

SDP-16 Treasure

Treasure Box Braille

User Manual

Table of Contents

Chapter 1: Getting Started.....	Page 3
Chapter 2: The Scenario Editor	Page 6
Chapter 3: Actions and Functionality.....	Page 9
Chapter 4: Sound Recorder.....	Page 16
Chapter 5: Common Scenario Creation	Page 18

Chapter 1

Getting Started

In This chapter you will learn:

- ✓ *How to install the Treasure Box Braille (TBB) program*
- ✓ *The general functions of the TBB*
- ✓ *Loading a previously created scenario on the TBB*
- ✓ *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 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 has two main functions. Running a scenario on the Scenario Player, or creating a scenario on the Scenario Editor. The Scenario Editor allows you to either start of fresh and create a brand new scenario, or load in a previously created scenario for modification. The easy to use UI for scenario creation makes accessing questions and editing responses simple and easy.

While modifying a scenario, the TBB allows the user to open a screen called the Sound Recorder (see chapter 4). The sound recorder is another one of the TBB biggest functions. Here, you can record audio via your microphone and use this audio in your scenarios. Additionally, the TBB allows for the user to import their own .WAV audio files for use in the Scenario.

For a full list and description of the SDP-16 TBB's functions, go to chapter 3: Actions and Functionality.

Installation

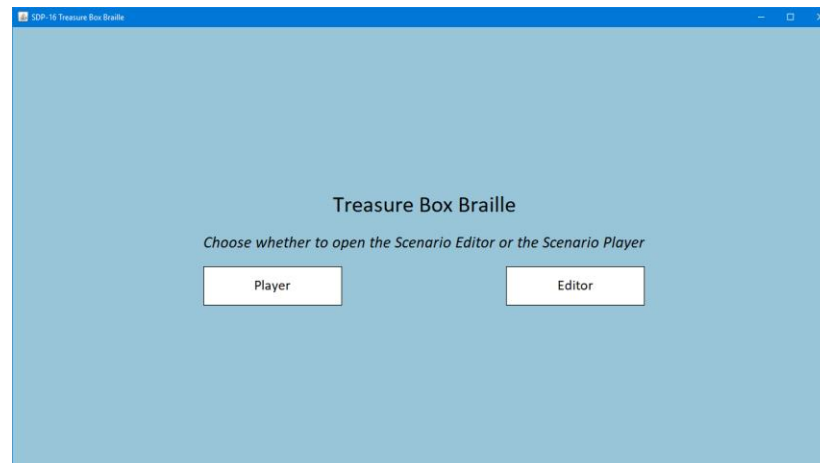
The Treasure Box Braille is designed for easy start up use. The entire program can be found in an archive file format (.zip). The name of the file is *result.zip*. You can get access to the contents of this file by right clicking *result.zip* and clicking on extract. Select where you would like to place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

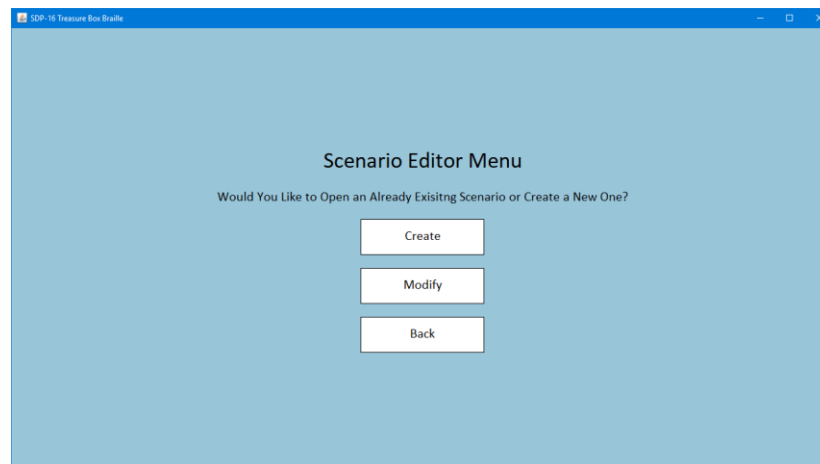
If you already have a scenario created and wish to run it, click on Player. Here you will be prompted with a file explorer to open the Scenario_#.txt text file. NOTE: It is important that the name of the scenario file follows this format. Otherwise you will receive an error from the program. If you would 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.

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

If you wish to modify an already created scenario, then click on the modify button of the Scenario Editor menu. A file explorer will open. Find the destination of where you saved your Scenario_#.txt file and click open. The Scenario Editor will now appear with all of the settings and configurations already set.

Creating a Scenario

If you wish to create a scenario, click on the create button. You will be directed to the scenario editor. A blank version of the Scenario Editor will appear. You can now begin creating your very own Scenario.

Chapter 2

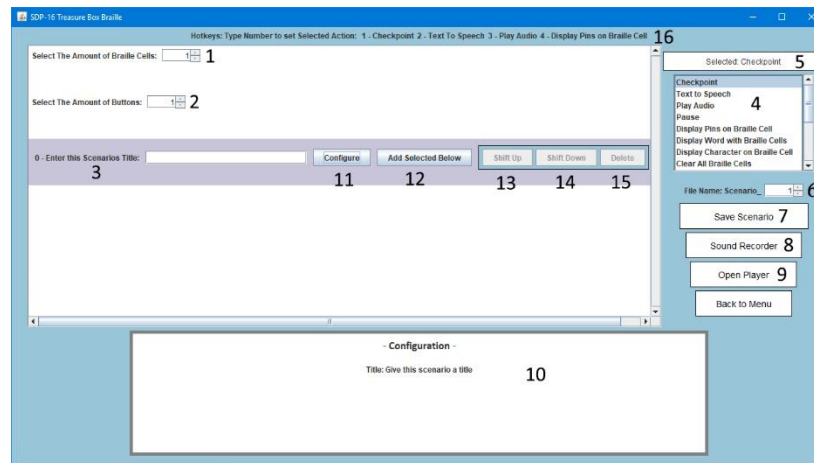
The Scenario Editor

In This chapter you will learn:

- ✓ The interface of the Scenario Editor
- ✓ The differences between a checkpoint and an action
- ✓ How to add, remove, and configure actions and checkpoints
- ✓ Hotkeys
- ✓ How to build your project

This chapter will give you the ins and out of the Scenario Creator interface.

The Scenario Creator Interface



This is where you:

Index	Description	Index	Description
1	select how many braille cells to use	10	Configure the selected action to do what you want it
2	select how many response buttons to use	11	Select an action to be configured from the configuration box
3	type the Scenarios Title	12	Add the selected action below this action
4	Select which action to add to the editor	13	Shift the selected action up one slot
5	See which action is currently selected to add to the editor	14	Shift the selected action down one slot
6	Select the Scenarios name	15	Delete the currently selected action
7	Build the Scenario to a Scenario File	16	View which hotkey to press for a specific action to be selected
8	Open the sound recorder to record audio		
9	Open the player to run a specific Scenario file		

The format of a Scenario

Every scenario consists of a series of checkpoints and actions. A checkpoint can be seen as a placeholder for a series of relevant actions. While an action is a specific task that the Scenario accomplishes when run. For example, if I wanted to ask the user, what character the braille cell shows, I would add the following:

- A checkpoint titled QuestionOne
 - Action: Text to Speech – Stating the Question
 - Action: Display character – ‘A’
 - Action: Go To Checkpoint with Button Clicks – One for each available button
 - Action: Wait for user input

Here you can see that a series of actions are clustered together under a checkpoint. This allows for proper organization of your scenario. Additionally, the Scenario Editor requires that checkpoints be used, as traversals throughout a scenario can only occur from one checkpoint to another. For example, QuestionOne will require 2 additional checkpoints:

- IncorrectAnswer
- CorrectAnswer

Depending on user input, QuestionOne will traverse to one of these 2 checkpoints and carry out their respective actions. It **cannot** traverse to another action directly.

Adding, Removing, and Editing an Action

After selecting the number of braille cells and response buttons you would like to use, and stating the title of the Scenario, your next step is to begin adding checkpoints and actions. To do this, the first thing you would have to do is select which action you would like to add. Select the action from the “action selection box” [4]. When you click a specific action, you should notice that “selected action” [5] changes to indicating the action you currently selected.

Now that you have selected a specific action, it is time to add it. Hover over the action you would like the new action to appear **under**. Now click on the “Add Selected Below” [12] button to create your new action. If you have not yet created an action, the first “Add Selected Below” button appears on the Scenario Title box. A new row will appear. This is a new action.

Every action and checkpoint appears as its own row in the editor. Each row contains its own index, the first number on the left of the row. This is the index of the action/checkpoint. The scenario title always contains index 0. Meaning it is impossible to delete or shift this row.

There are a lot of actions for you to choose from. This can make selecting common actions difficult at times. Thus, the TBB makes this easier. By using the hotkeys stated on the top of the editor [16], you can automatically select specific actions to add. Then click on the respective “add selected below” button to add your action. For example, if I wanted to add a “Play Audio” button, I would press the number ‘2’ on my keyboard. Notice how the “selected box” [5] changes to indicate you’ve selected the specific action.

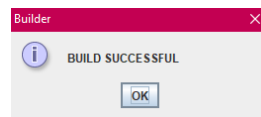
To configure the action you just added, click on the configure [11] button of that action. You will notice the configuration box [10] will appear. Follow the instructions on the configuration box to properly set up your action. For further instructions of configuring a specific action and recommendations, please see chapter 3: Actions and Functionality. If you wish to move the placement of an action, click on the shift up [13] or shift down [14] button of the respective action. If you no longer desire to have that specific action, click on the delete button [15] of the action.

To open the sound recorder and record audio, click on the “Sound Recorder” button [8]. See chapter 4 for more info on the sound recorder.

Building your scenario

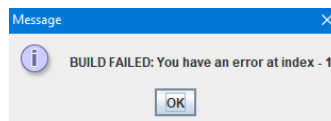
After creating your scenario, your next step is to give it a name. Go to the “scenario indexer” [6] and specify the name of the scenario you created. Note: The scenario indexer exists because a scenario can only have a name of format Scenario_#.txt. If the index you selected already exists, you will be prompted for overwriting the scenario when you try to save.

To save your scenario, click on the “Save Scenario” button [7]. If your scenario did not contain any errors, you will receive a build successful prompt as such:



You will then be able to run the newly created scenario on the scenario player.

If your scenario does contain errors, you will receive a prompt saying errors exits with the index of the error stated. So that you may revisit the action and fix any problems. The prompt for an error at action indexed 1 looks like this:



For quick access to the Scenario Player, to perhaps, test out your newly created scenario, click on the “Open Player” button [9].

Chapter 3

Actions and Functionality

In This chapter you will learn:

- ✓ *The functions of the SDP 16 Treasure Box Braille*
- ✓ *Setting up Actions*
- ✓ *Recommended times on when to use Actions*

Actions

Actions are the key components to creating a scenario. Every task, objective, or act that you wish to accomplish requires the use of a specific action. Thus, a complete scenario for use in the Scenario Player is just a collection of actions working together in chronological order of appearance. The TBB has a wide range of actions for users to use. The complete list of actions, and recommended usage can be found below:

Action: *Checkpoint*

Definition/Function:

The checkpoint is a header action which is used to describe the context of the next occurring actions. For example, a checkpoint called “QuestionOne” is used to describe that the next several actions will ask the user the first question.

Recommendations:

It is *HIGHLY* recommended that you utilize checkpoints. Not only are checkpoints useful for sorting your scenario into several tasks, they are also used by the TBB to indicate traversals to other tasks within the scenario. For example, checkpoint “QuestionOne” will have to reference checkpoint “CorrectResponse” or “IncorrectResponse” so that the TBB knows where to go depending on user input. Failing to provide checkpoints will cause the TBB to simulate the actions in chronological order which may remove any logical flow in the scenario.

Restrictions:

Checkpoint names must only consist of letters. Any special characters including spaces are not allowed. If two checkpoints in one scenario contain the same name (Not case sensitive), any reference to the checkpoint name from an action will traverse to the next occurring version of the checkpoint.

Action: *Text to Speech*

Definition/Function:

Text to speech is another commonly used action which takes the text written in its configuration and outputs it to the user during simulation. For example, you could use Text to Speech to ask the question: “Does the braille cell show the character a”.

Recommendations:

It is recommended that large portions of text appear as several text to speech actions occurring one after another. This way, the text can be broken up into several sections making it easier to make changes and locate errors.

Restrictions:

Text to Speech will only accept normal letters, spaces and numbers. Do not use any special characters.

Action: Play Audio**Definition/Function:**

The Play Audio action will play a selected audio file during the simulation. In order to select a specific .WAV audio file, click on the “Browse File” in the action configuration box. This will open a file chooser where you can select the desired audio file you wish to play.

Recommendations:

Use the sound recorder (Chapter 4) to create your own .WAV audio files which can be used in the scenario.

Restrictions:

The only allowed audio format is .WAV

Action: Pause**Definition/Function:**

This action causes the simulator to pause for a set amount of seconds. In order to use the pause action, click on the actions’s configure button and set the time that you wish to pause in seconds in the configuration box.

Recommendations:

None.

Restrictions:

The pause amount must be a positive whole number greater than 0. Invalid input is automatically handled as the value is set to the best valid representation of the invalid input.

Action: Display Pins on Braille Cell**Definition/Function:**

This action allows you to select a custom pin configuration for a specific braille cell. To do this, select the specific braille cell you wish to modify, then select which pins you would like to have appear on the braille image beside it.

Recommendations:

If you wish to have a specific character or word appear on the braille cell, it is recommended to use the display character or display word actions instead.

Restrictions:

None.

Action: Display Word with Braille Cell

Definition/Function:

This action will allow you to display any word of your choosing on the braille cells. To do this, simply type the word you would like to display in the text field of the action configuration box.

Recommendations:

Please use the “display character on braille cell” action if you wish to only display a single letter.

Restrictions:

Ensure that adequate number of braille cells are selected for the scenario so that the word can be properly outputted during simulation.

Action: Display Character on Braille Cell

Definition/Function:

This action will allow you to display any character of your choosing on a specific braille cell. To do this, simply type the character you would like to display in the text field of the action configuration box and select which braille cell you would like the character to appear on.

Recommendations:

If you would like to display an entire word on the scenario, use the “Display word on Braille Cell” action instead of repeatedly using this action.

Restrictions:

Ensure that the character you enter in the text field is a valid character from Aa – Zz.

Action: Clear All Braille Cells

Definition/Function:

This action clears any configurations set to all of the braille cells on the screen. Thus causing them to appear blank.

Recommendations:

The TBB does not automatically clear braille cells when a new checkpoint is reached. Please ensure you use this method after a specific checkpoint before traversing to another one. For example, if you wish to travel from QuestionOne to QuestionTwo, it is best to clear all braille cells first.

Restrictions:

None.

Action: Clear Specific Braille Cell

Definition/Function:

This action can be used to clear the configuration of a specific braille cell. To do this, simply select which braille cell you wish to clear in the action configuration box.

Recommendations:

If you wish to clear all the cells of the scenario, use the “Clear All” action instead.

Restrictions:

None.

Action: Lower Specific Pin

Definition/Function:

This action allows you to lower a specific pin on a specific braille cell of your choice. The indexes of the pins form as follows:

1	5
2	6
3	7
4	8

Recommendations:

If you wish to convert from one character to another one, which may take several pin lowers, use the “Display Character on Braille Cell” action instead.

Restrictions:

None.

Action: Raise Specific Pin**Definition/Function:**

This action allows you to raise a specific pin on a specific braille cell of your choice. The indexes of the pins form as follows:

1	5
2	6
3	7
4	8

Recommendations:

If you wish to convert from one character to another one, which may take several pin raises, use the “Display Character on Braille Cell” action instead.

Restrictions:

None.

Action: Button Repeat Text**Definition/Function:**

This action will repeat a specific block of text to the user using text to speech when a specific button is clicked. To use this, type the text in the text field using the same restrictions as text to speech, then select the button you would like to have clicked for the repeat to occur.

Recommendations:

Ensure that the button selected is not already in use by another relevant action. Doing this will cause the previous action’s button to be overwritten when the simulation reaches this repeat action.

Restrictions:

The text used must only consists of letters and numbers. No special characters.

Action: Go to Checkpoint

Definition/Function:

This action is used to traverse to a specific checkpoint. To use this, simply select the index of the checkpoint you would like to traverse to. The checkpoint index is the number found before the actions name. *Example: 4 – Enter a Checkpoint Name: QuestionOne*. This checkpoint has index 4.

Recommendations:

This action automatically traverses the simulator to the checkpoint. If you want to traverse after a button click, please use the “Go to Checkpoint with Button Click” action.

Restrictions:

The TBB can only traverse to checkpoints which occur after this specific action. Ensure that the checkpoint index used occurs after this one.

Action: *Go to Checkpoint with Button Click*

Definition/Function:

This action is used to traverse to a specific checkpoint after a specific button is clicked. To use this, simply select the index of the checkpoint you would like to traverse to. The checkpoint index is the number found before the actions name. *Example: 4 – Enter a Checkpoint Name: QuestionOne*. This checkpoint has index 4. After, select the button you would like to have clicked for the traversal to occur.

Recommendations:

This action traverses the simulator to the checkpoint after a button click. If you want to traverse automatically with no button click, please use the “Go to Checkpoint” action. Ensure that the button selected is not already in use by another relevant action.

Restrictions:

The TBB can only traverse to checkpoints which occur after this specific action. Ensure that the checkpoint index used occurs after this one.

Action: *Reset Button Configurations*

Definition/Function:

This action is used to remove all the configurations set for each button.

Recommendations:

The TBB does not automatically reset button configurations when a new checkpoint is reached. For example, if in QuestionOne, button 1 goes to an incorrect answer for QuestionOne, if button 1 is clicked in QuestionTwo, the same checkpoint will be traversed. Thus, it is important to reset button configurations at the beginning of new checkpoints, so that buttons start off fresh and do not conflict.

Restrictions:

None.

Action: User Input

Definition/Function:

User input is a very important action which halts the TBB simulator so that the user may enter a response through button click. Failure to include this action at the end of a question will cause the program to proceed to the next available action, without any user input.

Recommendations:

This action should occur at the end of a checkpoint, after all of the desired buttons have been configured. For example, QuestionOne should state the question, set the braille configurations, set a button for correct input and buttons for incorrect input. **After** this, the “User Input” action should appear to allow for the set configurations of the scenario to register.

Restrictions:

None.

Chapter 4

The Sound Recorder

In This chapter you will learn:

- ✓ The Sound Recorder Interface
- ✓ *The function of the Sound Recorder*
- ✓ *How to use the Sound Recorder*

The Interface

A visual representation of the sound recorder can be found below:



NOTE: In order to record audio, ensure that you have a working microphone attached to you device.

Index	Description
1	Name the audio file here. You should not include .WAV at the end
2	Click here to begin recording. If already recording, click here to stop recording

Using the Sound Recorder

NOTE: In order to record audio, ensure that you have a working microphone attached to you device.

The sound recorder can be accessed from the Scenario Editor (see chapter 2) menu. The button for the sound recorder can be found just under the “Save Scenario” button. When you click this button, a new window as seen above will open.

The TBB can only accept .WAV audio files. So the Sound Recorder automatically records audio in that format. To begin, type in the name of the audio file in the text field numbered [1]. You DO NOT need to include .wav at the end of the audio name. Including this will cause the file to be saved as “AudioName.wav.wav”.

After selecting a name, click on the record button [2]. If the audio name already exists, you will be prompted to overwrite. The record button will now change to say “Stop Recording”. You are now

recording from your input microphone. To stop the recording and save the audio, click on the new “Stop Recording” button now located at [2]. The audio file will save in the TBB default directory. To use it, simply “Browse” from a “Play Audio” action configuration.

Chapter 5

Common Scenario Creation

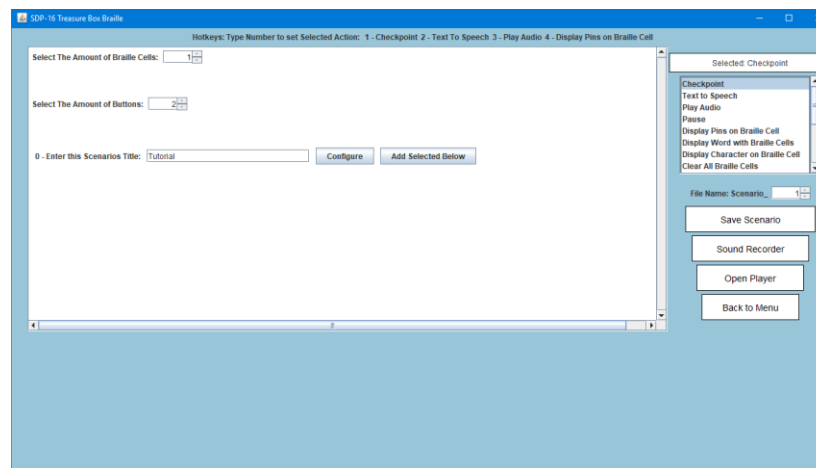
In This chapter you will learn:

- ✓ How to create a question in the TBB
- ✓ *Firsthand how to use the TBB*

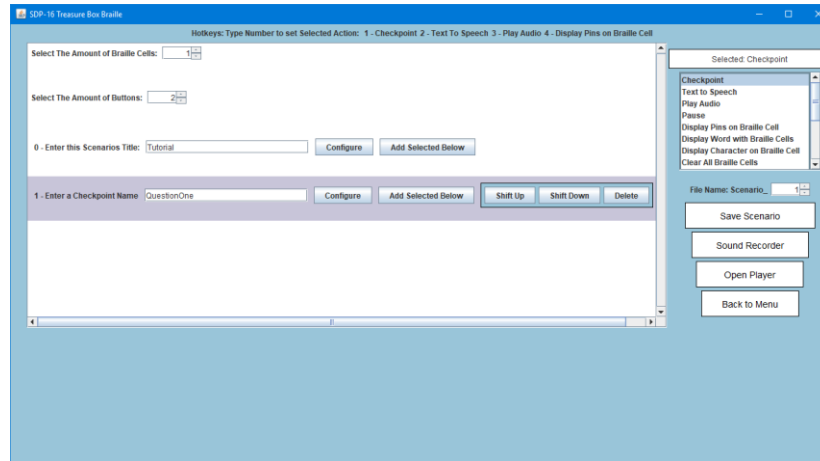
Creating a Question in the Scenario Editor

The most common use case of the TBB is creating a question asking the user what character appears on the braille cell. The following is a demonstration on how to create such a question. This can then be expanded to create your own questions using only slight modifications.

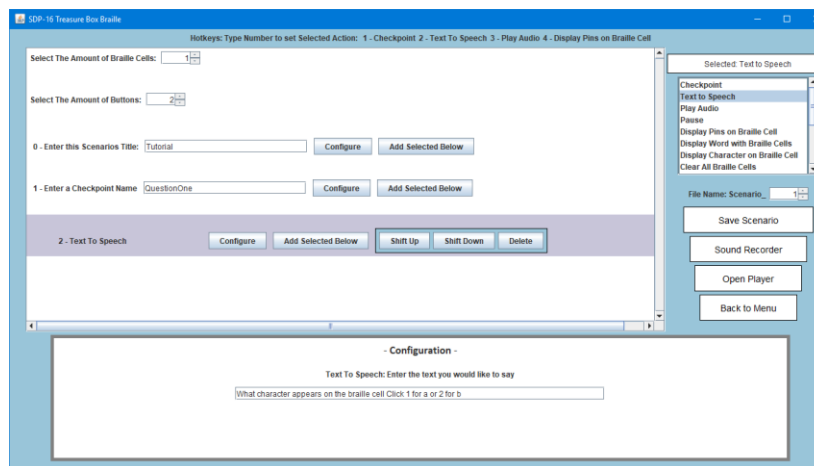
To begin open up the Scenario Editor (See Chapter 1: Getting Started). Set the number of braille cells to 1 and the number of response buttons to 2. For this case, the scenario title will be called “Tutorial”.



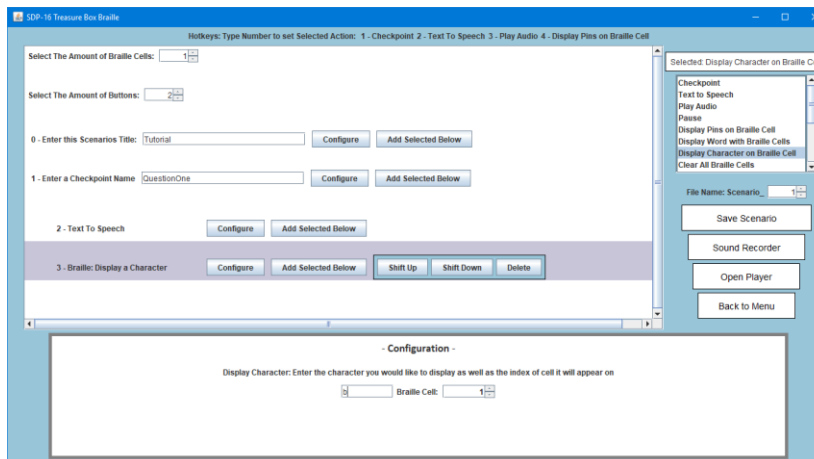
Add a checkpoint by selecting Checkpoint on the top right and clicking “Add Selected Below” on the title node. In our case we will call the checkpoint **QuestionOne**. Note: numbers and spaces are not allowed to be used when creating checkpoint names. Additionally, duplicate names are not allowed and are checked without case sensitivity.



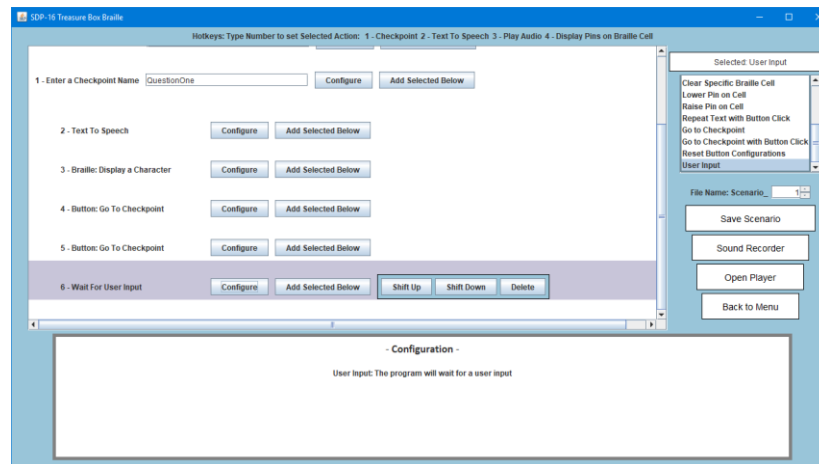
Now, add a “Text To Speech” action below the QuestionOne checkpoint. Click on the configure button of the newly created action. In the action configuration box, you will see configuration settings for TextToSpeech. Type in the question in the text field here: “What character appears on the braille cell Click 1 for a or 2 for b”.



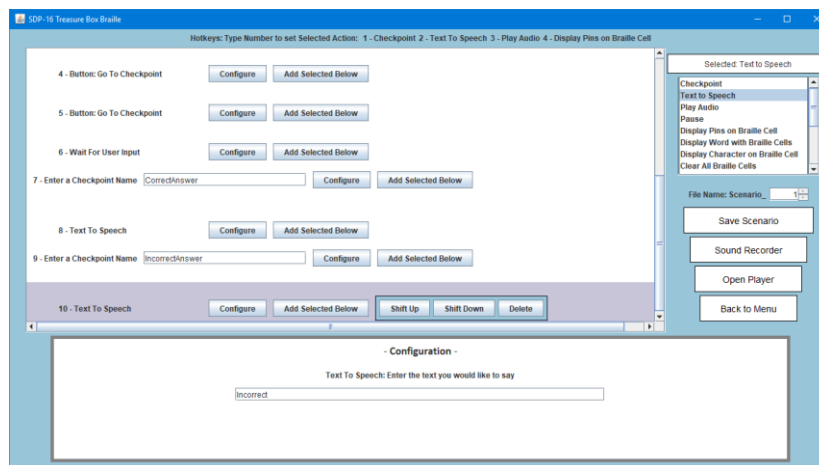
Now add a Display Character on Braille Cell action. Configure it to display the character b on braille cell 1 as below:



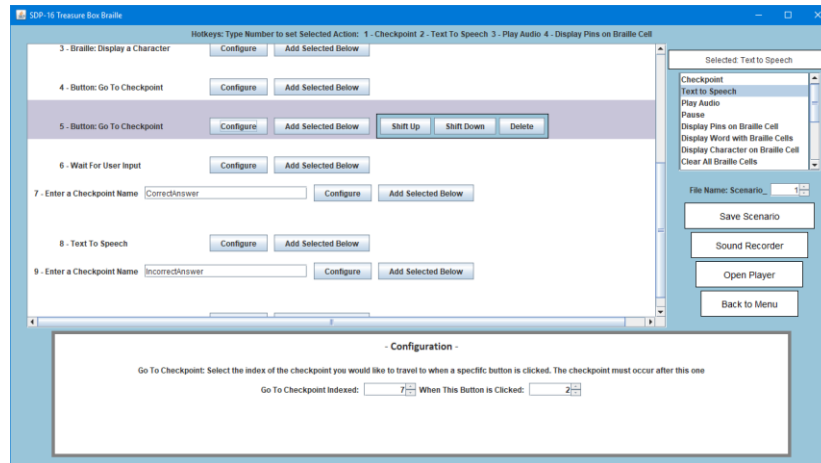
The next step is to add 2 “Go to Checkpoint with button click” actions. We will configure these later. Now add a “User Input” action so that the program knows to wait for user input.



The next step is to add 2 more checkpoints. One called “CorrectAnswer” and “IncorrectAnswer”. Each of these checkpoints will consist of a TextToSpeech action stating “Correct” or “Incorrect” respectively.



Once you have all of your actions placed, and are sure that the program is complete, go back to configure the 2 “Go To Checkpoint with Button Click” actions. Configure the first one as so: button 1, to go to checkpoint indexed 9 (Incorrect). The second one should be configured: Button 2 goes to checkpoint indexed 7 (correct).



This is a method for creating a simple question using the TBB. Most Scenarios will consist of a series of these types of questions. Which makes knowing this technique very important. Finally, save your Scenario (see chapter 2 for help).

Your final Scenario.txt file for this question will look like:

```
1 cell 1
2 Button 2
3 Tutorial
4
5
6 /~QUESTIONONE
7 What character appears on the braille cell Click 1 for a or 2 for b
8 /~disp-cell-char:0 b
9 /~skip-button:0 INCORRECTANSWER
10 /~skip-button:1 CORRECTANSWER
11 /~user-input
12
13 /~CORRECTANSWER
14 Correct
15
16 /~INCORRECTANSWER
17 Incorrect
18
```