

Software Requirements Document

Treasure Box Braille Authoring Application

EECS 2311 Group 3

1. Introduction

This section will give an overview of the product, its scope, potential use cases and acceptance cases.

1.1 Purpose

The purpose of this application is to be a tool to view, edit and create scenarios for the Treasure Box Braille (TBB) device. Through it, clients will be able open and edit existing TBB scenario files, and create new ones from scratch.

The target audience for this project are primarily teachers and educators of visually impaired students. The program will function as a tool for them to dynamically teach students braille through TBB scenarios.

1.2 Scope

The “Treasure Box Braille Authoring Application” is a desktop application which enables users to view, edit and create scenarios for the TBB in an intuitive and simple way. The scope of the project is as follows:

- View TBB scenarios that are in .txt format
- Identify if a text file is a TBB scenario
- Edit existing TBB scenarios that are in .txt format
- Create new TBB scenarios
- Save any changes made to an existing TBB scenario
- Save any created TBB scenarios as a text file that can later be reopened
- Test existing or created TBB scenarios using a software simulator
- Record audio
- Have accessibility support for visually impaired users

Mobile support is not planned at this point in time. Accessibility support is only available if the user has a screen reader already installed.

1.3 Use Cases

- Teachers creating scenarios for visually impaired students as a way to learn braille
- A dynamic way to create scenarios for teaching braille
- Viewing and editing existing scenario files
- Writing books/text in braille for those who are visually impaired

1.4 Acceptance Test Cases:

- Successfully opens and correctly displays a scenario text file
- Allows editing and saving of an imported scenario text file
- Throws error if imported file is not a scenario file
- New scenario files can be created, edited and saved
- Scenario files can be tested using virtual TBB
- Record audio from a user
- Usable by someone who is visually impaired

This section describes the functional and quality requirements of the TBB Authoring Application. It contains detailed descriptions of the system's features.

2.1 User Interfaces

A user opening the Authoring Application should be presented with a window containing the following buttons:

- Create Scenario
- Import Scenario
- Export Scenario
- (Greyed out) Test Scenario
- Create Audio File

See Figure 1.



Figure 1 - Opening Page



Figure 2 - Scenario Features Panel



Figure 3 – Importing/Exporting Files

2.1.1 Create Scenario

If the Create Scenario button is selected, the user should be prompted to first enter the number of TBB cells they are using, then the number of buttons they would like to include in their scenario. After both of these inputs are completed the application will update to show the Scenario Features panel (see Figure 2), which is used to add new elements to the scenario. The Test Scenario button should become active.

2.1.2 Import Scenario

If the Import or Export Scenario buttons are selected, a new window should open, allowing them to select the file they wish to import, or the location to export the scenario as a text file (.txt) to (see Figure 3). Once a Scenario file is imported, the Test Scenario button should become active.

2.1.3 Test Scenario

If the Test Scenario button is selected, the application should test whatever the current content is (either an imported or created scenario). It should run the scenario through a virtual TBB Simulator for the user to verify that this is the desired output.

2.1.4 Create Audio

If the Create Audio button is selected, the user should be prompted to enter the name of the audio file they wish to create and the duration of the audio file in seconds. Recording will start after they press yes to the confirmation shown after the previous two values are entered correctly. If they press no, a message should appear verifying that the recording was cancelled.

2.2 Functional Requirements

This section contains descriptions