Engine Instructions

# General Formatting

The engine includes a few folders and files that need to be included on the server. These include:

## Folders

|  |  |
| --- | --- |
| **Folder Name** | **Folder Location (currently hard coded)** |
| CapstoneCode | C:\Users\Taylor\Desktop\CapstoneCode |
| AllVoiceSamples | C:\Users\Taylor\Desktop\AllVoiceSamples |
| Downloads | C:\Users\Taylor\Downloads |

## Files

|  |  |
| --- | --- |
| **File Name** | **File Location** |
| input.txt | C:\Users\Taylor\Desktop\CapstoneCode\input.txt |
| output.txt | C:\Users\Taylor\Desktop\CapstoneCode\output.txt |
| InputOutputScriptPresentation.py | C:\Users\Taylor\Desktop\CapstoneCode\InputOutputScriptPresentation.py |
| InputOutputScriptAutomation.py | C:\Users\Taylor\Desktop\CapstoneCode\InputOutputScriptAutomation.py |
| blahblahblah.bat | C:\Users\Taylor\Desktop\CapstoneCode\blahblahblah.bat |
| SpeechRecognitionWorking1.exe | C:\Users\Taylor\Desktop\CapstoneCode\SpeechRecognitionWorking1.exe |
| word.wav | C:\Users\Taylor\Desktop\AllVoiceSamples\word.wav |
| MediaMenuGrammar.grxml | C:\Users\Taylor\Desktop\MediaMenuGrammar.grxml |
| phonemes.pls | C:\Users\Taylor\Desktop\phonemes.pls |

# File Coordination/Purposes

The engine that I customized is named SpeechRecognitionWorking1.exe. It takes a file name out of the input.txt file, looks for a file matching that name in the AllVoiceSamples folder, runs the speech recognition against that file, and then puts the word it recognized it as in the output.txt file. If the file isn’t recognized, then nothing is put into the output.txt file currently. The engine also gets the expected grammar words (the 42 words) and their associated phonemes from the MediaMenuGrammar.grxml file.

There are two instances of the MediaMenuGrammar.grxml file. There is one that only has one set of custom grammars per word (the “perfect” word phonemes that were emailed to us), and another where I added the standard words that the Microsoft Speech Recognition software interprets with its default phonemes along with a number of custom phonemes that I attempted to add to them. The second one is the beginning attempts to try to recognize as close to 100% of how anyone would say the words as possible. I have made two copies of the same file with a different naming system to make it clear which one represents what group of phonemes. The file that will be used needs to be titled “MediaMenuGrammar.grxml”.

As a side note, I added phonemes.pls to the list of files required. I don’t think this file is being used any more, but was being referenced through the MediaMenuGrammar.grxml file for a time. I kept the code there for future use if it becomes helpful in some way, and doesn’t seem to effect the program from running correctly if the .pls file isn’t present.

The blahblahblah.bat file is what the webpage is currently set to run when you tap on the “Run Script” link. The .bat file is set to run the InputOutputScript.py file after the user chooses to download the .wav file from the server.

The python script will search through the Downloads folder until it finds a .wav file, move it to the AllVoiceSamples folder (renaming it to word.wav), and then run SpeechRecognitionWorking1.exe against the word.wav file it creates. The original file in the Downloads folder must be a .wav file under the current coding, but that can be changed to accept any file type and then rename it to be a word.wav file if desired.

I used Microsoft Visual Studios 2012. The folder entitled “SpeechRecognitionWorking1” includes the Visual Studios project that was used for the presentation with the phonemes. This is the file we used in our final Capstone presentation. The folder entitled “WindowsSpeechVoicePresentation” referring to the presentation at the end of the first semester with live voices. This one is just an example code of how you would set it up to recognize live voice as the word given, and doesn’t include custom phonemes.