Table of Contents

Table of Contents .............................................................................................................. ii
Revision History .................................................................................................................. ii

1. Introduction ..................................................................................................................... 1
   1.1 Purpose ...................................................................................................................... 1
   1.2 Document Conventions .............................................................................................. 1
   1.3 Intended Audience and Reading Suggestions ............................................................ 1
   1.4 Product Scope ........................................................................................................... 1
   1.5 References ................................................................................................................ 2

2. Overall Description ........................................................................................................... 2
   2.1 Product Perspective .................................................................................................... 2
   2.2 Product Functions ...................................................................................................... 3
   2.3 User Classes and Characteristics .............................................................................. 3
   2.4 Operating Environment ............................................................................................. 4
   2.5 Design and Implementation Constraints ................................................................. 4
   2.6 User Documentation ................................................................................................. 4
   2.7 Assumptions and Dependencies .............................................................................. 4

3. External Interface Requirements ...................................................................................... 5
   3.1 User Interfaces .......................................................................................................... 5
   3.2 Hardware Interfaces .................................................................................................. 5
   3.3 Software Interfaces .................................................................................................. 5
   3.4 Communications Interfaces ..................................................................................... 6

4. System Features .............................................................................................................. 7
   4.1 Access to L'Or Acadie Game ..................................................................................... 7
   4.2 Choose How Many Monitors ................................................................................... 7
   4.3 Connect Phone to Computer Running the Game ...................................................... 8
   4.4 Pan Left and Right between the Monitors ............................................................... 8

5. Other Nonfunctional Requirements ............................................................................... 9
   5.1 Performance Requirements ..................................................................................... 9
   5.2 Safety Requirements ............................................................................................... 9
   5.3 Security Requirements ............................................................................................. 9
   5.4 Software Quality Attributes .................................................................................... 9
   5.5 Business Rules ....................................................................................................... 10

6. Other Requirements ....................................................................................................... 10

Appendix A: Glossary ......................................................................................................... 11

Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Purpose

The game “L’Or De L’Acadie” is seeking various improvements for increased player pleasure. One of these improvements is to incorporate an interstitial view which means integrating a handheld augmented reality view that shows a model of the game world. Currently this type of mobile application does not exist. By the end of this project, an intensive interstitial viewer for a mobile device will be running as part of the “L’Or De L’Acadie” game. The project is being built for the customer, Derek Reilly who has outsourced a series of projects using the “L’Or De L’Acadie” game.

1.2 Document Conventions

Bold face and bullet points are used to introduce a new topic or a specific subject matter. The rest of the document is written in regular Helvetica.

1.3 Intended Audience and Reading Suggestions

The SRS document is intended to inform the reader of the intended approach to creating an interstitial viewer. The reader will get a general understanding of the product including its functional and non-functional requirements and its features. This document will begin with an introduction of the product that is being developed and what uses cases are involved. From there the document will discuss any assumptions that have been made and what constraints currently exist. Four different types of interfaces has been defined and discussed in the third section of this document. All of the features that will be included as part of the product are outlined in great detail so the reader gets a good understanding of how it will work. Finally the SRS document will specify any non-functional requirements that relate to the application and that are important to many readers.

The SRS document has been developed for many readers of different backgrounds. Different sections of this document will prove to be more important to some than others. Developers will want to focus on sections 2, 3, and 4. Other teams working with the same game will likely want to focus on section 2. Project managers will want to understand the product (section 2) and will want to review the non-functional requirements (section 5) and ensure they are implemented. The user will want to understand how the product works so will be interested in section 2 and 4.

1.4 Product Scope

Our group will be creating an interstitial viewer for the game “L’Or De L’Acadie”. This application will run on a mobile platform (Android) and will be completed by the week of March 26, 2012. The interstitial viewer will provide a panoramic view of the game environment. The view of the environment will be mapped onto what is seen on a set of larger screens. By doing this, the areas of the user’s which are normally blocked by the frame of the screen will be displayed. The interstitial viewer allows the player to look around from his current stationary position. Player movement will not be included in the scope of this project.
1.5 References


2. Overall Description

2.1 Product Perspective

The interstitial viewer for Android mobile device platforms is to function as an extension to the pre-existing game “L’Or De L’Acadie”, a simulated 3-D world rendered with the Open Wonderland environment. Our interstitial viewer is intended to serve as a separate module which both integrates seamlessly with the existing product, but may also be transferrable for future applications.

As a “game world viewer”, our application is responsible for rendering areas of the 3-D world not immediately visible on a set of fixed-point monitors. The objective is to allow the user to “view” and render the segments of the game world not currently visible on-screen, by means of traversing the space between sets of monitors—corresponding to actual space in the game world—and rendering this view on to an Android mobile device. This will provide players with a 360-degree view of the game world, without the restrictions typically imposed by having a fixed number of perspectives as one would typically find in a multi-monitor setup.

The Open Wonderland Java environment is unable to render efficiently—or in some cases natively—on the Android or iOS operating systems. Therefore, it is necessary that we render the Open Wonderland software on a remote PC and utilize VNC Free Edition 4.1 to transmit this data to our mobile device. VNC is a remote control application that allows a mobile device to interface directly with a remote computer; this will serve as our primary means of interfacing between the software and hardware components of our products.
### 2.2 Product Functions

<table>
<thead>
<tr>
<th>Class of Use Cases</th>
<th>Use Cases</th>
<th>Description of Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use cases related to setup of Android devices</td>
<td>Installation of VNC / TwinView</td>
<td>Installation of necessary VNC or TwinView software to an Android mobile device.</td>
</tr>
<tr>
<td></td>
<td>Link devices</td>
<td>Connection between mobile device, computers, and monitors.</td>
</tr>
<tr>
<td></td>
<td>Login</td>
<td>User login to Open Wonderland game world.</td>
</tr>
<tr>
<td>Use cases related to viewing game world</td>
<td>Render game world</td>
<td>Render 3-D world from perspective of player.</td>
</tr>
<tr>
<td></td>
<td>Panning horizontal camera</td>
<td>Provide rendering along horizontal axis (no vertical axis).</td>
</tr>
</tbody>
</table>

#### Interstitial Viewer Use-Case Diagram

![Interstitial Viewer Use-Case Diagram](image)

### 2.3 User Classes and Characteristics

- The user should be familiar with the concept of a first or third-person perspective within a rendered 3-D game world. The concept of “turning” your character to reveal different areas of the game world will be intuitive to all users, however some basic familiarity with working in a 3-D environment will be required to use the device properly.
Setup and installation of rendering software on an Android mobile device will require a significant level of technical skill. Regular users of the software should be considered as distinct entities from the technicians that will be required to implement our system.

Instruction on the use of an Android mobile device as an interstitial viewer will require minimal expertise on the part of the user. Manipulating the phone to reveal areas of the game world “displayed” between sets of monitors will be intuitive.

2.4 Operating Environment

“L’Or De L’Acadie” operates within the Open Wonderland Java-based environment. The interstitial viewer we shall develop for this product will be designed for the Android mobile operating system, although potentially such a technology can be adopted for the iOS operating system in the future. To facilitate the rendering of Open Wonderland, which is not supported natively on Android systems, we will be utilizing VNC Free Edition 4.1 and TwinSpace, both of which will bridge a connection between a remote PC (running Open Wonderland) and our mobile Android device (rendering the game world). It will be necessary for our Android device to record distances between anchored pairs of monitors; additionally, our software must be receptive to accelerometer and compass data provided by the mobile device.

2.5 Design and Implementation Constraints

- The interstitial viewer only supports a single user at a time.
- Network or internet connections are required to interface between the phone and a remote PC.
- “Markers” between anchored pairs of monitors are necessary to provide a framework in which the interstitial viewer will render the game world.
- Android devices vary in capabilities / technology supported, and thus we cannot guarantee universal access to our application across all Android platforms.
- Android operating system is regularly updated by Google, and future iterations may not be compatible with current versions of VNC, TwinSpace, or Open Wonderland.
- The software will not be maintained by the developer following release, which may result in compatibility issues in the future.

2.6 User Documentation

- Procedures for setting up connections between Android mobile devices and a remote PC, through use of VNC and TwinSpace, will be provided as part of this project’s documentation.
- Documentation on the usage and functionality of Open Wonderland is available on the developer’s web page.

2.7 Assumptions and Dependencies

- Assumptions made regarding the version number of Open Wonderland; we are operating under the assumption that Open Wonderland will remain relatively unchanged from its current state, however the software is being continually updated.
- Similar risks of future incompatibility exist with VNC and TwinSpace, both of which may prove difficult to interface later in the project if their respective version numbers are changed.
- We approach the project with the assumption that the pre-existing game code for “L’Or De L’Acadie” is easily malleable and receptive to modular development, such as our interstitial viewer. Furthermore, we expect the game code to have been well-documented to allow an easy transition between understanding the pre-existing code and writing our own code.

3. **External Interface Requirements**

3.1 **User Interfaces**

Once the program has been launched, the user will interacting primarily with the Android device running VNC, and thus vicariously with the PC running “L’Or De L’Acadie”. Since the Android device will be receiving video from the PC, whatever interface is present in the game itself will also be the interface encountered by the player.

Since the program will be designed with simple mechanics and ease of use in mind, the GUI need not be overly complex. Whatever interface is used should in fact be kept readable and minimalist in order to accommodate the smaller screen size of the Android device. Possible added functionality to the GUI would be a button which re-centers the camera view in the case of lost synchronization.

3.2 **Hardware Interfaces**

At the centre of the interstitial viewer’s functionality is the connection between the PC running the game “L’Or De L’Acadie” and the Android device controlling it through VNC. As such, there are no specific hardware requirements for the personal computer used—only that

1. It must be compatible with some version of VNC, and
2. It must be capable of running “L’Or De L’Acadie” through a session of Open Wonderland.

As the program will allow the in-game camera to be manipulated through panning of the Android device, the Android device in question must

1. Contain motion-sensing hardware; an accelerometer, compass, and gyroscope, and
2. Also be VNC-capable.

The input created by the device’s panning will be sent to the connected PC through VNC, while a stream of the game screen will be fed back to the Android device through the same channel.

3.3 **Software Interfaces**

The product will be comprised of interaction between the following software products:
1. **Open Wonderland (version 0.5)**
   An open-source Java 3D framework used for the creation of virtual worlds.

2. **“L’Or De L’Acadie”**
   A collection of Open Wonderland modules that together compose the game known as “L’Or De L’Acadie”.

3. **TightVNC (version 2.0.4)**
   The PC-side VNC component. This program allows a computer to be configured so that it may be controlled remotely by an Android device via networking.

4. **android-vnc-viewer (version 0.5.0)**
   The Android-side VNC component. This program allows a compatible Android device to remotely connect to and control a specific PC. It contains built-in functionality which allows the PC’s mouse to be moved via panning of the smartphone. It requires Android OS version 1.5 or higher.

5. **Windows 7 OS**
   Though TightVNC is compatible with all versions of Windows from Windows 2000 and beyond, the device currently intended to be used in the project is running the Windows 7 operating system.

The main communication channels between the software components are as such:

- Mouse movement from the Android device to the PC, through TightVNC. The Android device must be capable of moving the PC’s mouse onscreen through panning movement.
- Camera movement within the game based on computer mouse movement. The program must be altered so that the in-game camera moves smoothly and at a realistic pace when the Android device is panned from side-to-side.
- Video streaming from the PC running “L’Or De L’Acadie” to the Android device through the android-vnc-viewer. This video must be rendered as smoothly as possible in order to improve the gaming experience for the user.

### 3.4 Communications Interfaces

In order for the product to be functional, the PC running “L’Or De L’Acadie” and the Android device must be connected through VNC. This system uses the RFB protocol to allow one device (the Android phone) to remotely control another (the PC) through a network. As such, the two devices must be able to share a network in order for the program to function. The configuration of VNC on the smartphone side requires the input of a) the PC’s IP address and b) a short password set by the owner of the PC prior to the connection. As such, while the program may not be controlled by any device immediately, it is not limited to only ever being controlled by a single Android device.
4. System Features

4.1 Access to L’Or Acadie Game

4.1.1 Description and Priority

This system feature allows the user to connect to a Wonderland server that has the modules of the game loaded into the server. The priority of this is high because without it the user will not be able to get access to any of the game feature or other system feature we implement.

4.1.2 Stimulus/Response Sequences

1. Go to the Wonderland server website.
2. Run the jnlp file to start Java Web Start.
3. Put in credentials if any is needed.
4. The system then gives access to the world to the user.

4.1.3 Functional Requirements

REQ-1: Must have access to a Wonderland server. If the user does not have access to one, they should find one they can access or download it for themselves.
REQ-2: The modules for the game need to be loaded into the server. If they are not then the user should load them into the server.

4.2 Choose How Many Monitors

4.2.1 Description and Priority

This system feature allows the user to choose how many monitors they will use to pan between. The priority for this system feature is medium because the risk trying to implement this is 7 because it might be harder then we think to implement it.

4.2.2 Stimulus/Response Sequences

1. Once they have access to the game they must choose how many monitors they want to use from one of the menus.

4.2.3 Functional Requirements

REQ-1: Must have the code that will show a menu that allows the user to choose how many monitors they want. If they do not have access to this menu they must load the right module into the server.
REQ-2: Must have the right algorithm so the feature will be able to determine where these monitors will be depending on how many the monitors the user chooses. The feature must be aware of how many monitors the user has so it can display an error message if the user chooses more monitors than they have access to.
4.3 Connect Phone to Computer Running the Game

4.3.1 Description and Priority
This system feature allows the user to view the game over their phone from the computer using a vnc viewer. The priority of this feature is high because without this feature the user will not be able to access the main feature of this part of the game.

4.3.2 Stimulus/Response Sequences
1. Use the phone to connect the computer’s viewer, the computer should know when a user is connected.

4.3.3 Functional Requirements
REQ-1: Must have TightVNC downloaded on the computer running the game and have android vnc viewer downloaded on the android phone. If they are not downloaded then the user needs to download them.
REQ-2: Must have the code so the game realizes that the user has connected and be able to recognize when is within the parameters of the monitors. If the user is not within the parameters, pause the game and show a message telling the user to be within the parameters.

4.4 Pan Left and Right between the Monitors

4.4.1 Description and Priority
This system feature allows the user to pan between the monitors using their phone to show the areas that are not being shown on the monitors. The priority is high because this is the main feature of the game and without it there would be no game.

4.4.2 Stimulus/Response Sequences
1. The game needs to notice when the phone is within the parameters of the monitors and where the phone enters the parameters, so it can show the correct area.
2. When the user pans left the game needs to know where the phone is going and what information its need to display to the phone.
3. When the user pans right the game needs to know where the phone is going and what information it needs to display to the phone.
4.4.3 Functional Requirements

**REQ-1:** Must have the correct algorithm to detect where the phone is in the beginning and then display the correct information. If the phone is not within the parameters then the game should pause and display a message telling the user to be within the parameters.

**REQ-2:** Must have the correct algorithm so the game knows when the phone is panning left or right and how much information the game should be displaying to the phone.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The objective of this project is to develop an interstitial viewer to display images of what is seen on a bigger screen of the environment, mapped to the position of the mobile device relative to the screen. When in use, the viewer will be moved horizontally by the player, panning across the main screens. As the device is moved, the image should follow what is seen on the main screens directly in front of it, also providing map views where the main screens’ frames block the image. To do this, it is necessary for the viewer to be capable of real time image rendering, as the image must continually update to seamlessly match the movement of the device relative to the main displays, using the accelerometer and gyroscope sensors of the device. The rendered images should also have image quality identical to that of “L’Or De L’Acadie,” so the interstitial viewer integrates flawlessly into the game play and feels like a natural extension of the game.

5.2 Safety Requirements

During use of the game, players experiencing eye strain should take a break from playing to avoid further strain and/or possible damage. Some players may be prone to seizures because of flashing or bright animations. It is therefore not recommended people who have a history of epileptic seizures or photosensitive seizures play “L’Or De L’Acadie.”

5.3 Security Requirements

The interstitial viewer, running as an application on the Android device should need no additional information other than collected data gyroscope and accelerometer data from the device for use in its image rendering. Wireless security settings on the device must allow for the application to connect to the game server so it can feed information to and from the server. Otherwise, access to the user’s personal information from other apps, i.e. calendar information, email, contacts, photos, etc. is under no circumstance necessary and should be considered a breach of privacy in the event it occurs.

5.4 Software Quality Attributes

The interstitial viewer should be able to work on any VNC-compatible Android device with an accelerometer and gyroscope. Installing the interstitial viewer should be a simple process, ideally identical to downloading and installing any application from the Android App Market. The software
should run smoothly without crashing or freezing, regardless of any game parameters, i.e. map, number of players, player type, etc. It should have a very intuitive interface that is easy to learn so the player can focus mostly on the game itself. Optimally, the architecture of the interstitial viewer might also be flexible enough to be easily adapted for an iOS device. At the end of the project, all source code, documentation, as well as any other material related to the development of the game may be made freely available to other developers of “L’Or De L’Acadie,” where it may be used as reference or for further development.

5.5 Business Rules

Any individual may have use of the interstitial viewer for academic or personal use. As the project is part of the development of the open-source project “Open Wonderland,” code, documents, or other materials used for this project cannot be used for commercial purposes. However, others wishing to further develop the code after the project’s completion are free to do so.

6. Other Requirements

Currently there are no other known requirements for the project; however this may change in the event of unforeseen circumstances encountered during the duration of the project.
Appendix A: Glossary

**Android** – an operating system designed for mobile devices (i.e. cell phones, tablet computers) by Google, Inc.

**Android device** – any device running Android. In this document, synonymous to "smart phone running Android."

**VNC** – (Virtual Network Computing) a cross platform remote control software created by RealVNC, Inc. It allows a user to fully control a computer remotely through use of another computer or mobile device.

**Open Wonderland** – an open source Java project which allows for the creation of virtual worlds that allows users to communicate in a collaborative environment. Originally started by Sun Microsystems in 2007 and later released to the public as an open-source project, it is build on the massively multiplayer gaming platform Project Darkstar.

“**L’Or De L’Acadie**” – a video game built using Open Wonderland as a platform. Currently under development.