Software Requirements Specification

for

Risky Business

Version 1.0 approved

Sloth Productions, Inc.

January 21, 2014

Prepared by:
Victor Cao
Craig Williams
Kyle Gabriel Maharlika
Jacob Wortman
Angela Niu
Joseph Pecoraro
Jon Claus
Ricardo Ivan Vieitez Parra
Jecob Lehman

Table of Contents

1. Introduction

1.1 Purpose of This SRS

The purpose of this SRS document is to give a detailed description of the specifications and requirements for the “Risky Business” Android software. This SRS will show the purpose and goals of this application, and will also detail the features and structure present within the game, including interactions between two Android devices. It will also explain the constraints and restrictions in developing and using this application, as well as potential issues that developers may run into during production.

1.2 Document Conventions

This SRS follows the guidelines specified in the RFC 2119 document "Key words for use in RFCs to Indicate Requirement Levels". The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in this document. A link can be found in section 1 part 5.

This document also uses terminology and abbreviations specific to the game itself that readers may not be familiar with. See Appendix A for a list of these terms and their definitions.

1.3 Intended Audience and Reading Suggestions

This SRS documentation is intended for the development team of this project, as well as for the instructor(s) of the University of Florida CEN 3031 Intro to Software Engineering course and for any individuals partaking in testing this software. For readers who wish to read the entirety of the document, the Table of Contents provides an order in which to read the document.

Users who are interested in the specifics of this software, such as the evaluators and testers of this project, SHOULD read on to Part 3 (System Features)and Part 4 (External Interface Requirements) to obtain a more detailed explanation of the features and interfaces that were introduced in the earlier sections of the SRS. In addition, section 5 (Other Nonfuctional Requirements) is RECOMMENDED to readers who may wish to know details about safety and security concerns associated with this software. Section 6 (Key Milestones) is also RECOMMENDED to such users if looking for a timeline of the development of this software.

Users who want to simply find a brief overview of this application and what it does should continue reading the remainder of Section 1 (Introduction) as well as Section 2 (Overall Description).

1.4 Scope of Project

The goal of this project is to create an application for tablets with the Android platform, utilizing elements from the popular board games RISK and Settlers of Catan to create a unique game that can be enjoyed by players of many different audiences. The game, titled "Risky Business", will incorporate a hexagonal board layout and resource system similar to that of Settlers of Catan while also tying in the system of building up a combat force and taking over the opponent's resources.

Players will have the option of playing locally with other players on a single tablet. Players will also have the option to communicate with other Android tablets and play with others through Wi-Fi connection.

1.5 References

This SRS incorporates the usage of key words detailed in the RFC 2119 memo written by S. Bradner of Harvard University, which can be found at the below link:

http://www.faqs.org/rfcs/rfc2119.html

2. Overall Description

2.1 Product Perspective

Risky Business is a self-contained Android application geared towards use on tablets. Gameplay is based off of the strategy board games Risk and Settlers of Catan.

2.2 Product Features

1. Strategic Turn-based Gameplay
2. Hex-Based Map with Various Resources
3. Single Player Versus an AI of Varying Difficulty
4. Multiplayer with up to 5 Friends via Local Pass ‘n Play, Bluetooth, and Wi-Fi
5. Collect Resources to Command your Nation
6. Build System to turn settlements into an Empire
7. Combat System to use an Army to Conquer the World
8. Mini-Games available to Gain an Edge in Defeating Your Friends
9. View Rankings and Compare Statistics amongst Friends

2.3 User Classes and Characteristics

Risky Business has a target age group of twelve and up, with the majority of users expected to be between the ages of fifteen and twenty-five. Due to the nature of the game being strategy based, the younger a child is the harder it will be to grasp the concepts of the game, and be able to play a competitive game.

Risky Business users need no technical expertise other than being comfortable using an android device. There is a small learning curve to the game, however after playing through the interactive tutorial, a user should be able to pick up a game and play without much difficulty. The more a user plays the game, the better they are expected to do.

2.4 Operating Environment

Risky business will be *supported* on all tablets running Android 4.X. This includes Android OS 4.0.3 (Ice Cream Sandwich) through 4.4 (KitKat). Although only directly supported on tablets, Risky Business may be available on other devices running Android OS 4.X, including phones and phablets.

2.5 Design and Implementation Constraints

Compatibility is a key design constraint to consider during game production. Risky Business must look good, and play well on a large number of devices with varying screen sizes, pixel densities, and internal hardware. Additional features may also require additional work or services to run as expected. An additional system to handle user data, scores, and rankings may have to be added in order to play over Wi-Fi, and create a global ranking system.

2.6 User Documentation

An interactive tutorial will be available in game to help new users overcome the learning curve of the game. In addition to the tutorial, there will be a “How to Play” option that will explain to the user how to play, while showing them pictures of the different components.

2.7 Assumptions and Dependencies

- An android game engine will be used for graphics rendering.
- Support of multiplayer over Bluetooth will require a tablet to have Bluetooth capabilities.
- Wi-Fi will have to be available to play over Wi-Fi.
- A screen resolution of 1024x728 Pixels is recommended to properly render and view the game.

3. System Features

3.1 Tap to Start Screen

3.1.1 Description

The Tap to Start screen will simply serve as a title screen for the game. This will display the name of our game and may or may not be animated for aesthetic enhancement.

3.1.2 Stimulus/Response Sequences

Stimuli:

- To display the screen, only opening the application is required

Responses:

- The user will tap anywhere on the Tap to Start screen to move to the Main Menu screen, in which they will be able to set up a game to be played.

3.1.3 Functional Requirements

REQ-1: When the player taps the Tap to Start screen, the Tap to Start screen **SHALL** transition to the Main Menu screen.
REQ-2: The Tap to Start screen **SHALL** display the name "Risky Business".
REQ-3: The Tap to Start screen **MAY** be animated for aesthetic purposes.

3.2 Main Menu Screen

3.2.1 Description

This screen will serve as the main point of navigation between various options for the game and will display the following menu items:

- Single Player
- Multiplayer
- Settings
- Credits

3.2.2 Stimulus/Response Sequences

Stimuli:

- Tap to Start screen must be tapped and the Main Menu screen will appear.

Interactions:

- Tap Single Player menu item to open the Single Player Game Setup screen
- Tap Multiplayer menu item to open the Multiplayer Game Setup screen
- Tap Settings menu item to open the Settings screen
- Tap Credits menu item to open the Credits Screen

3.2.3 Functional Requirements

REQ-1: Each menu item **MUST** be able to be tapped on screen.
REQ-2: When the player taps the Single Player item on screen, the Main Menu screen **SHALL** transition to the Game Setup screen.
REQ-3: When the player taps the Multiplayer item on screen, the Main Menu screen **SHALL** transition to the Multiplayer screen.
REQ-4: When the player taps the Settings item on screen, the Main Menu screen **SHALL** transition to the Settings screen.
REQ-5: When the player taps the Credits item on screen, the Main Menu screen **SHALL** transition to the Credits screen.
REQ-6: There **MAY** be a decorative background for the Main Menu screen.

3.3 Setup Screen

3.3.1 Description

This screen will display several settings for the game to be set up, such as:

- A mechanism for inviting players to the game based on username
- How many CPUs the player would like to have
- Which color the player would like to pick
- Entering the players' names
- Start Game

3.3.2 Stimulus/Response Sequences

Stimuli:

- In the main menu screen, tap the Multiplayer menu item to open this screen

Interactions:

- Tap the invite button and type in the name of a user you'd like to invite
- Tap object to choose the number of CPUs
- Tap picker for colors
- Tap text box to enter player name
- Tap the Start Game button to transition to the Game Board Screen

3.3.3 Functional Requirements

REQ-1: Each menu item **MUST** be able to be tapped on screen.
REQ-2: The number of players **MUST** be at least one.
REQ-3: The player **MAY** choose any number of players within the range of numbers we pose.
REQ-4: The number of players and bots **MUST** add up to the number of players chosen by the player to play the game.
REQ-5: The player **MUST** choose a color. If not, the game **SHALL** choose a default color for the player.
REQ-6: There **MUST** be components of a menu for allowing the player to choose each property of the game.
REQ-7: There **MUST** be components of a menu for allowing the player to choose each property of the game. Each of these components **MUST** be tapped to display a window to choose options.
REQ-8: The user **MAY** type a username to invite a user.
REQ-9: The user **MUST** enter a number of CPUs which they'd like to play with.
REQ-10: The Start Game screen **SHALL** transition to the Game Board screen when the Start Game button is tapped.

3.4 Settings Screen

3.4.1 Description

This screen will enable the user to change game settings, such as sound effects volume, game music volume, and more.

3.4.2 Stimulus/Response Sequences

Stimuli:

- The Settings screen can be accessed by tapping the Settings menu item in the Main Menu screen

Interactions:

- Drag the slider to adjust game volume
- Drag the slider to adjust sound effects volume
- Press back button to open Main Menu

3.4.3 Functional Requirements

REQ-1: The Settings screen **SHALL** have an option to change the game volume of the game.
REQ-2: The player **MUST** drag the slider to adjust the game volume.

REQ-3: The Settings screen **SHALL** transition back to the Main Menu screen when the menu item “back” is pressed.
REQ-4: Additional options **MAY** be added to the Settings screen.

3.5 Credits Screen

3.5.1 Description

This screen will display the names and roles of each of the team’s developers and give acknowledgment to whoever deserves it! The text will scroll in the style of the credit reel for a film.

3.5.2 Stimulus/Response Sequences

Stimuli:

- The Credits screen can be accessed by tapping the Credits menu item in the Main Menu screen

Interactions:

- Tap to exit the Credits screen

3.5.3 Functional Requirements

REQ-1: The Credits screen **SHALL** show the names of the people directly involved in the process of developing this game.
REQ-2: The Credits screen **SHALL** scroll through these names until the credits reel is finished or until the player taps again on the screen.

3.6 Game Board Screen

3.6.1 Description

The game board screen will be the main interface once a game is begun. The game board will fill the entire screen. On it, the player will be able to see the game board, all of the roads, settlements, and armies that have been built and fielded by all players, and the number of resources they have. The user’s resources will be displayed in the upper-right hand corner of the screen. The game board will have a certain number of hexagonal tiles based on the number of players; more players will require a larger board, and thus more tiles. Each tile will have a single resource assigned to it in addition to a number from two to twelve. The numbers on the tile represent the sum of the two dice that are rolled; and are involved with the gathering of resources. Initially, there will be no player units or buildings on the game board.

3.6.2 Stimulus/Response Sequences

Upon selecting “New Game” from the main menu, the user will be brought to the game board screen. When it is the user’s turn, they will be prompted to roll the dice. After the user has rolled, they will have the option to build, trade, or attack. There will be buttons for building or trading that the user can select; selecting one of these will take the user to the choices respective screen to perform that task.

3.6.3 Functional Requirements

REQ-1: The game board **MUST** be displayed when the user selects “New Game”.
REQ-2: A tile **MAY** only have one resource assigned to it.
REQ-3: A tile **MAY** only have one number assigned to it.
REQ-4: No player constructs or units may be on the game board when a new game is begun.
REQ-5: The game board **MAY** be randomly generated, meaning there will be no two game boards that are alike in their resource and number placement.

3.7 Trading Screen

3.7.1 Description

The trading screen will be used by the user when the trading of resources is involved. This screen will be displayed when a trade offer is either made by or made to the user. This screen will contain information such as the type and number of resources that are being offered in the trade. The trading player may accept, deny, or modify the trade. The user may trade as much as they like during their turn, provided they have ample resources.

3.7.2 Stimulus/Response Sequences

The user may make trade offers to other players through the game board screen. When it is the user’s turn, they may select the “Trade” button, and then select another player. The user will then be able to offer trade terms for resources through the trade screen. They can offer resources that they own, as well as ask for certain resources in return. If the offer is accepted, the user will give the resources they offered to the player, and get the resources they asked for from the player. If offer is declined, the user may still trade with other players, or the same player who declined. If the user is offered a trade, they may accept or decline.

3.7.3 Functional Dependencies

REQ-1: A player **MAY** only offer trades if it is his turn, and may trade as much as they likes during their turn

REQ-2: A player **MAY** trade resources with a ratio of 4:1 with the bank.
REQ-3: A player **MAY** only trade resources that they hold in their possession.

3.8 Building Screen

3.8.1 Description

The building screen will be used by the player to construct objects on the map. There will be four objects that can be constructed: roads, settlements, cities, and military units. There will be a list of items that can be built, the resources required, and whether or not the player has the necessary resources to build. A player may build as many items they want during their turn, considering they have the necessary resources to do so.

3.8.2 Stimulus/Response Sequences

The user will be able to bring up the building screen when it is their turn, after they roll the dice. This screen will appear when the user selects the "Build" button from the game board screen when it is their turn. The user can build items by selecting the desired item and tapping "OK". The user can then drag the item to a valid building location on the map.

- Roads can be built on tile edges that are already connected to one of the user's roads, settlements, or cities.
- Settlements can be built next to one of the user's roads
- Cities can be built on top of settlement, replacing it.

3.8.3 Functional Dependencies

REQ-1: The user **SHALL** only be able to build an item if they have the required number of resources.
REQ-2: The user **SHALL** only be able to build if it is their turn.
REQ-3: This screen **SHALL** display a list of the items that can be constructed.
REQ-4: A city **SHALL** only be built on top of other settlements.
REQ-5: A settlement **SHALL** only be built next to one of your own roads.

3.9 Interactive Tutorial

3.9.1 Description

Risky Business will include an interactive tutorial that will allow new players to become acquainted with the mechanics and rules of the game. The gameplay mechanics covered include the trading, building, and combat systems. Additionally, the game's interface will be explained, such as where the player can look on screen to find their current resources and gold.

3.9.2 Stimulus/Response Sequences

The user will be able to access the interactive tutorial from the main menu by selecting the respective option. When selected, the player will be brought to the tutorial, which will go through a predetermined sequence of events. A specific concept such as trading will be explained through text. Afterwards, the user will have an opportunity to practice the concept, by going step-by-step through the process.

3.9.3 Functional Requirements

REQ-1: The main menu **SHALL** contain an option for the interactive tutorial that the user can select.
REQ-2: If it is the user's first time playing the game, they **SHOULD** be asked if they want to go through the tutorial. After this initial prompt, they won't be asked again when they start a new game.
REQ-3: The text portions of the tutorial **MAY** be narrated.

3.10 Resource Dispersion System

3.10.1 Description

The Resource dispersion system will automatically add the proper resources to a player's inventory upon each roll of the dice. This is a core component of the overall gameplay, and is of high priority.

3.10.2 Stimulus/Response Sequences

The Resource dispersion system will be activated at the beginning of each player's turn after they roll the dice. Once the dice have been rolled, the resource dispersion system will use the total dice value to disperse the resources.

3.10.3 Functional Requirements

REQ-1: The Resource Dispersion System **SHALL** be triggered on dice roll

REQ-2: One resource **SHALL** be rewarded per settlement on a tile that has an equal value to that of the dice.
REQ-3: Two resources **SHALL** be rewarded per city on a tile that has an equal value to that of the dice.
REQ-4: If enemy military occupies a city or settlement, Resources **SHALL NOT** be dispersed for that city or settlement.
REQ-5: The resource dispersion system **SHALL NOT** interact with the user.
REQ-6: The resource dispersion system **MAY** notify users of the resources they have collected each turn.

3.11 Local Rankings and Statistics

3.11.1 Description

Rankings and Statistics system will display data and statistics for each player that has played on the device. Statistics are stored for the name of each user, as added at the setup of each game. Statistics will be displayed for a number of various items detailed below.

3.11.2 Stimulus/Response Sequences

The Rankings and Statistics will be accessible from the main menu. Upon tapping the selection, a user will be taken to the statistics page. If the number of players on the statistics page exceeds the length of the screen, then the user will be able to scroll through the page to see the other players' statistics. Upon tapping each column heading, a user will be able to sort ascending and descending by each of the statistic criteria.

3.11.3 Functional Requirements

REQ-1: The statistics screen **SHOULD** be scrollable to view more players and statistics
REQ-2: The statistics screen **SHOULD** be sortable by each criteria
REQ-3: Statistics **SHOULD** be available for the following items:

a. Wins/Losses
b. Number of Cities/Settlements Built
c. Highest number of Cities/Settlements Built in a game
d. Number of Cities/Settlements Conquered
e. Highest number of Cities/Settlements Conquered in a game
f. Longest Road built
g. Largest Army in a game
h. Number of each resource collected
i. Most gold collected in a game

4. External Interface Requirements

4.1 User Interfaces

The user will play Risky Business using the touchscreen features on the Android tablets. To navigate away from the application the user shall utilize the devices hardware buttons, detailed in the next section.

REQ-1: Risky Business **SHALL** interface with the user through the tablet's touch screen.
REQ-2: On each screen of Risky Business the user **SHALL** be able to access an options menu from which they can customize sounds, gameplay, etc.
REQ-3: The user **SHALL** be able to exit and return to the Risky Business interface anytime using the physical home button.

4.2 Hardware Interfaces

Risky Business will be physically limited to Android tablets, as standard cellular phone screens are not large enough to meet the interface needs of the application. Risky Business will utilize the standard Android hardware buttons for basic features involving navigating to and from the application.

REQ-1: Risky Business **SHALL** be limited to Android tablets.
REQ-2: Risky Business **MAY** accept input through the tablets home and settings buttons.

4.3 Software Interfaces

Risky Business will utilize multiple software interfaces to satisfy networking, operating, and graphical tasks. Risky Business is a multiplayer game and will have options for multiplayer or single player game types. As previously stated the application will run on the Android open mobile operating system. It will work in conjunction with the Android system API's so that it may momentarily suspend and restart operation smoothly. A graphics engine will not be required due to the 2D nature of the application; however, animations may be added to certain menu options or gameplay actions.

REQ-1: Risky Business **SHALL** be run on the Android 4.2.x mobile operating system.
REQ-2: The application's 2D graphics, including menu's and game board, **MUST** be preloaded into memory.
REQ-3: Risky Business **MAY** implement menu animations.
REQ-4: Risky Business **MAY** implement a variety of animated gameplay actions.
REQ-5: Risky Business will feature a pass and play multiplayer game mode.
REQ-6: Risky Business **MAY** feature a networked multiplayer game play option.

REQ-7: Risky Business **MAY** feature a single player versus AI game mode.
REQ-8: Risky Business **MAY** be able to sleep when signaled through the Android operating system by the user.
REQ-9: Risky Business **MAY** be able to transition out of sleep and resume previous gameplay.

4.4 Communications Interfaces

Risky Business has the capabilities to be played both online and offline through its different game modes. Communication with online opponents will be necessary for game play features like trade. In the offline version communication need not be done through the application.

REQ-1: Risky Business **SHALL** have the ability to be played offline with local opponents.
REQ-2: Risky Business **MAY** be able to utilize the available cellular Internet connection for online multiplayer.
REQ-3: Risky Business **MAY** be able to utilize the available Wi-Fi Internet connection for online multiplayer.
REQ-4: Risky Business **MAY** give users the ability to communicate with online opponents.

5. Non-Functional Requirements

5.1 Performance Requirements

5.1.1 Video Performance

Since Risky Business is presented as a two-dimensional board game it does not require a real-time video engine. However, Risky Business does have 2D textures and assets, and these **MUST** be preloaded into memory to facilitate access when the application is launched.
Some board game actions **MAY** be accompanied by animations or other graphical effects. These animations **MUST** finish within 3 seconds.

5.1.2 Network Performance

In those cases where Risky Business requires network access, either to the Internet or to a local network, the requests **MUST** timeout in five seconds. Failed requests **MUST** be reattempted three times, with each reattempt done three seconds after the failure was detected, after which the user **MUST** be prompted to conduct subsequent attempts. Multiplayer network games **MUST** be considered abandoned if the user is disconnected for at least three minutes.

Network activity **SHOULD** happen asynchronically by using threads and **MUST** be carried out such that it does not block or otherwise impair users' interaction with any part of the application other than those waiting on the network communication. The application **SHOULD** provide non-invasive feedback on the status of any network activity, such as by displaying a legend on a region of the screen left for application messages.

5.1.3 Error Conditions

The application **MUST** catch all the exceptions that it throws, including those that may be thrown by system or API calls. In the event of an uncaught or unknown exception the application **MUST** terminate without attempting to save any data.
The application **MUST** save the current game data periodically (which **SHOULD** happen at least once every five minutes) on a known location. Upon start, the application **MUST** prompt the user whether he or she wishes to resume the game from the current game data, if available.

5.2 Safety Requirements

5.2.1 Health Considerations

Risky Business carries the risks associated with the operation of a handheld computer and a touch screen, such as the carpal tunnel syndrome and the computer vision syndrome. The application **SHOULD** display a warning after it detects that the user has been using the application continuosly for over an hour. If a warning is displayed, it MUST briefly explain the importance of taking a break.

5.2.2 Financial Considerations

Risky Business **MAY** allow the user to make “in-app” purchases to have access to certain features. The users **MUST** be clearly promted about the transaction they are making, including the cost. The application **MUST** have a setting to require a password for making these purcheses.

5.3 Security Considerations

5.3.1 Physical Security Considerations

Risky Business **MAY** allow “Pass’N’Play” games, that involve that sharing of a common device amongst several users. Since handheld devices are often used to store personal information such as pictures or email messages, it is possible for a person to gain unauthorized access to this information upon his or her turn to play. Two measures SHALL be implemented to mitigate this risk.

1. The game will feature a soundtrack that **MUST** be played continuously for the duration of the game when allowable. However, users **MUST** be able to turn this

feature on and off throughout the game and similarly **MUST** be able to use the system sound volume controls.

2. A warning about the dangers of "Pass'N'Play" **MUST** be issued prior to starting one such game. This feature **SHALL NOT** be have an opt-out setting.

Additionally, the "Pass'N'Play" gameplay carries the risk of having unauthorized game movements, such as if a user plays for two consecutive turns, or plays a whole round. When the game soundtrack is activated, a recognizable sound **MUST** be played upon ending each player's turn.

5.3.2 Design Considerations

Risky Business SHALL access only its application directory and, in particular, **SHALL NOT** have access to any user data that is not stored there. Similarly, Risky Business **SHALL NOT** have access to the phone camera, the microphone, the SMS service nor the phone dialer. Risky Business **SHALL NOT** execute or call any code that is not part of its own interface, part of its libraries or part of the Android API.
When an internal data stucture is found to be misconstructed Risky Business **MUST** terminate immediately.

5.3.3 Network Considerations

All network traffic **MUST** be sent through a TLS pipe, and it **MUST** not include any information that is not part of the relevant client-server protocol. In particular, no circumstances allow for the transmission of user files over a network protocol.
Network communication **SHOULD** be handled by a separate process whenever possible.
Network messages **MUST** be parsed safely, and invalid or inconsistent messages **MUST** be discarded (although the application **SHOULD** notify the sender of the message about the failure to understand the message).
When playing in multiplayer mode, a secure random number generation system **MUST** be used. If Risky Business is played in client-server mode, the server **MUST** be the one providing the random numbers. If Risky Business is played in peer-to-peer mode, a collective, transparent and simultaneous random number scheme **MUST** be used for those gameplay operations that require random numbers, such as rolling dice.

6. Key Milestones

Milestone	**Deadline**	**Comment**
SRS document	1/21	Content should be done by respective members of the group by 1/20 so we can prepare the document for formatting.
Finalized Interface Design	1/24	Designs should be done on this date for revisions.
Finalized Algorithm Design	1/24	Designs should be done on this date for bug solving.
SDD document	1/27	Having final designs done by 1/24 should be plenty of time to put together the document.
Sprint 1 deliverables	2/24	End of Sprint 1. Start debugging project on 2/17
Spring 2 deliverables	3/31	End of Sprint 2. Start debugging project on 3/24
Interface & Algorithm Implementation	4/14	Project should be completed by this date. This milestone gives us 2 weeks to test and debug.
Software Festival	5/1	Completion and demo of project

7. Key Resource Requirements

Major Project Activities	**Skill/Expertise Required**	**Internal Resource**	**External Resource**	**Issues/ Constraints**
Design rules and guidelines to our game.	General knowledge of board games such as Risk and Settlers of Catan.	Jon and Joseph have solid knowledge in this field	Internet. Game manuals for said games.	Lack of game knowledge of some members.
Design the game board	Design & graphics experience.	Angela has digital design experience.	Internet. Lynda.	Limited amount of team members have experience.
Design interfaces	Design & usability experience; familiarity with Android GUI design	Angela has some UI design background. Members have aesthetic sense.	Internet. Android Developers community.	Scheduling conflicts. Screen resolution on tablets.
Design algorithms	Knowledge & experiences with data structures and algorithms	All members have taken data structures and algorithms	Android Developers community. Internet examples.	Potential scheduling conflicts
Implementing an algorithm	Familiarity with the Android development environment	Team members are familiar with Java programming	Android Developers community. Internet examples	Potential scheduling conflicts.
Implementing the interface and game board	Familiarity with Android GUI implementation	Team members have experience with Java.	Android Developers community. Internet examples.	Potential scheduling conflicts.

8. Other Requirements

Since our project may be time consuming to demo (typical games last from ~20 minutes to an hour) we may have to create an edited video showing how our game works. This would require some type of video editing work and experience. There are Lynda tutorials on how to complete this task.

The video will be created via AfterEffects and posted on Youtube or Vimeo upon completion.

9. Requirement Change Management

The following is the change management process that our team will be using for the SRS to reflect changes in project scope and requirements:

- **Identify:** our Software Engineering Facebook group, GroupMe, email
- **Log/Update:** Shared Google Drive for documents
- **Evaluate:** self-evaluation at the start of our weekly discussion session

In order for a proposed change to go into effect, there must be a two-thirds majority within the team (six out of the nine). Any opposing opinions can be voiced to the team as an appeal. This ensures that all decisions regarding any updates or changes to the project are done democratic fashion.

All changes to requirements will be submitted to the team via email or in person.

10. Restrictions, Limitations, and Constraints

10.1 Unfamiliarity with Our Development Platform

The members of our team have not worked with Android extensively in previous projects. None of us have extensive experience in GIT – although some of us do have a reasonable amount of experience – and very few of us have delved into the world of developing aesthetics in software. This project, therefore, is somewhat ambitious given the skillset we do have. Although we would love to have a beautiful interface within the Android platform and an amazing code base, it is more reasonable to say that we will simply perform to the best of our ability in developing the graphics, interface, and code base.

For the time being, we will keep the details simple and try to implement the core functionality as well as we can.

10.2 Time Constraints

Given that we only have 14 weeks to complete the project at hand, if we have underestimated the amount of time for development of any given component, the number of features we can reasonably implement must be limited. Thus, in our document, we have labeled some developments as optional, and in the future, we may label more developments as optional as more problems arise.

11. Appendix A: Glossary

BOARD

The surface on which the game is played upon, as in ‘board game’. It is comprised of hexagonal TILES in a configuration determined by the number of players.

CITY

A game piece that occupies a VERTEX. The user can exchange a given amount of resources in order to upgrade a settlement to a city. Its color denotes ownership.

EDGE

An edge of a HEX, possible shared between two of them.

HEX

A hexagonal piece with which the board is comprised of. Each one is assigned a given resource and number, which determines the in-game production associated with it. The number is written within a circle at the center of the figure and the resource is represented by the texture of the shape. See Appendix B for visual clarification.

MILITARY

A game piece that occupies a VERTEX. The location of a given military unit may change in game but it shall always occupy a vertex while in play. Its color denotes ownership and its number corresponds to its value in-game.

RESOURCE

A type of currency which is exchanged for certain actions in-game. There are six types: brick, ore, grain, wool, lumber, and, gold.

ROAD

A game piece that occupies an EDGE. The user can exchange a given amount of resources in order to construct a settlement at a viable location. Its color denotes ownership.

SETTLEMENT

A game piece that occupies a VERTEX. The user can exchange a given amount of resources in order to construct a settlement at a viable location. Its color denotes ownership.

TILE

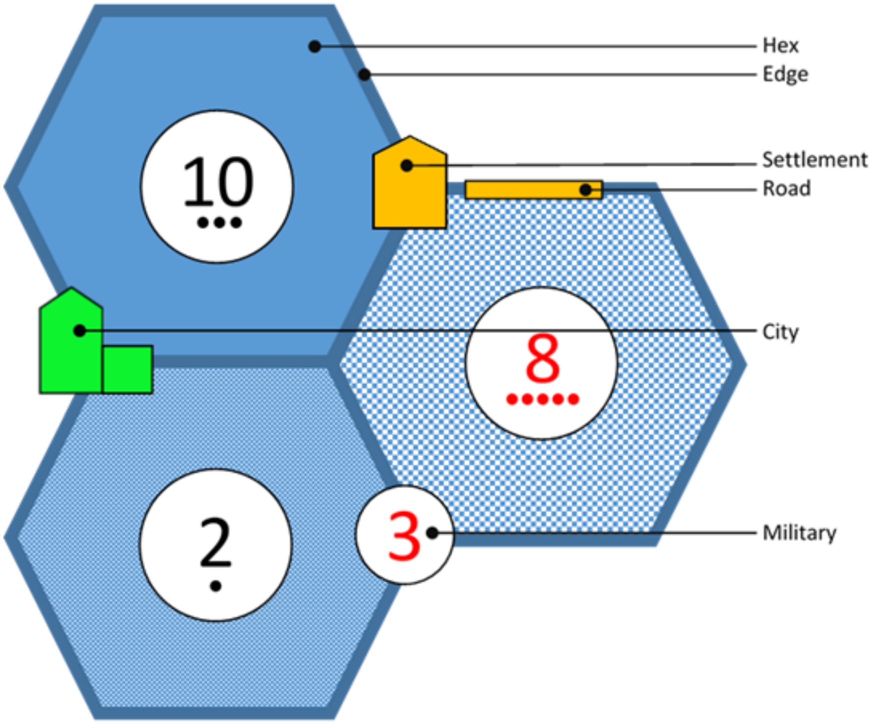
An autonomous piece used to construct the board. See HEX.

VERTEX

A vertex of a HEX.

12. Appendix B: Game Board/Figures/Diagrams

Example of the board structure:



13. Appendix C: Issues List

NOTE: *The list below is dynamic and therefore is subject to change*

13.1 Pending decisions to be determined:

- Multiplayer over Wi-Fi
  - If so, game servers
- Animation, such as:
  - Dice rolls

    - Movement of armies
- Artificial intelligence for single-player use
    - What is the feasibility of accomplishing this?
    - How “smart” should it be?
- Mini-games
    - Should we do this?
    - If so, how could they be implemented so that they would not take away from the main experience?
    - What kinds of games could we feasibly do?

13.2 Needed information:

- Devices (in this case, tablets) at end of term software festival
    - Will devices be provided?
    - If not, how many devices should be made available on the day?
- Cost of servers
    - Only if we choose to go with a Wi-Fi multiplayer model