Name:
Section:
Date Assigned:
Date Submitted:

EXPERIMENT 4
DIGITAL MUSIC SYNTHESIS

OBJECTIVE

Explore how mathematical operations will yield synthesized audio tones.

MATERIALS AND EQUIPMENT

Computer installed with Matlab R2018 or Octave 5.2.0

INTRODUCTION

In this experiment, we explore how to use simple tones to compose a segment of music. Each musical note can be represented by a sinusoid whose frequency depends on the note pitch.

DISCUSSION

Frequencies in Music

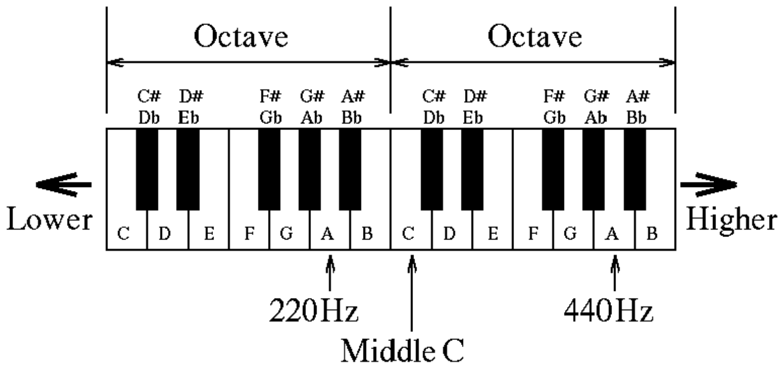
Musical notes are arranged in groups of twelve, called octaves. The twelve notes in each octave are logarithmically spaced in frequency, with each note frequency being $2^{1/12}$ times the frequency of the next lowest note.

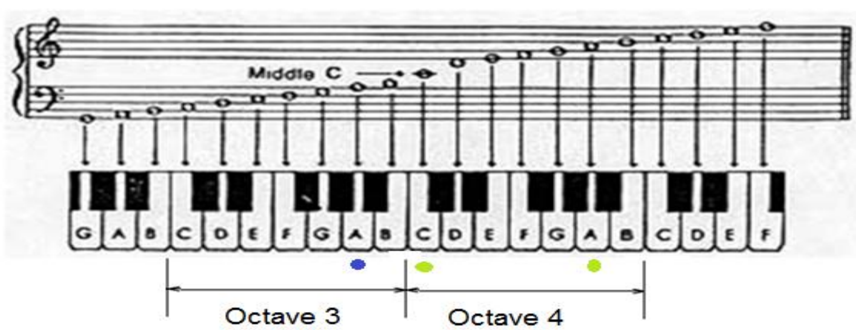
Notes in the 220-440Hz octave

Note	Frequency (Hz)
A	220
$A^{\#}$, B^{b}	$220*2^{1/12}$
B	$220*2^{2/12}$
C	$220*2^{3/12}$
$C^{\#}$, D^{b}	
D	.
$D^{\#}$, E^{b}	.
E	.
F	.
$F^{\#}$, G^{b}	
G	
$G^{\#}$, A^{b}	$220*2^{11/12}$

Table 1

The musical notation and the frequencies for each note are shown below.





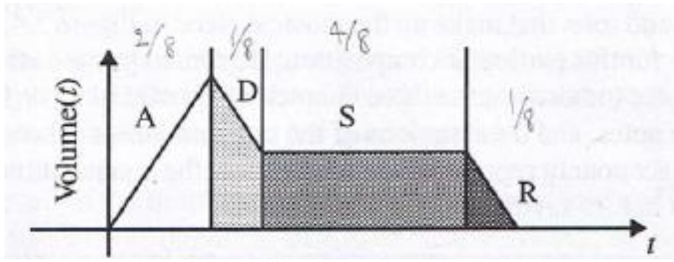
FREQUENCY (Hz)							
OCTAVE							
NOTE	1	2	3	4	5	6	7
C	32.703	65.406	130.813	261.626	523.251	1046.502	2093.005
C#/Db	34.648	69.296	138.591	277.183	554.365	1108.731	2217.461
D	36.708	73.416	146.832	293.665	587.330	1174.659	2349.318
D#/Eb	38.891	77.782	155.563	311.127	622.254	1244.508	2489.016
E	41.203	82.407	164.814	329.628	659.255	1318.510	2637.020
F	43.654	87.307	174.614	349.228	698.456	1396.913	2793.826
F#/Gb	46.249	92.499	184.997	369.994	739.989	1479.978	2959.955
G	48.999	97.999	195.998	391.995	783.991	1567.982	3135.963
G#/Ab	51.913	103.826	207.652	415.305	830.609	1661.219	3322.438
A	55.000	110.000	220.000	440.000	880.000	1760.000	3520.000
A#/Bb	58.270	116.541	233.082	466.164	932.328	1864.655	3729.310
B	61.735	123.471	246.942	493.883	987.767	1975.533	3951.066

Note Durations

Each note may be represented by a burst of a sinusoidal tone. The duration of each tone burst is determined by whether the note is a whole note, half note, quarter note, eighth note, etc. For example, a quarter note lasts one beat. For your experiment, 4000 samples will represent 1 beat, and should be about 0.5 second.

Improving Perceived Quality by Volume variations

Typically, when a note is played, the volume rises quickly from zero and then decays over time, depending on how hard the key is struck and how long it is depressed. The variation of the volume over time is divided into four segments: Attack, Decay, Sustain, and Release (ADSR). For a given tone, volume changes can be achieved by multiplying a sinusoid by another function called a windowing function. Shown below is a sample ADSR Envelope.



Sample ADSR Envelope

For your experiment:

Using Octave, construct a row vector that is a discrete-time representation of the notes that make up the musical piece that will be given to you. Let the sampling rate be 8 kHz for this particular composition. Try to improve the audio quality by having the volume for each note decay over time to make your music sound more interesting..