Portfolio Project

Gregory Norris

CSC 475 – Elchouemi

Colorado State University Global Campus

Portfolio Project

**Prospective Idea for Android Mobile Application**

This has been a fun, and challenging programming course in Android development. I have been excited to take this course since I see many ways to continue developing in Android for personal enjoyment and my list of potential Android projects is growing daily. I personally enjoy playing physical board games with friends quite often, and find the social aspect of tabletop gaming brings to the table many opportunities for developing relationships that online games lack. I have been looking at ways to use Android programming to help keep conversations going during social gatherings without becoming the focus of attention. In this, I realized that it is possible to create games which can only be played with other players are physically in the room (with an active internet connection). I wanted to create (or emulate) a social deduction game in which players are forced to communicate with one another to play without relying too heavily on their phones to participate. Enter: Spyfall.

Spyfall is an existing social deduction game which relies on hidden roles and the strategic release of information to allow one faction or the other to win. In Spyfall, every player (except the Spy) is given a shared location for a super-secret meeting place such as a Cruise Ship, Casino, Moon Base, Submarine, Carnival, etc. Players do not know who is an associate and who is the spy, and must take turns asking each other questions to determine if each player knows where they are. The spy must listen carefully to the questions and responses given to attempt to guess the location without giving away their role in the game. If the associates correctly guess who the spy is before a timer runs out, they win the game, if the Spy is able to remain undercover or guess the location of the meeting before the timer runs out, the Spy wins the game.

**General Implementation**

Implementation of a program to host this game should be relatively simple. Players only need to have a shared or generated lobby code (no user accounts or login) to join the game, and the application only needs to return four pieces of information: Their character, their location, a countdown timer, and a list of possible locations. Ideally, applications will be synchronized via a webservice to avoid the potential security issues of having the phones search for each other over the network and sharing data directly.

**Graphical Interface**

The graphical interface should be relatively straightforward as well. The game can be implemented completely with a handful of textview elements and using only three activities with 2-3 buttons each. The program will consist of the following activities: Join Game Activity, Lobby Activity, Main Game Activity, and (optionally) a Game Preferences Activity.

The Join Game Activity will allow players to either generate a new lobby code (creating a new instance of the game), or to join using an existing lobby code. It will have a text edit box which will by default generate a new lobby code every time the application is started. If a player clicks “create lobby”, the game creates a new instance of the game. The player may choose to enter in the name of an existing lobby and click “join lobby” to enter into an existing game. Either button will then activate the Lobby Activity.

The Lobby will have two text-view boxes to show the current lobby code. Below that is a text edit box and prompt for the user to select a codename. There will be three following buttons: “Ready”, “Change Codename”, “Quit”. Below these buttons will appear a list of player names and their ready status, as indicated by a non-editable checkbox. The game will periodically poll the server for the lobby status. When all players are ready, the game will proceed to the Main Game Activity.

The Main Game Activity will have a timer at the top, and text view boxes disclosing their role and location. Each client will receive their role, location and a signal to start the timer. Below the text view boxes will be a table layout containing toggle buttons for each of the possible locations or players. Clicking each button will change the color and can be used by the players to mark locations or players as new information is revealed as the game progresses. The buttons themselves do not have any real function other than facilitating deductive reasoning. When the timer reaches zero, the game will end and spy will be revealed to the players, while the location is revealed to the spy. After 15 seconds, all players will be placed back in the lobby with not-ready status. If either the players guess who the spy is, or the spy guesses the location, the players may select the quit button at the top of the screen to be placed back in the lobby with not-ready status.

**Web Services Implementation**

The game will contact a web service to synchronize data between the clients. On application startup, each client will generate a unique client ID string to help the server keep track of client connections. Most of the web activity will take place during the Lobby activity, which is designed to synchronize information between each player’s application and provide information on the number of players in the game before it begins. Android Developer (2019) states that the best practice for generating a unique identifier is to avoid using MAC based, hardware based, or non-resettable identifiers in applications such as the serial number. Currently, the application uses a randomly generated universal GUID as a session identifier.

To allow the programs to synchronize data between each other, the game will utilize HTTP requests to send and receive data from an application server. HTTP requests will send and receive data in JSON format. I plan on developing this application further after this course to utilize the Volley library for handling communications to the server. Volley itself is not suitable large downloads, but is suitable for rapidly exchanging smaller packets of information (Geeks for Geeks, 2019). This makes Volley perfect for synchronizing client apps to the server.  
The JSON strings between the client and server will look like:

{“client\_push”:[{“client\_id” : “21d31a2f”, “codename”: “Grey Tiger”, “lobby\_code”: “ACORN SIGINT”, “player\_status”: “ready”, “quit\_status”: “no”}]}

{“serv\_response”:[“lobby”: “My Lobby”, “user” : ”Ultra Buffalo”, “user” : “Danger Noodle”, “user”: “Not the Spy”, “user” : “E.D.K.H.S.”, “location”: “location\_12”, “spy” : “Grey Tiger”, “role”: “Spy”, “active\_Game”: “true”}]}

Although this application relies on passing bundles within intents whenever a new activity is started, for full functionality of the game, I will likely need to use a shared preferences storage on the device to pass information between each activity. This will allow for some useful options regarding the persistence of usernames and the GUID. I do not expect that it will be necessary to store information within a structured database like SQLite.

One important function which has not been developed in the UI is the list of available players and their ready status, which is set to appear in the Lobby activity. The plan is to add users to the application and utilize fragments to show usernames as they join the lobby. Android Developers (2019) highlights a property of fragments which makes them perfect for this application “You can manipulate each fragment independently, such as add or remove them”. This should allow the players list to be updated frequently so that individual player’s fragments can be edited independently without having to recompute the entire player’s list. The program will utilize the Fragment Manager to both add and remove fragments every few seconds as the program polls the web service for updated game information.

**Conclusion**

To end with, this application, while promoting verbal communication between players, relies heavily on unimplemented network connectivity to facilitate the game of Spyfall. While the basic mechanics of the game are currently in place and operational within the portfolio project, there is much room for improvement, and the game itself will be incomplete without multiplayer support and the development of an external web service. I plan on completing this program as an extra-curricular activity and publishing the source code to github when it is completed. The program as it currently stands does provide the user with a good idea of how this program will be implemented, and it contains the logic and UI required to demo the application without web services. The player role (spy / not spy) is chosen randomly, as are the locations. The application itself serves for an adequate proof of concept and lays the framework for developing a complete application.

References

Android Developer. (2019). Best practices for unique identifiers. Retrieved from: <https://developer.android.com/training/articles/user-data-ids>

Android Developer. (2019). Fragments. Retrieved from: <https://developer.android.com/guide/components/fragments>

Geeks for Geeks. (2019). Volley library in android. Web. Retrieved from: <https://www.geeksforgeeks.org/volley-library-in-android/>