Dr. Bil Tzerpos

EECS 2311

Authoring App Documentation

Group 6

Braille Authoring App Documentation

Team Members: Tedi Papajorgji, Kevin Suarez, Jessanth Harrymanoharan, and Ibrahim Manjra

Table of Contents

Needs of the Customer

Mission Statement:

The purpose of this project is to aid in the teaching of Braille to blind children. We seek to achieve this goal by creating a hardware and software system that will display Braille characters and contain buttons for the user to interact with the system. It will then be up to the software and each individual customer to create unique scenarios that will capture the children's attention and make it fun to learn the Braille alphabet.

Expectations:

The Customer at this stage expects the Authoring app to be able to aid them in the creation and implementation of the unique scenarios mentioned in the mission statement. This is the final stage of the software portion of the project so it needs to show the customer a completed product. The application needs to be easy to use and yet retain a premium quality to it. The user interface will be simple to use and intuitive; yet it will provide the customer with the means to accomplish any of the complex tasks required in the making of a unique teaching scenario. There will be two main focuses for the Authoring app, the first being to let the customer create scenarios from scratch on their own and the second to let them import existing ones from their computer.

Integration:

In order to deliver the customer with a complete product, it is necessary to integrate with the Authoring app all of the other work that was accomplished in both the Simulator app and the Player app. The Player app allows the customer to select and play existing scenarios so in this case the Authoring app will allow the customer to create new scenario files that will then be fed to the Player app for the children to use and learn Braille. Another option here is also to just allow the customer to import into the Player app existing scenarios that they have not necessarily created. While the goal of the project is to have this be an interaction between software and hardware, the Simulator app here is also important because it can help the customer visualize the outcome of the scenarios without needing to have the actual hardware component.

Required Features

Overview:

The main objective of the Authoring app is to build upon the work already done with the Player app, as an extension in the functionality and scope of the project. In order to do this we have the Authoring app communicate with the Player app and the Simulator app in order to test created or imported scenarios and visualize their outcome and performance. The Authoring app needs to be able to allow users to create scenarios that are then played and simulated using the other parts of the project. It also needs to be able to create the files that will become the stories fed to the Player app in order to help the children learn Braille.

User Interface:

Once the customer opens the application, they will be introduced to an easy to use and intuitive user interface. The user interface will be minimalist but retain a premium look and feel. In order to not overwhelm the customer the minimum amount of buttons will appear on screen at any point so that they can focus on their current task with ease. By having buttons lead to different selection screens we allow the user to be naturally guided in their creation process so that they are only exposed to the options that they need at that moment. It will be easy to go back and forth between any of the sections in the user interface should the customer want to redo or undo something. In addition the user will have the option of when to save a final file so that the Player app is not confused with erroneous input. By having the customer work on a scenario one step at a time the user interface will make it easy to create unique scenarios or import existing ones.

Audio Files:

The main feature of the Authoring app is to allow the customer to incorporate audio files that have been created with other application or to create those audio files using their computers built in microphone. These will be options in the user interface for the customer to select which one he will be using. If the customer wants to incorporate existing audio files into a scenario then the user interface will guide him into selecting the folder on his computer where the file is stored and then selecting the specific file he wants to use. It will then verify that it is an appropriate file and prompt the user for its insertion into a scenario. Should the customer need to create an audio file the Authoring app will then locate the users microphone and display a screen that will allows the customer to select when to start recording. The same screen will also display a stop recording, pause recording, play recording and save or delete recording button. In this way the customer has complete control in this creation process. After a recording has been made the customer has the option of inserting it into a scenario.

Implementation:

Once having either created or imported an existing audio file, the Authoring app will prompt the user to select to which of the available scenarios to insert that audio file or to save it into a new scenario. In addition an options screen will present the user with choices on where in the scenario to add this audio file. Apart from the audio file integration there will also be a menu for the customer to select which Braille cells to be displayed at certain points in the scenario. This will allow for the customer to create the tests that will confirm the children's choices in the story. This menu will also prompt the customer to select the correct interpretation of the Braille cell. After all of these choices are made, the Authoring app will create the file that will be fed to the Player app for the scenario to be played. The Simulator app at this point will also be able to show what the hardware should be doing when given the newly created scenario file.

Acceptance Test Cases

Test Case Scenario: User double clicks on the jar file with the York University logo.

Expected Behavior: The user is presented with a medium sized window that shows the user interface for creating scenarios.

Warnings: The jar file has been tested to work on Windows, Linux and Mac OS's so there should not be a problem opening the file.

Test Case Scenario: The user will input a file name, select a directory to save this new file and enter the number of braille cells and buttons that they require.

Expected Behavior: Once the user has selected all of those options and clicked on "Start Editing", the user interface will change to the next stage in which there will be new options for the user to make a custom scenario.

Warnings: Should the user not make a selection for any of the options they will be warned in a pop up message what they are missing. The directory selection screen also only allows the proper directories to be selected.

Test Case Scenario: The user will select the option to add a block to start creating a scenario.

Expected Behavior: A new window will pop up with options for the user to customize the scenario block.

Warnings: This is the only option when making a new scenario so the user will be guided to this in the beginning.

Test Case Scenario: The user will be able to choose the type of voice that will be used to create a text to speech sound file.

Expected Behavior: A new window will pop up after the voice and text to speech selection is made were the user will type in the desired text. When done the user will be brought back to the previous page.

Warnings: A voice is always preselected and the user only has the option to change the voice so there should not be a problem there. The text to speech is then done by the Player app.

Test Case Scenario: The user will select to add an existing sound file to the scenario.

Expected Behavior: A new window will open for them to select the directory where their sound file is located and then select the specific sound file they want added to the scenario. When done the user will be brought back to the previous page.

Warnings: They will only be able to select appropriate sound files as any other files will be greyed out and unselectable.

Test Case Scenario: The user will select to record the audio file that will be used in the scenario.

Expected Behavior: A new window will pop up that will display that the program has successfully identified the on board microphone. There will also be a filed for the user to name this audio file that will be created. The only button on screen will be the start recording button which will the changed to the stop recording button after recording has been initialized. After the file is created the user will be brought back to the previous page when clicking done.

Warnings: Once the user has recorded a file but failed to name this file a warning window will pop up reminding the user that a file name needs to be given in order to save the recording.

Scenario Test Case: The user will choose what each of their selected buttons will do when a braille cell is shown.

Expected Behavior: The user will select the button from a wheel which will have all of the buttons that were defined in the beginning of running the application. Once a button is selected the user will have a choice of action corresponding to that button. This can be done with all the buttons once they are selected.

Warnings: If the user does not select an action for the button then the button will do nothing.

Test Case Scenario: The user will select the finish block once having done any of the above test cases.

Expected Behavior: The scenario block will be saved and the user will be able to see it and have the option to add another block that will be concatenated after this one or finish the scenario.

Warnings: The user will be able to see any warnings given from this screen to avoid confusion and allow the user to fix any mistakes.

Justification:

The above test cases are reasonable acceptance test cases because they describe the process that a user will go through when using the Authoring app to create scenario files. They display the actions of the user along with the expected behavior of the software and the countermeasures implemented to catch any errors. For more extensive information on testing and the handling of mistakes made by the user, refer to the testing document. For simple user interactions the above acceptance test cases prove that the user should be able to use the app without encountering any problems and successfully make a scenario file.