Software Requirements Specifications

TreBBA: Treasure Braille Box Authoring

Version 1.0

EECS2311 Submission: Group 10

Amaan Vania
Amer Alshoghri
Arsh Punia

Table of contents

Downloading and Running:

At the time of the project deadline, my team was facing issues in deploying the project as a .jar file. Nevertheless, here is link to the Github of TreBBA: https://github.com/Shogri/EECS2311-TBB

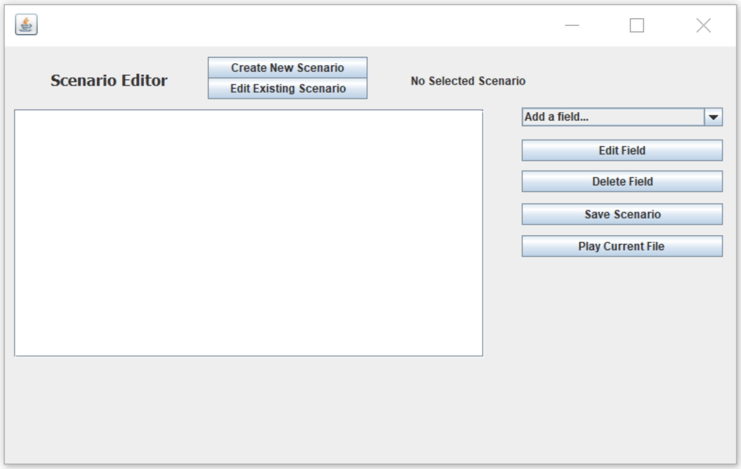
In light of present circumstances as we try and resolve the deployment issue, here is a brief tutorial on how to get TreBBA working if you have a Java IDE (example, Eclipse):

1. Navigate to the src folder: https://github.com/Shogri/EECS2311-TBB/tree/master/Enamel/src/enamel
2. Open and run the class EditorApp.java
3. You should now see a GUI like the one in the image below.

Documentation:

The documentation for version 1.0 of TreBBA can be found at: https://github.com/Shogri/EECS2311-TBB/tree/master/Enamel/Documentation

User interface:



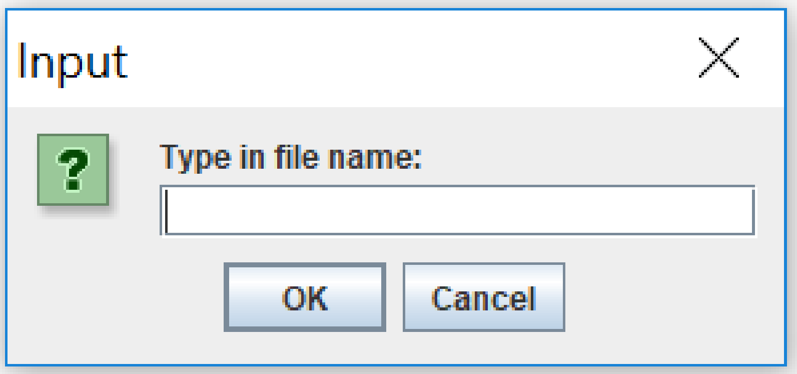
Upon launching the .jar file, the TreBBA user interface pops up. In the top left corner, one can see the option to either create a new file, or edit an existing file. To its right is the name of the chosen file. In this figure, that field is blank as no file has been chosen.

Below the Create/Edit buttons is a plain area which is where the user sees what they have input, and what their scenario file is structured like.

To the right of the plain area, one can see various buttons. On the top is a drop-down menu which enables you to select all the fields that you would like to add. These functions are explained in later sections.
To the bottom of the drop-down menu are buttons to edit a field, delete a field, save the scenario, and to play a given scenario. These functionalities too, are explained in later sections.

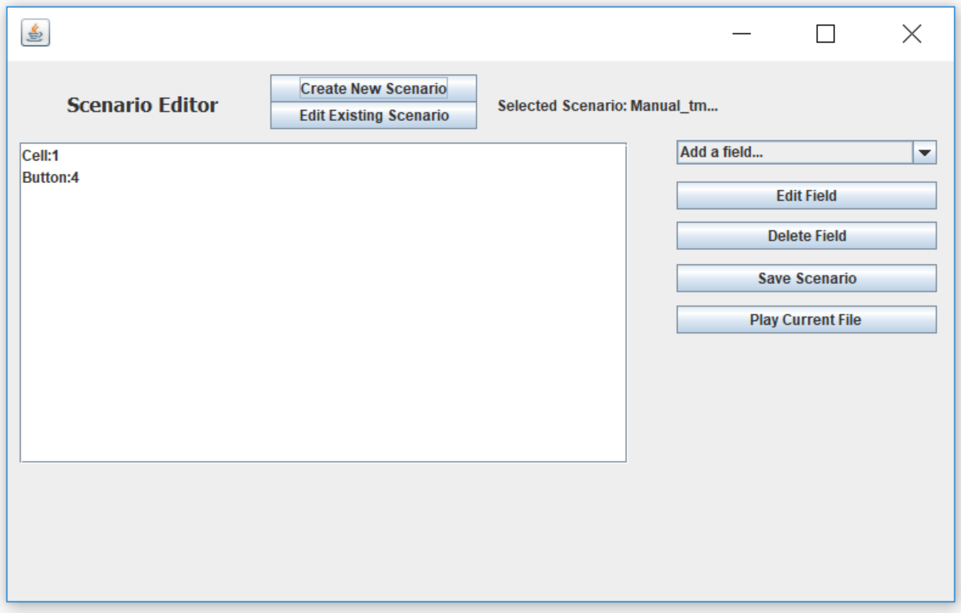
Getting Started

To get started, launch the.jar file. Once the TreBBA window pops up, click on "Create New Scenario" to get started on making a new scenario. This will lead to a popup asking for the file name. By default, files are saved in the same folder as the .jar file.



After you've entered the file name, there is another prompt for you to enter the number of cells and buttons.
'Cells' means the number of Braille cells that you would like to emulate.
'Buttons' Means the number of interactive buttons you would like to have in your scenario.

The number of cells and buttons that you would like has to be separated by a space.



Selected Scenario: Manual_tm...

Cell:1
Button:4

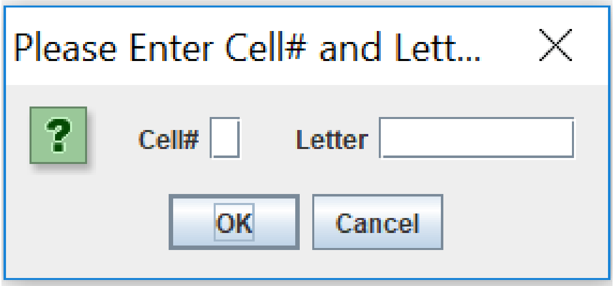
In the example above, the user has chosen to add 1 cell and 4 interactive buttons in a file called Manual.

DROP DOWN

The drop-down menu on the right of the plain area supports multiple operations which are as follows:

1. Display
2. Add Text
3. Ask Question
4. Specify Correct Answer Key
5. Begin Correct Answer Explanation
6. End Correct Answer Explanation
7. Specify Wrong Answer Key
8. Begin Wrong Answer Explanation
9. End Wrong Answer Explanation
10. Import Sound File
11. Display String
12. Add Pause
13. Start Repeat
14. End Repeat

Display: When the user clicks "Display", they are prompted for a cell Number and a Letter.
Cell Numbers start at 0. So if the user wants to display a certain string on cell 1, they would have to input 0 into the Cell Number. Efforts are underway to inculcate a more intuitive interface in later releases.
In the String field, the user has to input an 8-character string of 1's and 0's to display a certain string.



The prompt for Display

Once the cell number and text have been entered, the white text area is appended with the following line:
(In this example I am imputing character 11100000 into cell 0)
"Display cell #0 for the letter 11100000"

Add Text: This is a simple feature if the user wants the text to be read out as a part of the scenario. If clicked, the user is prompted for the string that they would like to enter, and that line is added to the white text area.

Ask Question: This Button leads to two prompts:
1. The Question
2. The buttons that would record the answers (These buttons are then activated)
When entering buttons, the user can input button numbers separated by a comma.

For example, if the user would like to activate buttons 1 and 2, the can simply type in “1,2”.
The question is then added to the interface.

Specify Correct Answer Key: This button helps the user keep track of the right answer. This key has to be one from the keys entered when the question was asked. When this option is chosen, and a valid key is entered (for example button 1), then the line “Correct Answer: 1” is added to the interface.

Specify Wrong Answer Key: Just like “Specify Correct Answer Key” except, this option is to specify the wrong answer key.

Begin Correct Answer Explanation: Choosing this option starts a user defined series of events if the right answer is chosen. For example, if the user would like the learner to say “Good Job!” after they have chosen the right answer, the user can click on “Add Text” after they have chosen “Begin correct answer explanation” and type in “Good Job”.
In addition to that, the user interface then has the line, “Correct Answer Explanation Starts here”
DISCLAIMER: **It is important that the user chooses “End Correct Answer Explanation” after the explanation has ended. If “End Correct Answer Explanation” is not chosen, this may lead to errors and the software might not be able to read the scenario file.**

End Correct Answer Explanation: This button is for the user’s reference to indicate that the correct answer explanation has ended. Note that of this option is not chosen the scenario file might not be able to be read by the software.

Begin Wrong Answer Explanation: Just like “Begin Correct Answer Explanation” except for the wrong answer. In this case, “End Wrong Answer Explanation” has to be chosen to signal the end of the explanation.

End Wrong Answer Explanation: Used to signal the end of the wrong answer explanation.
NOTE: If this question is the last question in the scenario, it is important that the user add some text after this, else there might be issues in the software not being able to read the scenario file.

Import Sound: Using this option, the user can play a given sound as a part of the scenario. When this option is selected, the user is prompted to select a file from a file chooser. TreBBA takes care of the rest.
NOTE: The file has to be a .wav file

Saving Files:

To save a file, click on “save File” Button and your scenario would be saved.

Playing a Scenario:

To play a scenario, click on “Play Current Scenario Button”