CSE 8A Lecture 13

- Reading for next class: 8.4-8.5
- Today’s topics:
  - Sounds!
- Finish PSA 6: Chromakey! DUE TUESDAY
- Interm exam 3 next Friday
CSE 8a Exam #3 Study Hints

1) **Reading Quizzes** (2/4-2/15)
   - Omit 2/6/13 #2

2) **Peer Instruction** questions

3) **Includes**: methods, loops, if, if else, Logical operators, mirroring, sound
   - Lecture7 (slide 14) through Lecture13 (slide 10)

4) **Discussion section**

5) **End of chapter problems**
   - Ch 5: 5.1-5.8     Ch 6: 6.2,6.8-6.9,6.17-6.19     Ch 8: 8.6-8.15

6) **Terms** from above
for (int x = 0; x < getWidth(); x++)
    for (int y = 0; y < getHeight() / 2; y++)
    {
        Pixel p = getPixel(x, y);

        if(x > getWidth() / 2)
            if(p.getRed() == 255 && p.getBlue() == 0 && p.getGreen() == 0)
                p.setColor(Color.BLACK);
    }

A. This code changes any pure red pixels in the top-right quarter of the picture to black.

B. This code modifies the top-right quarter of the picture.

C. This code loops over pixels in the top half of the array and, if the x coordinate is greater than half the width, checks if the red component is 255 and the blue is 0 and the green is 0, and if so, sets the pixel to black.
What are we doing next?

• Chapters 8,9,10: Sound!

• Continue with computational basics from Picture work:
  – Iteration/looping, if statements, arrays
  – Emphasis on deeper understanding
  – Emphasis on Java terminology, features, mental “model” of how code is represented in the execution on the machine

• When working with Sound/SoundSample, compare and contrast to Picture/Pixel examples.
Chapter 8: Sound!
Sounds

• Sound is a quasiperiodic pattern of waves of air pressure
  – Increase in air pressure is a compression
  – Decrease in air pressure is a rarefaction

• Strength of compression and rarefaction is the amplitude of the sound

• Number of compression/rarefaction cycles per second is the frequency of the sound
Digitizing Sounds

• To process sounds in a computer, sound must be digitized

• A microphone converts instantaneous sound pressure level into voltage
  (+ voltage for compression, - voltage for rarefaction)

• Then, amplitude of voltage can be converted to bits (digital integers)
  with an Analog-to-Digital converter (ADC)
Digital Sample Rate versus Sample Size

• Size determines max (and min) amplitude
  – CD audio: 16 bits per sample (per stereo channel)
  – Min: -32,768; Max: 32,767

• Rate is “how often we record an amplitude”
  – CD audio: 44,100 samples per second
  – A.k.a. 44.1 KHz sample rate
If the following sound were modified to be louder it would

A) Have **lower frequency** and stronger compressions/rarefactions
B) Have **smaller amplitude** and stronger compressions/rarefactions
C) Have **higher frequency** and stronger compressions/rarefactions
D) Have **larger amplitude** and stronger compressions/rarefactions
E) None of the above
1) Solo: (30 sec)
2) Discuss/Group: (2 min)
Making bad music: What’s wrong with these decisions?

- I decided, I don’t like the sampling rate and sample size provided by the book authors. Comment on my decision to use...

<table>
<thead>
<tr>
<th>Sampling Rate</th>
<th>Sample Size</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,000Hz</td>
<td>4 bits</td>
<td></td>
</tr>
<tr>
<td>100,000Hz</td>
<td>32 bits</td>
<td></td>
</tr>
</tbody>
</table>
String fileName = FileChooser.pickAFile();
Sound noise = new Sound(fileName);
SoundSample[] noiseArray = noise.getSamples();
noiseArray[3].setValue(0);

int foo = noiseArray[2].getValue();
```java
String fileName = FileChooser.pickAFile();
Sound noise = new Sound(fileName);
SoundSample[] noiseArray = noise.getSamples();
noiseArray[3].setValue(0);

int foo = noiseArray[2].getValue();
```
How would we fill in this SampleSound[]

1) Solo: (60 sec)
2) Discuss:/Group(2 min)
How would we fill in this SampleSound[]

1) Solo: (60 sec)
2) Discuss/Group (2 min)
According to Nyquist’s Theorem what is the minimum sampling rate?

A. 1.5 Hz
B. 3 Hz
C. 6 Hz
D. 10,000 Hz
E. 20,000 Hz
Write code which makes the following changes

• Here’s a Sound

```java
String fileName = FileChooser.pickAFile();
Sound noise = new Sound(fileName);
SoundSample[] noiseArray = noise.getSamples();

<<<< PICK SOME CODE >>>

for (SoundSample sample : noiseArray)
{
    int foo = sample.getValue();
    sample.setValue(foo/2);
}

for (int i = 0; i < noiseArray.length; i++)
{
    SoundSample sample = noiseArray[i];
    int foo = sample.getValue();
    sample.setValue(foo/2);
}
```
What does that code do

A. Makes a **lower pitched** sound during **first half** of play

B. Makes a **quieter sound** during **first half** of play

C. Makes a **lower pitched** sound during **second half** of play

D. Makes a **quieter sound** during **second half** of play

E. For each SoundSample element in **second half** of array it gets the Value and stores that into an int and then sets the Value with something that is half that
For you to practice

• Write code that reduces the volume of every other SoundSample.
  – What does that really sound like?
Summary of Concepts

- Digital representations of sounds
- Manipulating sounds using loops
• Reading for next class: 8.4-8.5
  – Read more about Sounds.

• Finish PSA6