OPSC7311

Duran Moodley 13016335

  Assignment 1

Portfolio of evidence

Contents

[Introduction 2](#_Toc452913112)

[Program Specification 2](#_Toc452913113)

[Game Description 2](#_Toc452913114)

[Programming Tools Used 2](#_Toc452913115)

[System Requirements 2](#_Toc452913116)

[Flow Chart 4](#_Toc452913117)

[UML Diagrams 8](#_Toc452913118)

[Class Diagrams 8](#_Toc452913119)

[Use Case Diagram 11](#_Toc452913120)

[Database Design 11](#_Toc452913121)

[ERD 11](#_Toc452913122)

[Table Structure 12](#_Toc452913123)

[Help File 12](#_Toc452913124)

[Screen shots/Test Data 13](#_Toc452913125)

[Lecturer Feedback & Corrections Made 18](#_Toc452913126)

[Enhancements 19](#_Toc452913127)

[Conclusion 19](#_Toc452913128)

# Introduction

This game is focused specifically on children. This app can be used to improve their vocabulary and help them learn. In addition, this game is not only educational but it provides some interactive game play that will appeal to children and it has a challenging gaming experience.

# Program Specification

## Game Description

“Word Maker” is a word game that tests your word making skills. The aim of the game is to make a word using letters that will fall/trickle from the top of the screen. Random letters will be generated. The user can select a letter by tapping it before it reaches the end of the screen. The player cannot deselect a letter. Selecting the letter will save it in the word making section of the screen. Once the player has created a word, clicking on the add word button will add this word to the word list. The player will have a restricted amount of time to make as many words as possible. After the time has ended, the word will be validated to see if it exists. Every word that the user gets correct, he/she will get 20 points per word.

## Programming Tools Used

* Use an API to check if the word exists
* Read and write to a text file (player words created)
* Make use of the animation class
* Using Threads and Async Task
* Incorporating Fragments in an activity
* Making use of classes such as Factory class
* Making use of the XML component in android(Strings and Styles)
* Database Adapter

## System Requirements

**Splash Screen**

*The user will be presented with a start-up screen. This will contain some images and a welcome message to the player.*

**Menu Screen**

*This will present the user with a set of options that he/she can select. These options are: Play Game, Game Play History, How to Play and about-us.*

**Play Game**

*This activity consists of two fragments. The first fragment shown on the first half of the screen displays the user selected letters and adds the user word to the list. The second fragment moves the buttons and randomly generates the letters on the buttons. First, the user will be shown three buttons containing random letters. These buttons will located at the second half of the screen. The countdown will start and each button will fall from the middle of the screen to the bottom throughout the game session. When the buttons reaches the end of the screen a new set of random letters will be generated each time. The user can choose a word by clicking the button before it reaches the end of the screen. The letter will be added to the word maker text field in the first fragment. Once the user feels he/she has created a valid word, the player can click on the add word button which will add that word to the list. After 15 seconds, the user created words will be validated and the user score will be generated. Each word will give the user 10 points. All valid words will be stored in a database and the user scores will be saved to a text file. After the game, the user will be prompted to play again or not.*

**How to Play**

*Information on how to play the game is shown to the user. This information retrieved by reading a text file. This gives the user detailed instructions on how to play the game.*

**About us**

*This gives the user added information about the development of the game and the games intended purpose. All information is retrieved by reading a text file and displaying the data.*

**Game Play History**

*This activity consists of a tab host containing 2 tabs. These tabs consists of fragments. This provides the user with 2 added options. They can view previous scores or created word. Selecting the scores tab will show all the user scores and the date and time that that score was achieved. This information is gathered by reading a text file and displaying the data. The user created words will show all words created by the user. This information is retrieved from the database table. If the user selects a word, he/she will be shown the definition of that word shown in an alert dialog.*

# Flow Chart











# UML Diagrams

## Class Diagrams









## Use Case Diagram



## Database Design

### ERD



### Table Structure

|  |  |  |
| --- | --- | --- |
| Id | Word | Definition |
| 1 | me | The person speaking regarded as an object |
| 2 | hip | The projecting region of the lateral parts of one side of the pelvis. |
| 3 | go | To pass from one side to another. |
| 4 | tam | A wooden cap of Scottish origin |
| 5 | geb | The god of the earth; father of Osiris and Isis. |

# Help File

Please put on your Internet/Wi-Fi on the device before playing.

1. The objective of the game is to make as many words as possible in the given time.

2. Random letters will glide down the screen and continuously Change.

You have to tap/select the letter.

Once the letter has been selected it cannot be unselected.

Choose your letters wisely!!!

3. Once you have created your word, you can click on the add word button.

This will keep a list of all the words you have created.

4. Once the time has ended, your score will be calculated.

Every valid word that is created, you will get 20 points per word.

Hint: Make sure you are quick at selecting your letters.

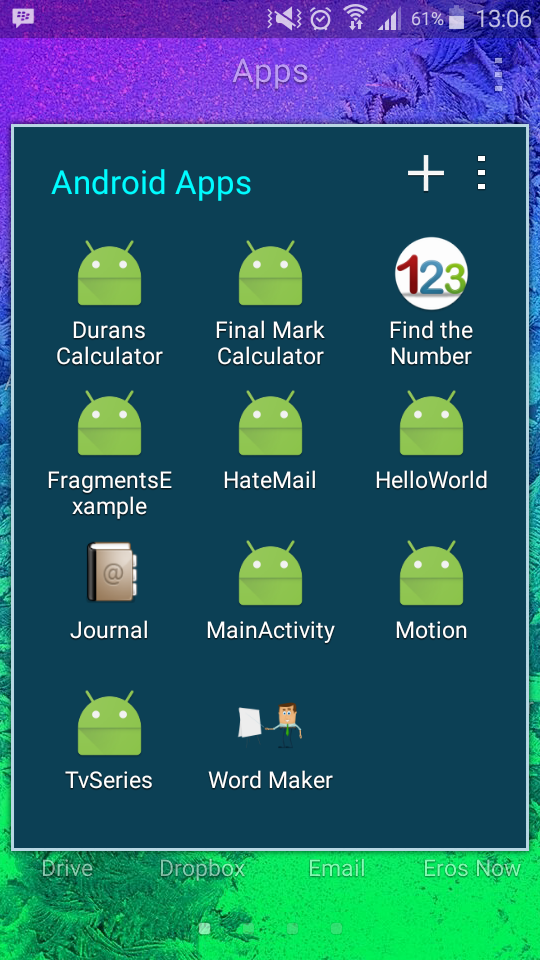
There are 26 letters in the alphabet, your desired letter may take a while to come up again.

Good luck!!!!

# Screen shots/Test Data

**Game Icon**

Icon and name of the game



**Splash Screen**

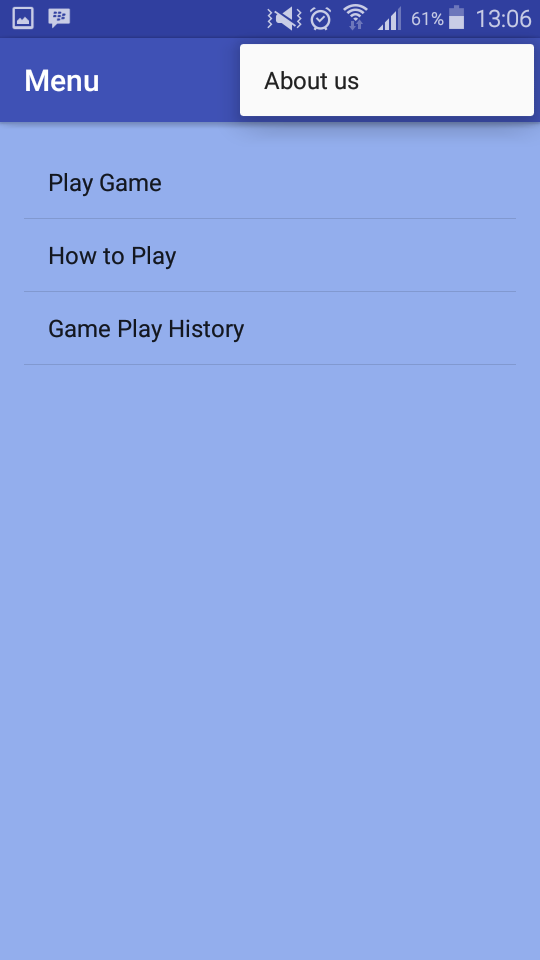
Splash screen of the game consisting of a progress bar.

This screen welcomes the player to the Game.

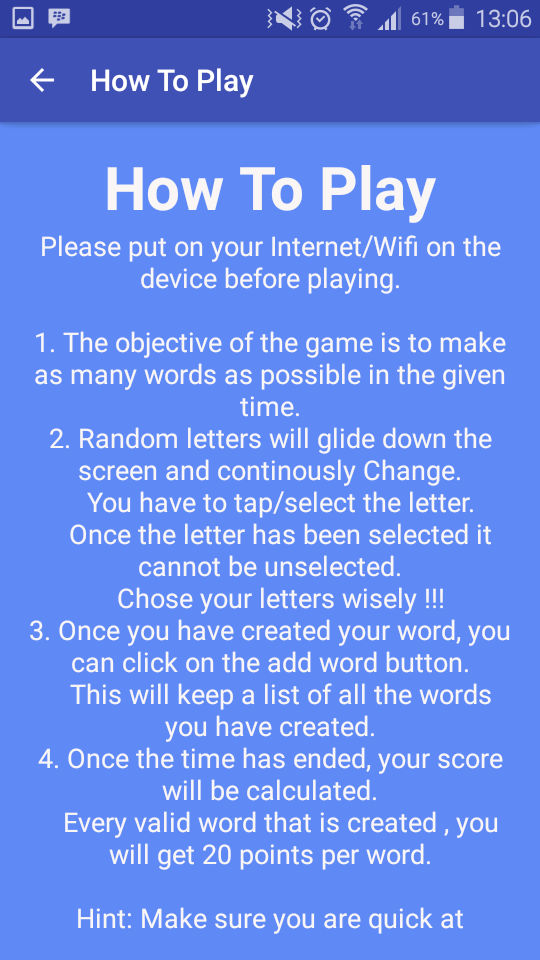


**Menu Screen**

Presents the user with a list of options to choose from



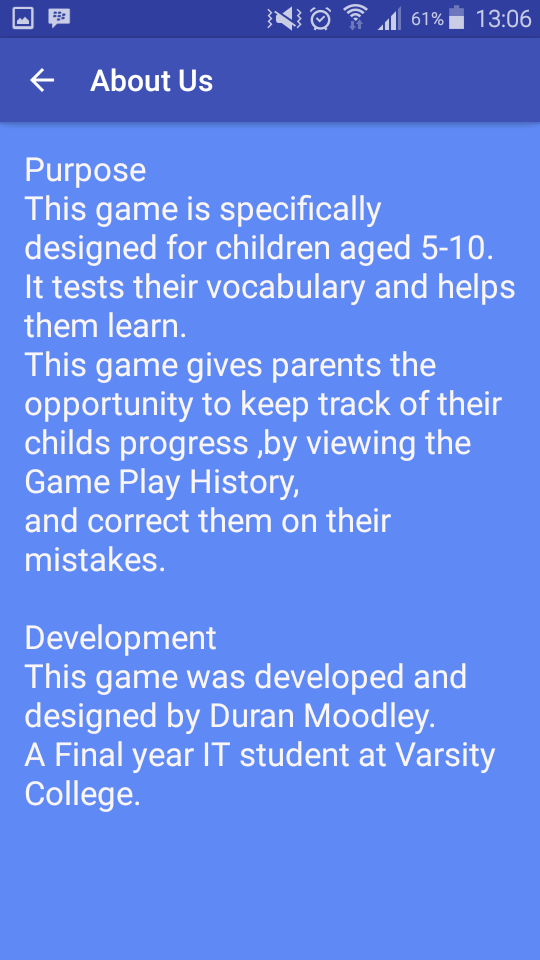
**How to Play**

 The player is shown a set of instructions on how to play the game.

**About us**

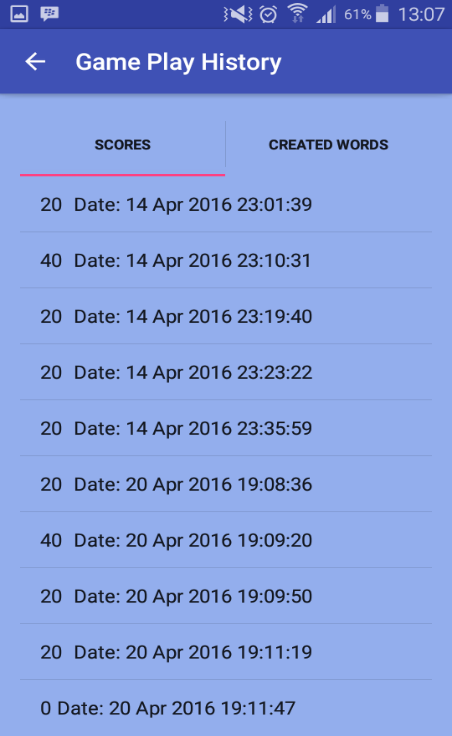
This sections provides some information about the development of the game

And the games purpose.



**Game Play History**

This consist of 2 tabs.

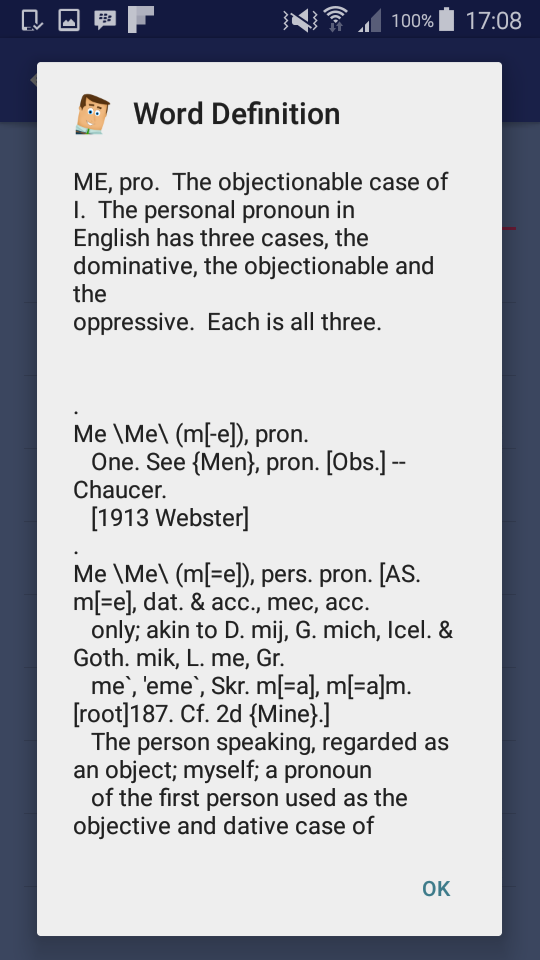
 The first tab, allows the player can view all previous scores that he/she has achieved.

**Game Play History – Created words**

The second tab allows the player to view all words created.



If the player selects the word. He/she can view the words definition in a dialog box

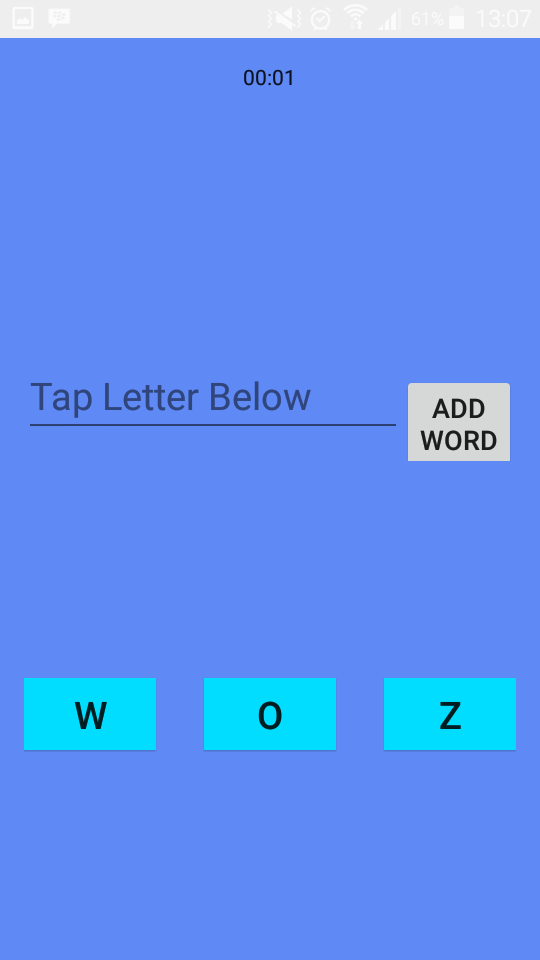


**Game Play**

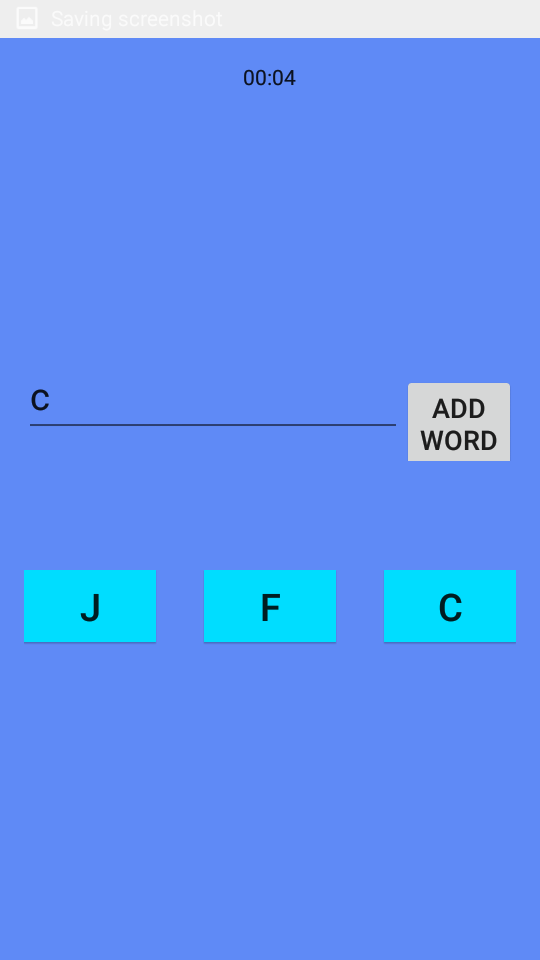
Game starts and countdown begins.

The buttons will trickle down the second half of the screen.

When the buttons reach the bottom, new letters will be generated

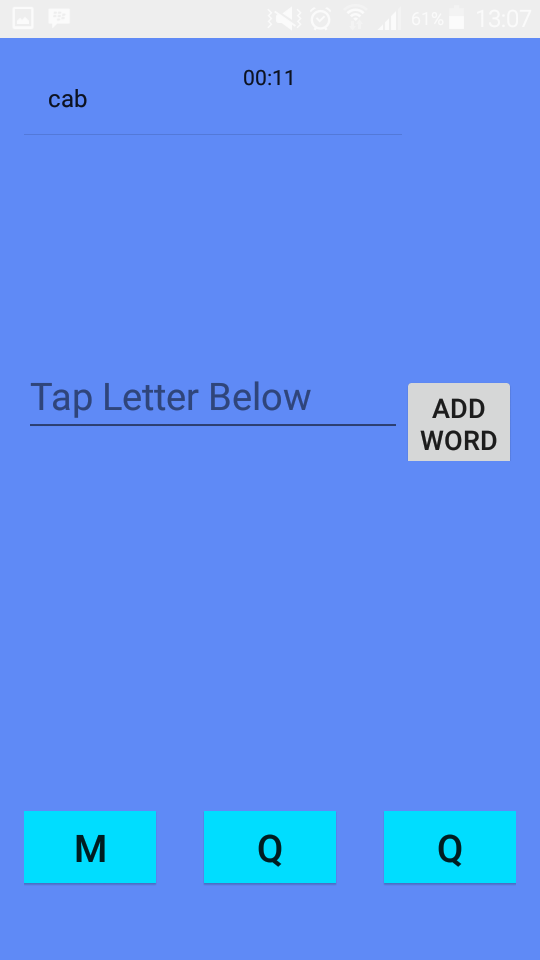
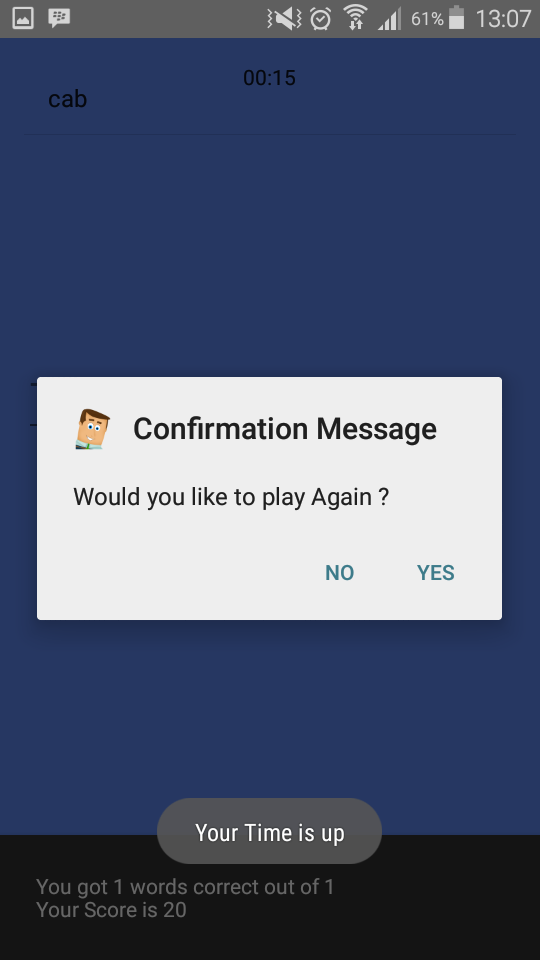


. Letters randomly generated. The player has selected a letter to use to create a word.



User has created a word and added it to the list by Time has expired. User result is displayed.

Clicking on the add word button. User can chose to play again or not.



# Lecturer Feedback & Corrections Made

|  |  |
| --- | --- |
| Lecturer Feedback | Corrections Made |
| Missing allow backup attribute | Allow backup attribute set to false |
| App is not index able by google search | Not fixed |
| Obsolete target param in scroll view | Attribute removed |
| R. Dimension - unused | Removed |
| Useless parent layout scroll view with relative layout | Not fixed |
| Using left right instead of start/end | Left-right has been replaced with start-end |
| Some declarations can be final | Made some declarations final |
| Declarations can be weaker and private access | Added a private access modifier to the declarations |
| Method invocation – get support action bar can produce null | An if statement was added to check for null value |
| Input stream is always null | Input stream variable deleted from method |
| Get view method might me null | An if statement was added to check for null value |
| Tab host. New tab spec can produce null | An assert statement was added |
| Spelling errors | Spelling errors corrected |
| This app can only run on Android 21. Errors will be produced if this app runs on Kit kat and lower | Not fixed |

# Enhancements

* A database table was included in the application. This table stores all user created words and the definition of each word. A database class was created to handle the database functions (Create and read).
* If the words creates the same word more than once, that word is not saved to the database or duplicated.
* Add a button style in the styles. Xml file. This style is called in the buttons.
* The user can now view the definition of a particular word. This can be done by viewing all the words in the Game Play History -> Created Words. When the user selects a word from the list, an alert dialog is shown that gives the definition of the word. This data is gathered from the database table.

# Conclusion

The development of this application has taken me to new level in my programming career. I have learnt a lot of concepts that were not covered in class such as: working with api’s, incorporating animations, adding threads (async task) and other android concepts. It tested my patience and perseverance. I am truly proud of my accomplishment and I hope to add this application onto the app store.