Authoring Application Manual

For

Player Audio Application for Braille-Based Device

Prepared by:
Dilshad Khatri
Drew Noel
Jonathan Tung
Alvis Koshy

Prepared on: April 4, 2017
For: EECS 2311 – Software Development Project

1. Introduction

This manual will provide documentation for users on how to use the Braille File Format Authoring Application (Authoring App). While it is entirely possible to create scenario files for the Braille player directly using a text editor, the Authoring App provides a more intuitive method for scenario creation through the use of a graphical interface. The Authoring App also provides some limited error checking for the scenario files it creates, which would not be available when directly creating a file from text. This manual will present a short tutorial on how to create a very basic scenario file using the Authoring App, before presenting more detail on the functions and abilities of the Authoring App.

2. Tutorial

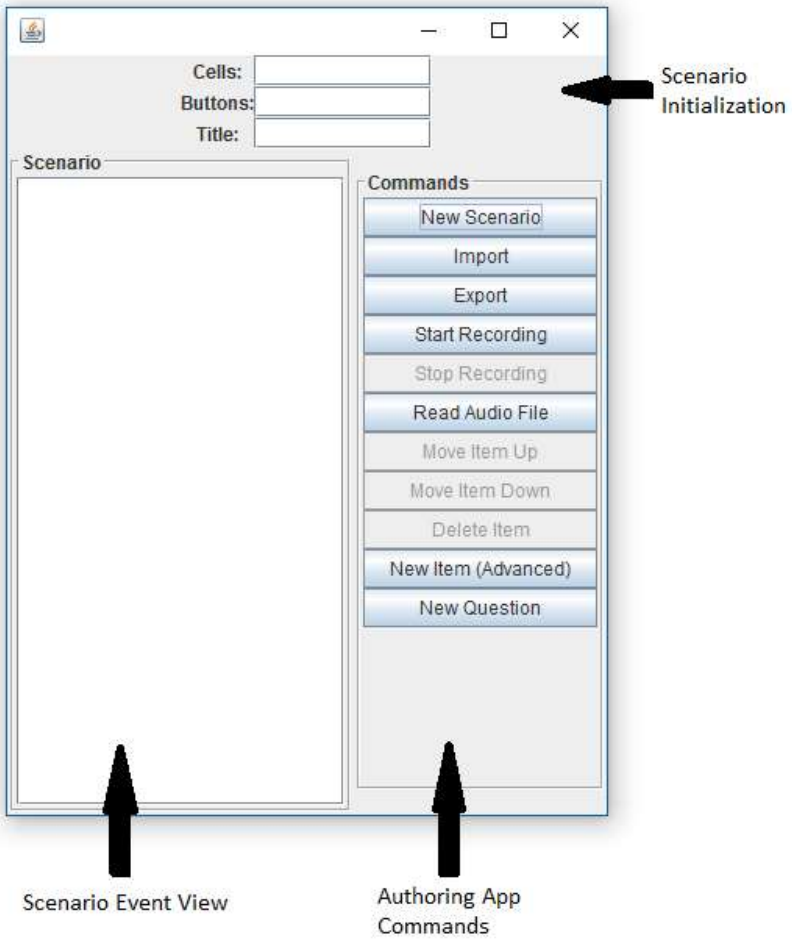
The objective of this tutorial is to present enough information to the reader so that they can be at a stage where they can create simple scenario files using the Authoring App. This tutorial will give step by step instructions on creating a scenario file describing a single quiz question. At the end of this tutorial, it is the goal that the reader is familiar enough with the Authoring App that they can then easily create more detailed scenarios with the information in the rest of this manual.

2.1. Installation and getting started

Extract the contents of the zip file ‘Authoring_App.zip’ to a desired directory. Navigate a command prompt to that directory, and run the command: ‘java –jar authoring_app.jar’.

2.2. Walkthrough

After the Authoring App loads, the window in the figure below should be displayed.



The GUI of the Authoring App consists of three main parts: an Authoring App commands panel contains buttons for commands used to create a Scenario file, a Scenario event view where the scenario event items that have been created can be seen, and a scenario initialization panel that contains important information such as the number of Braille cells, buttons and name of a scenario.

The scenario event view displays a vertical list of events that make up a scenario. When the scenario plays, these events will be executed sequentially top to bottom in the order that is displayed here. This view displays information about what type of events are in the scenario, as well as details about their execution that have been set. These details can be accessed and edited by double clicking on the event.

The Authoring App commands displays several buttons used to create, modify, and save a scenario. These will be explained in further detail in the Authoring App Commands section of the manual.

The information in the scenario initialization panel is set when a new scenario is created, and generally should not be changed once they are set.

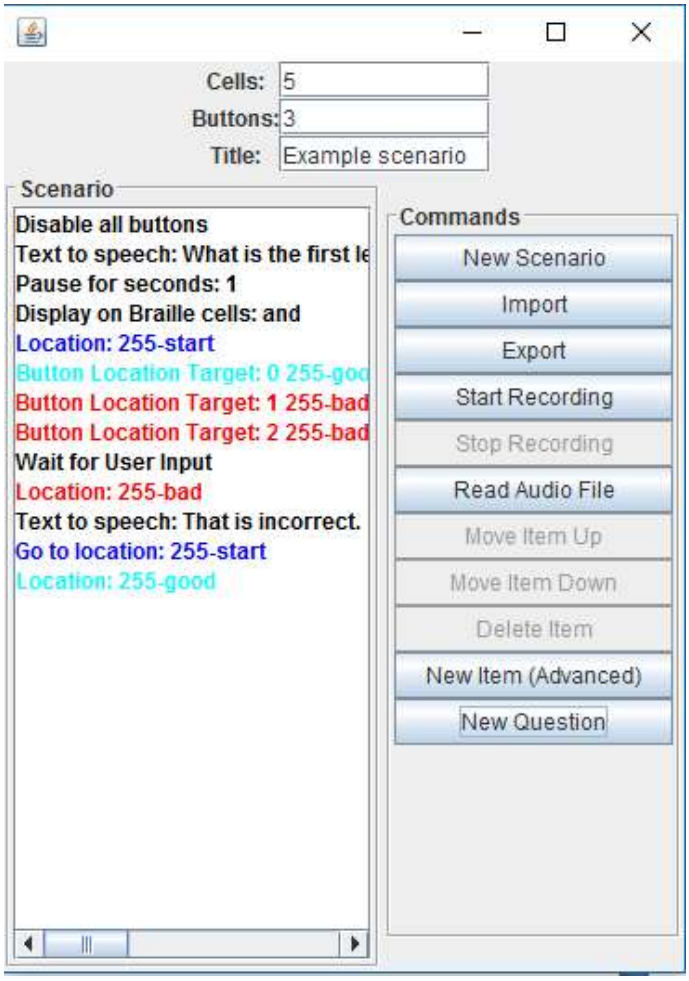
First click the ‘New Scenario’ button. Set the number of Braille cells and buttons this scenario uses by filling in the input fields in the dialog box that appears. In the field labeled ‘Cells’ enter ‘5’, and in the field labeled ‘Buttons’ enter 3. Name this scenario by entering ‘Example scenario’ in the ‘Title’ field, and then click ‘Ok’. The number of Braille cells and buttons is required in a properly formatted scenario file, so the input fields at the top of the application window must have values in them. The windows should look like the figure below.

We will use the New Question function to help us create the framework for a single quiz question. Click on the ‘New Question’ button. A dialog box

labeled ‘Enter Question Details’ should appear. Enter the following in the input fields:

Introduction Text – What is the first letter? Button 1 for a, button 2 for c, button 3 for c.
Braille Text – and
Correct Button – Button 1
Text For Incorrect – That is incorrect. Please try again. What is the first letter? Button 1 for a, button 2 for c, button 3 for c.

Hit ‘Ok’. The window should look like the figure below, except some of the numbers in the scenario event view may be different. These are auto-generated tags, so this is to be expected.

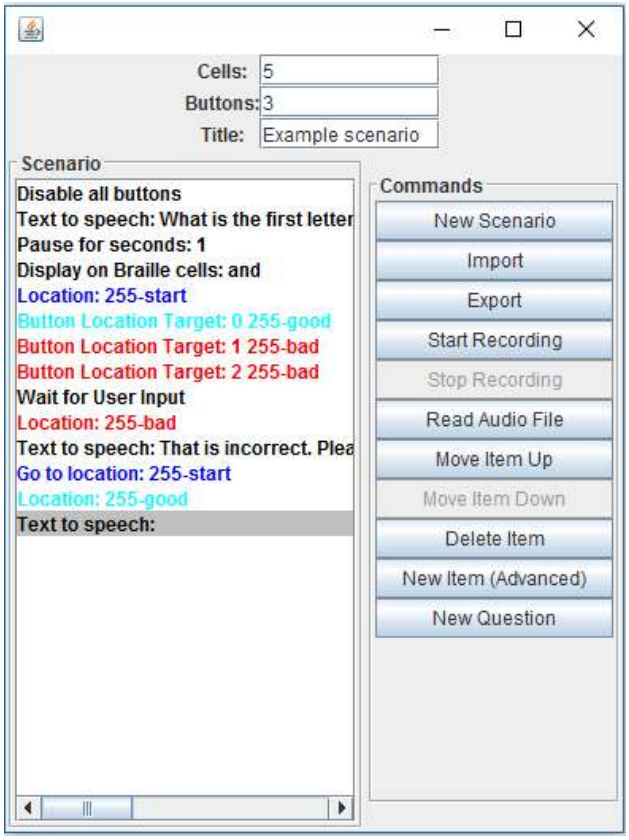


We now have a single quiz question scenario that asks what the first letter on the Braille cells is when it displays the word “and’. It gives 3 options:

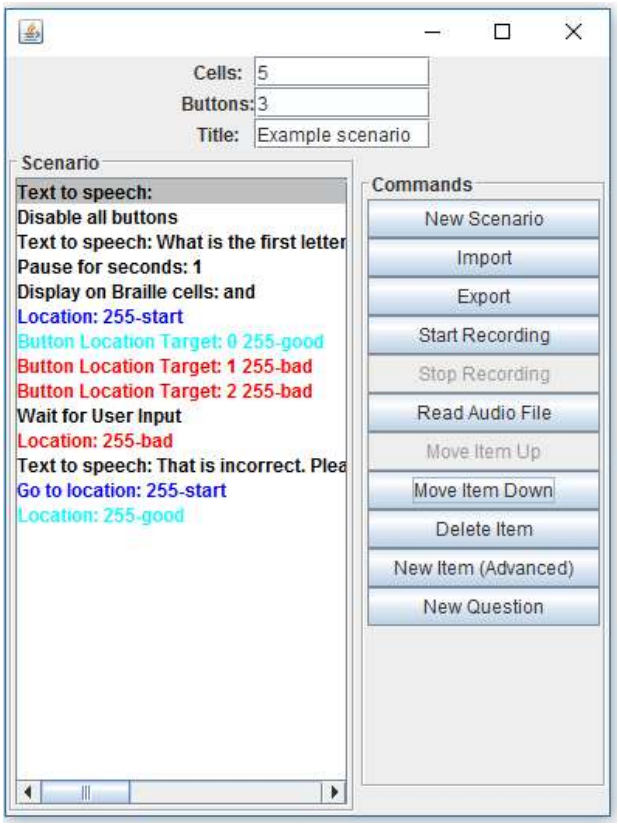
button 1 for a, button 2 for b and button 3 for c. If the wrong answer is given, the scenario indicates that this was a wrong answer, asks the question again and re-waits for an answer. If the correct answer is given, the scenario jumps to the end.

We should add a welcome message to this scenario. Click on the ‘New Item (Advanced)’ button, in the drop down menu select ‘Text-to-speech’ and click ‘Ok’

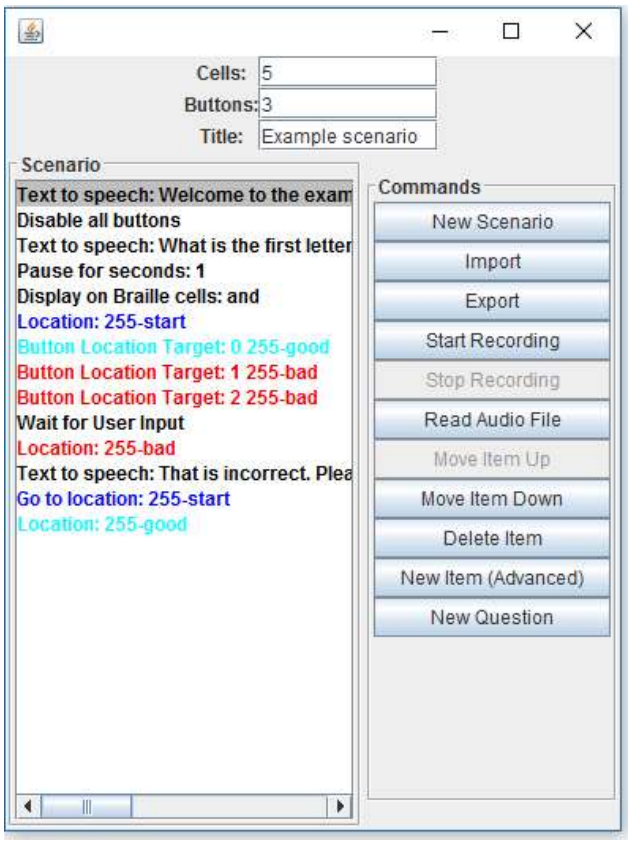
Single click on the ‘Text to speech:’ line at the bottom of the scenario event view to select it.



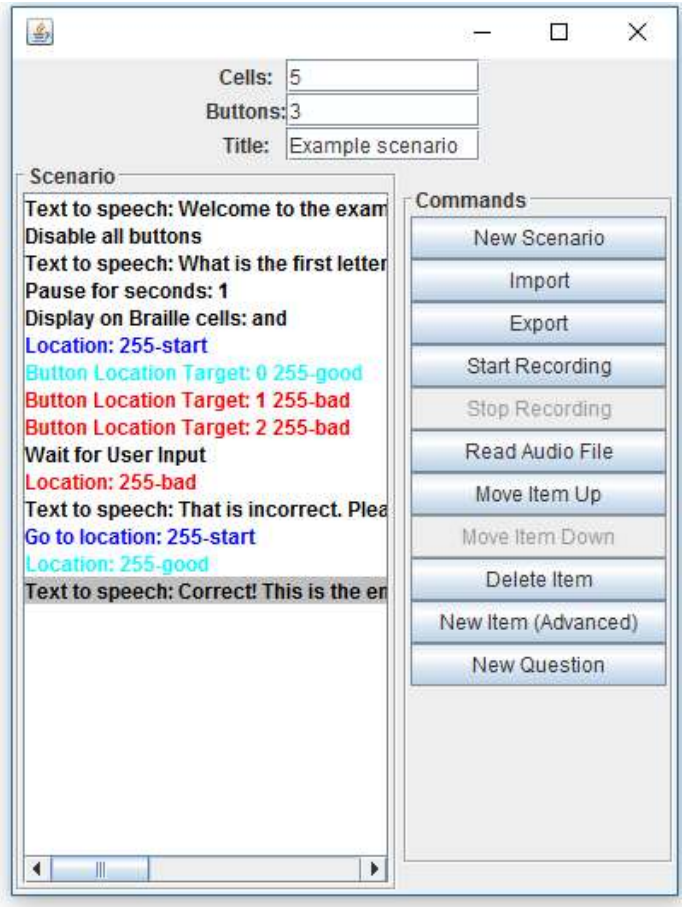
Click on the ‘Move Item Up’ button until this line is at the top of all the events.



Double click on this line and in the resulting dialog box fill in the input field with “Welcome to the example scenario”. Click ‘Ok’. We have just added a welcoming message to this scenario. The window should look like the figure below.



Next let's add a goodbye message for when the user gets the right answer. Click on 'New Item (Advanced)' and select 'Text-to-speech' again from the drop down menu. Click 'Ok'. Double click on the new 'Text to speech' line at the bottom of the scenario events view. In the resulting dialog box, in the input field enter "Correct! This is the end of the scenario, goodbye." and hit 'Ok'. The window should now look like the figure below.



To save this sample scenario, click the ‘Export’ button and in this dialog, navigate to the folder you want to save to, enter ‘Example scenario’ in the input field to name the file, and hit ‘Save’.

3. Authoring App Commands

This section will describe the commands available in the Authoring App Command panel on the right of the application window.

3.1 New Scenario

This button initializes a new scenario file. It is one of two commands that must be used before beginning to create a scenario file, with the other being the ‘Import’ command. Clicking the ‘New Scenario’ button brings up a dialog box to set the number of Braille cells and buttons this particular

scenario will work with. The number of Braille cells and buttons must be at least 1. This dialog also provides an input field to name the scenario. Be careful about clicking ‘New Scenario’ if there is already a scenario file in the Authoring App, as it will clear the scenario to a blank one. A warning dialog will be displayed if a scenario already exists in the Authoring App.

3.2 Import

This button is used to load pre-existing scenario files elsewhere on the computer running the Authoring App. Clicking the ‘Import’ button opens up an open file dialog that can be used to navigate to different directories in the computer. To import a scenario file, navigate to the directory that it is saved in and locate it in the main window of the open file dialog. From there, files can be opened by double clicking on them, or by selecting and then hitting the ‘Open’ button. The open file dialog by default will only display files with the ‘.txt’ extension. The Authoring App can attempt to open any file with the ‘.txt’ extension, though files that are not formatted properly will not result in a scenario showing up properly in the Authoring App.

3.3 Export

This button is used to save a scenario in the Authoring App. Clicking on the ‘Export’ button will open up a save file dialog that can be used to navigate to different directories in the computer. To save a scenario file, navigate to the directory you wish to save the file to. Enter the desired name for the file in the input field labeled ‘File Name:’, and click the ‘Save’ button. Scenario files are saved with a ‘.txt’ file extension. The Authoring App will do some basic error checking on the format of the scenario before saving, and display warnings if it detects any errors.

3.4 Start Recording and Stop Recording

These two buttons are used to record audio files that can be played back in a scenario. In order to use this command, the computer running the Authoring App must have a working microphone plugged into its line in. Clicking the ‘Start Recording’ button will start the recording of continuous audio from the microphone. To stop recording, click the ‘Stop Recording’ button. This will bring up a save file dialog, that can be used to navigate directories in the computer. Enter the desired name of the sound file in the input field labeled ‘File Name’, and click ‘Save’ to save the file. Audio

recording files are saved in the wav file format, with the '.wav' file extension.

3.5 Read Audio File

This button allows the user to select a .wav sound file that has been saved on the computer and create a sound event playing it in the scenario. Clicking the button will open an open file dialog, almost identical to when the 'Import' button is hit, as described above. The difference is that this dialog is set to display files with the '.wav' extension. When a file is opened with this dialog, a `Sound` event will be created in the scenario to play this sound file, and the sound file will be played.

3.6 Move Item Up and Move Item Down

This button moves scenario events up the list of events in the scenario event view on the left of the Authoring App. The order in which events appear here is the order in which they will be executed when the scenario plays. Events can be selected by single clicking. Clicking the 'Move Item Up' button will then move the selected event one event up.

Similarly to the 'Move Item Up' button described above, this button moves a selected event in the scenario event view down one event.

3.7 Delete Item

This button deletes an event from the scenario. To do so, select the event in the scenario event view, then click the 'Delete Item' button. Be careful when doing this, as there currently is no way to undo this change and there is no warning dialog.

3.8 New Item (Advanced)

This button is used to insert new events into the scenario, where they will show up in the scenario event view. Clicking the 'New Item' button will bring up the 'Add Item' dialog, which contains a dropdown menu listing all the different event types that can be put in a scenario. Selecting the desired event type and hitting 'Ok' enters that event into the scenario. For most events, they will be entered without their necessary details, so these will need to be set by double clicking on the event in the scenario event view.

These details vary from event to event, and will be explained in the next section.

4. New Question

The new question function provides a simplified way to introduce a quiz question into a scenario. It is accessed by hitting the ‘New Question’ button on the Authoring App command panel. The resulting dialog box provides input fields to configure a simple question format. This question format will first read the entry in input field ‘Introductory Text’ using text to speech. It will then pause for 1 second before displaying the entry in the input field ‘Braille Text’. It will then configure one of the initialized buttons to be a right answer, while all the rest will be associated with wrong answers. If a button associated with the wrong answers is pushed, the scenario will read the entry in input field ‘Text for Incorrect’ using text to speech and jump back to wait for the same button input again. If the button associated with the right answer is pushed, the scenario jumps immediately out of this question. This function creates this quiz question using standard event types accessible from the ‘New Item’ button, and thus is editable once it is created. A user could for example, replace the auto generated text to speech events with sound events to play pre-recorded sound files instead of using text to speech.

5. Event Types

All the different event types that can make up a scenario file can be accessed through the ‘New Item’ button, as described in the section above. Each event type has a description field that describes what that particular event does, and in most cases must be set by the user. This is done by double clicking on the event in the scenario event view, and entering an appropriate description in the resulting ‘Edit Item Details’ dialog box. The following sections elaborate on how to format the descriptions for each event.

5.1 Pause

A `Pause` event makes the scenario pause for a number of seconds, which is determined by the pause event's details. Valid descriptions are positive integers.

5.2 Text to speech

A `Text to speech` event makes the scenario read out some text using a computerized voice. A text to speech event's description is the text that is to be read out. Valid descriptions are single or multiple sentences written in English.

5.3 Display String

A `Display String` event will display a given input on the Braille cells. A `Display String` event's description is the word or letters that are to be displayed on the Braille cells. A valid description is a word or set of characters that will fit on the number of Braille cells that have been initialized in the scenario, which is visible at the top of the Authoring App.

5.4 Repeat

A `Repeat` event makes the scenario read out some text using a computerized voice. It differs from a `Text to speech` event in that the text to speech reading of a `Repeat` event can be set to be repeated when a button is pressed through the first `Button Repeat` event entered below it in the scenario. Valid descriptions are single or multiple sentences written in English.

5.5 Button Repeat

A `Button Repeat` event sets up a button to read out the text in the first `Repeat` event above it when it is pushed. A valid description for a `Button Repeat` event is a number representing a button initialized in the scenario. Button numbers start from 0, rather than 1, thus a scenario with 4 buttons would have them numbered 0, 1, 2, 3. `Button Repeat` events require a `User Input` event after them in the scenario to fully execute, which makes the scenario wait for user input from a button. This is expanded on later in the `User Input` event section.

5.6 Button Location

A `Button Location` event sets up a button to jump the scenario to a specified point when pushed, indicated by a `Location Tag` event, rather than having it continue sequentially down the events in the scenario event view. `Location Tag` events are described further down in their own section. A valid description for a `Button Location` event is a number indicating an initialized button, a space, and then the name of the `Location Tag` event you wish to jump to. `Button Location` events require a `User Input` event after them in the scenario to fully execute, which makes the scenario wait for user input from a button. This is expanded on later in the `User Input` event section.

5.7 User Input

A `User Input` event pauses the scenario while it waits for input from the scenario's buttons. The scenario would be resumed when a button is pushed. `User Input` events do require a description, and it should be left in their blank default state. A `User Input` event require at least one `Button Repeat` event and/or a `Button Location` event above it in the scenario, or else it will pause the scenario indefinitely, as no buttons are set up to receive input.

5.8 Sound

A `Sound` event plays a .wav sound file specified by the user. A valid description is the file path to a .wav file. Another way to create a `Sound` event is through the 'Read Audio File' button described above, which will automatically fill in the description with a file path.

5.9 Reset Buttons

A `Reset Buttons` event undoes the actions of any `Button Repeat` and `Button Location` events above it. It essentially puts all the buttons in a scenario back to their default state, where they are not set to expect any input. `Reset Button` events do not require a description and should be left in their blank default state. A typical use for this event is to reset the buttons between questions in the scenario.

5.10 Go to Location

A `Go to Location` event makes the scenario jump to a specified point, indicated by a `Location Tag` event, rather than having it continue sequentially down the list of events. A valid description is the description of the location tag event you wish to jump to. `Location Tag` events are described further in their own section.

5.11 Clear All

A `Clear All` event will clear anything displayed on all the Braille cells used by the scenario. These events do not require a description and should be left in their blank default state.

5.12 Clear Cell

A `Clear Cell` event will clear a single Braille cell used by the scenario. A valid description is a single number, corresponding to one of the Braille cells. Like buttons, Braille cells are numbered from 0. Thus a scenario with 8 Braille cells would have then numbered 0, 1, 2, 3, 4, 5, 6, 7.

5.13 Set Pins

A `Set Pins` event is used to raise or lower specific pins in a specific Braille cell. A valid description consists of a single number, a space, and then an 8 character sequence consisting of 0's and 1's. The number corresponds to a Braille cell, while the 8 character sequence corresponds to pins on that cell. The first character in this sequence corresponds to the top-left pin, with the next character corresponding to the next pin in a left-to-right and then a top-to-bottom fashion.

5.14 Set Char

A `Set Char` event displays a single specified character on a single specified Braille cell. A valid description is a number representing the Braille cell, a space, and then a single character that is to be displayed.

5.15 Raise Pin

A `Raise Pin` event raises a single specified pin on a single specified Braille cell. A valid description is a number representing the Braille cell, a space, and a single number between 1 to 8 representing the pin to raise. 1

corresponds to the top-left pin, with each number after corresponding to the next pin in a left-to-right and then a top-to bottom fashion.

5.16 Lower Pin

A `Lower Pin` event is very similar to the `Cell and Pin Raise` event described above, except that it lowers a specific pin on a Braille cell, rather than raising it. A valid description is a number representing the Braille cell, a space, and a single number between 1 to 8 representing the pin to raise. 1 corresponds to the top-left pin, with each number after corresponding to the next pin in a left-to-right and then a top-to bottom fashion.

5.17 Set Voice

A `Set Voice` event selects 1 out of a choice of 4 computerized text to speech voices to use in text to speech and repeat events that follow it. Voices 1, 3 and 4 are male voices, while voice 2 is a female voice. A valid description for a voice event is a number between 1 and 4.

5.18 Location Tag

A `Location Tag` event marks a place in the scenario that can be directly jumped to from `Go to Location` and `Button Location` events elsewhere in the scenario. In order to make the jump, the descriptions of the `Location Tag` event and the `Go to Location` event must match. For `Button Location` events the second part of the description after the button number must match. For example, a `Location Tag` event with description “Test” can be jumped to from a `Go to Location` event with description “Test”, or a `Button Location` event with description “1 Test”. A valid description for a `Location Tag` event can be any sequence of characters.