User Manual

Jinho Hwang

Authoring App | EECS 2311 |

Table of Content

Table of Contents

1. Introduction

Welcome to Authoring app user manual. In this document, we will cover the basic usages and functions of Authoring app.

1.1 What is Authoring app?

The Authoring app is a visually-impaired educator assistant application which give an easy way for the educators to create scenarios to provide an entertaining braille learning experience to the students the educators are teaching. This program will give the educators tools to import, export, create, and edit the scenario and import custom voice lines.

1.2 Where and how do I get the Authoring app?

Here is the official recommended installation guide:

1. **Install Java:** You need Java version 8 or higher to execute our application, so install the latest Java from the following site:
    https://www.java.com/en/download/
2. **Install Authoring app:** Now that you installed the Java, navigate to the Authoring app download site here:
    https://github.com/DandyEnders/Braille-Project/releases
3. **Put it in a folder:** Put "authoring_app.jar" in a folder you wish to store. The application will create two extra folders side by side for its use when executed it later.
4.   **(Optional)** If the potential user lacks optical perception and the user has no personal visual aids, we recommend you install a free TTS program which is compatible with Authoring app, called NVDA. You can download it from here: https://www.nvaccess.org/download/ NVDA is for Windows users.
5. **(Optional)** If you are using GNU Linux system, we recommend you to install Orca, which is a Linux version of TTS. You can install it from here:
    https://wiki.gnome.org/Projects/Orca
6. **Execute it:** Double click the "authoring_app.jar" to execute it.

2. So how exactly does Authoring app work?

Authoring app uses a formatted scenario file to let possibly visually impaired educators to create a certain scenario for students to learn about braille. This scenario will allow students to interact with braille machine like following in a way educator wants students to learn.:

On Authoring app, you can create, edit, load, save, and simulate Scenario with a window like so, with Text to Sound reader reading lines set by app user.
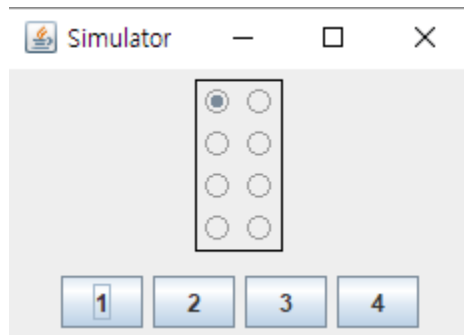
3. Layouts

The Authoring app is usable regardless of the presence of visional perception, as the TTS program will read out which element you are on, and the intuitive design of each menu will let you do engage on the tasks you want to do.

You can navigate through the menu with the arrow keys, tab key, and enter key,

The letter with brackets is the hotkey you can press to use the feature.

Exiting by pressing the X button on the right corner is allowed but is not recommended.

3.1   Main Menu
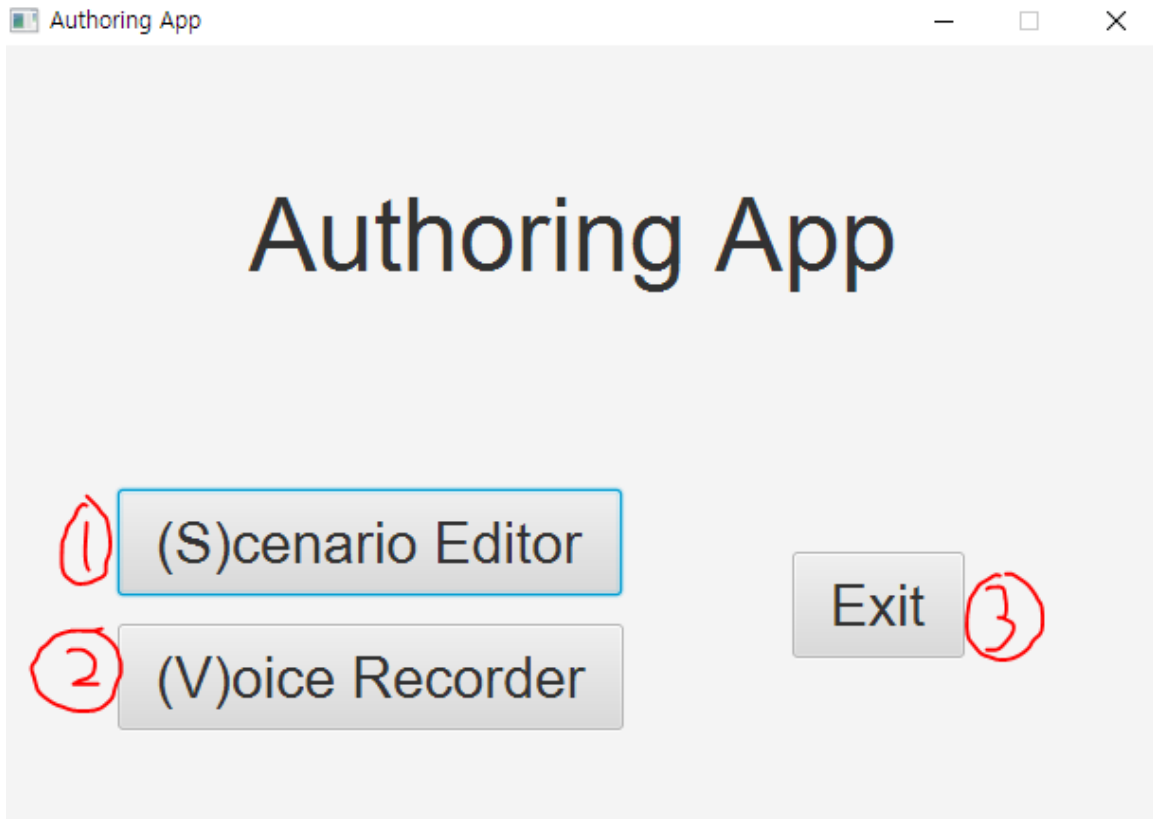
This is the main menu that lets you choose which task you would like to perform.

1.   **Scenario Editor**

This button pops up the scenario editor menu. This menu has all the operation needed to deal with scenarios.
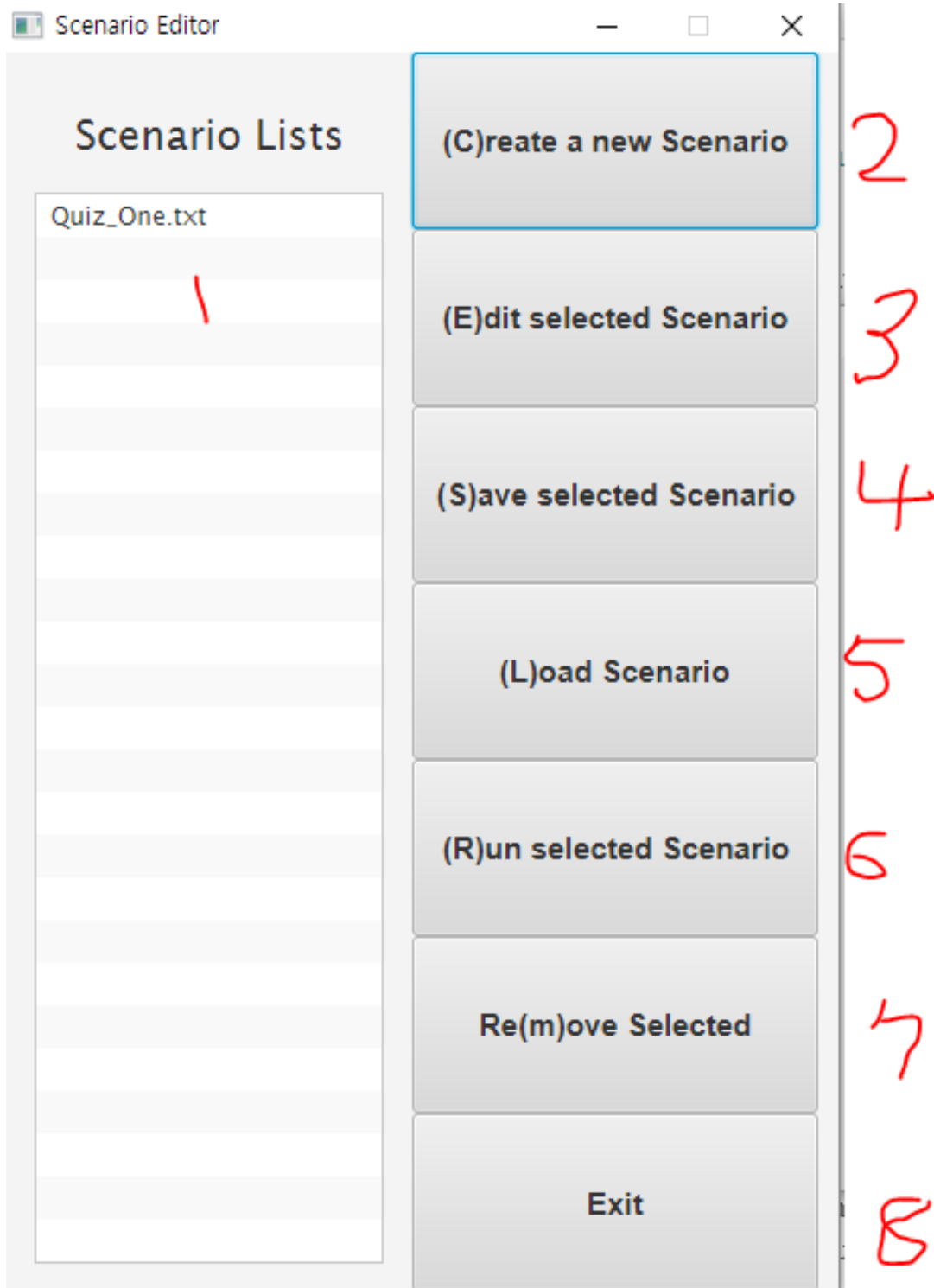
2.   **Voice Recorder**

This button pops up the voice recorder menu. This menu allows you to manipulate with sounds you will import or record.

3.   **Exit**

This button lets you exit the program.

3.2   Scenario Editor

This is Scenario Editor menu. You can manipulate Scenarios on the list.

1. **Scenario Lists**

This is the list of Scenarios loaded by the program. If you installed recently, this list will be empty. You can fill this up by either load Scenarios with Load Scenario button or create new one by pressing the create a new Scenario button.

Each Scenario is a txt file, and is editable with a text editor, but is not recommended to edit with text editor, as doing so without a proper care will result in unusable Scenario.

2. **Create a new Scenario**

This opens a Scenario maker with an empty scenario file to use.

3. **Edit selected Scenario**

This opens a Scenario maker with selected scenario on the list. You must select the Scenario on the Scenario Lists before you press this button, by mouse click or keyboard navigation.

4. **Save selected Scenario**

This lets you save Scenario in a place you want other then this program locally stores. Simply choose where to save, name the scenario file, and hit save to save it.

5. **Load Scenario**

This lets you load Scenario that you have not made with Scenario maker tool provided by this application. You may load multiple Scenarios. The application will check if the Scenario's syntax is valid by checking each line in the Scenario file.

6. **Run selected Scenario**

This lets you simulate the Scenario you selected from the Scenario Lists. There are visual version and audio version.

7. **Remove Selected**

This lets you remove a Scenario from the list and local storage. This will remove the Scenario file permanently, so double check before you remove the Scenario.

8. **Exit**

This lets you exit from the Scenario editor and go back to main menu.

3.3   Scenario Maker

This is Scenario maker. You can create a Scenario with lines of commands.

1. **Scenario Name**

This field appoints the name of Scenario you are dealing with. You may put ".txt" but is not necessary as the program will detect and will add it if it is missing.
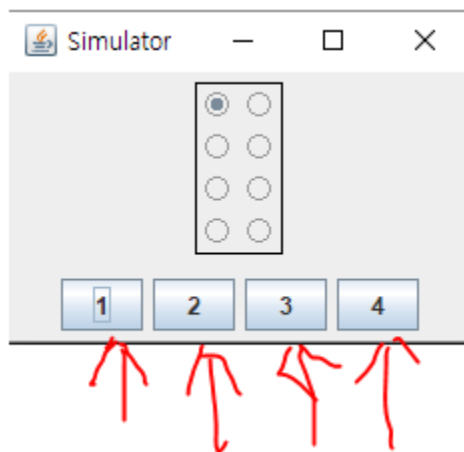
2. **Number of Braille Cells**

This field lets you choose number of Braille Cell you are going to use on Scenario. (Red arrow on the following diagram is the Braille Cell)
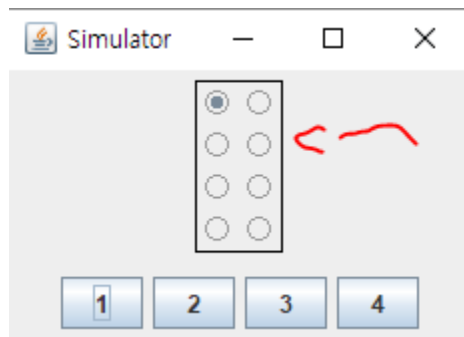
3. **Number of Buttons**

This field lets you choose number of buttons you are going to use on Scenario. (Red arrows on the following diagram are the buttons)

4. **Create a command**

This button opens command maker; the command maker allows user to add commands to make Scenario.

5. **Insertion position**

These three radio buttons show where on the command list choice (the blue or dark grey line on the command list) the command is inserted in the command list.

If Above is selected, the command inserted by command maker will be inserted on top of selected command.

If Replace is selected, the new command will replace the selected command.

If Below is selected, the new command will be inserted after the selected command.

6. **Remove the command**

This button removes selected command.

7. **Move up / Move down**

These buttons move either up or down the selected command.

8. **Command List**

This list has commands for a Scenario.

9. **Save and Exit**

This button checks if the Scenario built is formatted right, and if it is, it will save a Scenario into the Scenario Lists on Scenario Editor and close Scenario Maker. It will pop up error window if the Scenario is not formatted right.

10. **Exit**

This button exits Scenario Maker without saving.

3.4 Command Maker
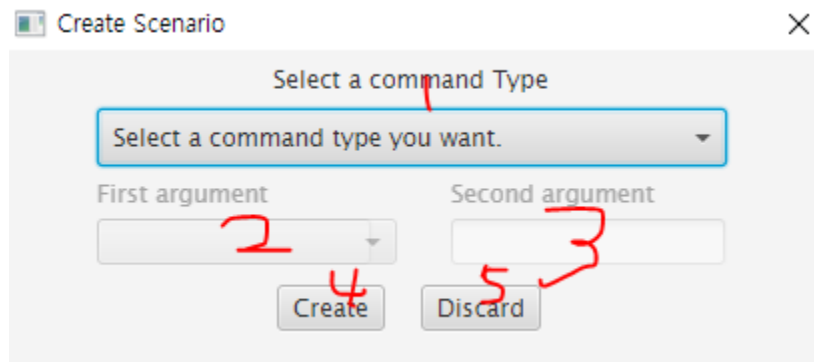
This is a Command Maker. You can make a single command with arguments of it.
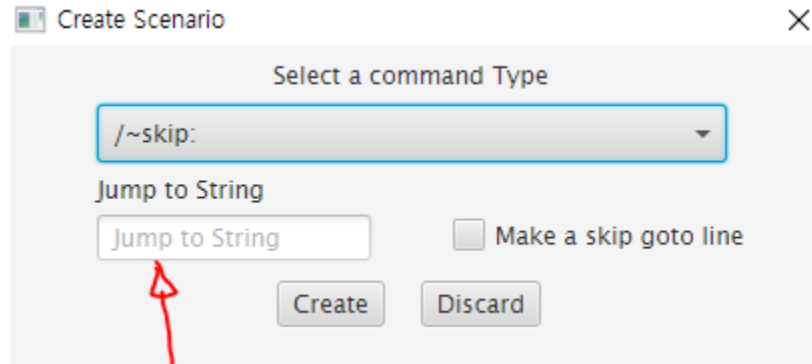
1. **Command list box**

This box contains a list of commands to make a Scenario. For each command's description, please look also : https://wiki.eecs.yorku.ca/course_archive/2017-18/W/2311/_media/scenarioformat.pdf
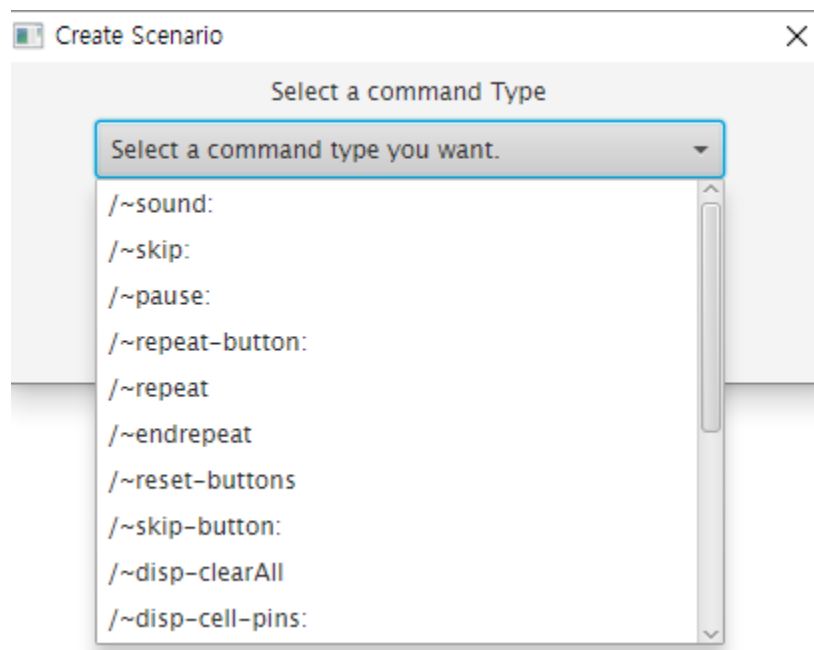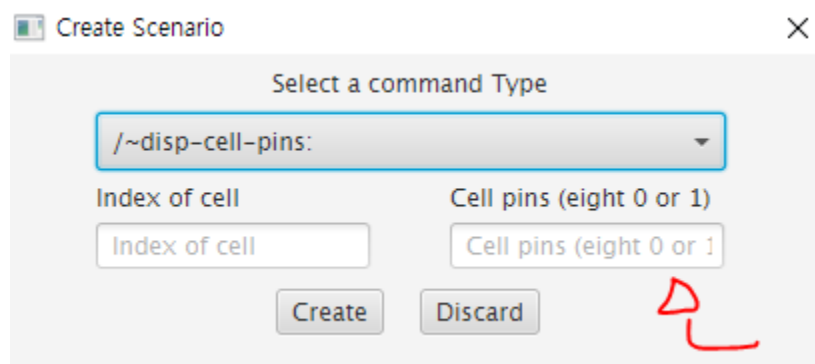
2. First argument box

This box is available for a command that needs first argument.

3. Second argument box

This box is available for a command that needs second argument.
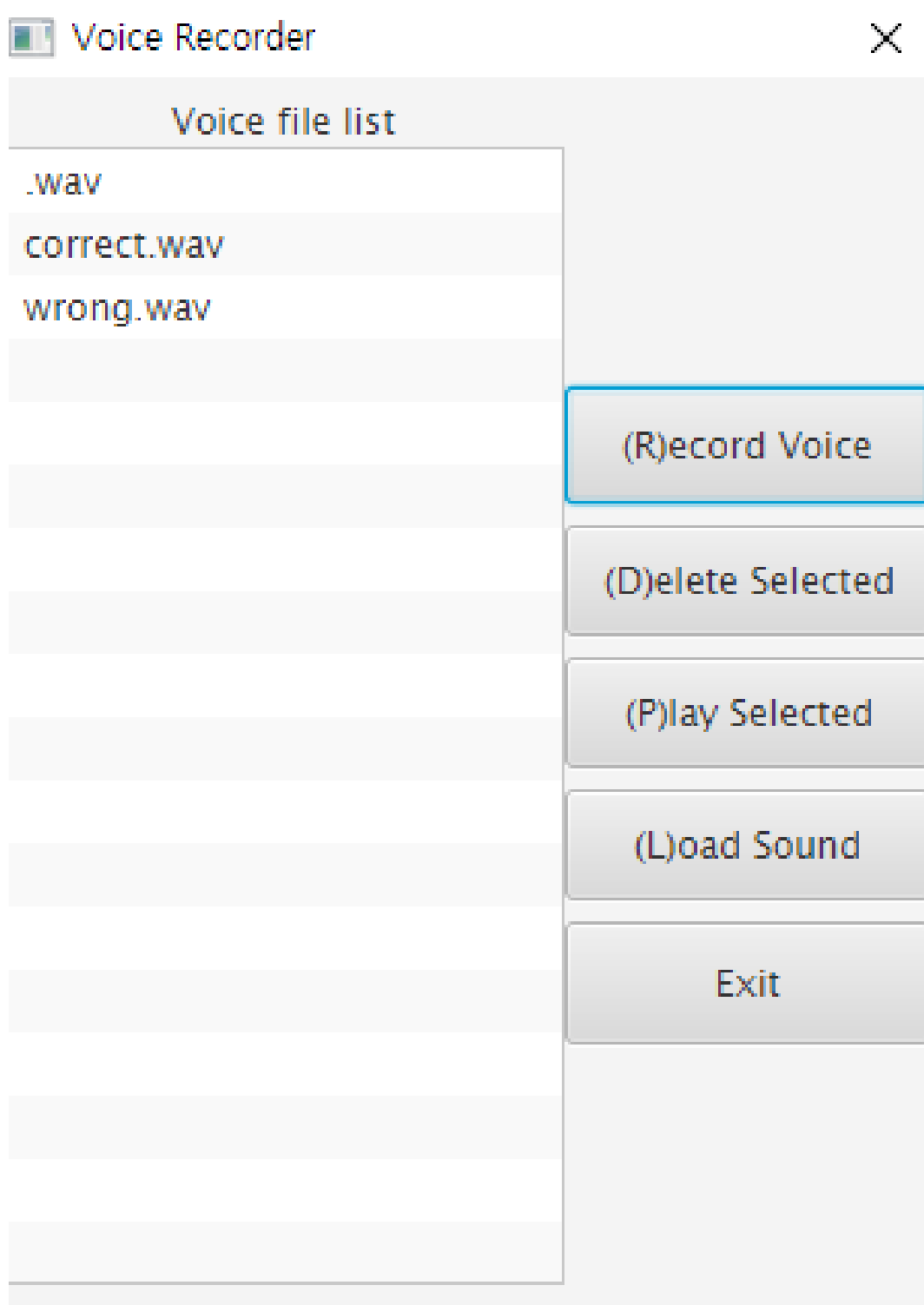

4. Create

This button will check if the arguments for selected command is formatted right and will insert command on the command list.

5. Discard

This button will exit Command maker without adding another command on the command list.
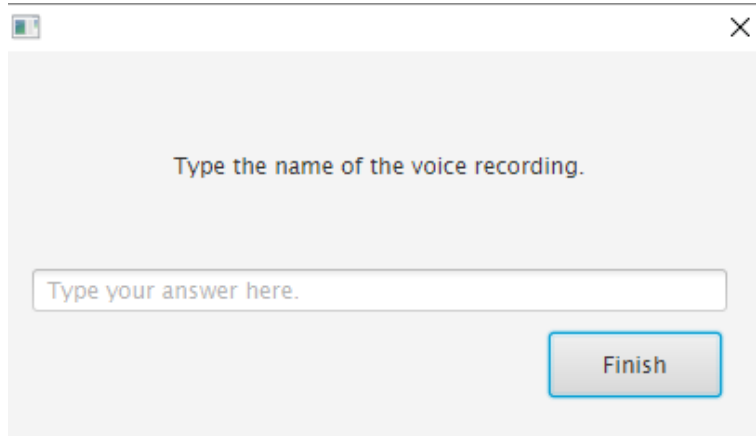
3.5   Voice Recorder

This menu lets you manipulate sounds to use on the Scenario.

1. **Record Voice**

Once you press this button , the program will ask you to name the recording's name:

Once you press Finish, the recording will start. To finish recording, you simply press record voice button again. The recorded sound will be saved locally and on the Voice file list.

2. **Delete Selected**

This will delete the selected sound file from the computer and the list.

3. **Play Selected**

This will play the selected sound file.

4. **Load Sound**

This will let you import a custom sound track or sound file for you to use.

5. **Exit**
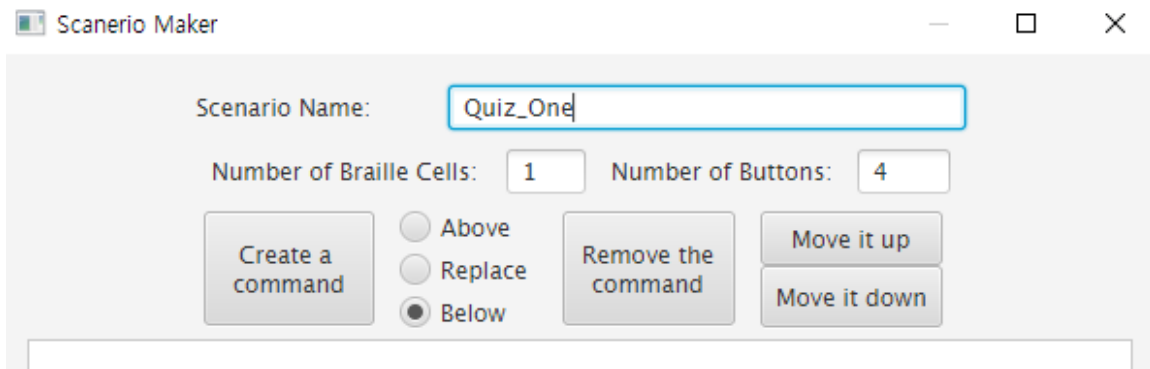
This will exit the Voice recorder.

4. Usage Examples

Here are some of common usage examples of the Authoring app.

4.1 Create a brand-new Scenario

In this example, we are going to make a simple Scenario that shows a braille cell composition and tell student what it is and simulate it.
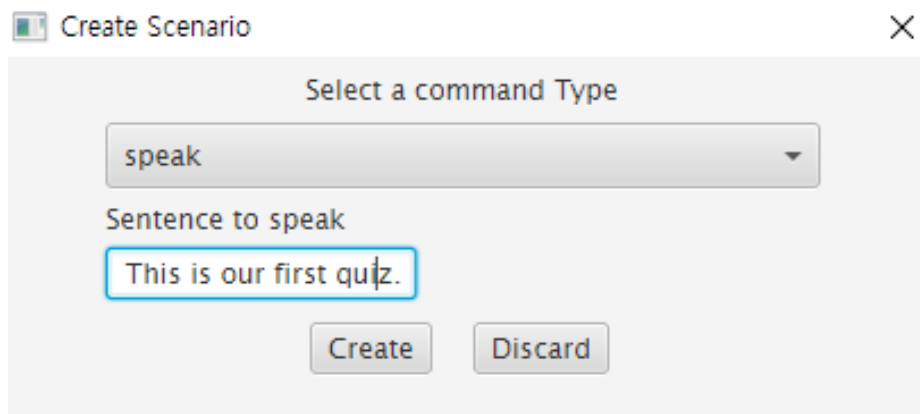
1. Open Authoring App.
2. Go to Scenario Editor.
3. Press Create a New Scenario button.
4. Name the Scenario. In our case, I will name it as Quiz_One.
5. Fill up Number of cells to work on and number of buttons. In our case, we will have one cell with four buttons.
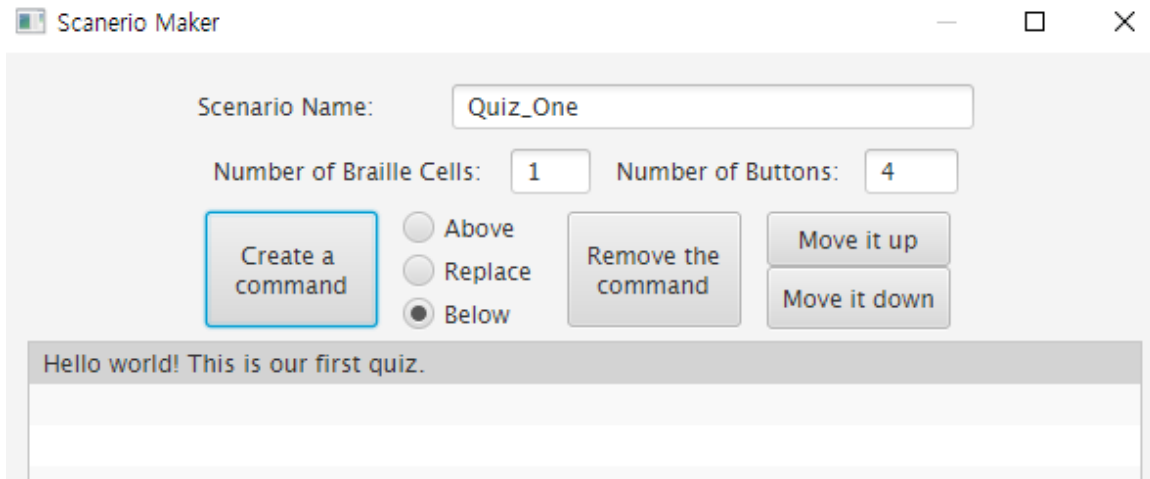

6. Press Create a command.
7. Select Speak on the command box, and type "Hello world! This is our first quiz."
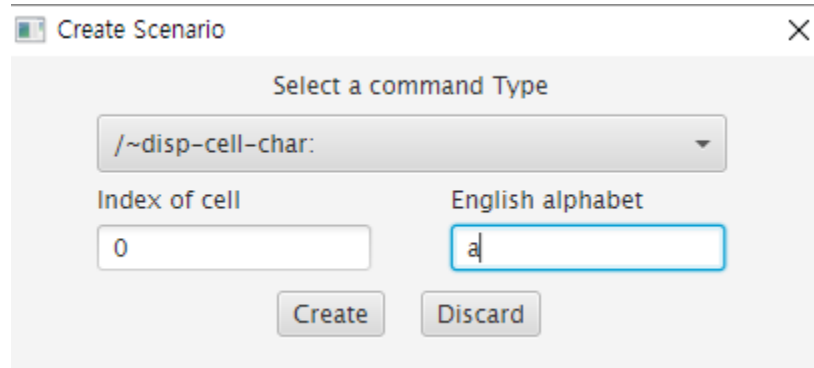
8. Press Create. This should have created a speak command line which simulator will read and speak.

9. Press Create a command again.
10. Select "/~disp-cell-char:". This sets the index of cell to the representation of a single alphabet. Type "o" on the index of cell and "a" on English alphabet.
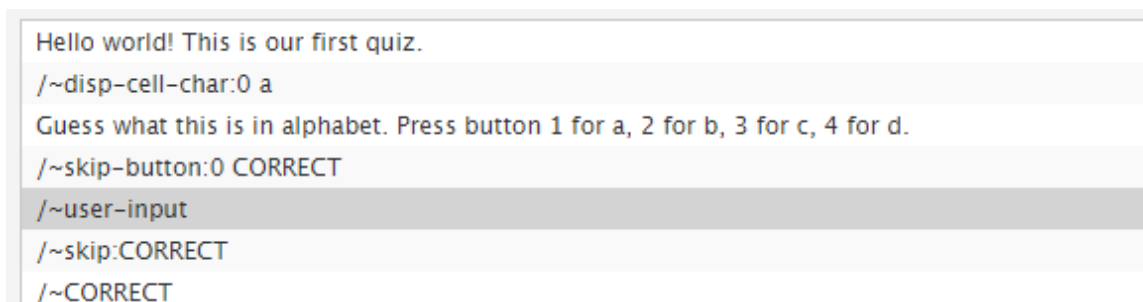
11. Now, add another speak command with "Guess what this is in alphabet. Press button 1 for a, 2 for b, 3 for c, 4 for d."

Scanerio Maker — □ ✕

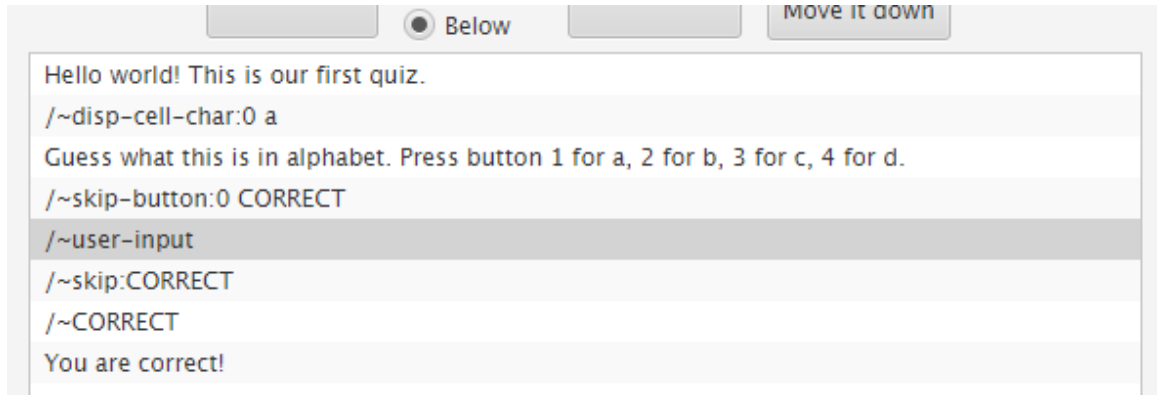Scenario Name: Quiz_One

Number of Braille Cells: 1   Number of Buttons: 4

○ Above   Move it up
Create a   Remove the
command   ○ Replace   command   Move it down
● Below

Hello world! This is our first quiz.
/~disp–cell–char:0 a
Guess what this is in alphabet. Press button 1 for a, 2 for b, 3 for c, 4 for d.
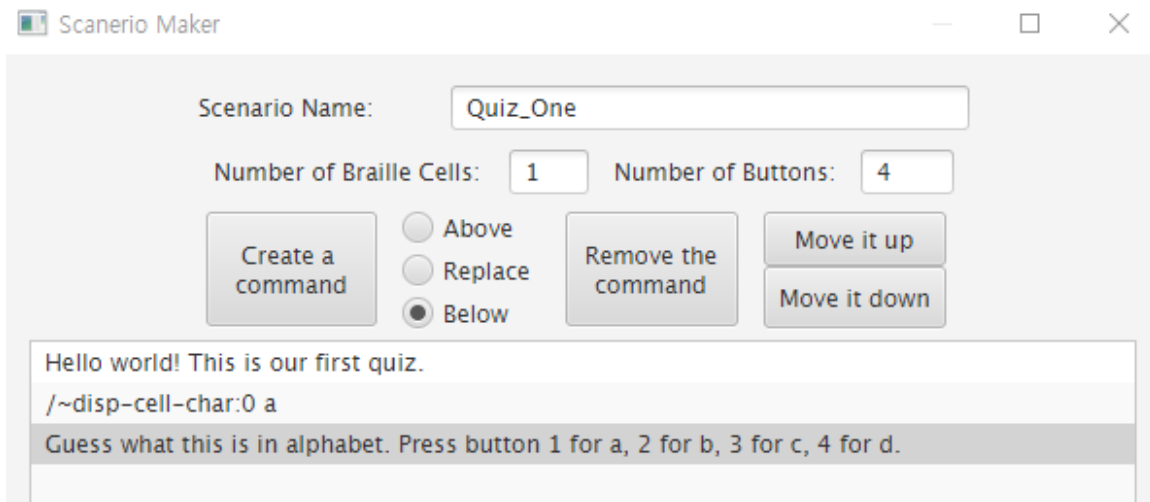
12. Now we need to make a button interaction. Create a command with "/~skip-button", with index 0 and Jump to String "CORRECT"

Create Scenario   ✕

Select a command Type

/~skip–button: ▼

Index of button   Jump to String

0   CORRECT|

Create   Discard
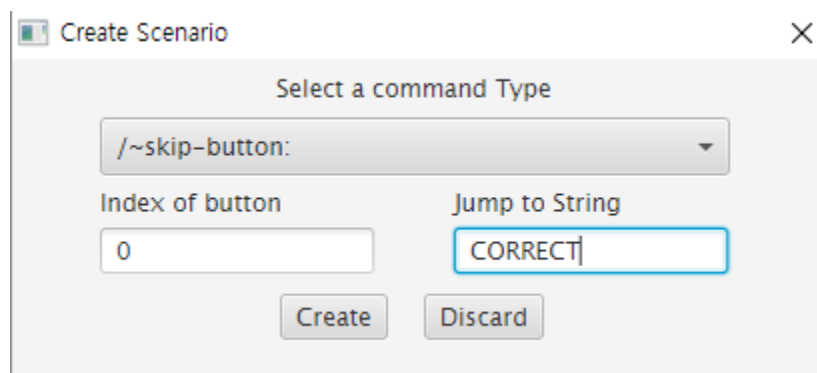
13. Let's make program wait for user input. Create a command "/~user-input"
14. Also, we need to create the point where the button click yields correct or incorrect. Make a new command with "/~skip:" with Make a skip goto line enabled.

Hello world! This is our first quiz.
/~disp–cell–char:0 a
Guess what this is in alphabet. Press button 1 for a, 2 for b, 3 for c, 4 for d.
/~skip–button:0 CORRECT
/~user–input
/~skip:CORRECT
/~CORRECT

15. Now, we have to tell students that their selection was right when selected. Let's add speak line saying the student was correct. Once you are done, it should look like this:


```
                                    ● Below                        Move it down

Hello world! This is our first quiz.
/~disp-cell-char:0 a
Guess what this is in alphabet. Press button 1 for a, 2 for b, 3 for c, 4 for d.
/~skip-button:0 CORRECT
/~user-input
/~skip:CORRECT
/~CORRECT
You are correct!
```
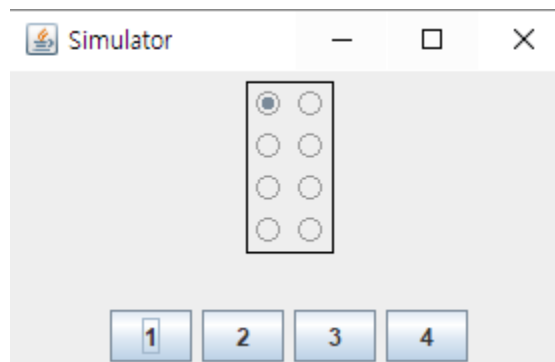
16. Let's save our progress. Press Save and Exit button.
17. Now we have our Scenario set up on Scenario Editor. Let's select it and run it.
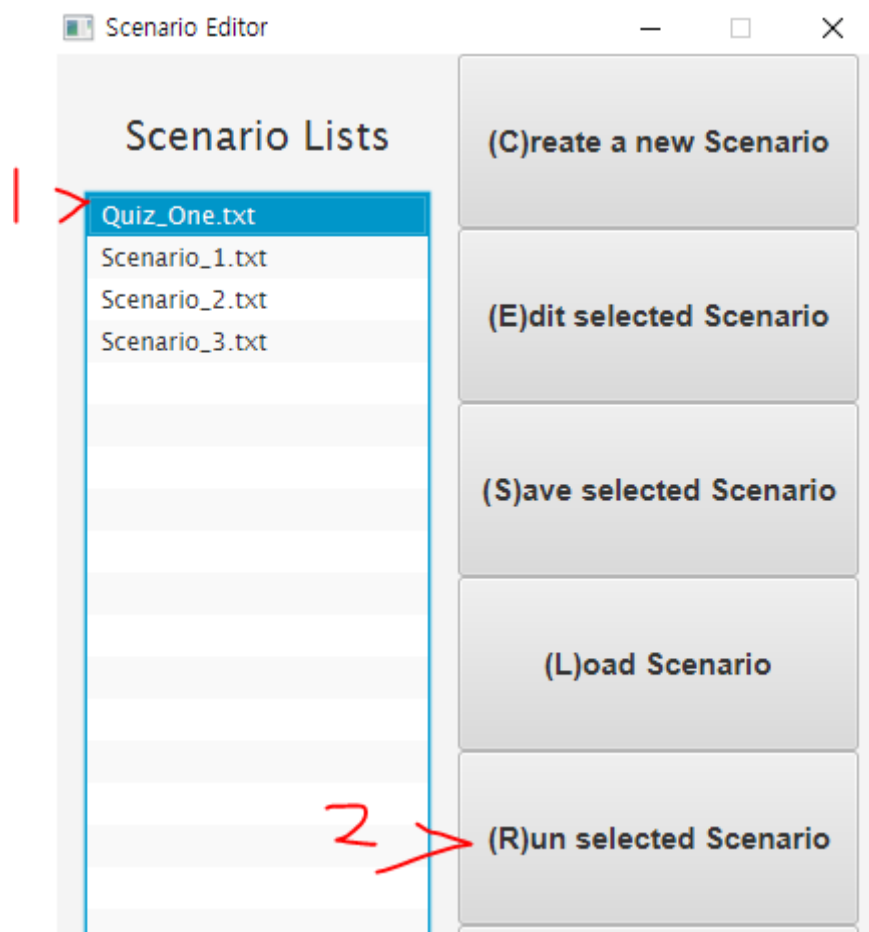
18. You can simulate your Scenario with either visual mode or audio mode. Let's do Visual mode for now. Press Visual Player.
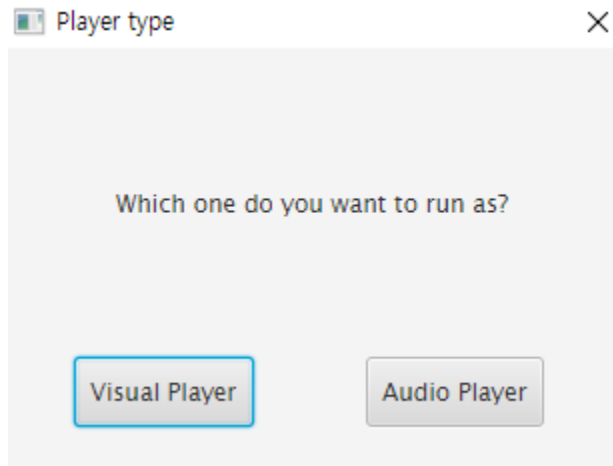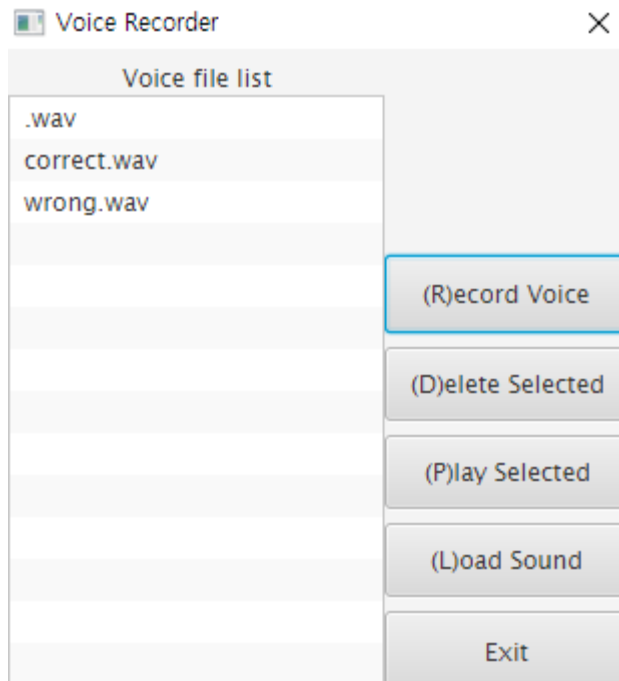

19. It will start saying the line we set. Press the button 1 to check if we yield correct answer.
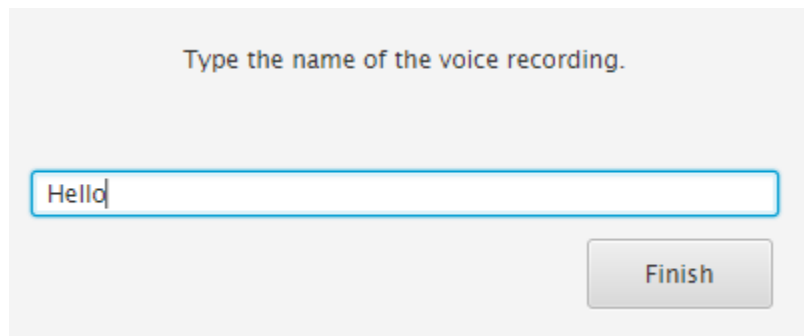
4.2   Record Voices

1. Open Authoring app
2. Open Voice Recorder.

3. Press Record Voice.
4. Type in the sound file name you want to put. Say, "Hello".

5. Press Finish, and the recording will start. Press record voice again to finish recording.
6. Now you can use this sound file on the scenario with play sound command, or you can export it for your own use.