

SDP-16 Treasure

Treasure Box Braille

User Manual

Chapter 1

Getting Started

In This chapter you will learn:

- ✓ *How to install the Treasure Box Braille program*
- ✓ *The ins and outs of the functions in the TBB*
- ✓ *Loading a previously created scenario on the treasure box*
- ✓ *Setting up a brand new Scenario*

Welcome

Welcome to SDP-16's Treasure Box Braille (TBB) system. A user friendly program designed to create educational scenarios for the visually impaired. These scenarios enable the visually impaired to learn braille through the use of a series of generated questions created by the instructor/user. Let's get started on how to use the Treasure Box Braille system.

Common User Functions

The TBB has wide range of functions for you to apply. The program allows for you to either open scenarios on a simulator to run, or create a brand new scenario from scratch. Additionally, you can open unfinished or outdated scenarios to edit and modify. The easy to use UI for scenario creation makes accessing questions and editing responses simple and easy.

Furthermore, the TBB also allows the instructor/scenario creator to upload audio which can be played at specific times during the scenario.

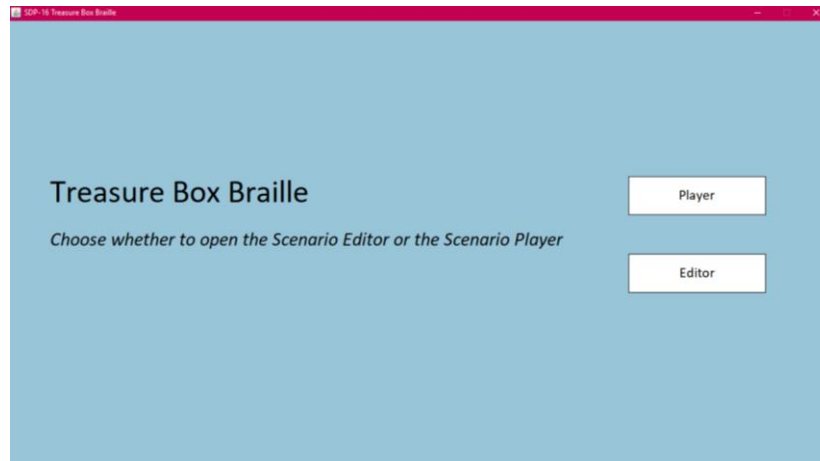
Installation

The Treasure Box Braille is designed for easy start up use. The entire program can be found in an archive file format (.zip). The name of the file is *result.zip*. You can get access to the contents of this file by right clicking the *result.zip* and clicking on extract. Select where you would like to place the contents of *result.zip* by clicking on the 'browse' icon.

The file containing the program will now appear on that directory. After opening this folder, there should be a runnable file called *tbb.jar* and another folder. Ignore the folder and double click to run *tbb.jar*. The program should now start.

Opening or Creating a Scenario

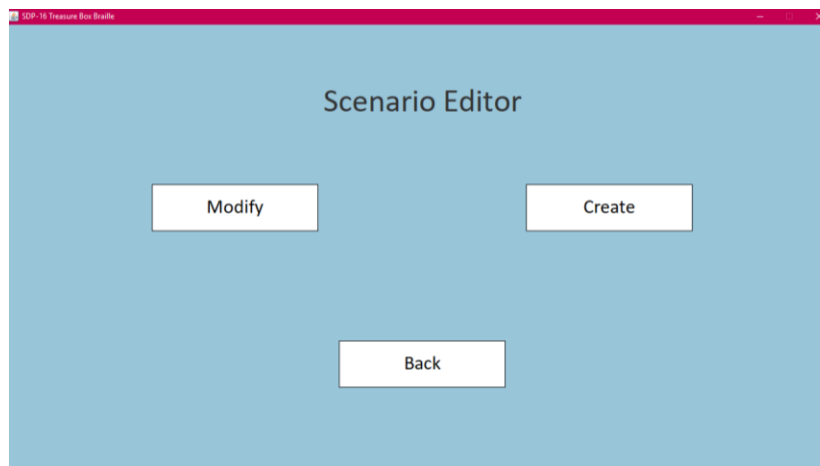
After installing and opening the program, the first screen you will see is the main menu. It consists of the option to run the Player or the Editor. If you already have a scenario created and wish to run it, click on Player. Here you will be prompted with a file explorer to open the Scenario_#.txt text file. NOTE: It is important that the name of the scenario file follows this format. Otherwise you will receive an error from the program. If you would like to test the player, feel free to open one of the factory scenarios provided in the FactoryScenarios folder.



The main menu of the TBB which consists of a Player and an Editor Selection

If your scenario has not yet been created, then proceed to click the Editor Button. A new menu called the Scenario Editor will pop up. This menu will ask you whether you would like to create or modify a scenario. If you would like to modify an already made scenario, then click on modify. A file explorer will open. Find the destination of where you saved your Scenario_#.txt file and click open.

If you wish to create a new scenario, then click on create.



The Scenario Editor Menu which gives options for modifying an already created scenario, or creating a brand new one. Additionally, the back button is present for returning to the main menu.

Modifying a Scenario

The ability to modify an already made scenario is still in production by the development team. Thus, no instructions have yet been made and will appear here on this document after release.

Creating a Scenario

If you wish to create a scenario, click on the create button and you will be prompted for 3 different inputs. Follow chapter 2 for further instructions on creating a new scenario.

Chapter 2

Creating a New Scenario

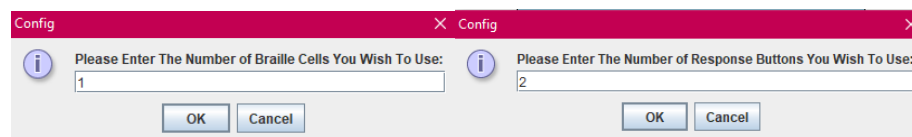
In This chapter you will learn:

- ✓ How to initialize a brand new scenario
- ✓ The differences between an event and an action
- ✓ How to add, remove, and configure actions and events
- ✓ How to build your project
- ✓ Sample Project built by the developers

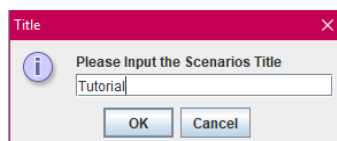
The following chapter is a tutorial on how to create a new Scenario using the Scenario Editor. The chapter will re-enact the creation of a simple scenario that will ask the user what character appears on the braille cell.

Initializing your new Scenario

Once you have clicked on the create button in the Scenario Editor Menu, you will be prompted to input the number of Braille Cells and Response Buttons you would like to have appear in the scenario. Furthermore you will be prompted to input the title of the scenario to be read at the beginning of the simulation.



Note: The Number of Braille Cells and Response Buttons must be a positive integer greater than 0.



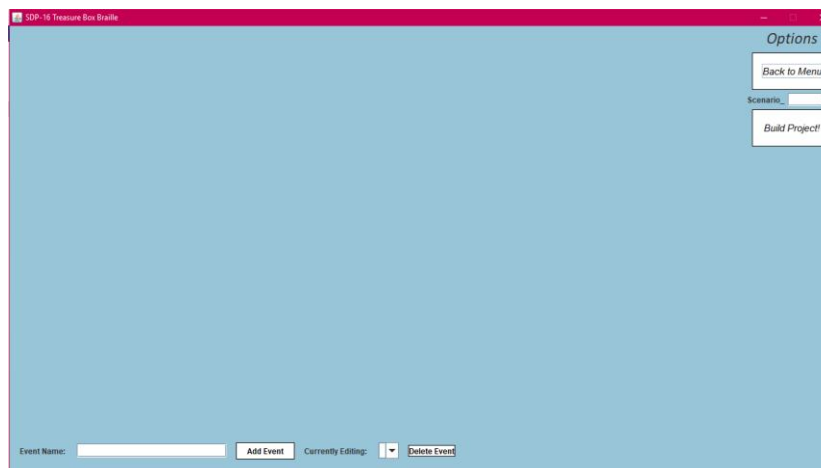
Note: The title of the Scenario can be left blank should you not desire one

Creating your first Event

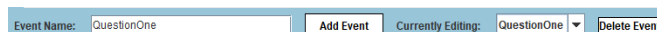
It is important to note the difference between an event and an action. An **Action** can be seen as a specific command that you would like to have run. An example would include the use of “text to speech” to output a question. Or the command “user input” which tells the program to wait for the user to click a button. The list of all possible actions along with their descriptions can be found in chapter 3. An **Event** is a combination of actions used to complete one desired task. For example, if you would like to ask the question, “What character is being displayed on the braille cell?, followed by the responses for correct or wrong answers, you would create 3 events called, QuestionOne, CorrectR, and IncorrectR.

Note: Naming is up to the user’s choice. These specific names are not mandatory.

When you have completed typing the scenario title, you will come across the Scenario Editor screen. The screen will originally look like this:



On the bottom left, there is a text field called Event Name. Here type in the name of your first event. In our case we will call it: **QuestionOne**. Note: numbers and spaces are not allowed to be used when creating event names. Additionally, duplicate names are not allowed and are checked without case sensitivity. After typing the event name, click **Add Event**. The event will appear on the drop box beside **Currently Editing**.

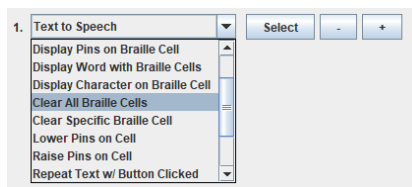


You can cycle through your events and edit the actions associated to each one via that drop box. Additionally, the delete button can be used to remove any unwanted events. Note that removing an event will cause any previous references to that event from actions to return an error. Thus causing the build to fail. Please fix these references accordingly.

A new screen will now come up containing a section with a drop menu and addition/removal signs.

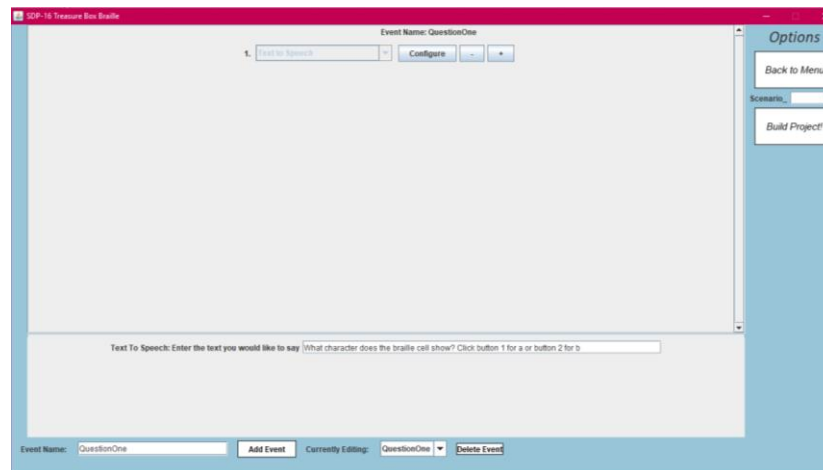
Creating your first Action

The following is an example of what a single action looks like. By Clicking on the drop box, you will be given an option to select any of the possible commands.



We will select the “Text To Speech” command. After selecting your desired action, click on the *Select Button* to confirm. Once confirmed, the only way you can change a command is to remove it via the – *Button* and re input the action via the + *Button* of the previous action. After clicking select, the select button will turn into a *Configure* button. Click configure to change the settings of the action. A

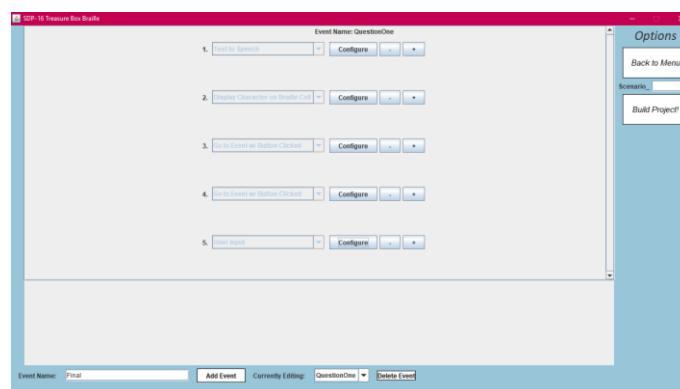
menu will pop up at the bottom of the screen showing the action's unique configuration settings. You can go in depth with the functions of each command in chapter 3.



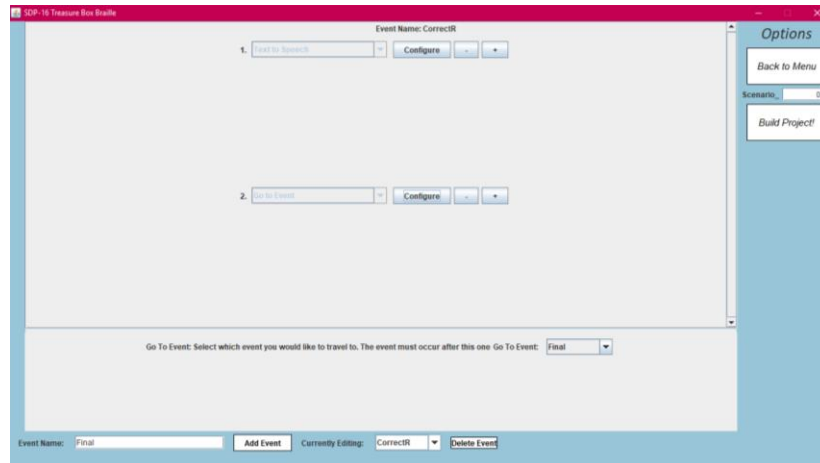
Creating your first Program

You will input the question in the text field of the Text To Speech configuration screen. The simulator will output the text as speech when its action is reached. Every action is listed in order starting from one. Four more actions are now added using the *+Button* found on the first action. Additionally, 3 more events called correctR and IncorrectR and Final are added.

The first action in QuestionOne is a text to speech stating the question to the user. The second action is a configuration of the braille cell. The action “Display Character on Braille Cell” allows a specific alphabet character to be added. In our case we chose the character ‘a’. Additionally, the 3rd action sends the program to the event CorrectR if button one is clicked. Action 4 sends the program to IncorrectR if button 2 is clicked. The last action is “user input”. This action tells the program to halt and wait for the user to click a button.

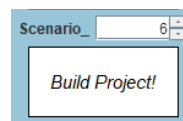


CorrectR and IncorrectR contain “textToSpeech” which simply state ‘correct’ or ‘wrong’. Each event then travels to a new event called “Final”. Final outputs as speech “You are now done”.

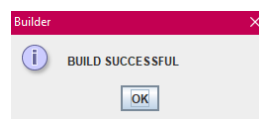


Building your first Program

Before building your Scenario, you need to give it a name. Since Scenarios must follow the *Scenario_#.txt* style, all you need to do is hover over the *Scenario_Selector* located just above the *Build Project* button on the top right corner. Select a number that does not already exist in the project folder. For example, selecting 6 will create a file called *Scenario_6.txt* in the program directory.



Now you are ready to build your project! Click the *Build Project* Button. If errors exist, the builder will output a message indicating an event name and action index which has not been configured properly. The built text file will then be deleted. Fix these errors and try again. If everything worked, you will receive a message saying *BUILD SUCCESSFUL!* Your scenario has now been built and can be accessed from within the TBB program files.



CONGRATULATIONS! You Have Just Built Your Very First Treasre Box Braille Scenario

```

1 Cell: 1
2 Button: 2
3 Tutorial
4
5 What character does the braille cell show Click button 1 for a or button 2 for b
6 /~disp-cell-char:0 a
7 /~skip-button:0 CORRECTR
8 /~skip-button:1 INCORRECTR
9 /~user-input
10
11 /~CORRECTR
12 Correct
13 /~skip:FINAL
14
15 /~INCORRECTR
16 Incorrect
17 /~skip:FINAL
18
19 /~FINAL
20 You Are now done
21

```

An example of what your txt file should look like

Chapter 3

Actions

In This chapter you will learn:

- ✓ *About all of the existing actions*
- ✓ *How to use them*
- ✓ *Recommended times on when to use them*

THIS SECTION OF THE USER MANUAL WILL BECOME AVAILABLE DURING THE FINAL LAUNCH OF THE TBB