cs171 Introduction to Computer Science
Sounds Assignment

1. Suppose we wish to take a sound recording of something someone has said, and change it into some-thing that they *did not* say. One of the simplest variants of this idea is to logically remove one or more words from a digital audio recording. That is our goal in this problem.

    Recall that the data that comprises a digital sound can be represented as an array of positive and negative integers. The values of the integers represent the amplitude of each of the digital audio samples, and the samples may be "played back" at a fixed rate to reproduce the sound.

    Suppose the original sound, represented as an integer array, ranges from index 0 to index $n - 1$ (i.e. the sound has n samples). Further suppose that we wish to remove a single word that begins at index j and ends at index k. We would want to create a new integer array that has samples 0 through $j - 1$ from the original sound (call this the prefix) immediately followed by samples $k + 1$ through $n - 1$ (call this the suffix).

    Your sound editing application must retrieve three inputs from the user: (i) the pathname of the origi-nal sound, (ii) the start index (j) of the portion to be removed, and (iii) the end index (k) of the portion to be removed. It should create a MySound object based on the original using the given pathname and determine the number of samples in the original. It should then create a new sound object of the appropriate size and copy the prefix and suffix samples into the newly created sound and finally to play the sound. You should save your creation to a `.wav` file.
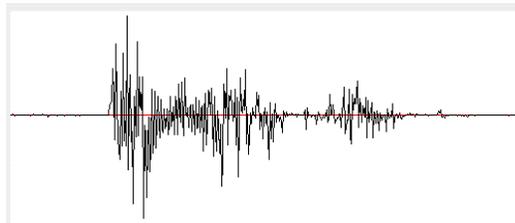

    Figure 1: Original Audio

    The sound file you are to use for this problem is in the Media directory within the Resources directory of the Share volume, and its name is `prob1.wav`. The word you are to remove is **not**. You may use the SoundExplorer to find the starting and ending indices of the word to remove. Figures 1 and 2 show a picture capture of the original audio and one with the word "not" removed.
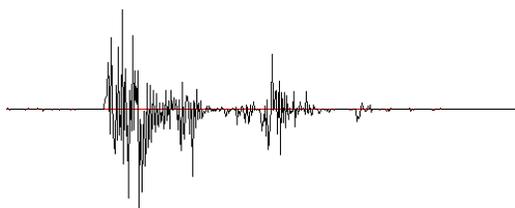

    Figure 2: Audio with"not" Removed

2. In this problem, we are going to use the MySound class to create a new MySound object from an existing Sound object. The idea is to create an "echo" sound from the original. The echo sound will repeat the original a fixed number of times, with each repetition at a diminished amplitude of the original.

For an initial version of the Echo application, we will create a new sound with a total of 4 repetitions of the original. The repetitions will have amplitude scaling of 100%, 75%, 50%, and 25% of the original.

(a) Obtain the path for the original base sound and create a Sound object from the file.

(b) Determine the number of samples in the original sound, call it n.

(c) Create a new sound capable of holding $4n$ samples.

(d) Copy the original from indices $[0..n-1]$ to the destination beginning at index 0 of the destination.

(e) Copy samples of the original scaled at 75% from indices $[0..n-1]$ in the original to the destination starting at index n.

(f) Copy samples of the original scaled at 50% from indices $[0..n-1]$ in the original to the destination starting at index $2n$.

(g) Copy samples of the original scaled at 25% from indices $[0..n-1]$ in the original to the destination starting at index $3n$.

Figures 3 and 4 demonstrates the use of this program on a simple square wave audio.
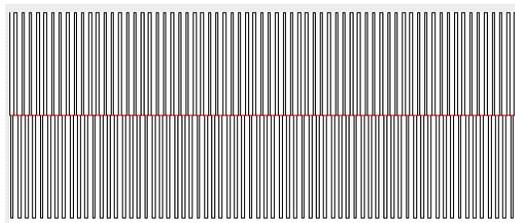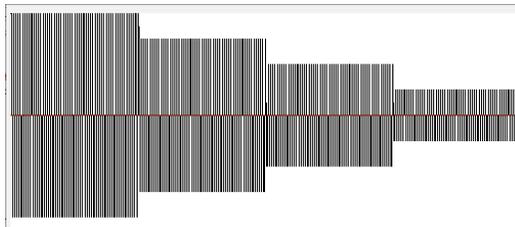

Figure 3: Original Square Wave Audio

Figure 4: Echo Square Wave Audio

Helpful APIs

FileMgr class

- static String FileMgr.selectExistingFilename ( )

- static String FileMgr.selectNewFilename ( )

- static String FileMgr.setCurrDirectory (String s)

MySound Class

- MySound (String path)

- MySound (int numSamples)

- MySound (int [] samples)

- void play ( )

- void blockPlay ( )

- int getNumSamples ( )

- int getSample (int index)

- int [ ] getSamples ( )

- void setSample (int index, int sample)

- void setSamples (int [ ] values)

- void writeFile (String path)

SoundExplorer Class

- SoundExplorer (Sound s)