

Authoring Application

Test Document

Group 2

EECS 2311

Midterm Submission

February 23rd 2018

1. Table of Contents

Table of Contents	2
Purpose	3
Definitions	3
Test Requirements	3
Scenario Requirements	3
Open Scenario File	3
New Scenario File	4
Edit Scenario File	4
Run Scenario File	4
Save Scenario File	4
Interaction Requirements	4
Read	4
Voice	4
Pause	4
Display Braille	5
Test Cases	5
Test Coverage	8
References	9

2. Purpose

The purpose of this document is to detail the testing specifications of our Authoring application.

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*

3. Test Requirements

The following testing requirements are derived from the requirements document and will serve as the baseline for determining our test cases in section 3 below.

3.1. Scenario Requirements

3.1.1. Open Scenario File

The user shall have the capability to open an existing, properly formatted, scenario text file into the application.

3.1.2. 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.

3.1.3. 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.

3.1.4. Run Scenario File

The user shall be able to run their scenario in the simulator in order to test their scenario out.

3.1.5. 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.

3.2. Interaction Requirements

3.2.1. 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.

3.2.2. 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.

3.2.3. Pause

The pause interaction shall be an interaction where the simulator pauses for a user-defined number of seconds.

3.2.4. 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.

[1]

4. Test Cases

ID	Description	Procedure	Expected Output	Result	Notes
Testing main panel					
TC1	Validating requirement 2.1.1 - Open Scenario File	<ol style="list-style-type: none">1. Click Open button from main page2. Choose a scenario file from the file dialog3. Click Open	<p>If a valid scenario file was selected, the editor screen should appear with all the interactions listed and the configurations of the selected interaction.</p> <p>If an invalid scenario file was selected, an error message dialog will notify that an invalid scenario file was chosen. User will remain on the startup screen on dialog close.</p>	Pass	File menu items are not yet implemented and thus user cannot use the menu to create/open scenario files. However, the behavior should be the same and thus they simply need to be wired to the correct methods.
TC2	Validating requirement 2.1.2 - New Scenario File	<ol style="list-style-type: none">1. Click New button from main page2. On the New scenario page, enter the scenario title, and set the number of cells and buttons as desired.3. Click Create	On clicking Create, the editor page should appear with an empty interaction list.	Pass	Editor window appears even if no title was set.
Testing editor panel					

TC3	Validating requirement 2.1.3 - Edit scenario file	<p>Create / edit:</p> <ol style="list-style-type: none"> 1. Create a new scenario from the main page 2. On the editor page, select an interaction from the drop-down box (on the bottom with the other controls) 3. Click Add <p>Load / edit:</p> <ol style="list-style-type: none"> 1. Open a scenario file from the main page 2. Select any interaction from the list 3. Click any of the controls / fields associated with the interaction on the configuration panel on the right to change them 	The configuration options for the selected interaction should appear on the right. The fields associated with the particular interaction are editable.	Pass	BUG: When creating a new scenario, the first item that is added does not show the configuration panel. This is likely because the list selection listener is not being invoked.
TC4	Validating requirement 2.1.4 - Run scenario file	<ol style="list-style-type: none"> 1. Create a new scenario or load an existing one from the main page 2. If a new scenario was created, add any interaction to test and click Save 3. Click Run to launch the simulator 	The simulator window should appear with the computerized voice playing audio.	Pass	
TC5	Validating requirement 2.1.5 - Save scenario file	<ol style="list-style-type: none"> 1. Create a new scenario or load an existing one from the main page 	If a new scenario was created, a new .txt file with the title of the scenario will be saved under a folder called	Pass	

		<ol style="list-style-type: none"> 2. Make changes or add new interactions to the scenario 3. Click the Save button 	<p>Scenarios in the working directory of the application.</p> <p>If a scenario was loaded, it will overwrite the existing .txt file.</p>		
Testing interaction configurations and behavior					
TC6	Validating interaction requirement 2.2.1 - Read	<ol style="list-style-type: none"> 1. Create a new scenario from the main page 2. Select Read from the drop-down list of interactions 3. Click Add 4. On the configuration pane on the right, modify the Data field on the right to the text that simulator should read out 5. Click Save 6. Click Run 	The simulator should open and read out the text data that was specified for the Read interaction.	Pass	
TC7	Validating interaction requirement 2.2.2 - Voice				Voice interaction, ie. the ability to record an audio and have it played out during simulation, is to be implemented in the next release.
TC8	Validating interaction requirement 2.2.3 - Pause	<ol style="list-style-type: none"> 1. Create a new scenario from the main page 2. Add a Read interaction and set the desired text to be read out 3. Select Pause from the drop-down list of interactions and 	The simulator should open and read out the first set of text that was set, and then wait for the selected amount of seconds, and then finally read the second set of text.	Pass	

		click Add 4. Select the number of seconds to pause for 5. Add another Read interaction and modify the data field 6. Click Save 7. Click Run			
TC9	Validating interaction requirement 2.2.4 - Display braille	1. Create a new scenario from the main page 2. Select Display braille from the drop-down box of interactions and click Add 3. Select / deselect the pins (combo-box items) to raise on the simulator 4. Click Save 5. Click Run	The simulator should open with the selected pins being shown on the braille pad.	Pass	

5. Test Coverage

The metrics available at this time for our testing include the test coverage results from running the authoring application and creating a new scenario and running the simulation. Note that the coverage results are lower than what it actually is since there are several deprecated files that were not removed in this release. The deprecated files are noted in the screenshot below with a red arrow.

Element		Coverage	Covered Instructions	Missed Instructions	Total Instructions
Enamel		48.9 %	4,389	4,590	8,979
src		48.9 %	4,389	4,590	8,979
authoringApp		50.0 %	3,377	3,379	6,756
App.java		83.3 %	30	6	36
CellClearInteraction.java		0.0 %	0	56	56
CellClearInteractionView.java		0.0 %	0	118	118
ConfigurationViewTest.java		0.0 %	0	19	19
CustomListModel.java		69.1 %	85	38	123
DisplayBrailleInteraction.java		83.6 %	143	28	171
DisplayBrailleInteractionView.java		89.4 %	169	20	189
EditorController.java		81.5 %	172	39	211
EditorPane.java		89.7 %	807	93	900
EntryPoint.java		0.0 %	0	493	493
Interaction.java		93.7 %	164	11	175
InteractionList.java		0.0 %	0	87	87
InteractionView.java		86.3 %	202	32	234
KeywordInteraction.java		100.0 %	36	0	36
KeywordInteractionView.java		75.2 %	106	35	141
ListItemTransferable.java		0.0 %	0	27	27
ListItemTransferHandler.java		0.0 %	0	263	263
MainFrame.java		95.8 %	251	11	262
MainViewController.java		29.8 %	51	120	171
MainViewModel.java		73.1 %	19	7	26
NewScenarioConfigPane.java		99.2 %	508	4	512
NewScenarioInfo.java		0.0 %	0	340	340
PauseInteraction.java		70.5 %	31	13	44
PauseInteractionView.java		88.7 %	102	13	115
ReadInteraction.java		54.8 %	23	19	42
ReadInteractionView.java		67.2 %	131	64	195
ResetButtonInteraction.java		0.0 %	0	9	9
ResetButtonInteractionView.java		0.0 %	0	21	21
Scenario.java		33.9 %	191	373	564
ScenarioEditorController.java		0.0 %	0	3	3
ScenarioEditorView.java		0.0 %	0	585	585
ScenarioTest.java		0.0 %	0	83	83
SkipButtonInteraction.java		0.0 %	0	54	54
SkipButtonInteractionView.java		0.0 %	0	234	234
StartupPane.java		84.3 %	156	29	185
TextInputInteraction.java		0.0 %	0	11	11
TextInputInteractionView.java		0.0 %	0	21	21
enamel		45.5 %	1,012	1,211	2,223
AudioPlayer.java		0.0 %	0	7	7
BrailleCell.java		75.6 %	232	75	307
Player.java		51.5 %	88	83	171
ScenarioParser.java		31.3 %	392	860	1,252
ToyAuthoring.java		0.0 %	0	12	12
VisualPlayer.java		63.3 %	300	174	474

Figure 1. Screenshot of test coverage results

6. References

1. C. Dear, *Authoring Application Requirements Document*, 1st ed. 2018.