

Software Requirements Specifications

TreBBA: Treasure Braille Box Authoring

Version 1.0

EECS2311 Submission: Group 10

Amaan Vania
Amer Alshoghri
Arsh Punia

Table of contents

| | |
|-------------------------------|---|
| Part 1: Application Overview | 3 |
| Part 2: General Description | 4 |
| Part3: Specific Requirements | 6 |
| Part 4: Acceptance Test Cases | 7 |

Part 1: Application Overview

The purpose of this document is to give a detailed description of the requirements for the “Treasure Braille Box Authoring” (TreBBA) Software. Not only will this document serve the purpose of clarifying the software’s capabilities, it will also serve as a guide for its design and the acceptance test cases.

1.1 Overview

TreBBA helps educators teach braille to learners. Educators can save time by writing scenarios using TreBBA which are interpreted by the Braille Box, which then acts as an interactive medium of learning.

TreBBA is an offline, desktop application and can be deployed on desktop machines. In terms of a brief overview, the software has been written in Java and has been designed so that it can be used by visually impaired instructors.

As will be elucidated in later sections, instructors can set a cell size, relay instructions, and create questions to help keep learners engaged using text based or voice-based media.

TreBBA assumes some understanding of Braille to be used effectively.

A short Braille Tutorial can be found here: <https://brailletutorial.com/tutorial/>

The applet enables the user to conduct a variety of operations on a standard Treasure Box Braille device.

The following sections illustrate the functionality and the design of the software.

The second chapter provides details on TreBBA’s functionality, and user experience.

The third chapter dwells into the specific requirements and different user classes with examples.

The fourth chapter expands upon acceptable test cases for the software.

Part 2: General Description

This section provides a brief overview of the applet. The functionalities of the system will be elucidated and brief examples of user interaction.

2.1 Product overview

TreBBA is an authoring app that lets educators create, edit, and run different scenarios on Treasure Box Braille (henceforth referred to as TBB) devices. There are multiple functionalities that help the user create scenarios and edit them. TreBBA is designed to be able to be used by visually impaired individuals.

The software interacts with the user via a standard interface which support the various functions of TreBBA through buttons and a text area.

2.2 Product Functions

Following are some of the things that the user can do with TreBBA:

1. Create Scenario: Enables the user to create a new scenario
2. Edit Existing Scenario: Enables the user to edit an existing scenario
3. Cells and Buttons: Upon choosing to create a new scenario, the user is automatically prompted to enter the number of cells and buttons that they need to be able to work with
4. Display: Displays a character on the braille cell by parsing the input by the user
5. Add text: Adds text to the file, and speaks it out when the scenario is run
6. Ask Question: Prompts a question to the learner. TreBBA user has to specify the question and the buttons that they would like to activate as answer keys
7. Specify Correct Answer Key: Specifies the button the learner needs to press for the correct answer
8. Begin Correct Answer Explanation: Allows the user to start an explanation if the learner has pressed the right key.
9. End Correct Answer Explanation: Ends the explanation for the correct answer
10. Specify Wrong Answer key: Like “Specify Correct Answer Key”, except for the wrong answer
11. Begin Wrong Answer Explanation: Allows the user to start the explanation if the learner has pressed a wrong key
12. End Wrong Answer Explanation: Allows the user to end the explanation for the wrong answer
13. Import Sound File: Allows the user to import a sound file and for it to be played as a part of the scenario
14. Delete Field: Deletes the selected field from the chosen file
15. Save Scenario: Enables the user to save a scenario
16. Play Current File: Launches a TBB emulator and enables the user to play the selected file

2.3 User Characteristics

TreBBA has been built with a Braille Educator in mind. In addition to that, it also has been built to work in conjugation with the NVDA Screen Reader to aid visually impaired educators.

As developers of TreBBA, we have assumed that the software would be used to create scenarios to enable the learning process of students of Braille through an interactive medium of TBB.

TreBBA supports functionality of asking questions, raising pins, displaying characters on the cell, and playing sound files.

As the user can be visually impaired, TreBBA supports keyboard and mouse navigation functionality.

2.4 Constraints

As TreBBA has been built with a visually impaired user in mind, it is not graphics intensive and focuses the majority of its resources on the core functionality of creating scenarios.

It also assumes a basic understanding of Braille and experience with TBB devices for character input.

2.5 Apportioning of Requirements:

There are some features that are still being worked on and would be a part of a later release. These include a better question-answer interface and to make the software more user friendly for people who have previously not worked with TBB.

3. Specific Requirements

3.1 Interface Requirements:

The interface is one of the critical components of TreBBA. A user should be able to create a new file and open an existing file to edit. The interface should encompass all of the functionality of TreBBA and enable the user to add, edit, and delete fields from the scenario file.

For all the features to work, the user is required to either select an existing file or create a new one. For example, if the user clicked on ‘Add field’ without loading or creating a file, they would be prompted to do so first.

3.2 Hardware Requirements:

The user is required to run TreBBA on a desktop machine.

3.3 Functional Requirements

3.3.1 The user should be able to create new scenarios and save them

The user should be able to create a new scenario file for the emulator, starting off by specifying the number of cells and buttons they want, followed by the suite of functionalities offered by TreBBA. This can include adding text, displaying certain pins, playing sound files and recorded questions and their answers.

3.3.2 The user should be able to edit an existing scenario

The user should be able to load an existing scenario into TreBBA and add or delete some features depending upon need.

3.3.3 The user should be able to play sound files

The user should be able to import a sound file into the scenario for the learners to be able to hear.

3.3.4 The user should be able to ask questions and record answers

The user should be able to ask questions, record answers, tell if the answers are right or wrong, and explain further concepts

3.3.5 The user should be able to read out simple text to the learners

The user should be able to input text into the file such that the TBB device can read it out and ensure an interactive experience for the learners

3.3.6 The software should be usable by visually impaired users

The user's visual impairment status should not be a factor to decide if they can use TreBBA or not.

4. Acceptance Test Cases

| Feature | Expected | Output |
|--|---|---|
| Create new scenario file with 1 cell and 4 buttons. File should be named "test.txt" | A new scenario file name "test.txt" with 1 cell and 4 buttons | Scenario file created with 1 cell and 4 buttons created. The file is named "test.txt" |
| Edit Existing File | File gets loaded into the interface in a readable format | File gets loaded into the interface in a readable format |
| Display 11100000 on Cell Number 0 | Popup asking for cell number and the character to be displayed | /~disp-cell-pins:0 11100000 appended to the file |
| Add Text "Test Text" | "Test Text" to be written to the file | "Test Text" written to file |
| Ask Question "Are you human? Press 1 if yes, press 2 if no" | Question: "Are you human? Press 1 if yes, press 2 if no" with buttons 1 and 2 activated | Appends question: "Are you human? Press 1 if yes, and 2 if no" and activates buttons 1 and 2 |
| Set Button 1 as the button for the right answer and say "Congratulations on being human" | Button 1 is set as the right answer and text "Congratulations on being human" would be appended to the file | Scenario File recognizes Button 1 as the right answer and appends the text "Congratulations on being human" to itself |
| Set Button 2 as the button for the wrong answer and say "I believe you are an alien" | Button 2 is set as the wrong answer key and text "I believe you are an alien" is appended to the file | Scenario File recognizes Button 2 as the wrong answer key and appends "I believe you're an alien" to itself |
| Import and play sound file "AfricaByToto.wav" | Scenario file would recognize and play sound file | "AfricaByToto.wav" is appended to the end of the file and is played when the scenario is loaded into the simulator |

References:

The following document was used as a rough template for the requirements document:

http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf