WORD JUMBLE GAME

MODULE: Mobile Application Development

YEAR: Software Development Year 3 GMIT

LECTURER: Damien Costello

STUDENT: Ciaran Whyte

STUDENT ID: G00254624

PUBLISHER NAME: Ginga Ninja

STORE LINK: <https://www.microsoft.com/en-us/store/apps/word-jam/9nblggh4qrwm>

GITLINK: [https://github.com/whytekieran/MobileAppProject](https://github.com/whytekieran/MobileAppProject%20)

(App is currently in a private repository, I will make the repository public after the due date)

INTRODUCTION

The following application is a Windows 8.1 application which has been tested primarily using a Nokia Lumia 520. The application is a word jumble game that offers the user jumbled words and gives him/her a time period in which to decode as many of them as possible, for each word the user manages to decode he/she gets points. When the time is up the user is navigated to a page where they can enter a name to go along with their score, which will be saved.

THE APPLICATION STRUCTURE

The application consists of various pages, classes and additional resources, the following is a brief outline of each of them:

The Pages

--**MainPage.xaml:** This page is the opening page of the application and is first loaded when it starts providing the user with a list of options (Binded Options) that allow a user to choose a game of different word lengths or also to navigate to the high scores menu.

**Game.xaml:** This page contains the actual game which the user plays, the words for the game are stored in an SQLite Database.

**GameOver.xaml:** This page is navigated to when the game ends, it is used to allow the user to enter a name and save it along with the score they got. These scores are stored in an SQLite Database

**Highscores.xaml:** This page outputs all the high scores for the game

**HighScoresMenu.xaml:** This page allows the user to choose which set of high scores for which type of word game they wish to see.

The Classes

Firstly, every page listed above comes with its own class file with the extension .xaml.cs .....These classes will not be listed here as they come with any .xaml page. Other classes created for the application are as follows:

**DataPasser.cs:** This class is used to pass data from one page to the next. it contains just one instance variable which is an int. Depending on the int passed the receiving page can make a decision on what it must do.

**GameType.cs:** This class is similar to DataPasses.cs in structure except it works with strings. Its used to bind lists of game options for the user.

**HighScore.cs:** This class holds high score objects which contain data like rank, name and score. This object is used to bind data to the high score lists.

**ScoreInformationPasser.cs:** This class is used to pass the score and type of word jumble game played to the Game over page. This way the score can be inserted into the correct table in the database

**Words.cs:** This class is used to hold words retrieved from the database. Using this class we can create a table of Words which we can read from the database.

The Database

**GameDatabase.db:** This is an SQLite database file and contains all the data needed for the different word games and all the high score lists which the user can view. In addition to the resources listed above, this application also contains resources generated by Visual Studio.

eg

--Assets Folder of default images

--Package.appxmanifest

To include SQLite for Windows phone 8.1 into the project I used the SQL Server compact toolbox to edit the database and insert rows. This was installed into Visual Studio by myself. Here is the tutorial I used to do it. [https://www.youtube.com/watch?v=UM5qf\_3Go-0](https://www.youtube.com/watch?v=UM5qf_3Go-0%20)

It must be installed through the Tools->Extensions and Updates menu.

Also in the Tools->Extensions and Updates menu we install SQLite for Windows Phone 8.1, then we must right click on references and add a references.

We add SQLite for WP8.1 and Microsoft Visual C++ 2013 Runtime Package for Windows Phone.

Then we go to manage nuget packages and install sqlite-net. We also need to change the settings in the configuration manager. Here is a link to a tutorial i used that help me set it up <https://www.youtube.com/watch?v=nVk8jkicphQ>

LOCALIZATION

A simple example of localization was included for the title of the main page, eliminating a hardcoded string.

RESOURCES

These resources are inserted manually into the project

--Images folder containing images for the application

--References (SQLite for WP8.1 and Microsoft Visual C++ 2013 Runtime Package for Windows Phone (Needed for SQLite))

--SQLite.cs contains SQLite functionality

--SQLiteAsync.cs contains SQLite functionality

The two SQLite classes are installed when you include the packages described above in the database section.

RUNNING SUGGESTIONS

I would suggest running this application on a Nokia Lumia 520 as this is the platform the application was run and tested on. The code itself is heavily commented and contains more depth about the actual implementation of the application.

Enjoy :)

References

1. Handling Orientation

<http://blog.jerrynixon.com/2013/12/the-two-ways-to-handle-orientation-in.html>

1. How to install the packages needed for SQLite and WP8.1

<http://www.c-sharpcorner.com/UploadFile/59b9d6/sqlite-for-windows-phone-app/>

1. SQLite tutorial

<http://www.tutorialspoint.com/sqlite/index.htm>

1. Five, six and seven letter words

<http://www.yougowords.com/5-letters>

<http://wordfinder.yourdictionary.com/letter-words/6>

<http://wordfinder.yourdictionary.com/letter-words/7>