Software Requirements Specification

For

Scenario Building Application for Braille-Based Device

Prepared by:
Dilshad Khatri
Drew Noel
Jonathan Tung
Alvis Koshy

Prepared on: April 4, 2017
For: EECS 2311 – Software Development Project

Table of Contents

1. Introduction

1.1 Purpose
This document entails the requirements for a scenario building application that complements a player application that parses the created scenarios. In the scenarios, there will be events that pertain to a certain function and the player will recognize these events and provide the needed functionality. The purpose of this project is to give a client a user-friendly and easy to use tool to author these scenarios while also giving them the ability to record and incorporate their own sound files into the scenarios.

1.2 Intended Audience
This project is intended for people who wish to make use of a braille-based teaching hardware by allowing them to create and save scenarios that can be utilized by the player application.

1.3 Project Scope
The scope of our project is to have a fully functioning application that incorporates events for every functionality that the player application has. This is so that when a user creates and saves their scenario, they can parse the scenario into the player application and it will recognize all the events that were in the scenario file.

2.Overall Description

2.1 Product Perspective
This product is handy and convenient tool that will allow users who are helping people - especially children - with visual impairment, by letting them have the flexibility of creating and saving scenarios for the braille-based teaching device.

2.2 Product Features
This application allows the user to create scenarios with a multitude of features, allowing the user to fully customization, change and add events to their scenarios.

2.2.1 New Scenario
This allows the user to initialize the bounds of the cell and button parameters of the application as setting a title for the scenario. It forces the user to create the bounds first before being able to add other features to the scenario by disabling the buttons until they are set.

2.2.2 Import
This feature lets the user import scenarios, allowing them to further customize the scenarios if needed. (As long as the scenarios still have the same format as the application)

2.2.3 Export
This feature lets the user save the current scenario that they were working on. Only active after the scenario has been initialized with some number of buttons and cells and a title.

2.2.4 Start Recording
This feature lets the user record their voice through a microphone when clicked. Only active after the scenario has been initialized with some number of buttons and cells and a title.

2.2.5 Stop Recording
This feature lets the user save their recorded voice and lets them save in any path on the computer. Only active after Start Recording is pressed.

2.2.6 Read Audio File
This feature allows the user to listen to audio files. Only active if the scenario has been initialized.

2.2.7 Move Item Up
This feature allows the user to move an event in the scenario up one place, making that event run first. Only active if an event is clicked and if it's not the topmost event scenario.

2.2.8 Move Item Down
This feature lets the user move an event in the scenario down one place, making that event run later. Only active if an event is clicked and if it's not the bottommost event in the scenario.

2.2.9 Delete item
This feature lets the user remove an event in the scenario leaving that place to be replaced by the event below it. Only active if an event is clicked.

2.2.10 New Item(Advanced)
This feature allows the user to add events to the scenario and it provides more of a hands-on level of customization to the scenario creation. It is only active after the scenario is initialized. It contains the following events:

-Pause: An event that makes the scenario pause for a specified amount of time
-Text-to-speech: An event that converts the text into audio
-Display String: An event that displays a string on the braille device
-Repeat: An event that repeats other events
-Button Repeat: An event that corresponds to repeating events with a button press
-Button Location: An event that assigns functionality to the buttons
-User Input: An event that corresponds to the user's input (which button pressed)
-Sound: An event that corresponds to a wav file in the directory
-Reset Buttons: An event that clears all functionality of the buttons.
-Go to Location: An event that lets you jump to other events
-Clear All: An event that clears the cells and button assignments

-Clear Cell: An event that clears the cells assignments
-Set Pins: An event that sets the pins of a cell at some defined position
-Set Char: An event that sets a character to a cell
-Raise Pin: An event that raises a pin of a cell
-Lower Pin: An event that lower a pin of a cell
-Set Voice: An event that sets what voice
-Location Tag: An event that holds the position of events

2.2.11 New Question

This feature lets the user create questions by being provided a field where they can input their question, a field where they can specify what the braille text on the device will be, an option to dictate which button corresponds to the right option and a field that lets the user know if the option they choose was incorrect. It is only active after the scenario is initialized.

2.3 Product Specification

-Hardware: A braille-based teaching device provided by the professor
    -Ability to play audio
    -Display braille cells
    -Contains a number of buttons
-Software: Graphic User Interface – Authoring Scenario Application
    -Actual GUI to be load
    -Allows user the ability to create and make scenarios for the hardware

2.4 Product Information

-Operating Environment
    -Java Runtime Environment 1.7 or above
-Design and Implementation Constraints
    -Software
        -Development language: Java
    -Hardware
        -Constraints provided through specifications provided by the professor
        -Limited amount of braille cells and buttons

3.Acceptance Test Cases

We feel that these acceptance tests cases are sufficient because they mainly dealt with GUI functionality, such as the Move Item Up or Move Item Down commands. There are only a very limited number of ways a user is permitted to use the GUI, and our tests covered all those situations that we can foresee a user encountering with reasonable operation of the Authoring App.

Case 1:

Test Case Scenario: Initialize with a New Scenario
Expected Behaviour for Success: Fields are properly filled out and the fields in the Settings Panel is filled as a result. Enables the rest of buttons to be able to start creating scenarios.
Expected Behaviour for Failure: One or more fields are missing. An error message will pop up informing the user that there is an empty field. Buttons will still be disabled.

Case 2:

Test Case Scenario: Import a scenario file
Expected Behaviour for Success: Successfully imports the file and displays its content in the scenario panel. Enables all the buttons that allow for customizing scenarios.
Expected Behaviour for Failure: No implementation has been made if the file does not follow the format of the scenarios. Tries to forcefully populate the scenario creating unintelligible events.

Case 3:

Test Case Scenario: Exporting a scenario file
Expected Behaviour for Success: Successfully exports and saves the current scenario in a text file in the computer's directory.
Expected Behaviour for Failure: Catches any errors that occurs if any events surpass the parameters defined when the scenario was initialized.

Case 4:

Test Case Scenario: Recording an Audio file
Expected Behaviour for Success: Properly records the audio that is transmitted through a microphone.
Expected Behaviour for Failure:

Case 5:

Test Case Scenario: Saving the recorded Audio file
Expected Behaviour for Success: Successfully saves the audio recorded as a wav file and stops anymore recording of the microphone. The user is then prompted to save it somewhere in their directory.
Expected Behaviour for Failure: If the user does not save the file then the file will be disregarded.

Case 6:

Test Case Scenario: Read an Audio File
Expected Behaviour for Success: Opens a path to the directory to find a suitable audio file. The file will be played through once and an event will be created with a reference to the directory from where the wav file currently is.
Expected Behaviour for Failure: The only expected failure is if the wav file doesn't contain any sound. In that scenario, the file would still create an event in the scenario, however since the file contains no sound, the scenario would simply jump to the next event.

Case 7:

Test Case Scenario: Moving an Item up
Expected Behaviour for Success: Successfully able to move items up by one, each time the button is pressed.
Expected Behaviour for Failure: If an event is at the top, the button no longer is enabled and the user cannot move the event any higher.

Case 8:

Test Case Scenario: Moving an Item down
Expected Behaviour for Success: Successfully able to move items down one at a time with each button press.
Expected Behaviour for Failure: If an event is at the bottom, the button no longer is enabled and the user cannot move the item any lower.

Case 9:

Test Case Scenario: Deleting an item
Expected Behaviour for Success: Able to remove the item from the scenario.
Expected Behaviour for Failure: Only expected failure that can happen if the user tries to delete an event that is not there, and in such a scenario, nothing happens because there is nothing to delete.

Case 10:

Test Case Scenario: Creating a new item
Expected Behaviour for Success: Successfully able to go through the drop-down menu and select an item. The item is then displayed at the bottom of the scenario.
Expected Behaviour for Failure: Only expected failure is if the user does not choose a scenario. In that scenario, nothing happens because there will be nothing to add.

Case 11:

Test Case Scenario: Creating a new question
Expected Behaviour for Success: Upon filling the fields, the scenario panel is auto-generated with the parameters that it was given and populates it.
Expected Behaviour for Failure: If there are potential fields missing, then the skeleton is still auto-generated but the parameter will be missing in the scenario panel, however since the panel contains selectable and editable events, the user may simply fill it in later.