

# **Developer Documentation**

---

## **Group 7 - Drive'n'Quiz™**

---

**29 - 05 - 2015**

---

## Contents

<b>1. THE APPS BUILD PROCESS.....</b>	<b>2</b>
1.1. DEVELOPING ENVIRONMENT AND TOOLS.....	2
1.2. PROJECT ARCHITECTURE .....	2
1.3. CLASSES DEVELOPMENT .....	2
<b>2. APPS MAJOR COMPONENTS .....</b>	<b>3</b>
<b>3. DESIGN DECISIONS.....</b>	<b>4</b>
<b>4. PROJECT UML CHART .....</b>	<b>5</b>
<b>5. PROJECT FLOWCHART.....</b>	<b>6</b>
<b>6. TESTING .....</b>	<b>6</b>
<b>7. EXTERNAL DEPENDENCIES .....</b>	<b>7</b>

# **1. The Apps Build Process**

## **1.1. Developing environment and tools**

The following tools were used to develop this project:

- Android Studio – IDE specially developed by Google Inc. for creating applications that are meant for Android mobile platform
- AGA (Automotive Grade Android) – on-board automotive computer simulator
- GitHub – web based tool for code sharing and synchronizing between group members

The project was developed in the Java programming language and in XML, as a standard Android Studio language for the application components.

## **1.2. Project architecture**

After the environment build-up, the design decisions are made, making sure that the user requirements will be met in the final product. Main ideas for the classes as well as the connection between them is being evaluated and later on used for the development process. In addition, for the successful build-up of the project and brainstorming on the important features of the program is required.

## **1.3. Classes development**

After the app architecture was agreed upon, we decided to start with the most important activity, the starting activity (StartmenuAct), which would introduce the player to the app and allow interaction with the program. This activity was connected to a layout specifically designed for it. Also, after this activity was done, the android manifest was updated so this activity became the main activity. After this, we designed two other activities (MainActivity and TOActivity), which will give the user different options for how to use the app. The connection between activities was made and the manifest was

updated. Next, we implemented the AGA and decided how it will affect the app. Then, the database was designed and connected to the activities. Finally, we designed the ScoreActivity, which will present the result to the user.

## **2. Apps Major Components**

The app consists of three different modes, whose functionality depends on AGA (on the distraction level variable). The app is presented by three categories of functional screens. The first one is the start screen which changes according to the signal received from AGA. The button of this screen changes according to the signal level and its output designated the version of the next screen. The second category of presented screens is only displayed in two of the first screen cases (Yellow for TextToSpeech and Green for TextOnly). The TextToSpeech screen consists of three buttons (Yes, No and Repeat). The TextOnly is composed of two buttons (Yes and No) and one text view field. The third category of screens is presenting the score and includes a Back button that enables the user to return to the first screen manually.

The fun facts presented in the TextToSpeech option and the questions presented in the app are part of an integrated database.

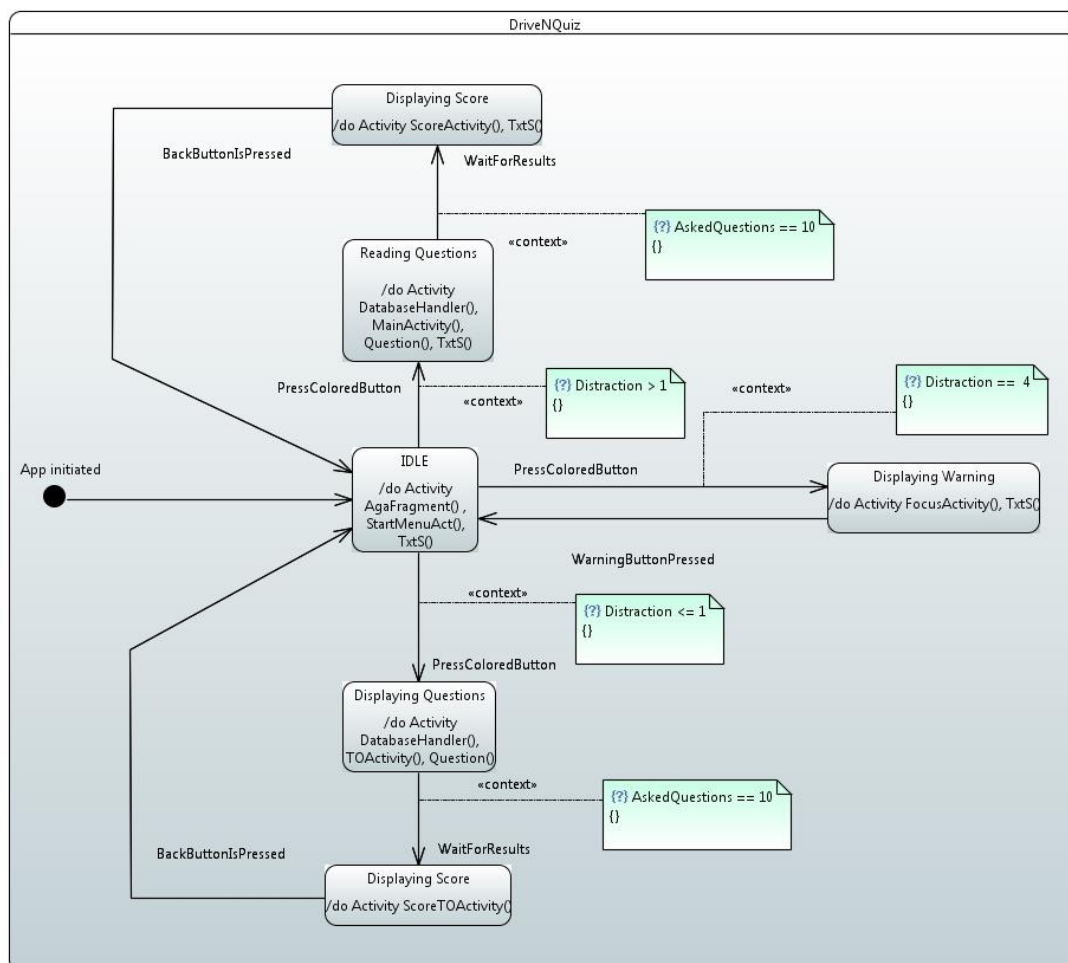
### 3. Design Decisions

The overall design of the App is focused on the safety aspect of the driving experience. During the development we kept into consideration the safety lecture held by Claudia Wege, from the Volvo Group, as a part of the course that this project is part of. We reviewed all the presented risks and keep in mind all of them. We paid special attention to the risk described in the table below.

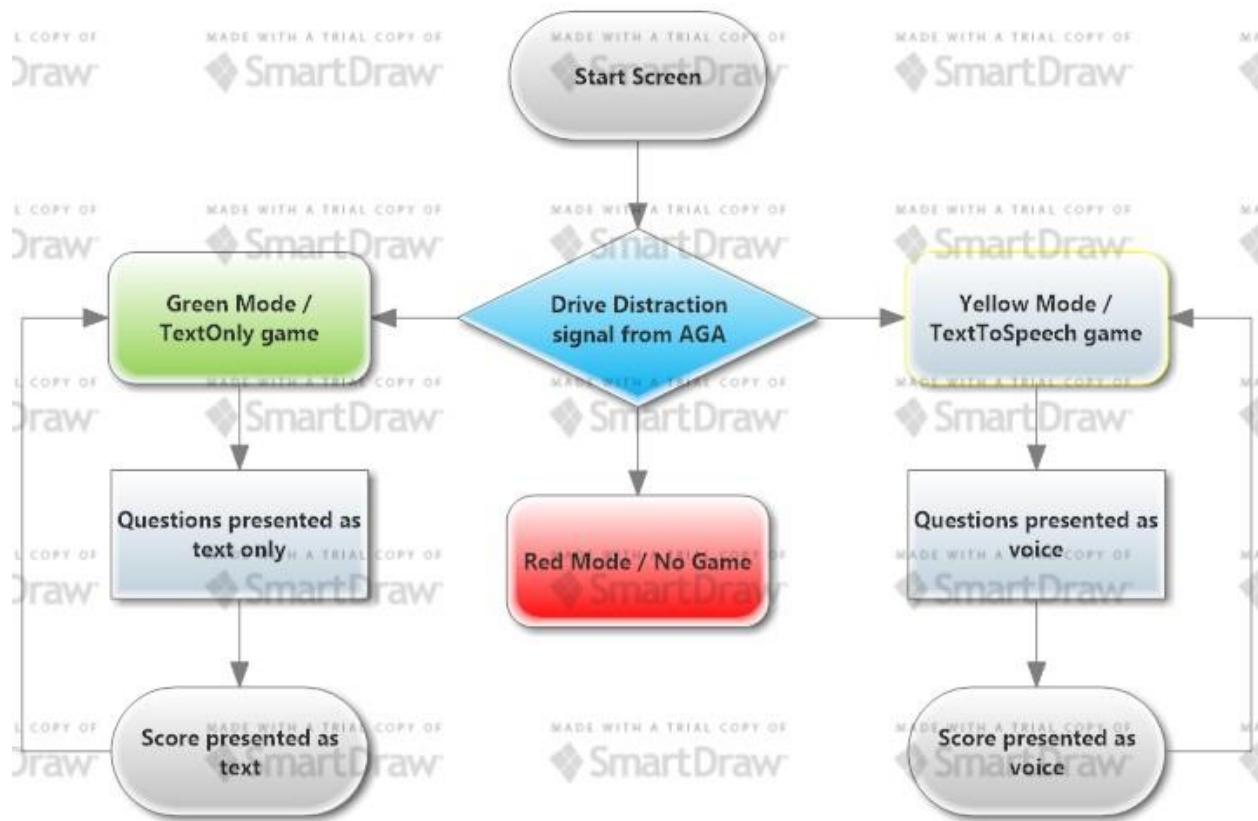
Risk name	Crash risk level	In App solution
<b>Reading various text on the mobile device</b>	High	We introduced a TextToSpeech game option so the user doesn't have to visually interact with the device
<b>Daydreaming during long routes</b>	High	One of the main purposes of the quiz game is to entertain the driver during long courses. Additionally, we have implemented a 'play the game' prompt message if the driver distraction is unchanged for a longer period.
<b>Talk or listen to hands free phone</b>	Low	Hands-free interaction with a mobile device is a low risk action. This was our guidance for developing the TextToSpeech option.
<b>Interact with other vehicle occupants</b>	Low	The quiz game can be a good substitute for a driver/co-passenger interaction and keeps the driver alert
<b>Manual interaction with device and reading</b>	High	To prevent additional focusing on the device we designed large and clear to read button with sharp color contrast

## 4. Project UML Chart

When the user initiates the application, the IDLE state is entered by default. From that point the app can transfer to three other states depending on the Driver Distraction level. If the distraction level is less or equal to one, the state of the program changes to Displaying Question. In other words, questions are only shown on the screen. After the tenth question the state changes to Displaying Score (correct answers are shown to the user) and then the game returns to IDLE. If the Distraction Level is larger than one, the state is changed from IDLE to Reading Question. In that phase the question are read to the user with the assistance of Text to Speech. When the tenth question is asked the program enters the state Displaying Score and the results are read to the user. When text to speech finishes reading, the game returns to IDLE. If the Distraction Level is equal to four, the Display Warning state is initiated where the user is not allowed to play the game, because their distraction level is very high. The application also plays a warning message to the user.



## 5. Project Flowchart



## 6. Testing

Throughout the development process of the App testing was done continuously by all members. With every functional change, testing was done using the Android Studio build in emulator. Based on the subjective impressions, the result of the testing was discussed at the group meetings and additional ideas were developed and implemented.

In addition to the manual testing conducted on a daily basis, automated testing was implemented for testing the beta version and for the final submission. For automated testing Sikuli was used. A file including the Sikuli log for the final version is included in the final submission.

## 7. External Dependencies

For proper functionality and full utilization capture of the app, it is essential the app to be connected to a vehicles computer that will send a signal with the actual driver distraction level. For the project development, testing and presentation AGA is used to simulate the vehicles on-board computer.

From the AGA simulator, we are retrieving the driver distraction variable. Based on the received signal the game is entering three different modes. Two of them are leading to a game mode. The first one, or the 'Green mode', enables the user to play the game as a text only option. The second one, or the 'Yellow mode', enables the user to play the game as a voice presentation of the questions option. The third option, or the 'Red one', is the no game options. All of these is highly dependent of proper functionality of AGA. The use of it is related to the safety requirements of the developed app. More details of the AGA dependence to the application can be seen on the project UML chart and the flowchart presented above.