Requirements Document

By: Zaeem Israr, Kevin Banh, Amir Khademi

Contents

Introduction

This document describes the required features of the Authoring App. This app is the third part of a three-part project in creating software for a hardware device used to help kids read Braille. In the second part of the project, a Player App was created to play stories for the users. The "stories" are text files with special keywords that allows display of Braille characters onto the hardware, input from the hardware (and based on the input, go to a different branch of the story) and more. The purpose of the Authoring App is to allow educators to create scenarios as they see

Definitions

For the purpose of this document, scenarios are text documents (.txt extensions) with keywords in them. Scenarios are played by the Player App and based on certain keywords, do certain actions. A list of keywords can be found at https://wiki.eecs.yorku.ca/course_archive/2016-17/W/2311/_media/playerfileformat.pdf.

Customer Requirements

1. The program should allow users to create scenarios that can be played by the Player App without knowing the underlying keywords via a GUI.
2. The program should have a GUI.
3. The program should allow users to create scenarios with branches based on user input.
4. The program should run on Windows and Linux.
5. The program should have audio support. This means the GUI should allow scenarios to incorporate audio files that have been created with other apps. It should also provide a way to record audio clips during running the program, store the clips and allow them to be used in the creation of the scenario. All audio clips should have a .wav extension as required from the Player App File Format.

6. The GUI should be able to support a majority of the keywords in the Player App File Format not limited to repeating certain text based on user input, changing voice available when reading, creating branches in story based on user input, playing audio, pausing playing of the scenario for a certain amount of time and playing audio clips.
7. The users should be able to generate the scenario based on what the users put on the GUI. Basically this means actually creating the .txt for the Player App.
8. Users should be able to specify the amount of buttons and cells the Braille hardware has.
9. Edit a previously made scenario.

Acceptance Test Cases

The general idea of how the GUI works is the user opens the program. There should be three options. The first option should be allowing the creation of the scenario, the second option is a way to record audio and the third option is a way to edit the scenario. Upon clicking the first option, the program will go into scenario creation mode. There will be a button to add nodes to the GUI. A node consist of multiple fields and a unique ID. There should be a field allowing user to apply a "tag" which is a keyword in the File Format, a field to provide more information as required by the specific tag and the text that is to be spoken. A scenario is a sequence of these nodes and is generally created by following a linear progression starting from the highest node to the bottom. For more complex scenarios like branches, there will be fields allowing specification of how the nodes should be linked. After finishing adding information to the GUI, there should be a button to generate the scenario. Pressing it will create a .txt file (scenario) based on how the nodes are structured (branches), text inside each node and tag in the GUI.

Test Case Scenario: Creating Scenario with no Keywords

Users should first click on "Create a Scenario." Users should then fill out two textboxes prompting for how many cells and button the hardware has. Users should click on a button to create a node. Do not alter the tag field (default no keyword). Click on the larger textbox below and type in the text the story should have. To add more nodes, simple click add node and another node should appear below the first node. If the story has a linear progression, do not worry about the fields specifying how the nodes should be connected. Continue adding nodes (and filling out the fields inside a node) until satisfied. Users can generate the file by clicking the "Save" button.

Test Case Scenario: Creating Scenario with Keywords

Users should first click on "Create a Scenario." Users should then fill out two textboxes prompting for how many cells and button the hardware has. Users should click on a button to create a node. Instead of leaving the tag field along, click on it and select required tag. There should be a field next to that for users to input more information regarding the tag. For example, selecting "Pause" and filling out "3" in the rightmost text field to create a pause for 3 seconds. The function should be applied before the text in general when running the Player App on the file generated. Click on the larger textbox below and type in the text the story should have. Continue adding nodes (and filling out the fields inside a node) until satisfied. Users can generate the file by clicking the "Save" button.

Test Case Scenario: Creating Scenario with Branches

Users should first click on "Create a Scenario." Users should first fill out two textbox prompting for how many cells and button the hardware has. Users should click on a button to create a node. To create branches, users should add multiple nodes. There should be a field on each node allowing users to specify the node preceding it. For a linear story, the program, to make it user-friendly, should always assume the node preceding it is the last node added before the current node (basically automatically generating the value in the field). For branching, the users should change these values so multiple nodes have the same node preceding it. Fill out the rest of the nodes' fields as normal. Users can generate the file by clicking the "Save" button.

Test Case Scenario: Recording Audio

The user should click on a button on the initial screen to go to recording mode. The user should fill out a textbox specifying what the name of the file should be called. There should be a button that when the users clicks on it, start recording audio from users' microphone. Click the button again to end it and it saves the audio file with the name specified in the textbox.

Test Case Scenario: Editing File

The user should click on a button to go to editing mode. The app should open the text file created by the Authoring App. The text is editable and when done editing, users can hit the save button.

The above test cases cover all possible scenarios when using this GUI based on customer requirements.