EECS 2311

Authoring Application

User Manual

Jinho Hwang

Rajvi Chavada

Andrew Maywapersaud

Table of Contents

Pages	Description
4	1.0 Revision History
5	2.0 Program Overview
6-8	3.0 Getting Started
9-12	4.0 Load Scenario
13-14	5.0 Creating and editing a Scenario
15	6.0 Screen Reader Compatibility

List of Figures/Media

Pages	Figure Description
6	Figure 1: Project Imported into Eclipse
7	Figure 2: Launching the Authoring Application
7	Figure 3: Command line launch of Authoring Application
8	Figure 4: Scenario Editor
9	Figure 5: Load Scenario
10	Figure 6: Scenario editor after Load scenario
11	Figure 7: Simulated braille cells
11	Figure 8: Pins 1,3 versus 4,6
12	Figure 9: Pins 2,5 versus 1,4
13	Figure 10: Scenario Maker
14	Video 1: VoiceOver demo

1.0 Revision History

Date	Revision
02/23/18	Revision 1. Midterm Design and Implementation

2.0 Program Specification

Welcome to the Authoring Application. This application simulates a treasure box braille with the purpose of allowing the educator to convey text to the visually impaired. In this context, the user is both the educator and the visually impaired. With this software, the user can load, create and save scenarios to be presented. The scenarios involve a set of instructions or passages. These instructions are illustrated through the braille letters. Buttons allow a response to the instructions.

The program was constructed using the Java programming language. Specifically, SceneBuilder was installed on the Eclipse IDE to design the application's user interface using JavaFx libraries. Most of the functionality is implemented with JavaFx. The Swing and Awt library comprise the remaining code.

It must be noted that this application is still in the beta stage. Thus, this manual will be continuously revised until the finalization of the application. Further, this manual was created using a MacBook. Therefore, the graphics will be iOS related. This should not effect the use of the application.

3.0 Getting Started

The prerequisites to this section are that the readers of this manual have successfully extracted the program. In addition, it is assumed the user has the current version of java installed, the Eclipse IDE or a terminal that can compile and run java applications.

After importing the project to Eclipse, the following directory structure should be seen:

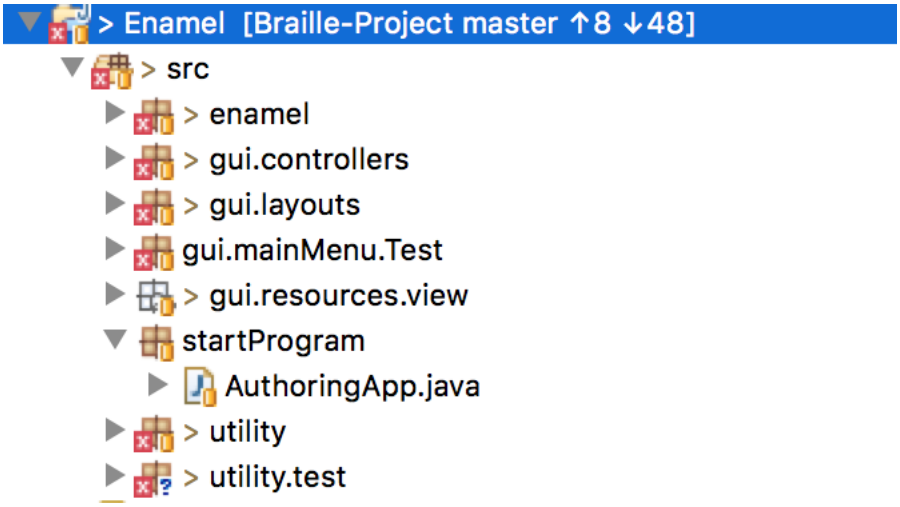

Fig.1 Project Imported into Eclipse

Ignore the errors on the packages. This snapshot of the project was taken during a testing phase. Our focus is on the startProgram package.

To launch the application, click on the AuthoringApp.java class and then run the program. The following user interface will display:

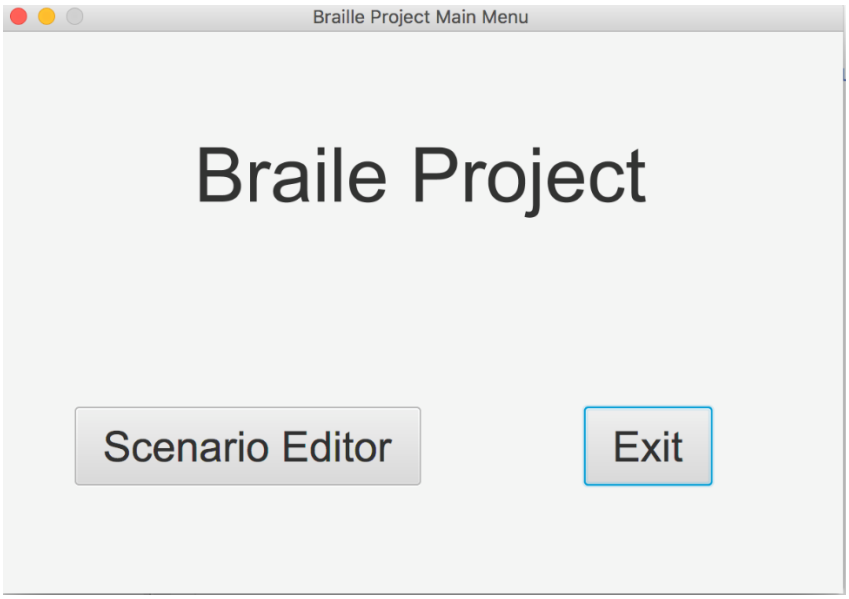

Fig.2 Launching the Authoring Application

Alternatively, the application can be run from the command line using the following commands. Ensure your current directory is correctly set to the startProgram package.

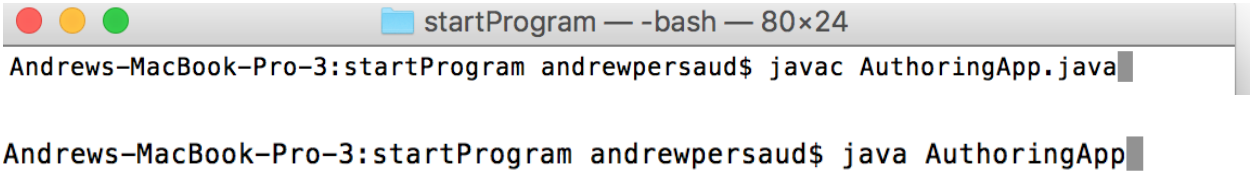

Fig.3 Command line launch of Authoring Application

After confirming this display successfully launched, click on 'Scenario Editor'. The interface should change to the following window:

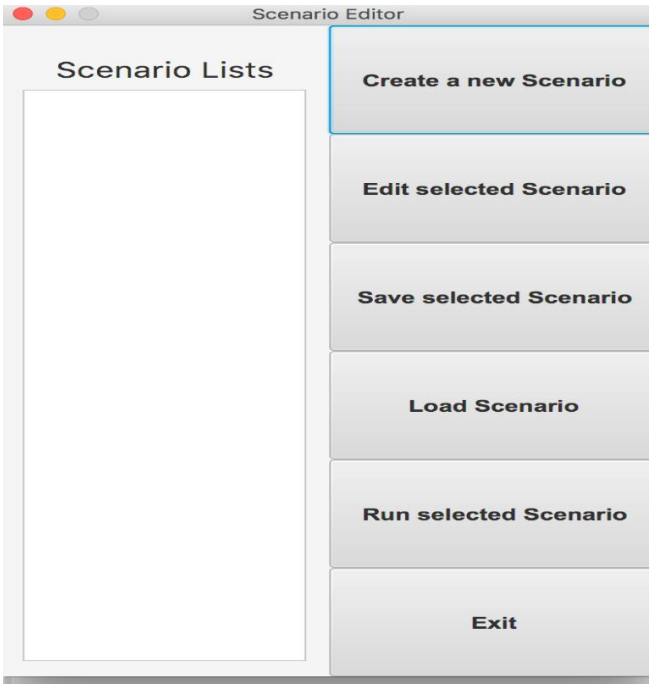

Fig.4 Scenario Editor

This interface will allow the educator to perform the actions specified above. The following section will traverse through these actions.

4.0 Load Scenario

The first section we will explore is the ‘Load Scenario’. Note that the project will include scenario files for you to load and run to get a perception of the application. It is assumed the reader of this manual is competent in creating their own scenario files to be loaded.

After clicking on ‘load scenario’, the following window should appear allowing the user to select a file.

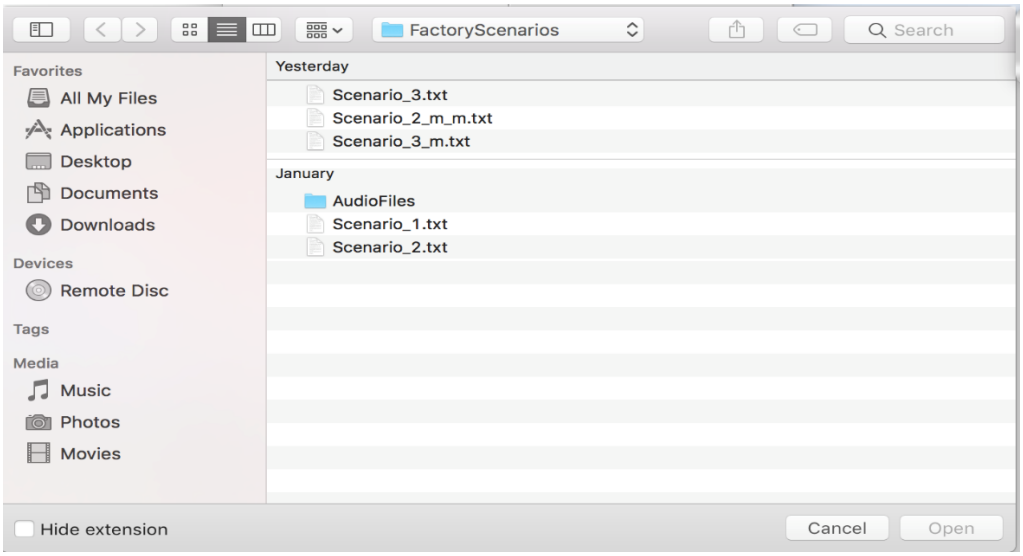

Fig 5. Load Scenario

Your folder may not look exactly as the above, however what should be there are Scenario_1, Scenario_2 and Scenario_3 text files.

Click on Scenario_1 and then ‘Open’. Repeat the above for Scenario_2 to ensure that multiple files are shown on the Scenario Editor. After these steps, the following should display:

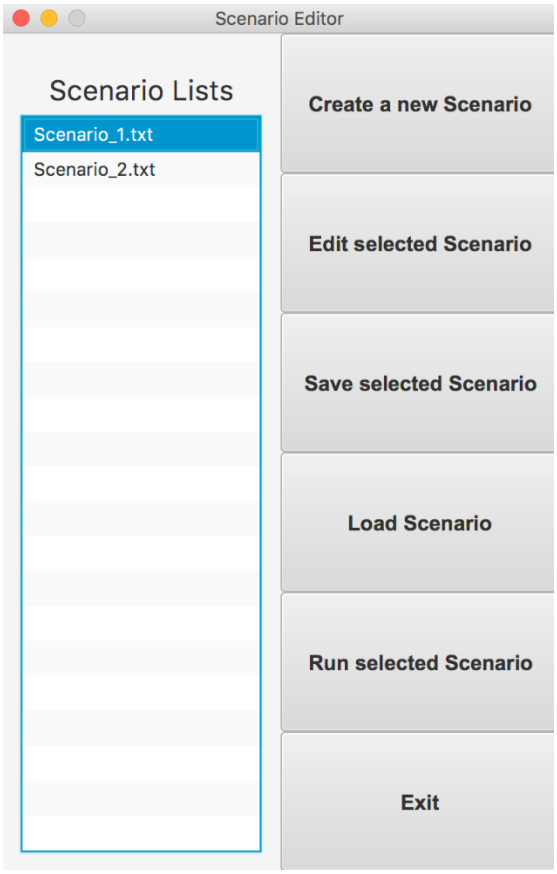

Fig 6. Scenario Editor after Loading Scenarios

Then, with Scenario_1.txt highlighted, click on 'Run selected Scenario'. Alternatively, you may run Scenario_2.txt. After clicking on 'Run selected Scenario', a simulated braille cell should appear with buttons labeled 1,2,3 and 4 at the bottom for user interaction. The following figure illustrates this:

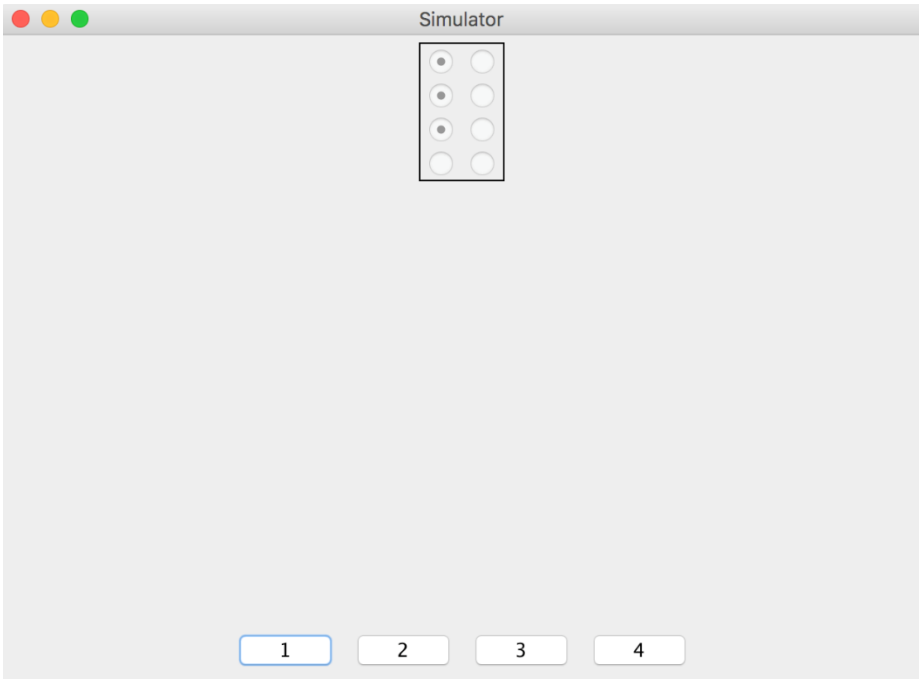

Fig.7 Simulated braille cells

At this point you may try this scenario and play around with the input to familiarize yourself with how a basic run of the application may proceed.

For instance, let us quickly run through Scenario_2. First, note that pins 1-3 are the first 3 cells on the left and pins 4-6 are the first three on the right.

After running Scenario_2, you will be asked if pins 1,3 versus pins 4.6 are highlighted:

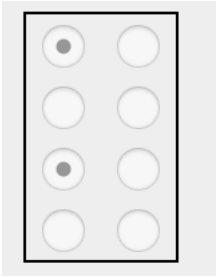

Fig.8 Pin alignment for 1,3 versus 4,6

These are pins 1,3. Therefore, click on button 1. A bell sound should be heard. The bell signifies correct answers.

Next, you will be asked about this configuration:

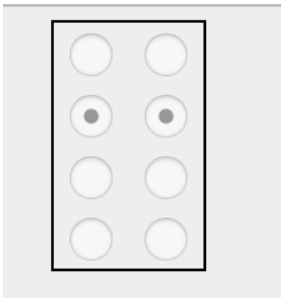

Fig.9 Pins 2,5 versus 1,4

These are pins 2 and 5. However, let us choose the incorrect option of pins 1 and 4 by clicking button labeled ‘2’. You should hear an unappealing buzzer sound. This buzzer sound signals an incorrect answer. The rest of the scenario runs similarly. You may finish it if you desire.

The following section will introduce the capability of creating your own scenario.

5.0 Creating and Editing a New Scenario

After exiting the simulated braille cells from section 4.0 by clicking the red circle or the analogous button on your system, you will be able to use features of the Scenario Editor from figure 4.

The Authoring Application will allow the user to create scenarios such as those seen in section 4.0. Click on 'Create a new scenario'. The following display should appear:

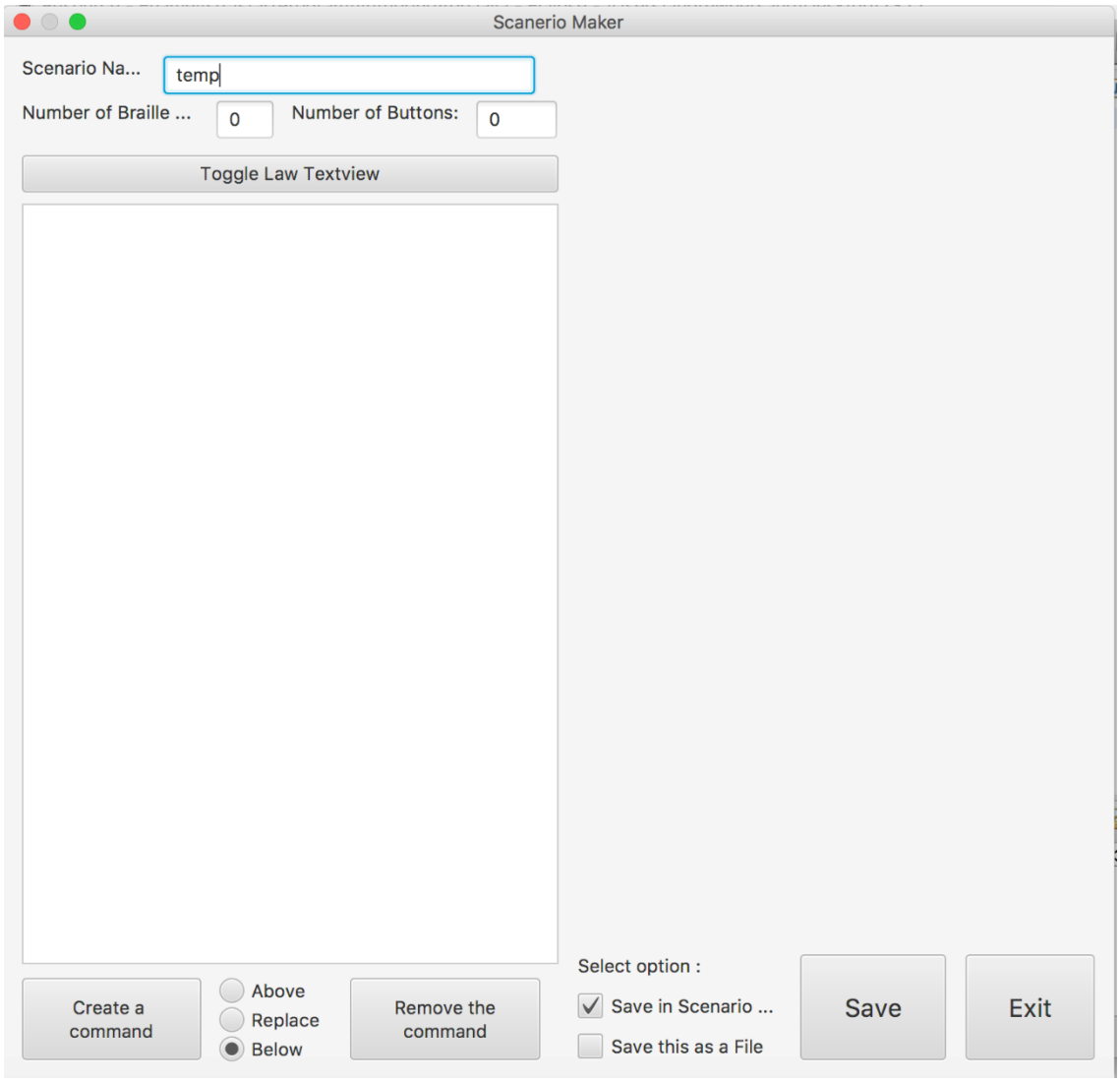

Fig.10 Scenario Maker

This portion of the application will allow the user create their scenario in the text box located on the left of the display. For convenience, the user can select one of ‘above’, ‘replace’, or ‘below’ for where their command will be inserted. Additionally, the user can save their file. Options include saving as a scenario in the editor or creating a new scenario file.

If a command needs to be removed, simply highlight the command and click ‘Remove the command’.

Similarly, after loading a scenario (section 4.0), instead of running the scenario, the user can use the application to edit the file. With the loaded file highlighted, click on ‘edit selected scenario’.

The display from figure 10 will appear and the user can mutate the file as specified prior.

6.0 Screen Reader Compatibility

Note that the manual will be revised with a full specification for this section. This functionality is not fully implemented as the application is currently in a beta stage. However, the application will enable the operating system's built in screen reader. Specifically, this application will support Windows NVDA and the Mac's Voiceover.

The following video we created illustrates how text can be read by the VoiceOver on the iOS operating system. This demo uses a JFileChooser from the Swing library and will not represent implementation of the finished application. Please follow the link below or copy and paste it into a browser. Connection to the internet is required:

http://www.youtube.com/watch?v=RcSPowKO8Yc

Video 1. VoiceOver iOS