CS 1120: Media Computation Spring 2018
Lab 8: Introduction to digital sound

Activity A: Connecting the past to the future

To begin exploring sound I want you to look backwards at the unit you just finished: digital images.

[Q1] Think back over everything we did with image manipulations. List as many manipulations that you learned/studied as you can remember.

Looking back at the list you made for Q1, think about how you could organize those manipulations into groups of similar actions. I wouldn't necessarily use the structure of the book as your guideline for this because the book organized manipulations based around the things you needed to know about the code in those manipulations more than on what the manipulations actually did. This question wants you to think about what the manipulations DO to the pictures and come up with categories of similar actions.

[Q2] Identify 3-5 categories that classify the manipulations in your answer for Q1 into groups of "similar actions." List those categories, and the manipulations that go in each category.

You read chapter seven for this week which talked about how sound is encoded. Brainstorm as many manipulations as you can come up with that you think could be performed on sound files. Don't worry about whether or not they can actually be done at this point, but think about things that seem like you should be able to do to sounds.

[Q3] List as many sound manipulations as you can come up with.

Activity B: Introduction to the Explorer Tool

When we first started working with pictures, a significant part of our first lab was spent exploring image files and how they are represented in the computer. We will begin this unit in a very similar manner. Download the mediasources directory (available on the resources page). This directory contains a variety of sound files distributed by the authors of your textbook. Let's begin by exploring three of these. Down Launch JES, and set the media path to point to the mediasources directory. Once that is complete, load three sounds using the following commands:

```
>>> s1 = makeSound('c4.wav')
>>> s2 = makeSound('e4.wav')
>>> s3 = makeSound('g4.wav')
```

Notice that this is very similar to loading images where you first set the media path and then used makePicture() to load an existing picture file.

To play a sound in JES you need to use the play() command:

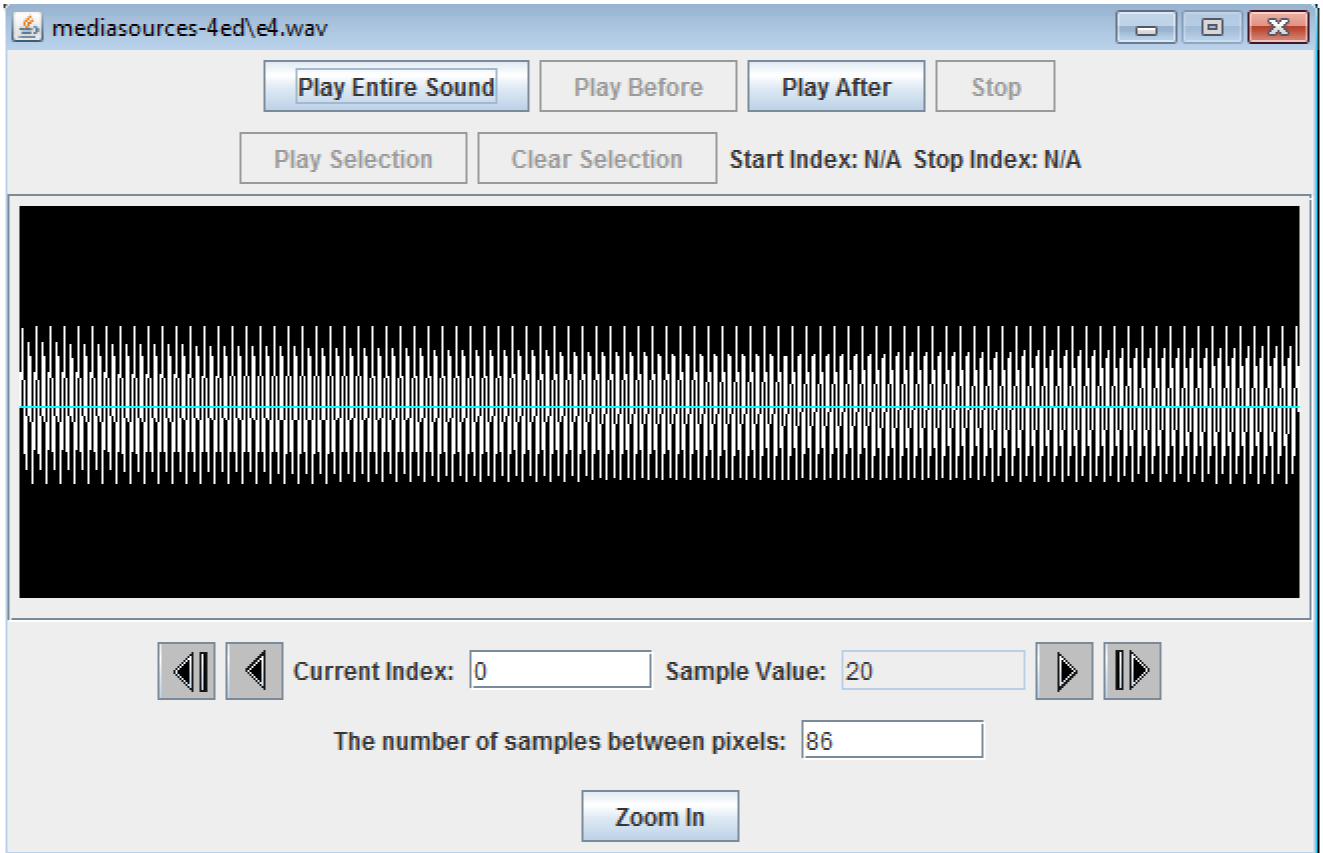
```
>>> play(s1)
```

Again, this is very similar to the show() command. The show() command let you see a picture while the play() command lets you hear the sound. If your speakers are on and set to a reasonable volume you should hear a very simple tone that is playing at the frequency of middle C (this is a computer generated sound/tone known as a "pure" tone). This tone will play for 2.5 seconds. Once the tone has finished playing, repeat the above command with s2 and s3 respectively. These are the E and G notes that come above middle C.

In order to understand these sounds better it is helpful if we can explore them so that we can both hear them and “see” them so we know what is actually happening in a sound file on the computer. To do this, open s2 in the sound explore tool using the command:

```
>>> explore(s2)
```

When you do you get a tool that should look like this:



This probably doesn’t look anything like you might have expected that sound to look. Play it again (You can do this by typing play(s2) or by pressing the “Play Entire Sound” button in this explore window.). That is a nice gentle sound and this image is all rough and sharp looking. Not what we expected.

Well, that is because what you are viewing is a compressed representation of the entire sound. It has been “squished” together so that it fits in that window all at once. If you were paying attention when doing the reading in the chapter you learned that sounds are stored digitally with some number of samples per second. You can find the sampling rate of this particular sound by typing this command:

```
>>> getSamplingRate(s2)
```

[Q4] What is the sampling rate of sound 2?

This means that there are literally thousands of samples that make up the sound. In fact, we can find out how many sound samples there are by typing:

```
>>> getLength(s2)
```

[Q5] How many samples are there in sound 2?

We can use this information to determine how long a sound is when we play it. To do this you have to divide the length of the sound by its sampling rate.

[Q6] Divides the length of s2 (the answer to Q5) by the sampling rate of s2 (the answer to Q4). What do you get? [If you don't get an answer of 2.5 seconds you did something wrong].

So what does this have to do with why the image in the explore window looks so funny? Well, JES is using a representation in the explore window where they show one sound sample per pixel of the width of the image. Since they are trying to make over 55,000 sound samples fit into a window that is only about 650 pixels wide, they clearly can't show you every sound sample. Instead they show you a small number of samples.

To help make this point, click your cursor in the black area immediately above or below the sound wave in the explorer window. If you do this correctly a vertical cyan line should appear.

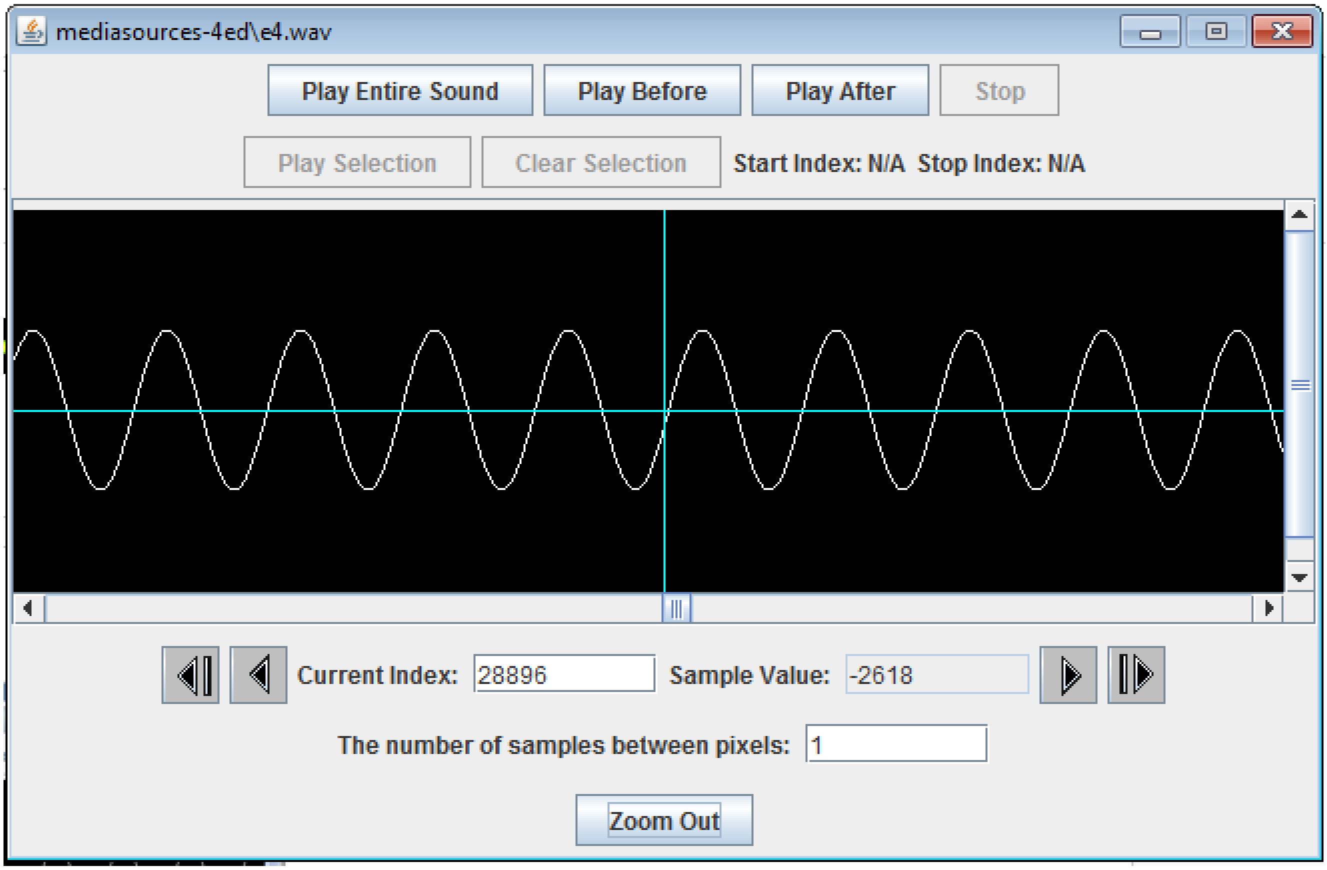
[Q7] What number appears in the box labeled "Current Index" immediately below the sound wave?

Now press the left arrow button located just to the left of the "Current Index" box.

[Q8] Now what number appears in the "Current Index" box?

Notice that these numbers are not consecutive values. In fact, they should be approximately 86 samples apart (it may be possible to have a value one or more less than this value depending on where you select). This is because in order to fit 55,000 samples in only 650 pixels the computer needs to look at approximately every 86th sound sample.

But it is easy to ask the computer to show you the entire sound representation. Notice that at the very bottom of this window there is a button that says "Zoom In" Clicking this button allows us to see every individual sound sample



This is a lot more like what you might expect this sound to "look" like. It's a very gentle, consistent sound wave.

Now let's repeat what we just did. The vertical cyan line bar should still be on the screen. If not, click the cursor in the middle of the black window again.

[Q9] What is the value listed in the Current Index box?

Press the left arrow button again.

[Q10] Now what value is listed in the Current Index box?

This time you should observe that their difference is only one index value. Of course we are only looking at one small portion of the sound wave at any time. We can't possibly fit all 55,000 sound samples on the screen at once. Notice that the scroll bar underneath this window has become active. You can drag the bar all the way to the left or the right to see if and how the sound changes over time.

Several times in this lab we've made connections between what you did with pictures and what you are now doing with sounds. In the last couple of activities we have been concentrating on the value in the "Current Index" box. This value is the closest equivalent of the (x,y) location of a pixel in an image. It indicates WHERE in the sound a particular sample exists. We haven't actually yet looked at the values of these samples. This information is located just to the right of the Current Index and is given as the Sample Value.

Use the explorer tool to navigate around inside of sound sample s2 (which should be e4.wav).

[Q11] What is the value of the sample at index #10?

[Q12] What is the value of the sample at index #110?

[Q13] What is the largest value that you can find? Which sample # is it located at?

Activity C: Using the Explorer Tool to consider a sound's "sound"

In chapter seven you've read that sound is made by waves of air particles and that a digital recording of a sound is an attempt to represent that wave in a long sequence of numbers. How a sound wave actually "sounds" is based on the shape of that wave. There are two main attributes of waves that affect how they "sound" to us. These attributes are frequency and amplitude.

Make sure that you have the c4, e4, and g4 loaded into JES.

Play each of these sounds and listen carefully.

Compare c4 to e4.

[Q14] Is one higher or lower in pitch or are they basically the same? Which one is higher in pitch?

[Q15] Is one louder than the other or are they are basically the same volume?

Compare e4 to g4.

[Q16] Is one higher or lower in pitch or are they basically the same? Which one is higher in pitch?

[Q17] Is one louder than the other or are they are basically the same volume?

A sound's pitch is related to its frequency. Frequency in sound is defined as the number of waves per second. While it would be difficult and time consuming to count the number of waves in these sounds, it is relatively easy to calculate them. By measuring the distance between two wave peaks (distance measured using number of sound samples not inches or centimeters) we can calculate the number of sound samples per wave. By using the knowledge that there are 22050 samples per second we can divide sampling rate by samples per wave to arrive at waves per second.

Let's explore the frequency of these three sounds. Use the explorer tool to complete the following table. **Make sure that you zoom in** and are looking at the whole wave before completing this activity.

[Q18] Use the explorer tool to complete the first two columns in the table below. Then use math to complete the second two columns

Sound	Index # of the first peak	Index # of the second peak	Distance between the two peaks = col3 – col2	Frequency 22050 / col4
C4.wav				

E4.wav				
G4.wav				

If you want to check the accuracy of your calculations you can check the table located at http://en.wikipedia.org/wiki/Piano_key_frequencies to see what frequencies you should have calculated. If you are off by 1 or 2 that is due to basic sampling error and you shouldn't worry. If you are off by more than that then you have done something wrong in completing the question.

[Q19] Rank the three tones from lowest pitch to highest pitch.

[Q20] Rank the three tones from lowest frequency to highest frequency.

[Q21] What can you conclude about the relationship between pitch and frequency?

A sound's volume is related to its amplitude. Amplitude is defined as the "height" of the wave from its lowest point to its highest point. Let's explore the amplitude of these three sounds. Use the explorer tool to complete the following table. **Make sure that you zoom in** and are looking at the whole wave before completing this activity.

[Q22] Use the explorer tool to complete the first four columns in the table below. Then use math to complete the final column.

Sound	Index # of the first peak	Sample Value of that peak	Index # of the first valley	Sample Value of that valley	Amplitude col3 – col5
C4.wav					
E4.wav					
G4.wav					

You should observe that these three sounds all have nearly identical amplitude. This explains why they appeared to have nearly identical volume.

Before we move on, it's probably a good idea to look at one or two sounds that aren't "pure" sounds.

Open the following sound file that is in your mediasources directory. This is a middle c but played on a bassoon rather than as a pure tone:

```
>>> s4 = makeSound('bassoon-c4.wav')
>>> explore(s4)
```

Make sure you zoom in to look at the full sound file. Drag the scrollbar out to the middle of the sound somewhere.

[Q23] Describe the general shape of this wave. How is it different from c4.wav?

[Q24] Press the "Play Entire Sound" button. What does it sound like?

[Q25] If you were going to try to measure the frequency of this sound, how would you go about doing it?

[Q26] If you were going to try to measure the amplitude of this sound, how would you go about doing it?

Finally, load one additional sound from the mediasources directory:

```
>>> s5 = makeSound('thisisatest.wav')
>>> explore(s5)
```

[Q27] Describe the general shape of this wave. How is it different from the previous sounds we have looked at?

[Q28] Press the "Play Entire Sound" button. What does it sound like?

[Q29] I normally have you zoom in to see a wave, this time make sure you are zoomed out. Describe how the shape of the wave matches what you heard in Q28.

[Q30] If you were going to try to measure the frequency of this sound, how would you go about doing it?

[Q31] If you were going to try to measure the amplitude of this sound, how would you go about doing it?

Activity D: Writing code to modify a sound

It's time to actually write some code and see what happens to a sound when you attempt some basic modifications. To begin with, copy Program 93 into the Program Area of JES and save this as lab8.py. Once you have this typed in correctly load c4, play it at the original volume, run this program, and then replay the sound:

```
>>> s1 = makeSound('c4.wav')
>>> play(s1)
>>> increaseVolume(s1)
>>> play(s1)
```

[Q32] What do you notice when you complete these actions?

[Q33] Open s1 in the explorer window and zoom in again. What do you notice about the shape of the sound wave?

How does this work? The write up in your textbook is pretty good at explaining this, so I won't repeat the full discussion here. But the simple answer is that this code takes every value, multiplies it by 2 – making it twice as big – and then saves it with that new value. The end result is that we have a sound whose positive sound samples are now MORE positive, and whose negative sound samples are MORE negative. The result is a wave with a larger amplitude.

In fact, it's worth looking at that specific issue. In addition to s1 that you just increased in volume, reload s2 (e4) and s3 (g4) and run them through the increaseVolume() program. One at a time open each sound in the sound explorer and repeat what you did in Q22 with these new sounds.

[Q34] Use the explorer tool to complete the first four columns in the table below. Then use math to complete the final column.

Sound	Index # of the first peak	Sample Value of that peak	Index # of the first valley	Sample Value of that valley	Amplitude col3 – col5
C4.wav					
E4.wav					
G4.wav					

[Q35] What do you notice about the values in the final column in Q22 compared to those in the final column in Q34?

Add the decreaseVolume() program (Program 94) to lab8.py. Run each of the three sounds through decreaseVolume(). Finally, open each in the sound explorer and consider the shape of the resulting sound waves.

[Q36] What do you observe when you look at these new sound waves?

[Q37] Compare Program 94 with Program 34 from chapter 4. In what ways are these programs similar?

Actually, if you think about Q35 and the questions I asked you at the very beginning of this lab you should be able to make comparisons between what you have done with pictures as you move forward into sounds. MANY of the things you will do with sounds are nearly identical to things you did with pictures. In some ways they are actually easier because while pictures were two dimensional, sounds are mostly one dimensional (we only can look at a linear progression of sound samples). Furthermore, while pictures had three values to consider (red, green, and blue) sounds only have the single sound sample to work with.

Activity F: Finishing up

[Q38] Copy Program 96 into lab8.py. What happens when you run it with a factor of 2? What happens when you run it with a factor of 0.5?

[Q39] Copy Program 97 into lab8.py. What happens when you run it on c4.wav? Use the explorer to explore the resulting sound to help explain your answers.

[Q40] Copy Program 98 into lab8.py. What happens when you run it on c4.wav? What happens when you run it on thisisatest.wav? Use the explorer to explore the resulting sounds to help explain your answers.

Submit your work

And you're done! You must submit two things to eLearning:

1) Your file lab8.py with the following functions:

- increaseVolume()
- addToSamples ()
- changeVolume ()
- normalize()
- onlyMaximize()

2) Your answers to the questions in this lab using the response word document (you may use as many lines per question as you like).