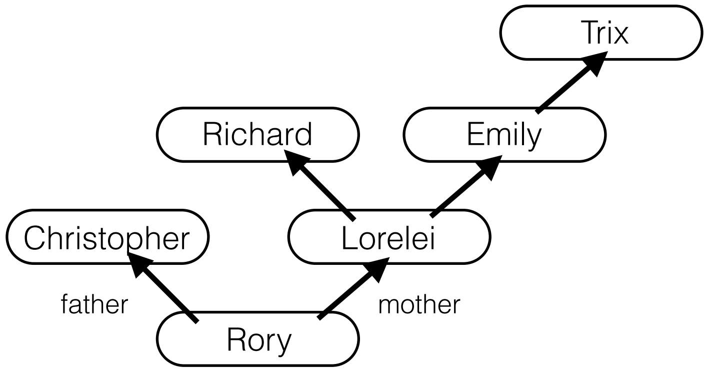
CS 125 - Lecture 35

Objectives: recursive trees; selection sort; desktop apps;

Up next: MP6 - due Monday (8PM);



3. Write a recursive method to return the total number of people in this family tree:

4. Write a tail-recursive method getFL2 to return the female lineage in reverse: *"great-grand-mother, grand-mother, mother, p." (Your method will be called with an empty string as the parameter)*

```
public String getFL2(String result) {

}
```

1. **Create an activation diagram for** `f3(31373)`:

```
public static int f3(int x) {
   if (x == 3) return 1;
   if (x < 10) return 0;

   return f3(x/10) + f3(x%10);
}
```

`What does this function do?`

6. **If each link has a larger value than the previous, will the following** `getMax()` **create a tree or chain of activations?**

```
class LinkedList{
   int value;
   LinkedList next;
}
public int getMax() {
   if (next == null)
      return value;  // BASE CASE

   int result = next.getMax();
   if (result < value) return value;
   else return next.getMax();
}
```

9. **Discuss with a neighbor your favorite desktop application's graphical user interface. Why do you like it? How does it help facilitate using the application?**

10**. Desktop user interfaces in Java:**

- Composed of graphical elements or components
- Elements are all objects: properties, behavior, inheritance

11. **JFrame**

- Subclass of Container;
- Defines a rectangular area on screen to hold components (graphical objects like buttons, sliders, text labels, etc.)
- To use, import graphics packages:

```
import java.awt.*
import javax.swing.*
```

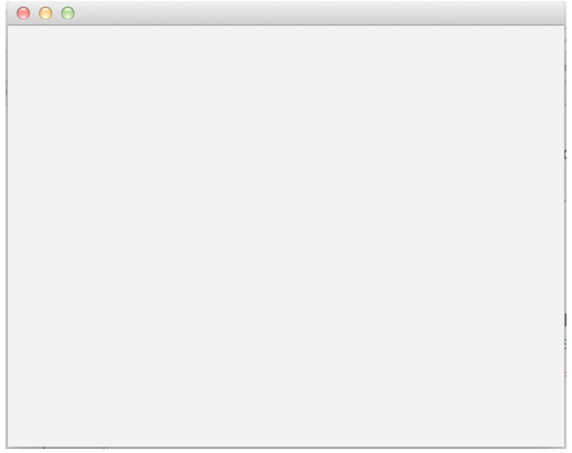
- Usage:

```
JFrame frame = new JFrame("Test Frame 1");
frame.setSize(200,100);
frame.setVisible( true );
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
```

docs (hover):

```
public void setBounds(int x, int y, int width, int height)
```

6. **Whack-a-mole windows!** `Demo desktop app`



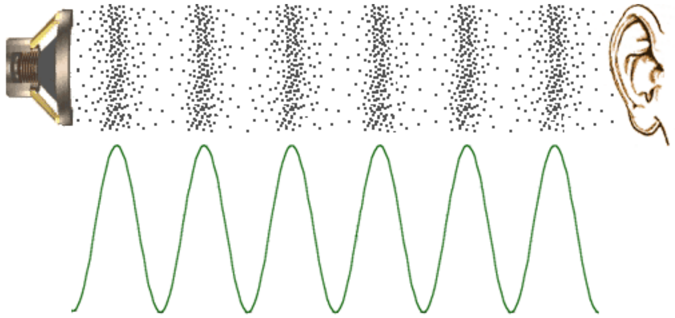
CS 125 - Lecture 35

Objectives: sound representation

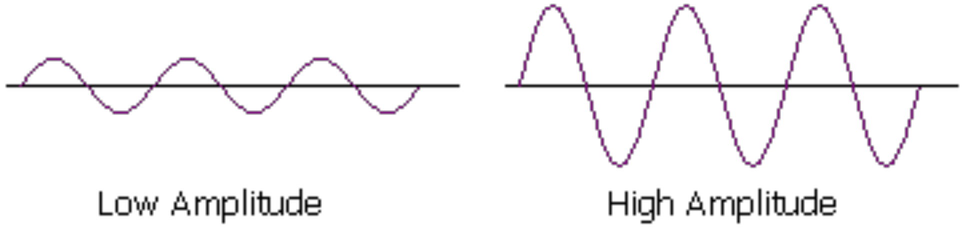
1. How can we create custom voices for the voiceless?

http://www.npr.org/2014/03/07/283452215/how-do-you-construct-a-voice

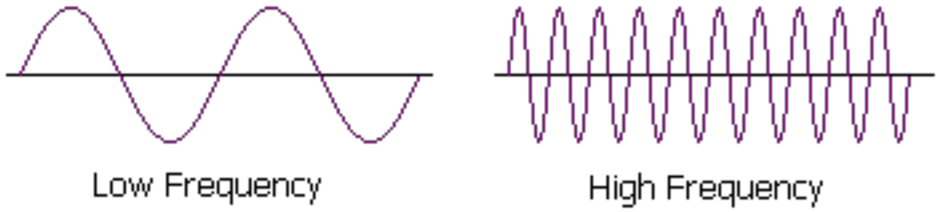
2. What is sound?



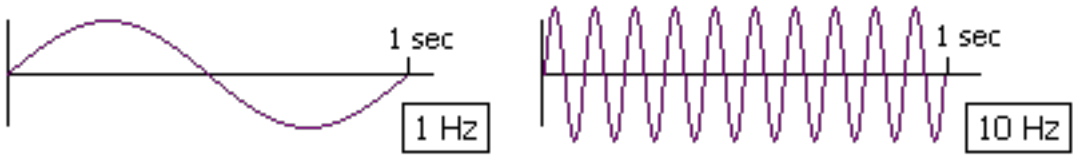
3. Characteristics of sound?



Amplitude is the size of the vibration, and this determines how loud a sound is.



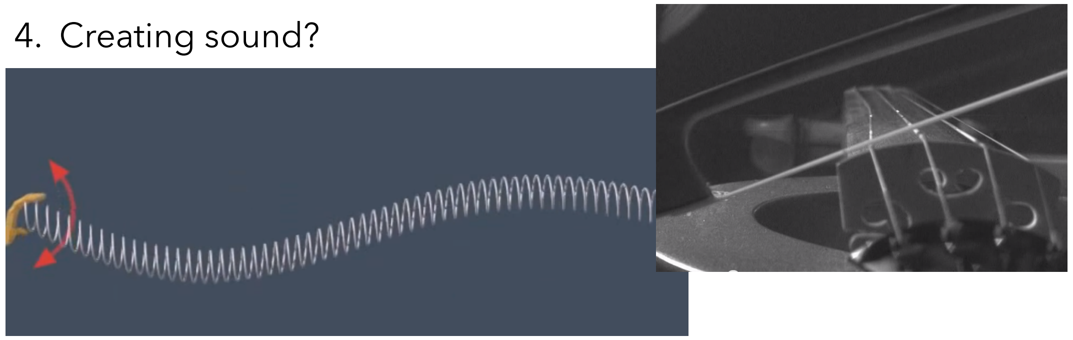
Frequency is the speed of the vibration, and this determines the pitch of the sound.



Frequency is measured as the number of wave cycles that occur in one second. The unit of frequency measurement is Hertz (Hz for short).

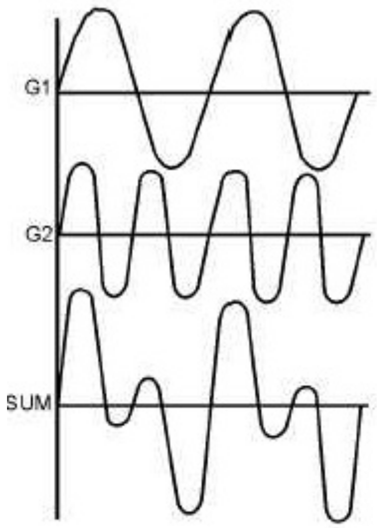
Timbre - shape, harmonics, overtones: http://www.animations.physics.unsw.edu.au/jw/sound-pitch-loudness-timbre.htm

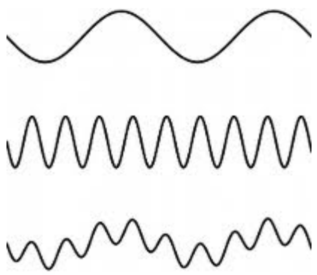
4. Creating sound?

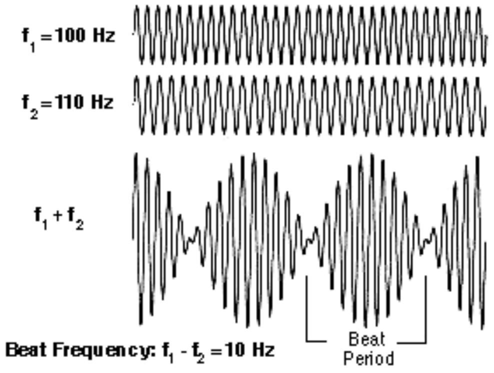


http://www.youtube.com/watch?v=6JeyiM0YNo4#t=40
http://www.youtube.com/watch?v=3BN5-JSsu_4

5. Combining waves…







There is a reason why some combinations of notes seem to go together. When you play several notes at once, you are no longer just playing the notes, the instrument you are playing is the listener's eardrum. The eardrum will vibrate with a pattern that is some complex combination of the wave-forms coming from the two (or more) notes. When you add notes together you are adding waves together and you get an interference pattern - the interference may create pleasing new sounds.

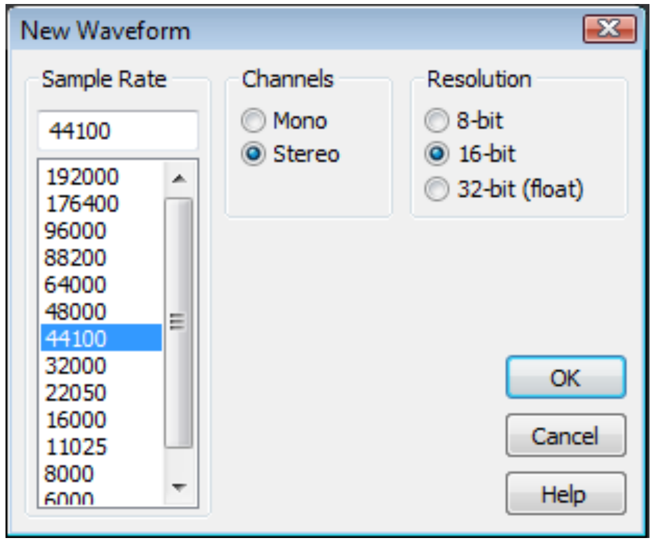
6. Diversion - Harmonics - cultural preferences: 12 pitch scale (ala piano)

http://www.youtube.com/watch?v=7_AiV12XBbI

7. Discuss with a neighbor: How would you represent sounds so that they could be worked with like other data?

Sampling is a matter of measuring air pressure amplitude at equally-spaced moments in time, where each measurement constitutes a sample. The number of samples taken per second (samples/s) is the sampling rate. Units of samples/s are also referred to as Hertz (Hz).

Quantization is a matter of representing the amplitude of individual samples as integers expressed in binary. The fact that integers are used forces the samples to be measured in a finite number of discrete levels. The range of the integers possible is determined by the bit depth, the number of bits used per sample. A sample's amplitude must be rounded to the nearest of the allowable discrete levels, which introduces error in the digitization process.



New Waveform

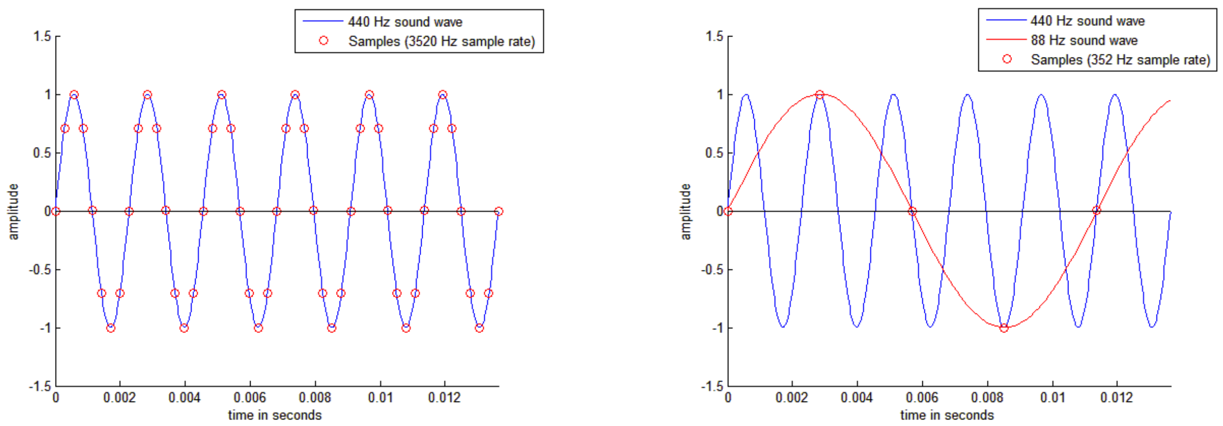
Sample Rate: 44100

192000
176400
96000
88200
64000
48000
44100
32000
22050
16000
11025
8000
6000

Channels
○ Mono
● Stereo

Resolution
○ 8-bit
● 16-bit
○ 32-bit (float)

potential problem: aliasing



8. How can we construct a custom voice? Take a page from the music industry? How do music synthesizers work?

A) Wave synthesis

B) Wave sampling

9. Listening: Switched on Bach

10. jMusic Library Demo:
http://explodingart.com/jmusic/

jMusic is a project designed to provide composers and software developers with a library of compositional and audio processing tools. It provides a solid framework for computer-assisted composition in Java™, and is also used for generative music, instrument building, interactive performance, and music analysis.

jMusic supports musicians with its familiar music data structure based upon note/sound events, and provides methods for organising, manipulating and analysing that musical data. jMusic scores can be rendered as MIDI or audio files for storage and later processing or playback in real-time. jMusic can read and write MIDI files, audio files, XML files, and its own .jm files; there is real-time support for JavaSound, QuickTime and MIDIShare.

5. Goal: Create a recursive hit song; perform; retire to tropical island;

6. **Main program: Make an array of 81 pitches; call recursive pitch generating algorithm; create a note for each pitch; record;**

```
public class Recursound implements JMC{
   public static void main(String[] args) {
      Score s = new Score(“My Hit Song");
      Part p = new Part("Flute", FLUTE, 0);
      Phrase phr = new Phrase("Chromatic scale", 0.0);

      double[] myNotes = new double[81];
      double startingPitch = 440.0;  // freq. in Hz

      //  call recursive method to create song



      for (int i = 0; i < myNotes.length; i++) {
         Note n = new Note(myNotes[i], 0.2);
         phr.addNote(n);
      }
      // add the phrase to a part
      p.addPhrase(phr);
      // add the part to the score
      s.addPart(p);
      //write a MIDI file to disk of the score
      Write.midi(s, "TestUpDown.mid");
}
```

```
public static void createSong(double[] pitches,
                              int lo, int hi, double augment)  {

//  divide the range of subarray into thirds and work on each third









}
```

5. The method below finds the smallest value in an array of doubles. Modify it to return the **index** of the smallest value.

```
Example use: int value = findMin(data, 0, data.length-1);


public static double findMin(double[] array, int lo, int hi) {
   if(lo == hi) return array[lo];

   double result = findMin(array, lo + 1, hi);

   if( result < array[lo] ) return result;

   return array[lo]; // my value wins!
}
```

6. **Use findMin() above and swap() to implement a recursive selection sort:**

```
 public static void sort(double[] data) {





 }

 public static int findMin(double[] data, int lo, int hi) {
   // returns index
    see code above
 }

 public static void swap(double[] data, int posA, int posB) {
   double temp = data[      ];
   data[     ] = data[      ];
   data[     ] = temp;
 }
```

7. You need to climb a flight of stairs with N steps. You can climb one or jump three steps at a time. How many different ways are there to ascend the stairs?

***Process**: i) Identify the sub-problem; ii) Choose parameters and temp variables; iii) Write the base cases; iv) Write the recursive case;*

Extend your solution above to include:
1) an optional elevator 200 steps from the top.
2) a missing/broken step 15 steps from the top.
3) a non-optional worm-hole exactly 211 steps from the top.

Extend your solution above so that we only count paths that have a maximum of M moves (single steps or jumps).

6. **Use findMin() above and swap() to implement a recursive selection sort:**

```
public static void sort(double[] data) {






}

public static int findMin(double[] data, int lo, int hi) {
   // returns index
    see code above
}

public static void swap(double[] data, int posA, int posB) {
   double temp = data[      ];
   data[     ] = data[      ];
   data[     ] = temp;
}
```

7. You need to climb a flight of stairs with N steps. You can climb one or jump three steps at a time. How many different ways are there to ascend the stairs?

***Process**: i) Identify the sub-problem; ii) Choose parameters and temp variables; iii) Write the base cases; iv) Write the recursive case;*

Extend your solution above to include:
1) an optional elevator 200 steps from the top.
2) a missing/broken step 15 steps from the top.
3) a non-optional worm-hole exactly 211 steps from the top.

Extend your solution above so that we only count paths that have a maximum of M moves (single steps or jumps).