EECS2311 – TalkBox Application

Testing Document

Group 6

Created: Feb 1, 2019

Last Updated: Feb 23, 2019

Author: James Kong

1. Introduction

This test document aims to describe the implementation and framework that will be used to facilitate the testing of the TalkBox configuration application and the TalkBox simulator. This includes:

* **Test coverage**: a comprehensive list of all tests used with a brief summary will be included for both the configurator and simulator respectively
* **Test case derivation & sufficiency**: how the test cases were derived and why they are sufficient will be explained
* **Testing implementation**: an explanation will be provided on how exactly how the test cases are implemented
* **Testing coverage metrics**:

1. Test Coverage

| **Application Testing Checklist** | | | |
| --- | --- | --- | --- |
| Application Name | TalkBox Configurator | | |
| Procedure | | Pass/Fail  (P/F) | Actual Results/Comments |
| Application Functionality | | | |
| Application executable launches successfully | |  |  |
| UI Layout dynamically adjusts to screen resolution/size | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| **Application Testing Checklist** | | | |
| --- | --- | --- | --- |
| Application Name | TalkBox Simulator | | |
| Functionality being tested | | Pass/Fail  (P/F) | Actual Results/Comments |
| Application executable launches successfully | | P | Running .jar file works properly with no errors |
| Upon running, application should display a file browser to user where they can select a configuration file to load into the simulator | | P | File browser opens after running .jar file |
| After selecting a configuration file, the simulator loads the right amount of icon panels | | P | Correct number of icons are loaded |
| After selecting a configuration file, the simulator loads the right amount of audio buttons | | P | Correct number of audio buttons are loaded |
| After selecting a configuration file, the simulator loads the right amount of toggle buttons | | P | Correct number of toggles are loaded |
| Accurate icons are loaded from the appropriate filepath | | P | Accurate icons are displayed |
| Accurate audio sounds are played respective to each button from the filepath | | P | Clicking each button plays the correct audio |
| Toggle buttons have accurate preset name as labels | | P | Toggles are named appropriately |
| Audio buttons have accurate button names as labels | | P | Audio buttons are named appropriately |
| GridLayout is properly loaded with the icon representations at the top and the audio buttons to the bottom | | P | GridLayout is setup correctly |
| When there is 1 preset, application should not load any the toggle buttons | | P | Tested with one preset config file, toggles did not appear |
| If there is more than 1 preset with toggles, the toggles should appear to the right | | P | Tested with a two preset config file, toggles did appear appropriately |
| When there is less toggles than audio sets, the last toggle should be named “next” and traverse through the presets | | P | Tested with a three preset config file and two toggle buttons, first toggle lead to first preset, second toggle was labeled ‘next’ and traversed through the 3 presets one by one |
| When there is equal number of toggles and audio sets, the toggles should bring users to the named presets | | P | Tested with a two preset config with two toggle buttons, first toggle leads to the first preset and the second toggle leads to the second |
| When there are less audio buttons than spots in a preset, the icon should be blank and the audio button should be empty with no action when clicked | | P | Tested with a three audio button configuration with only two buttons set and the empty button shows no icon and throws the proper exception when pressed with no error |
| TBA | |  |  |

1. Test Case Derivation & Sufficiency

The methodology used to obtain the test cases was to go through the user flow of both the configurator and simulator application from a user perspective. By going step by step through the applications, comprehensive testing can be guaranteed as each step in the user flow is accounted for.

For configurator testing, each possible flow will be walked through, and each step will be documented as a test. For example, there is a flow for creating a preset, loading a preset, and testing a preset. In the case of creating a preset, each step from clicking each button to go to the next screen to clicking a button to upload a sound file will be noted as a test.

The simulator testing will be similar, albeit with less user flows. The simulator tests are derived from the first step being loading in a configuration file from a file browser to the last step being clicking the simulator’s buttons to emit audio.

1. Testing Implementation

The testing will be implemented in the JUnit 5 framework within the Eclipse IDE.

The software has been tested and functions on macOS Mojave version 10.14.2.

1. Testing Coverage Metrics