Project Proposal: Hangman using Wi-Fi Direct

By: Asma Mehjabeen ([amehjabeen@luc.edu](mailto:amehjabeen@luc.edu)),

Keerthana Aravamudhan (karavamudhan@luc.edu)

Date: 22 Oct 2013

Updates: 26 Oct 2013, updated numbered requirements and the plan and schedule, updated peer review comments.

Overview: Interconnect two devices using Wi-Fi direct and allow the users of the devices to play the classic hangman game; and a game session starts. The players take turns in guessing words. The player who initiates the connection goes first and gets to specify a word, which the second player has to guess, and a hint (displayed to player 2). The second player tries to guess the letters of the word and player 1 decides whether the letter is in the word or not. With every wrong guess, a part the man is hanged. The game ends if the second player guessed the word or if the hangman diagram is completed. The scores are stored in an SQL Lite database in each player’s device separately and are synched after each game finishes. The total score is displayed at the end of a game session. Any player can end the game session at any time.

Operating System(s):

1. Android 4.x on device

Links to Documentation and Tools to be Used:

Wi-Fi P2P in Android - <http://developer.android.com/guide/topics/connectivity/wifip2p.html>

Animation and Graphics - <http://developer.android.com/guide/topics/graphics/index.html>

OpenGL ES - <http://developer.android.com/guide/topics/graphics/opengl.html>

The application will be developed using Android Studio and will be tested using real devices.

Numbered Requirements

1. Allow the user to search for peer devices on the same W-Fi network and display the list of available devices. (Mandatory)
2. Interconnect the devices when a user selects a device from the list and start a game session. (Mandatory)
3. Send a system notification to the second player; inviting him/her to play the game. If the invitation is accepted start the game session. (Mandatory)
4. When the game session starts, make the person, who initiated the connection, player 1 and display the screen for player 1 where he could specify the word and the hint. Till then display a screen with the appropriate message to player 2. (Mandatory)
5. After the word is specified by player1, display the game screen to both players. Display dashes corresponding to the number of letters in the word to player 2. Display the complete hangman figure that is initially greyed out. Display buttons for each alphabet which the player 2 can use to guess letters. (Mandatory)
6. Do not update the UI if player 1 presses any of the alphabet buttons. (Mandatory)
7. When player 2 presses an alphabet button, send the corresponding alphabet to player 1 and let player 1 decide if the letter is in the word or not. (Mandatory)
8. If it’s in the word then insert in the appropriate location and disable the button for that alphabet on the UI’s of both players. If it’s not in the word then disable the button on the UI of both players and change the color of the button to red. Also prompt player1to touch a grayed out part of the hangman (if not already done). On touching a greyed out part of the hangman figure, change the color so it stands out. Update the UI in both devices accordingly. (Mandatory)
9. Disable on touch events on the hangman figure for player 2. (Mandatory)
10. Check if player 1 makes a mistake in deciding if the letter is present in the word or not and display a message accordingly. (Optional)
11. After a game ends (i.e. if the word is guessed or the hangman figure is completed) , update the score for the players and store the updated score in the database. Display the score only after the game session ends. (Mandatory)
12. Allow any user to end the game session at any time and display the scores. (Mandatory)
13. Limit connection to only one device during a game session (i.e. unless the game session ends don’t let any user access the list of available devices. (Optional)
14. Always display the score on the UI. (Optional)
15. Allow the players to chat with each other during the game. (Optional)
16. Implementing separate modules for the hangman game logic and the communication logic (wifip2) so that the game logic could be reused to communicate via different channels such as Bluetooth.

Plan and Schedule:

As per the current planning, the work for the project would be divided as following:

* Implementation of Wi-Fi direct : Asma Mehjabeen
* Design of the app : Keerthana Aravamudhan
* Code for player 1 : Keerthana Aravamudhan
* Code for Player 2 : Asma Mehjabeen

Week 1: Understanding the Usage of OpenGL and wifip2p and SQLite – 11 hrs

* Understanding the Open Graphics Library for 2D and 3D graphics and the related openGL ES API for creating graphical user interfaces.
* Understanding the android.net.wifi.p2p package to create and manage peer to peer connections using Wi-Fi direct.
* Exploring the sample apps available on the android tutorial website to understand the device compatibility, coding convenience etc. for openGL using the sample available apps.
* Exploring the sample apps for understanding how to let an application discover available peers, set up connection with the peers and query the list of peers.
* Creating and updating the database using SQLite in the android application

Week2: Implementing and understanding using the sample demo apps – 15 hrs

* Implementing small graphical user interfaces using openGL
* Establishing a peer to peer communication between devices i.e. discover available peers, set the connection between interested peers.
* Designing of the user interface to see how the app would look like and check whether the design is in-line the requirements stated.

Week3: Incremental Development - Phase 1 – 20 hrs.

* Implementing Requirements
* Perform small manual tests to validate the app and to verify if the requirements have been implemented as per the design

Week4: Incremental Development - Phase 2 – 20 hrs.

* Implementing the remaining requirements
* Perform small manual tests to validate the app and to verify if the requirements have been implemented as per the design
* Integrating all the modules and perform an end to end validation of the app

Week5: Verification and Validation - 12 hrs.

(If possible): Implementing the optional requirements specified.

* Mapping the requirements against the app and check if all the requirements have been addressed and implemented
* Validate the actual results with the expected results
* Perform a peer review of the app and the code
* Create a report detailing the approach and the outcome the app development. Include the future work (features that could be added to the existing app), challenges faced, defects (bugs) if any in the current version of the app.

Estimate of Effort:

* Requirement Analysis – 10 hrs.
  + Scope and Approach
  + High Level Application Idea
  + Application Flow finalization
* Planning and Design – 20 hrs.
  + Graphical User Interface design
  + Design finalization
* Build and Test – 30 hrs.
  + App Development Phase 1 & Testing
  + App Development Phase 2 &Testing
  + Integrate and testing
  + Verification and validation
* Implementation – 5 hrs.
  + Release
  + Analysis and reporting

Metrics for Code and Defects:

* Lines of Code including executable lines, data definitions, and comments.
* Defect Density i.e. Number of Bugs / KLOC

Software Metrics to be Gathered and Reported (including PSP)

* Source Lines of Code (SLOC) or effective lines of code( ELOC) ( Code complexity)
* Number of defects found
* Number of defects found by severity
* Testing efficiency
* Reliability and quality
* Ease of use/User satisfaction
* Schedule

PSP:

Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Program#:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructor:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Language:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Planned Actual Date

Summary

LOC \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Time for

Each phase Plan Actual Date

Planning \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Design \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Coding \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total

Implementation\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Defects

Injected Actual Date

Planning \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Design \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Coding \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total

Implementation \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total Defects \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Defects

Removed Actual Date

Planning \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Design \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Coding \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total

Implementation \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total Defects \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_