**TalkBox Testing Documentation**

EECS 2311

Bryan Barcos Caro, Param Balaganeshan, Dhruv

**Table of Contents**

**CongifurationAppGUI**

1. Test Click Play Button

A test case that checks for the functionality of the Play JButton. The purpose of this button is to begin playing an audio file that has been selected by the user.

To ensure a audio file is correctly played, the setSelectedIndex() method is used to select an arbitrary audio file. Once selected, the Play button is clicked using the doClick() method. When running the Junit test, one can observe the sound being played if the test is passed.

To pass this test, the audio must begin playing after the Play button is clicked.

1. Test Click Stop Button

The functionality of the Stop JButton is tested in this test case. The purpose of the Stop button is to stop any audio that is currently playing.

To test if the stop button is fully functional, a file is selected and played. Once played, before the audio ends, the Stop button is clicked using the doClick() method. One can observe the audio file stop suddenly during testing.

To pass this test, the Stop button must successfully stop any file currently playing. If the Play button is clicked after the stop button, the audio file must be played from the start rather than resumed.

1. Test Click Pause Button

The purpose of the Pause button is to stop the audio file from playing, but contains the ability of resuming the audio file instead of starting from the beginning afterwards.

To test the functionality of the Pause button, a file is selected and played as done in the Play button test. Once an audio file is playing, the Pause button is clicked before the end of the file has been reached. After the audio output has stopped playing, the play button is clicked once again to check if the output is played from the correct place.

To pass this test, the Pause button must stop audio output when clicked without resetting the audio file to the beginning, providing the user with the ability to resume the audio output.

1. Test Click Resume Button

The purpose of the Resume button is to allow the user to continue playing an audio file once it has been paused.

Testing the functionality of the Resume button requires for an audio file to be played using the Play button, the audio file to be paused using the Pause button before reaching the end of the file, and then finally click the Resume button using the doClick() method.

To pass this test, the audio file must continue playing from the same place where it was paused rather than playing from the beginning.

1. Test Click Reset Button

The purpose of the Reset button is to allow the user to return to the default setting of their audio list. The default setting is having only 3 buttons available and an empty final list.

To test the Reset button, the application must return to default settings after being clicked. The methods getNumberOfAudioSets() and getNumberOfAudioButtons() are used, asserting that the values are 4 and 3, respectively. To further test the Reset button, 2 audio files are added to the final list and 2 buttons are also added. Once added, the Reset button is clicked and we once again assert the number of set and buttons return to their default. Also after clicking the Reset button for a second time, the audio array list is compared to the default list, and check that the final list is empty.

To pass this test, the number of audio sets after clicking reset must be 4, the number of audio buttons must be 3, and the final list must be empty. The initial list must also return to its default setting even after adding buttons.

1. Test Click Swap Button

The Swap button allows the user to cycle to the next audio files, where is equal to the number of buttons currently available.

To test the Swap button, the reset button is clicked to bring the application to its default setting of 3 audio files in the initial list. The first three audio files are added to a list prior to clicking the Swap button then compared to the files currently in the final list to ensure equality. Next, the following 3 audio files are placed in a list. This list should be equal to the final list after the Swap button is pushed. This same process is done for a second time to ensure accuracy and consistency.

To pass the test, the Swap button must correctly cycle through to the next available audio files when clicked.