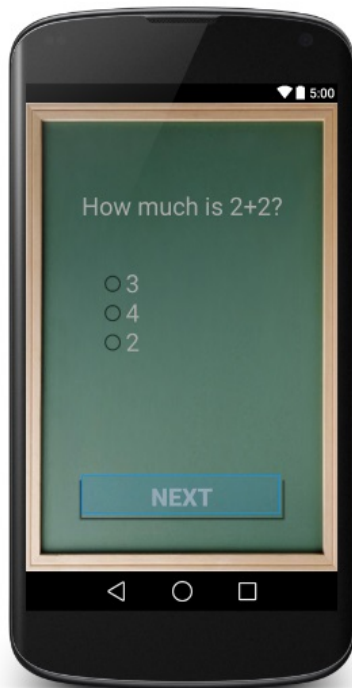
C'est la classe avec laquelle on crée les questions.

On utilise une méthode constructeur et des méthodes get et set pour initialiser les attributs de la classe.

La classe a 6 attributs:

- L'id: pour différencier les questions entre eux
- String Question: c'est la question
- 3 attributs STRING pour les réponses
- un attribut STRING pour la réponse qui est vraie

Pour chaque attribut on a utilisé une méthode set et get qu'on va utiliser dans les autres classes implémentées.



2. Database Helper

La Classe DBHelper est la classe qui crée la base des données avec les questions, et avec laquelle on introduit des nouvelles questions.

On a utilisé des constantes String pour définir les elements de la base des donees:

- String KEY_ID
- String KEY_QUES
- String KEY_ANSWER
- String KEY_OPTA
- String KEY_OPTB
- String KEY_OPTC

Et puis on a utilisé un attribut de type SQLiteDatabase, une classe qui a des méthodes pour créer et gérer les elements d'une base de données. On a besoin de cet attribut dans la methode onCreate(), qui va créer la base de données.

Avec la méthode addQuestions() on ajoute des questions a la base de données.

3. Le Layout

Le layout est simple. On a utilisé un RelativeLayout pour distribuer les elements qu'on a utilisé. Un TextView qui reçoit les questions, trois RadioButton pour les variantes de réponse et un bouton "Next" qui est appuyé après la sélection de la variante qu'on croit être correcte.

4. MainActivity

On l'utilise pour prendre les questions et les réponses de la base de données et les mettre dans le layout de l'application. En plus, le score est au début égal à zéro.

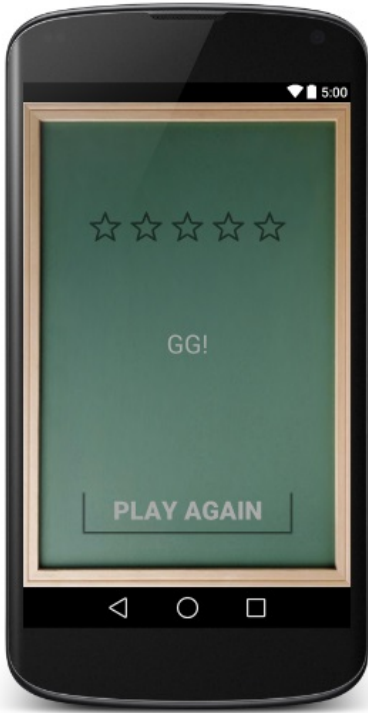
On a utilisé une classe QuizActivity qui est une extension de la classe de base Activity. Elle a les attributs suivants:

- List<Question> quesList - l'array qui va recevoir les éléments de la base de données créée avec SQLite
- int score=0;
- int qid=0 - le compteur utilisé pour parcourir quesList
- Question currentQ - la question que l'utilisateur a devant soi en ce moment, du type Objet, la classe qu'on a déjà définie
- TextView txtQuestion - l'élément du layout qui va recevoir la question à laquelle on doit répondre maintenant à l'aide de l'objet Question
- RadioButton rda, rdb, rdc - ils vont prendre les valeurs de l'objet Question
- Button btnNext;

La méthode onCreate() qui prend les informations de la base de données et les met dans une liste. Avec la méthode setQuestionView() on crée le layout avec les questions et les réponses que les utilisateurs voient devant soi. La méthode utilise les méthodes getQuestion(), getOPTA(), getOPTB(), getOPTC() de l'objet Question pour actualiser le layout.

5. ResultActivity xml

Le xml qui va afficher le score final. Il est sous la forme des étoiles. On a utilisé un TextView qui va prendre le texte de la classe ResultActivity et un RatingBar pour les étoiles.



6. La Classe ResultActivity

Elle prend le résultat de la QuizActivty classe et affiche un texte selon le score obtenu.

On crée le RatingBar avec elle et un TextView. Selon le résultat, on a plusieurs réponses:

- "Oopsie! Better Luck Next Time!"
- "Hmmm.. Someone's been reading a lot of trivia"
- "Who are you? A trivia wizard???"

Pour sélectionner les réponses on utilise une condition de type switch().

Conclusions

Pour la création de ce projet, la classe qui nous a aidé beaucoup est SQLite. Sans elle, on devrait faire une activity pour chaque question, mais avec elle, on a pu prendre les éléments de la base de données pour updatier les fichiers XML.

Initialement, le concept de l'application a été différent. On pense à faire une application qui teste la vitesse de réaction des étudiants et leur connaissance de mathématiques divisées sur trois niveaux:

- premier année de l'école primaire
- 2eme et 3eme année
- 4eme année

Mais à cause du temps mal géré, on n'a pas pu mettre en application le concept initial et maintenant, le quiz a un seul niveau de difficulté. sans le classement final et des prix spéciaux pour les scores très, très bons.

Futur

FindX a été initialement comme une application qui teste la vitesse de réaction des étudiants. On n'a pas implémenté cette fonction, ni les 3 niveaux différents de difficulté et alors, dans le futur, l'application pourrait se développer dans ce sens-là. En plus, elle va bénéficier d'un dictionnaire avec les plus importantes opérations mathématiques.

Et alors, au niveau de design, voilà la direction dans laquelle l'application va évoluer:

