

Music Challenge 5: A general music program**Due:** April 9, 2008

The goal of this assignment is develop a music program so that it can play any song given a separate input file. The input file can then be changed so that your program plays a new song. This will build on concepts used in all of the previous challenges and will use the `note2freq` subroutine you just developed. This is to be done as a group project.

The inputs to this program should be (1) the file name of the file where the song data is located and (2) the sampling frequency desired for the resulting song. The program needs to read in this user specified file, convert the note names to frequencies, create notes based on the information read in and play these notes. There will be bonus points for being able to save a wav-file of the song created.

Input File: Your program will need to read a tab-delimited ASCII file that has columns for (1) the note, (2) the amplitude (0-1) and (3) the duration. The note should be written as a capital letter for the note, a number for the octave, and a sharp(#) or flat(b) if necessary. Your program should be capable of accepting notes from octaves 3, 4 & 5. Example files are on the Music in MATLAB page.

To read these files you will need to use MATLAB's "textread" function. For example, your read statement might be:

```
[note,amp,dur] = textread('filename', '%s %f %f')
```

this will create three column vectors: one a cell array with the text for the notes, the second with the amplitude, and the third with the duration. Be careful to notice the literals (single quotes) in this statement.

Converting Note Names to Frequencies: Use the subfunction you created last time to convert the notes to frequencies. The note vector created by the `textread` command will be a cell array. To address the cells in a cell array you should use braces { } instead of parenthesis (). To address the k^{th} note you would use:

```
n = note{k}
```

Then you can call your subroutine, to convert the k^{th} note to a frequency:

```
freq = note2freq(n)
```

Remember your subroutine was not designed to handle vectors so you must convert one note at a time.

Deliverables (to be emailed to moor@engr.ipfw.edu):

(1) A separate user introduction. This document should explain to a potential user what the purpose of your program is, how to use the program including a description of all inputs and outputs. This introduction must describe the nature of the text file required as input. This document should also describe your program logic. This can be done using a flowchart, a detailed list of steps or a detailed narrative description.

(2) An electronic copy of the program (.m file) and a `note2freq` subfunction (also as a .m file). These programs should follow usual conventions for commenting.

If you have created a new song data file in developing this program please send that file to me as well.

Music Challenge 5 Ruberic:

| | |
|--|--|
| | <i>Criteria (one point each) – numbers indicate relation to the steps in table 1.7-2</i> |
| 1 | Problem clearly presented and follows the instructions for submittal including submittal of 1. a separate user introduction, 2. a note2freq subfunction m-file and 3. a general song player file. |
| Separate “User Introduction”/problem setup covering: | |
| 2 | Goal of program presented (1) |
| 3 | Instructions for use of program including all required files, the form of the main program call including appropriate format information on inputs and outputs (2 & 3) |
| 4 | Program Logic: The logical steps the program uses are presented in detail as a list or a flowchart. This presentation demonstrates clear understanding of how the program accomplishes its goal. (4) |
| Program comments: | |
| 5 | Clear introductory comments that form a useful help file (try >> help ‘filename’ to check) (5) |
| 6 | Comments identifying <u>all</u> variables and their units (5) |
| Program logic and performance: | |
| 7 | .m file electronic included with essentially correct logic that can play any file (5) |
| 8 | Electronic (soft) copy provided runs (5) |
| 9 | Program (soft copy) produces desired output (6) |
| 10 | Output (soft copy) completely fulfills the goal of the program. Including no unexpected or extraneous output not described in the user introduction. |
| 11 | Bonus: included text file represents a pleasant song of at least 35 notes (a completely random or arbitrary list of notes does not qualify) |
| 12 | Bonus: the played song is saved to a single .wav file |
| 13 | |