TOBB UNIVERSITY OF ECONOMICS AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

PROJECT TITLE

WIGGLE (ANDROID WORD GAME)

BİL 496 Graduation Project

SDD DOCUMENT

2014-2015 Spring Semester

Supervisor:

OSMAN ABUL

Team Members:

Utku GÜLEVİZ, Ahmet Abdulkadir ÖZSOY, Arascan ÖNAL

Date: 09 February 2015

Software Design Document

1-Introduction

1.1 Problem Definition

In this project we will implement an android word game. This game can be played single player or multiplayer. Also players can share their score in Facebook and Twitter.

1.2 Purpose of the Project

The fundamental aim of this game, user sees a shuffled word and tries to find the actual word. User can read the clues which are bottom of the word. Also in multiplayer games, user can gain experience points if he/she wins the game. Also, with these experience points user can level up. The difficulty of word is directly proportional with user's level.

1.3 Scope of the Project

All words in the game are English. The game will have 60.000 different words. The xml files will include all words and their meaning or meanings in specific format. User can answer the word in two ways. In first way, user will select the first character and then select second character. These chosen characters' change with another one's. In second way, user will press the microphone button which is in our application and says the word. Also this game is not only a single player game, but also is a multiplayer game. Multiplayer games will have three difficulty level. Each level will save different score. Users will match with each other as far as their level. Finally users can share their score in Twitter or Facebook.

1.4 Overview

This project is an android game and will be developed with Android Studio. First step is that finding an English dictionary and parse it. After that, we will prepare three xml files in a specific format for each level. Secondly, we will create our layouts. Before creation of these layouts, we must design them. After these steps, we will implement our java (source) files. These files include animations, speech to text property and all other features. Also in these files we need to connect database, Facebook-Twitter API's and server side to our application. In server connection we will use Google App Engine services.

1.5 Definitions and Abbreviations

Android Studio: is an integrated development environment (IDE) for developing on the Android platform that launched by Google.

Layout: is all screen in the game.

Clue: is the meaning of the word.

Animation: is all transitions between events.

Speech to Text: is that sound is converted word with the aid of the microphone.

Experience: is earned level and point.

API -> Application Programming Interface

AS -> Android Studio

SDK -> Software Development Kit

1.6 References

- http://android-developers.blogspot.com.tr/2014/12/android-studio-10.html
- http://www.mybringback.com/series/android-basics/
- http://www.mybringback.com/series/android-intermediate/
- https://cloud.google.com/appengine/docs
- https://developers.facebook.com/
- https://dev.twitter.com/
- http://www.tutorialspoint.com/android/android_ui_deign.htm
- https://gelecegiyazanlar.turkcell.com.tr/konu/android

2. System Overview

In this project we will use Facebook - Twitter API's for sharing score in these APPs, Google App Engine for multiplayer gaming and using our database. Also, we will override some Activity class' methods for using android features.

3. Design Considerations

3.1 Design Assumptions, Dependencies, Constraints

We will build this game only for Android platform. With user feedback, we may build this game for IOS, Windows Phone. Our game language is only English. If user don't know English, unfortunately cannot play this game. Our game has three difficulty level. After some point user will get bored because of the same level. With user feedback, we may increase our level number.

3.1.1 Hardware Constraints

The device has Android operating system and max 50 MB free hard disk space. If devices don't have Internet, user cannot play multiplayer. If devices don't have microphone, user cannot use speech to text too.

3.1.2 Software Constraints

The computer must have some SDK for building in computer. If users don't allow the application to use its microphone, user cannot use speech to text. Android's speaking language must be English for speech to text feature.

3.1.3 Time Constraints

Game's response time must be fast enough to play smoothly.

3.1.4 Security Constraints

There isn't any security constraints for single game platform. In multiplayer games we will need to have user name and password to connect each other. Because of this, our database systems must be reliable.

3.2 Design Goals and Guidelines

Our main goal is that we will put our games in Google Store. We will design our game in accordance with this purposes. If our game will be liked other Android users, we may improve our games.

3.2.1 Portability

This game won't be portable. (Game's features cannot be used in another APPs or Systems.)

3.2.2 Speed Considerations

We will do some specifications for this issue. Parsing word files (.xml files which have nearly 60000 words-clues) will be only one time. Also, in this game server sided operations will be minimized.

3.2.3 Reliability

In this game server sided operations can be risky sometimes (which uses internet). These operations as reliable as Google. Because we will use Google App Engine in these operations.

3.2.4 User Interactivity

User interactivity will be provided by touching android screen (pressing buttons or characters of words) and saying words (in speech to text).

4. Data Design

4.1 Data Description

Our game has two data types which are the word and its meaning/meanings. We will use an English dictionary which have 60.000 words for this game. In multiplayer games we will need to have user to connect each other.

4.2 Data Dictionary

Our English word database mainly bases on three separate xml files. The first xml file consists words which contains three to four letters. Similarly second and third xml files consist words containing five to six and seven to eight letters respectively. As to multiplayer database, we will use App Engine for all of our database needs provided by Google.

5. System architecture

5.1 Architectural Design

This content is in our Extra's. (Picture 1)

5.2 Description of Modules

We have four modules for this game. It includes game layouts, interaction with player, word files and multiplayer database. We have given information about these modules before.

5.3 Design rationale

We need to decide two points. Firstly, how to parse xml files without big performance drops and lag. Our solution is making all parse process at the beginning of the game start. Thus, user will not encounter any lag after game loading process. Another issue is deciding which database solution will be used for that project. Our choice is App Engine provided by Google. Big database space along with good database access speed, Google is the best solution we found.

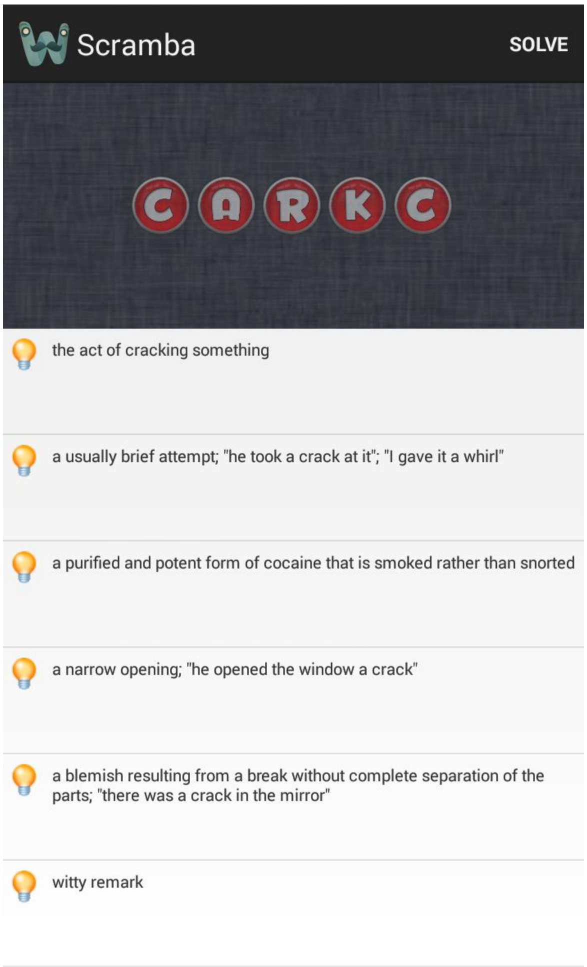
6. User Interface Design

Android Studio has its layout creator. Our all layouts will be prepared with Android Studio. We decided that firstly we will design our layout basically for example layouts will have only basic button image in the beginning. After the project's main components finished, we will add our application and edit our layouts visually.

6.1 Overview of User Interface

Designing is important for this project because visually effect is one of the most important part of the project. For each stage, we will implement a layout. We will decide later the visually effect of this project.

6.2 Screen Images



the act of cracking something

a usually brief attempt; "he took a crack at it"; "I gave it a whirl"

a purified and potent form of cocaine that is smoked rather than snorted

a narrow opening; "he opened the window a crack"

a blemish resulting from a break without complete separation of the parts; "there was a crack in the mirror"

witty remark

6.3 Screen Objects and Actions

We have four screen layouts. Firstly, we will create main menu window. Secondly we will create single and multiplayer layouts. Lastly we will create pause layouts for single game. All of the layouts connect with each other.

7. Libraries and Tools

7.1 Libraries

We will use some android SDK's for android development. For gaming data, we will use XMLPullParser Object for parsing words.xml files. In server connection we will use Google Services or other services like Google's. For Twitter and Facebook connection, we will use Facebook and Twitter API. Lastly, we will App Engine for multiplayer connection.

7.2 Tools

We will need Android Studio and some included SDK. Maybe we will receive support from some layout design sites.

8. Project Timeline

19 January 2015- 31 January 2015

SRS and SDD documents are prepared along with parsing English dictionary and preparing it for further usage.

1 February 2015 - 8 February 2015

Layouts and xml word files will be created.

9 February 2015 - 14 February 2015

Menu Activity and single player's main activity will be added.

14 February 2015 - 28 February 2015

Multiplayer gaming and database will be added.

1 March 2015 – 4 March 2015

Preparation of demo.

4 March 2015 - 11 March 2015

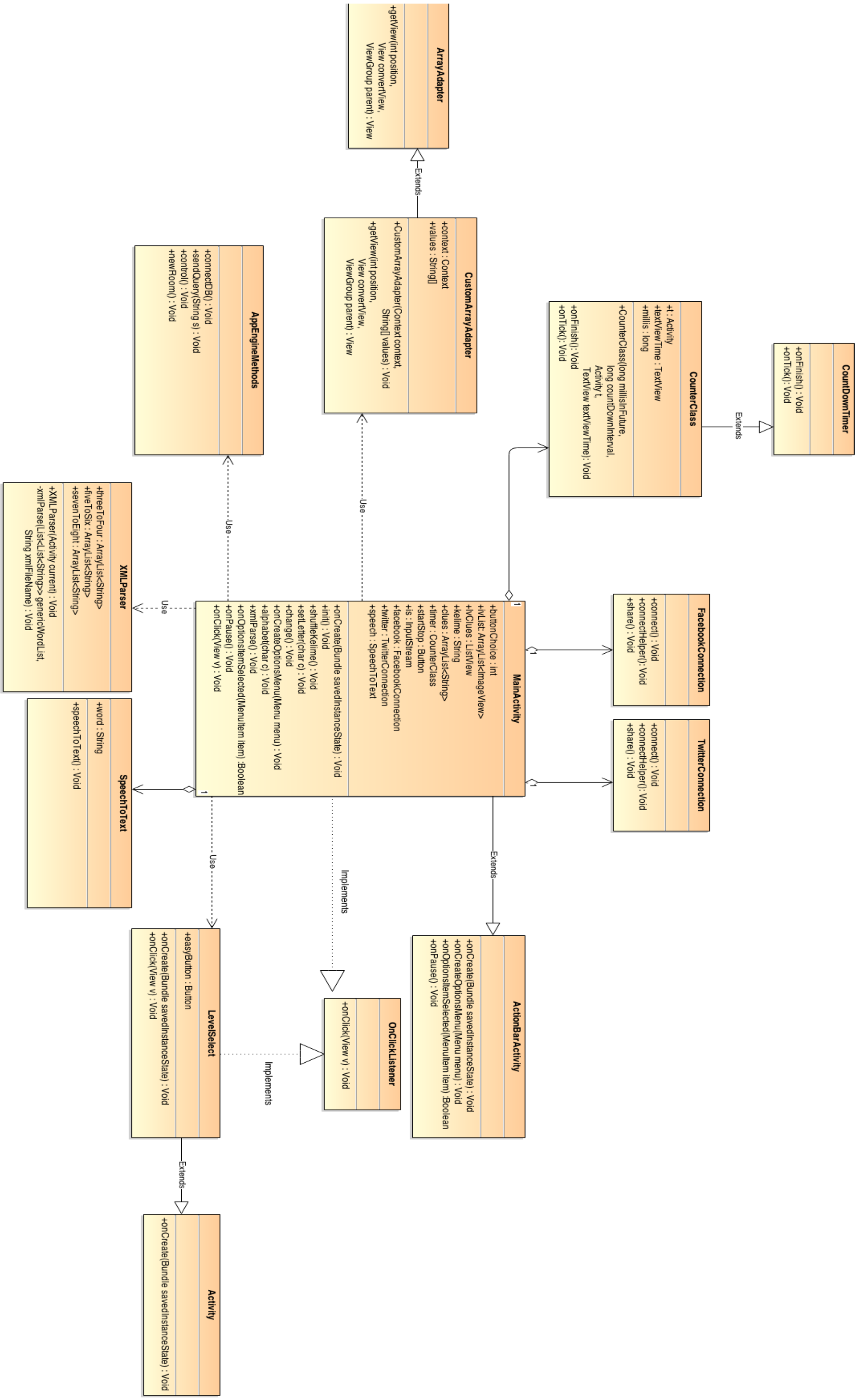
Facebook and Twitter connections will be added.

11 March 2015 - remaining time until demo day

All improvements and missing parts will be developed and game reaches the testing stage. The final report will be prepared.

9. Conclusion

In conclusion, we are implementing an enjoyable game that users guess the words with the meaning of the word. The goal of the user is that to reach the highest level of the game. The level increases, the length of the word increases and earns more point to the user. In Android devices, there are some similar games but they don't support multiplayer gaming. We think that this attribute get us one step forward. Also we will connect our game to the Facebook and Twitter. Thus we will learn Facebook and Twitter API for Android. Furthermore, users who play our game will learn more word and their meaning. They will use synonym words, antonym words and other meaning of the words. They will improve their vocabulary knowledge.


ArrayAdapter
+getView(int position, View convertView, ViewGroup parent) : View
Extends
CustomArrayAdapter
+context : Context
+values : String[]
+CustomArrayAdapter(Context context, String[] values) : Void
+getView(int position, View convertView, ViewGroup parent) : View
CountDownTimer
+onFinish() : Void
+onTick(): Void
Extends
CounterClass
+t : Activity
+textViewTime : TextView
+millis : long
+CounterClass(long millisInFuture, long countDownInterval, Activity t, TextView textViewTime): Void
+onFinish(): Void
+onTick(): Void
AppEngineMethods
+connectDB() : Void
+sendQuery(String s) : Void
+control() : Void
+newRoom() : Void
Use
Use
XMLParser
+threeToFour : ArrayList<String>
+fiveToSix : ArrayList<String>
+sevenToEight : ArrayList<String>
+XMLParser(Activity current) : Void
-xmlParse(List<List<String>> genericWordList, String xmlFileName) : Void
Use
MainActivity
+buttonChoice : int
+ivList: ArrayList<ImageView>
+lvClues : ListView
+kelime : String
+clues : ArrayList<String>
+timer : CounterClass
+startStop : Button
+is : InputStream
+facebook : FacebookConnection
+twitter : TwitterConnection
+speech : SpeechToText
+onCreate(Bundle savedInstanceState) : Void
+init() : Void
+shuffleKelime() : Void
+setLetter(char c) : Void
+change() : Void
+onCreateOptionsMenu(Menu menu) : Void
+alphabet(char c) : Void
+xmlParse() : Void
+onOptionsItemSelected(MenuItem item) :Boolean
+onPause() : Void
+onClick(View v) : Void
1
1
1
1
FacebookConnection
+connect() : Void
+connectHelper(): Void
+share() : Void
TwitterConnection
+connect() : Void
+connectHelper(): Void
+share() : Void
SpeechToText
+word : String
+speechToText() : Void
Use
Implements
Extends
ActionBarActivity
+onCreate(Bundle savedInstanceState) : Void
+onCreateOptionsMenu(Menu menu) : Void
+onOptionsItemSelected(MenuItem item) :Boolean
+onPause() : Void
OnClickListener
+onClick(View v) : Void
Implements
LevelSelect
+easyButton : Button
+onCreate(Bundle savedInstanceState) : Void
+onClick(View v) : Void
Extends
Activity
+onCreate(Bundle savedInstanceState) : Void

Picture -1: UML Diagram