Software Requirements Specification

for

Treasure-Based-Braille

Version M approved

Prepared by Md Nasiful Haque

And Marlin Lachance

February 5th, 2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Marlin Lachance	Feb 23, 2018	Midterm Submission	vM
Md Nasiful Haque	Feb 5, 2018	Lab Task Draft	v1.0

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Text-Based Braille” (TBB) Scenario Editor software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Intended Audience and Reading Suggestions

This document or the Software Requirement Specification is intended to be read by other developers, EECS 2311 Professor, Teaching Assistants, Users and Testers of the software. The rest of the SRS (Software Requirement Specification) document is organized nicely and you would have no trouble going through all the subtopics sequentially, but you can always jump towards a section that you would like to specifically read.

1.3 Product Scope

The purpose of this software is to allow educators, including those who are visually impaired to help create interactive story scenarios that can be shared and/or taught to students and help them learn to read Braille. This software was made in part of a University project which was intended to help both Visually Impaired and non-Visually Impaired educators be part of an interactive teaching experience with their students.

1.4 References

The entire project can be found on this GitHub link:
https://github.com/marlinla/Treasure-Box-Braille

2. Overall Description

2.1 Product Perspective

This piece of software is going to emulate a real TBB (Treasure-Box-Braille) Player.

2.2 Product Functions

The Treasure-Box-Braille software is going to be used for a variety of reasons:

- *It can be used to create the flow of the story/scenario; the educator can use this piece of software to create and ask questions which they can get an answer to from his or her students.*
- *It will allow the educators to record audio to be put in their story scenarios*
- *It will allow the educators to save the scenario in an appropriate format*
- *It will allow the educators to test the scenarios they created and/or prepared which they can test to see if it works.*

2.3 User Classes and Characteristics

Anticipated users of the Treasure Box Braille Scenario Editor software primarily include educators regardless of visual disability. These educators have a need to educate students using Braille.

2.4 Operating Environment

- *This software has been fully tested for the Windows 10 Operating System.*
- *Preliminary testing has been performed on the Linux distribution centOS.*

2.5 Design and Implementation Constraints

Software is primarily constrained by time and manpower. Early in our development, one engineer left our group, leaving the two remaining engineers to cover the work to implement this application. The application must be able to be used with a screen reader. Consequently, NVDA was chosen as the supported screen reader.

2.6 User Documentation

An User Manual for the Treasure Braille Box Scenario Editor can be found at https://github.com/marlinla/Treasure-Box-Braille/blob/master/User%20Manual.pdf

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>
No Assumptions and/or Dependencies
The following libraries have been used in the Treasure Box Braille Player bundled with the Treasure Box Braille Scenario Editor;

- *JavaSE-1.8*
- *cmu_time_awb.jar*
- *cmu_us_kal.jar*

- *cmudict04.jar*
- *cmulex.jar*
- *cmutimelex.jar*
- *en_us.jar*
- *freetts-jsapi10.jar*
- *freetts.jar*
- *jsapi.jar*
- *assertj-core-3.6.2.jar*
- *commons-io-2.5.jar*

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>
The swing GUI widget toolkit for Java was used. It is not the most modern widget toolkit for Java. But it's a proven toolkit and will provide a consistent look and feel with the bundled Treasure Box Braille Player.

3.2 Hardware Interfaces

- *Mouse and keyboard are used for conventional inputs.*
- *Monitors are used for conventional outputs.*
- *Speakers are used for conventional playing of scenarios and for screen reader output for visually disabled users.*

3.3 Software Interfaces

The Treasure Box Braille Scenario Editor interfaces with the Treasure Box Braille Player and Windows operating system using Java JRE 1.8

3.4 Communications Interfaces

The Treasure Box Braille Scenario Editor does not communicate using network interfaces.

4. System Features

These are the requirements for acceptance of our software project.

4.1 Create the flow of the scenario (ask questions, receive answers)

4.1.1 Ask questions

Our Treasure Box Braille has a question template for prompting the reader with questions.

4.1.2 Receive answers

Our Treasure Box Braille has a question template that accepts answers in the form of 4 possible answers. If the answer in incorrect, the player will ask the question again until the correct answer is received.

4.2 Record Audio

Our Treasure Box Braille Scenario Editor has a question template that records a .wav audio to be used in a scenario file. In addition, the user can choose a .wav file from their file system. Also, these audio files can be removed, and the audio template can be deleted entirely.

4.3 Save the scenario in the appropriate format

Our Treasure Box Braille does not have a save scenario button. This was a oversight due to time limitations. We could have quickly implemented a saving function in the reverse order that we opened scenario files. Instead we focused on finishing the GUI elements to showcase our vision of a template based Treasure Box Braille Scenario Editor.

4.4 Test the scenario using the provided software

Our Treasure Box Braille provides an option to play an opened scenario file using the Treasure Box Braille Player by pressing the Play Scenario button.

5. Other Nonfunctional Requirements

5.1 Version Requirements

The submissions specifications state that the release to be marked for the midterm is vM on our team's github repository https://github.com/marlinla/Treasure-Box-Braille/

5.2 Deadline Requirements

The deadline for the EECS 2311 midterm submission is Febuary 23, 2018.