Authoring Application

Requirements Document

Group 2

EECS 2311

Final Submission

August 21st 2018

1. Table of Contents

2. Purpose

The purpose of this document is to detail the requirements of our Authoring application. First it will go over a brief description of the needs of the user, what the system does to address the needs of the user, and finally about the specific test requirements that the application shall meet.

Authoring Application is an application that enables users to create unique scenarios to be used in conjunction with the Enamel Braille Cell simulator. The application shall allow users to create “scenario files”: files that contain interactions that will be run through the Braille Cell simulator. Through use of the application, users will be able to edit and modify these interactions to create unique and interesting lessons to their students.

2.1. Definitions

In this document, the following terms are defined as follows:

Scenario: *an interactive activity or lesson to be performed with the braille cell.*
Interaction: *any of the features outlined in the scenario file format document.*
User: *a user of the authoring application software. E.g. a braille instructor*
Student: *a user of the braille cell; i.e. a student learning braille through the braille cell, one whom is the intended user of the lessons created through this app.*
Braille Interface: *the braille cell simulator*
Authoring Application (AA): *the application where instructors can create or edit scenarios*
Entry Point (EP): *the initial window that pops up upon executing the program*
Interaction List (IL): *a list containing the various interactions in a given scenario*
Configuration Panel (CP): *a pane in the AA that allows users to configure the interaction based on their input. E.g. specifying text to be read to a*
Blind*: *those who are blind or with low vision*

2.2. Background

As defined by the American Foundation For The Blind, “Braille is a system of raised dots that can be read with the fingers by people who are blind or who

have low vision" (*What is Braille?,* 2019). It is not a language, rather a code that can be used by those who are blind* to read in their language.

Traditional braille lessons include using a braille writer (Fig. 1) to type out lessons and then guide the students through the different symbols. As with any kind of learning, having an interactive and effective lesson is essential to better long-term comprehension and understanding. A well-designed authoring program is important to help empower teachers to create more effective lessons. Our system is designed to meet these needs of our users, and provides teachers with the tools needed to create these kinds of lessons.

Fig. 1 - a Braille writer (image from http://www.dancingdots.com/prodesc/Cosmo.htm)

2.3. Application Overview

The application can be run on any personal computer running Mac, PC, or Linux with a monitor, keyboard, and mouse. The system is also built to be accessible for visually impaired users.

This software is meant to be used with a Braille interface which will be the target device for the lessons to be experienced through. This software comes with a simulator that allows the user to test out their lessons with.

3. User Requirements

3.1. Product Perspective

From a broader perspective, this application shall provide the capabilities to a user to create and edit scenarios to be ran on a Braille Interface. It will be a

desktop application that will run on any Java-application compatible computer. There will be a user interface for the user to interact with their scenario, and the user shall have the capability to test their scenarios in an integrated simulator as well.

This section breaks down the use cases that are required for our software to be able to perform. The user in these cases holds the perspective of an instructor/teacher of the Braille system, looking to create an interactive lesson for students to learn with the use of braille cell(s).

3.1.1. User Interfaces

The requirements of the user interface (the Authoring App) can be broken into four sections: the entry point, the authoring app, the interaction list, and the configuration panel.

3.1.1.1. Entry Point (EP)

The EP shall be a window that opens up when a user double-clicks the executable file. This window shall contain the option to create a new scenario file with no interactions in it, or open an existing scenario file to edit the contents of that scenario.

Upon selection of one of the above options, the EP shall open the Authoring App with an empty Interaction List (if a new scenario), or loaded with the interactions from a past saved file (if opened an existing scenario).

If the user exits the EP, then the application will close.

3.1.1.2. Authoring App (AA)

The AA is the window which contains the Interaction list and the Configuration Panel. Within the AA, users can also save or test their scenario by clicking the ‘Save’ or ‘Test’ button. Upon clicking the ‘Save’ button, the application shall create a text file in the Factory Scenarios directory which can be ran through the simulator. Upon clicking the ‘Test’ button, the application shall open up the Braille Interface, and run the current scenario in the simulator.

3.1.1.3. Interaction List (IL)

The IL is a list that contains all of the interactions in the currently opened scenario. The user shall be able to reorder the interactions currently loaded into the list, and shall be able to add or remove interactions currently in the list by pressing the corresponding buttons.

The IL also shall control which interaction configuration is currently being displayed in the configuration panel; by clicking on different interactions, users will be able to view the details of each interaction in the Configuration Panel.

3.1.1.4. Configuration Panel (CP)

The CP is the panel that contains the details for a selected interaction out of the interaction list. It will show the configurations corresponding to the currently selected interaction and will allow users to edit the configurations of that interaction. Upon editing an interaction, the CP will track these changes, but not save them to the current scenario file until the user presses the save button.

3.1.2. Hardware Interfaces

The application is meant to be used with a keyboard, mouse and monitor. The application can also be used through just a keyboard with a text-to-speech software (e.g. Apple Voiceover, Orca) to guide the user through the application.

3.1.3. Software Interfaces

This application shall be able to interface with the given Braille Simulator, allowing users to test their scenarios with different numbers of braille cells, buttons, and scenarios.

3.1.4. Communication Interfaces

The application shall allow users to record their voice into a ‘Voice Interaction’, which will be an interaction for a Scenario that will playback the recorded file.

3.1.5. Memory Constraints

There should be at least 200MB of free storage space in the directory that the application is run in. Larger scenarios will require more space to save, and more audio files will require more memory as well.

3.2. Additional System Requirements

3.2.1. Text-to-speech capabilities

The application shall be able to read out text written by the user to the students.

3.2.2. Screen reader compatible

The application shall be compatible with both Apple VoiceOver, and Orca.

4. Specific Requirements (Test Requirements)

4.1. Scenario Requirements

4.1.1. Open Scenario File

The user shall have the capability to open an existing, properly formatted, scenario text file into the application. This will be tested by verifying the correct display of a scenario that was made previously.

4.1.2. New Scenario File

The user shall have the capability to create a new scenario file, specifying the number of cells, buttons, and title of the scenario. This will be tested by creating a new scenario and verifying the scenario properties were kept track of when saved.

4.1.3. Edit Scenario File

Upon opening or creating a new scenario file, the user shall have the capability to reorder, add, and remove interactions from the scenario. The user shall also be able to select interactions in the scenario and edit the contents of the selected interaction. This will be verified by opening a

previously created scenario adding an interaction, updating one of the interactions, moving it, and deleting another interaction. Then these actions will be verified by the tester.

4.1.4. Run Scenario File

The user shall be able to run their scenario in the simulator in order to test their scenario out. This will be verified by running an existing scenario file and verifying that the simulator has run nominally.

4.1.5. Save Scenario File

The user shall be able to save the scenario they have created or edited within the application to a valid scenario text file. This will be verified by attempting to save a scenario file through the application.

4.2. Interaction Requirements

4.2.1. Read

The read interaction shall be an interaction where the simulator reads out the user's specified text to the student. The user shall be able to edit the text read out to the student. This will be verified by creating a new read interaction and writing sample text into it, and finally ran and correctly read out by the simulator.

4.2.2. Voice

The voice interaction shall be an interaction where the simulator plays a recorded message to the student. The user shall be able to record from within the application and delete/re-record if they choose to do so. This will be verified by creating a new voice interaction and recording a small sample recording, then checked by testing it in the simulator.

4.2.3. Pause

The pause interaction shall be an interaction where the simulator pauses for a user-defined number of seconds. This will be verified by timing the length of the pause interaction in a sample scenario and ensuring it aligns with the specified number of seconds in the interaction.

4.2.4. Display Braille

The display braille interaction shall be an interaction where the simulator displays a user-defined Braille symbol on a cell of the user's choosing. This will be verified by seeing whether the simulator displays the correct braille symbol based on the user generated interaction.

4.2.5. Keyword

The keyword interactions links the application between different parts of the scenario file, linked by a user-specified keyword. This will be verified by attempting a skip/keyword interaction in order to skip over a certain read interaction. If the read interaction was not read, then it will be determined successful.

4.2.6. Skip Button

The skip button interaction is used to link a button press with a certain keyword interaction This interaction is used in combination with the user-input interaction. This will be verified by creating a test scenario where only a specific read interaction is read when button 1 is pressed, and only a specific read interaction is read when button 2 is pressed.

4.2.7. Skip

The skip interaction is the second half of the skip/keyword interaction, where users can use this to skip to a different section of an interaction. See Keyword's test case for this interactions acceptance case.

4.2.8. User Input

The user input interaction is how users can pause the program and wait for a student to push a button in the simulator. The program will skip to the linked skip-button interaction. This will be tested as outlined in the skip-button section.

4.2.9. Reset Button

The reset button interaction is used to clear the listeners for all the buttons in the simulator. This will be tested by verifying that the simulator doesn't respond to further button presses after a skip-button/user-input combination is used with a reset-button afterwards.

4.2.10. Cell Clear

The cell clear interaction is used to clear the display of all the braille cells. This will be verified by inserting a reset button interaction into a scenario where display braille was called before it, and viewing the braille-cells clearing their raised pins.

5. References

Scenario Format File (contains descriptions of different interactions: https://wiki.eecs.yorku.ca/course_archive/2017-18/W/2311/_media/scenarioformat.pdf

"What Is Braille?" *What Is Braille? - American Foundation for the Blind*, www.afb.org/info/living-with-vision-loss/braille/what-is-braille/123. Accessed Feb 19 2018