John Sun, Kaya Ota, Frank Li, Eric Vuu
CS 161

Project Proposal - Mobile Game – *Spy*

Description of Product

Overview

Spy (based on the card game Spy Fall) is intended as a party game where a player will create a public game for his friends to join. When everyone has joined the game, a location and a spy are chosen at random. The spy does not know the location while the other players do know it. No one knows the identity of the spy. The objective of the spy is to deduce the location of the group by carefully listening to the questions that other players are asking each other. Other players are trying to figure out who the spy is, by asking questions that someone who knew the location would answer a certain way. If a player suspects that another player is a spy, he or she may start a vote that another player is a spy. The only way that vote does anything is if most the players also think he or she is a spy. Suppose the vote passes and the accused player is not a spy, then the spy wins. If the spy wins, they earn a SpyPoint. If the spy loses, other players gain a SpyPoint each. SpyPoints may be spent on items in the game store such as cool wallpapers and player avatars. Games continue if there are enough players.

Core Features

- Mobile optimized: Does not rely on a web browser to run
- Persistent world: Players can earn rewards outside of the individual games
- Player-run experience: Players decide how the game is run
    - E.g., how long games are, what the game room is titled, etc

Need for the product

This product is a fantastic way for groups of friends to socialize at parties and other gatherings. Games like *Spy* also make judicious use of the bluffing, spies/secret agents and deception mechanics (these mechanics can be found on boardgamegeek). These mechanics are the best ones for a party game. Although there is a web app for this game, no one has created an Android version of it that I am aware of. Our version of the game will add RPG elements to turn the game into a kind of MMORPG (Massively Multiplayer Online Role Playing Game). In that way, *Spy* becomes more interesting because players are more invested in winning the games they play. (Since winning has an effect beyond any one game of *Spy* that they are playing.)

Potential Audience

The potential audience for our game is relatively large. In our audience, we have: people who love playing card games that have been implemented in digital form, people who love playing mobile games in a party/group setting, and people who love playing games like Spy Fall. Anyone can play the game without any understanding of technology or coding. Some of our players will have experience designing and coding games of their own, and this is great because their feedback will help us improve the game faster. We want to get this game in front of as many people as possible, as quickly as possible. (To make the best game that we can.)

Competing Products

There are already several implementations of this card game on the internet. One of these websites implements the game as a mobile-friendly, web app. Players can create a game and then share the game join code. Players with the game join code can join the game, and once the game lobby is full, the game begins.

Our version of the game will be a native, mobile-optimized application that will include more locations and some RPG elements. Users will be able to spend their SpyPoints to purchase goodies from a store.

High-level Technical Design

Our team will program this game in Java using the Android Studio development environment. To support multiplayer functions, we will use MySQL and the Connection class in Java to get and set data between the mobile client and the database.

We will have a table for leaderboards, current games in progress, and player status. The player status tuple might contain a game id attribute. That way, we can know which game the user is playing or if the user has yet to join a game. All players will go into the same table. To make managing our SQL database more manageable, we will utilize the MySQL Workbench software. (We will use this to create our schemas, tables, and perform other database administrative tasks during the development and production phases of our project.) To test our game, we will use the Android Studio emulation of Android on the Nexus 6P with Android 7.1.1 (Nougat).

Resource Requirements

Our project does not require any additional resources in the developmental phase. If we were to make this a real game that hundreds of people play, we would need a dedicated server to run a MySQL database server. We would also have to change the way that our application communicates with the MySQL database. (Since our current implementation plans involve storing database login credentials on the app itself, which is a security risk.) Any challenges related to keeping the project running (during production phase) would be related to ensuring that the service is secure and reliable.

Potential Approaches

There are many ways to satisfy the need for our product. The reason we have chosen to write this application in Android, is because it is often nicer to have a native mobile app rather than a web one. This will provide the user with a nicer experience; one that is custom-tailored for the device they are using. Furthermore, our game will add additional features that may be difficult to add in a web app version.

Assessment of Risks

The biggest risk for our project is the danger that we might not complete it in time. Most of our group members have only a limited experience with MySQL and relational database in general. A good way to ameliorate these risks somewhat, is to schedule regular group meetings where everyone works on the project together. This way, whenever someone has a problem, it can be addressed right away and as a group.

Another risk that we face is the fact that we're using JDBC to connect the Android client to the MySQL server. This means we will have to store database login information on the client. In the future, we can use other technologies to connect our app to relational database. Until then, the risk associated with this method of connection is acceptable.

Next Steps - Looking Forward

Overview

To begin building our project, our team needs to design a user-interface for the activities that our game will require. Someone will have to create those activities and write the Java code and XML required for them to be functional. The game requires MySQL backend support, so someone will have to create the various schemas that will be required. Then everything must be committed, then pushed to our GitHub repository. Regular in-person meetings will make the whole process smoother probably.

Backlog

- Social features: Players should be able to message each other within the app
- Customizability: Players should be able to customize their in-game personas
    - E.g., player avatars and titles, different skins for the app, etc
- Better graphics: Rich, 3D graphics for players and the world they inhabit
- Better scoring system: Game outcomes is currently self-reported; people might cheat