

Words Quiz

21st February 2018

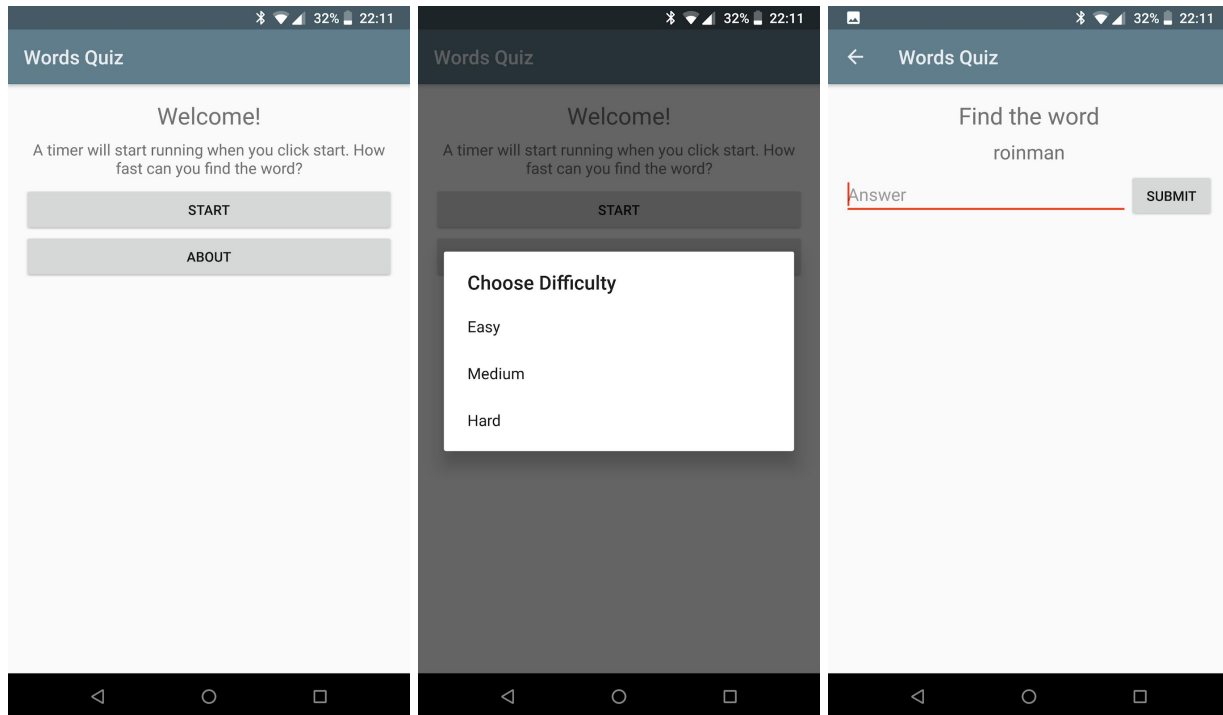
Kostas Andrianos | CS151018

OVERVIEW

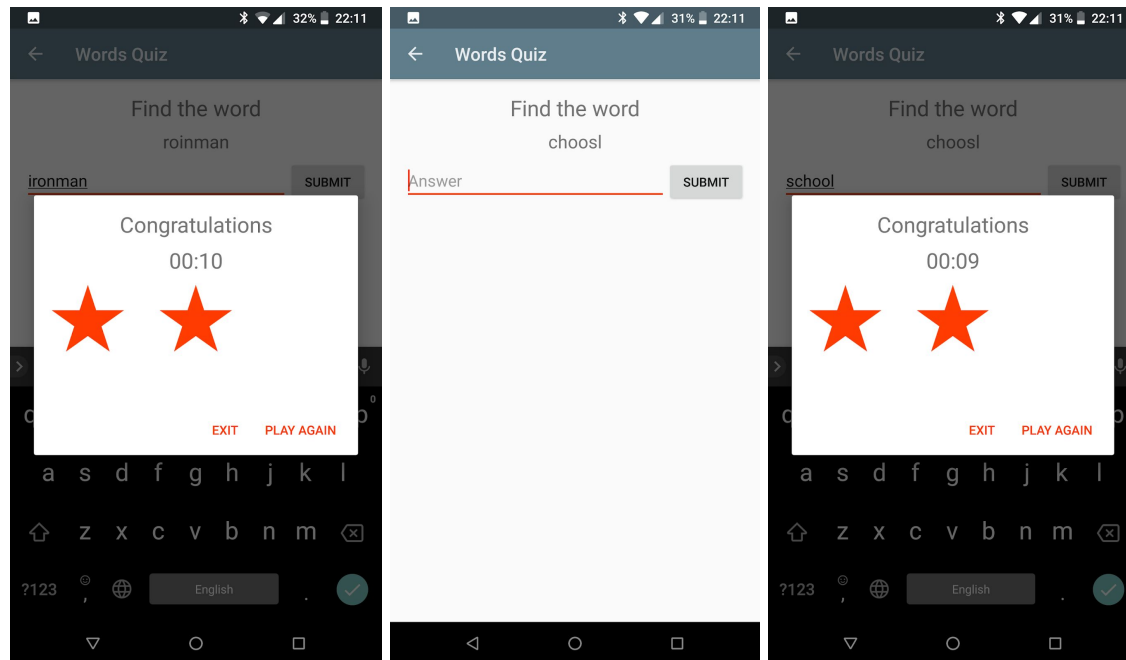
This project is a simple game with shuffled words that aims to be challenging but fun. There are three levels of difficulty. There is a score system ranging from one to three stars based on the time it takes the player to find the shuffled word. Built with Material Design in mind, strictly following the design standards provided.

Typical playing scenario

The player is greeted with a screen where they can choose to play or view some details about the programmer. Choosing “Start”, pops up a dialog where the player needs to choose the level of difficulty. After that, the game starts. Below the prompt to “Find the word”, there is a random word from a pool of words of that difficulty, shuffled.



The player has to answer correctly and click submit. If the answer is correct, they are greeted with the following screen, showing the score, the time and giving them a choice to play again or exit to the main menu. If they choose “Play again”, then the activity is re-created and the player has a new word to find.



After clicking “Exit”, the player is presented with the main menu again. Choosing “About”, takes the player to the About activity where a picture of the programmer is shown, along with some information about him and a link to his website.

Nielsen rules.

Visibility of system status [x]

The system always keeps users informed about what is going on, through appropriate feedback within reasonable time.

This rule does not apply as the game is very simple.

Match between system and the real world [✓]

The system speaks the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follows real-world conventions, making information appear in a natural and logical order.

User control and freedom [✓]

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.

Consistency and standards [✓]

Users do not have to wonder whether different words, situations, or actions mean the same thing. Follows platform conventions.

Error prevention [x]

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Eliminated error-prone conditions, checking for them and presenting the users with a confirmation option before they commit to the action. **There were no such scenarios to begin with or those where handled in the back-end, hence these are not handled in any other way**

Recognition rather than recall [✓]

Minimizes the user's memory load by making objects, actions, and options visible. The user does not have to remember information from one part of the dialogue to another. Instructions for use of the system are visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use [x]

Accelerators may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. **Since this is a game, this rule does not apply and therefore, is not met.**

Aesthetic and minimalist design [✓]

Dialogues do not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Helps users recognize, diagnose, and recover from errors [✓]

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation [x]

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

This rule does not apply as the game is very simple.