SOFTWARE REQUIEREMENTS DOCUMENTATION

TBB Authoring App

Group No. 13

Arshdeep Singh Saini
Daniel Kozlovsky
Kirollos Kamel

TBB Authoring App

Purpose

The core purpose of this software is to facilitate educators of Braille in an institutional environment. It will aid teachers to formulate and manufacture routines, called **Scenarios**, for a special piece of composed hardware to play on. This special hardware will then provide a medium for learners to engage in creative activities through user input and get relevant feedback from it, all designed by the educators.

Scenario Creation Features

1. Braille cell manipulation

The software includes commands to allow the client to interact with the Braille cells present on the hardware. The client can either set each pin on the cells individually or set an English alphabet on the cells in the Scenarios they create. Individual pins on the cells can be lower and raised by the client through commands in the Scenario.

2. Voice Commands

The software allows the client to write commands to order the hardware to speak text, ask questions, and provide auditory feedback based on the user's input. These voice commands can be triggered at any point in the Scenario's lifecycle as desired by the client.

3. Audio playback

The software allows the client to include audio in their Scenarios to deliver auditory feedback for the users. The user, through the software, has the freedom of either including existing audio files or record them on the spot.

4. Command Management

The software provides additional commands to manage other commands as well. The client can use these to repeat a set of commands at a point in the Scenario or skip certain commands entirely.

5. Lifecycle management

The software gives the client several tools to help manage the Scenario's lifecycle. These tools allow the client to pause the program, reset the Scenario and clear all Braille cells at moments the client sees most fit.

6. Scenario management

The software allows the client to quickly reopen and improve previously made Scenarios. The client can easily import existing Scenario files and edit them as they see fit. When the Scenario editing it done, it can be conveniently exported as a file by the client to be worked on later or to be simulated and played onto hardware.

7. Accessible interface

The software has a robust interface to facilitate an engaging experience for both normate users through an intuitive GUI, and for visually disabled users through screen reader support. The software gives the client contextual feedback through screen reader services to allow a hassle-free experience.

Use cases for TBB Authoring App

1. **Educational tool**

The software can be used to teach the Braille language to kids and illiterate people through creative activities and routines. These routines can be distributed and freely edited to improve their effectiveness and refine the experience. The software provides teachers with many tools to build very detailed Scenarios to be used in classrooms and helps craft new ways of learning that's more engaging to the learner.

2. **Auditory games**

The software can be used to build intricate games that use Braille cells and audio playback to interact with the players. The games can have several Scenarios played conditionally according to the games rules to provide an engaging experience. The buttons can be used to take player input while playing the games.

3. **Daily News displays**

The software can be used as a medium to deliver News and other daily info to users through daily updating Scenario files that periodically rotate between different topics to be displayed like Sports, Politics, Local and international News etc.

4. **Smart home device**

The software can be used to control smart home appliances and even as a smart home device itself using Braille cells as a display and Scenarios triggered using available buttons.

Acceptance cases for TBB Authoring App

1. **Reusability**

Feature: Importing and exporting scenarios

When the client makes a new Scenario using the Scenario editor, it is saved used an Export feature which converts the Scenario into a commands file for the hardware to read. However, the client can import this file and edit, delete, and modify the commands at will. This gives the client a freedom of continuously revising, editing, and previewing the Scenarios that are made.

Steps:

- ⇨ Open the software and select *Create New Scenario*
- ⇨ Build the Scenario by adding commands
- ⇨ When done, export the Scenario using *Export Scenario*
- ⇨ Preview the Scenario on the hardware
- ⇨ When revision of the Scenario is required, open the program, and select *Import Scenario*
- ⇨ The Scenario will be read from the file and opened in the Scenario Editor
- ⇨ Edit the Scenario as required
- ⇨ Repeat the above steps until the resulting Scenario is satisfactory

2. **Robust accessibility**

Feature: Screen reader support

The software has full support and usability for visually impaired users. Each element of the GUI is given proper names and descriptions that the client can listen to and use understand their purpose. This allows the client to have an unrestricted user experience.

Steps:

- ⇨ Open the software and select *Create New Scenario*
- ⇨ Hover over different elements of the GUI to trigger the screen reader
- ⇨ The screen reader will read the name and description provided by the software
- ⇨ As the mouse hovers over different elements and windows, the screen reader gives auditory feedback to the client

3. Command management

Feature: Command reordering and removal

The software allows the client to reorder and remove commands made previously. This helps the client edit the Scenario to better suit their needs or to fix mistakes and inconsistencies in their Scenarios.

Steps:

- ⇨ Open the software and select *Create New Scenario* ***OR*** *Import Scenario*
- ⇨ Select a command to be reordered/ removed from the commands list on the left
- ⇨ Use the controls on the right to move the selected command Up or Down, or remove it
- ⇨ The commands list will be updated to reflect the user's action