TalkVox

Javier Thiele-Ruiz

Computer Engineering

June 2014

Project Advisor

John Oliver

Winter 2014 – Spring 2014

Table of Contents

Introduction 3

Project Overview 3

Need Statement 3

Project Definition 4

Project Goals 4

Project Objectives 4

Project Deliverables 4

Marketing Requirements 5

Engineering Requirements 5

User Experience 6

Project Planning and Design 7

Overview 7

Project Schedule 7

Software Design 8

Closing Statements 9

Sustainability Impact 9

Health and Safety Impact 9

Conclusion 10

Introduction

Project Overview

The problem I am trying to solve is the difficulty associated with editing the pronunciations of words for Text-to-Speech (TTS) programs in a fast and efficient manner. When using certain text-to-speech software, the pronunciation must first be converted to a different format, and then a command line argument is used to create a .WAV file with the pronunciation of that word. This is incredibly inefficient and time-consuming so I set out to develop a program which would be able to remove the majority of the tedium involved with testing how a given word is pronounced by TTS programs. With fewer steps required from the user, more progress is able to be made and more words are able to be edited.

Need Statement

TTS programs are built upon large dictionaries of words with the phonetic spelling of each word alongside them. When editing these pronunciations, many steps are required in order to change, test, and save the changes to the dictionary. This makes it extremely time-consuming to actually edit these dictionary files and results in either an incomplete TTS program or too many hours spent working on ensuring the pronunciations are correct. I took on this project as a way to be able to streamline the process of editing and saving these phonetic pronunciations.

Project Definition

Project Goals

- Develop a program which would remove the intermediate steps required to listen to a given word's pronunciation when read from a file.
- Reduce time required to edit a dictionary file of words.

Project Objectives

- Develop a User Interface using Java Swing.
- Use the Pico2Wav executable to play the phonetic pronunciations.
- Save and revert changes made to dictionary file.

Project Deliverables

Upon completion, the project will include a program which is capable of reading in a dictionary file with a list of words and their given pronunciations. The program will then allow the user to navigate the dictionary with ease while listening to and editing the words which are included. The source code, a sample dictionary, and all of the utilities required to run the program will also be included.

Marketing Requirements

- Load and save an existing dictionary file
- Display word and associated pronunciation
- Mark which phonemes are invalid in pronunciation field
- Display current word number as well as total word count
- Play button which plays current word's pronunciation
- Next and Previous buttons which update with new word's information and play its pronunciation
- Word/Phoneme search to navigate dictionary easier
- Change language TTS voice uses

Engineering Requirements

Category	**Engineering Requirement**	**Justification**
Functionality	The program must be able to support different phonetic mappings	In order to accommodate different phonetic alphabets, the ability to change the mappings is necessary
Functionality/Usability	The program must be able to play the updated word pronunciations instantly	If this is not done instantly then
Usability	The program must display which phonemes are invalid in the current pronunciation	Without this feature, incorrect pronunciations could be saved to the original file
Usability	The program should have a word/phoneme search	This allows for the quick navigation of large dictionary files

The User Experience

The primary user to use this would most likely be a developer trying to build up a better TTS dictionary for modifying existing TTS engines or creating their own. They would begin by loading a dictionary file containing a list of words whose pronunciations would need to be added or have their accuracies verified. Once the list is loaded they would then proceed to go through every word as quickly as possible to ensure each word's accuracy, or add pronunciations where needed. Once they reached the end, they would save the changes to the file and open a new file to repeat the process.

Another user could potentially be someone who is only looking to add certain words to TTS engines in order to fix how certain words sound, like acronyms or foreign names. They would directly contribute to helping a TTS engine improve its accuracy by crowdsourcing its resources.

Project Planning and Design

Overview

When I began this project I laid out a timeline with my advisor John Oliver which broke down every section of the project and when each one should be done. Included in the timeline were prototypes, basic features, extra features, debugging, and testing. Throughout the project there were also regularly scheduled meetings with Dr. Oliver to update him on my status along the program.

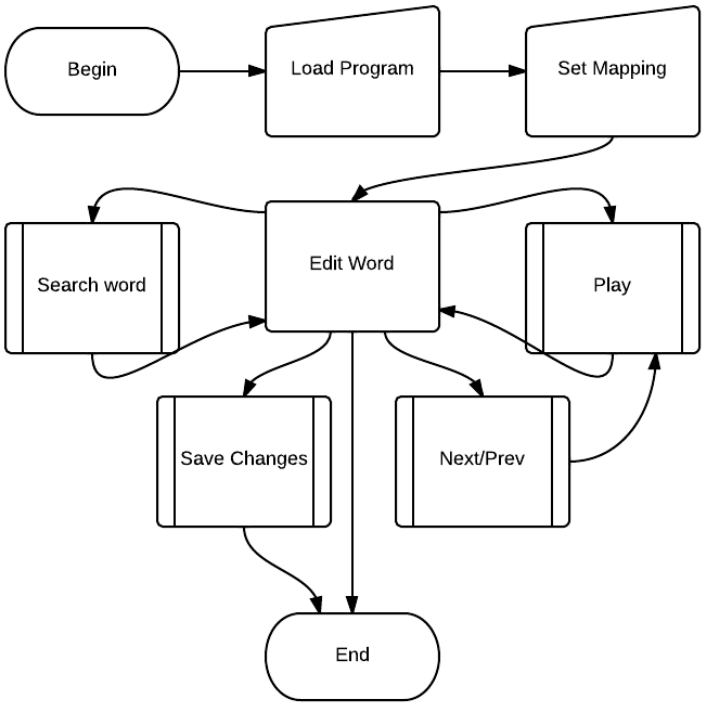
Project Schedule

The project will be broken down into several phases. The first phase is a prototype phase where the viability of the chosen TTS engine (Pico SVOX) is examined and a small, simple program is written with it to demonstrate it working. The next phase is one where the foundation of the program is built, with only extremely basic functionality present such as loading and parsing a dictionary file, translating the pronunciations so they can be read by Pico, and playing how they currently are. After that phase is complete, the next phase commences where the majority of the program is completed. This includes pronunciation editing and display, several visual indicators to mark where in the file the current word is located, and the ability to save the new pronunciations back to the file. The following phase includes the adding of several features such as a display of invalid phonemes for each word, a word and phoneme search, an option to choose which language to run the TTS engine in, and the ability to revert changes made to the file. The final phase is the testing phase where the viability of the program is tested among several different conditions and the functionality of each feature is tested.

Software Design

I initially decided on using Java Swing and Pico2Wav for this project. I decided on Swing because it is what I have the most experience in when building UIs, and because running system commands from Java programs (which is necessary in this program) is very simple and straightforward. Pico2Wav was chosen as a result of its very clear mapping of IPA to X-Sampa, two different phonetic notations, and its ability to play words based on their phonetic spelling, rather than their actual spelling. While I had some difficulty getting the Pico2Wav command to run from within the program initially and debated using a different language to overcome that issue, I eventually realized my problem and was able to fix it without having to start over in a different language.

The diagram (right) illustrates the user-software interaction by outlining the basic flowchart the user would take when initally starting the program. It follows a straightforward path designed for simplicity and immediate usabilty.



Closing Statements

Sustainability Impact

Because this project is 100% software, the sustainability impacts are very minimal as there is very little hardware to worry about other than the computer running the program. With that said, it should still be mentioned that as a result of the reduced time it now takes to alter the dictionary files, it is possible the computer will remain on for less time thereby reducing overall energy consumption for those using this program.

Health and Safety Impact

This project has a very minimal impact on health and safety because of the fact that it is purely software. It is possible however that due to the decreased time required to edit these files, overeager users could have so much fun editing pronunciations they neglect their physical health.

Conclusion

The final implementation at the end of the project time did meet all of the initial expectations which were set.

The following requirements were met:

- Load and save an existing dictionary file
- Display word and associated pronunciation
- Mark which phonemes are invalid in pronunciation field

- Display current word number as well as total word count
- Play button which plays current word's pronunciation
- Next and Previous buttons which update with new word's information and play its pronunciation
- Word/Phoneme search to navigate dictionary easier
- Change language TTS voice uses

The progress made on this program was the result of a familiarity with the programming aspects involved with this project as well as a solid conceptual understanding of utilizing the Pico SVOX TTS engine. Overall this project was a success as all of the requirements which were initially declared were met, and the final product is a stable working program.