Table of Contents

Preface

This module is part of a collection titled **Accessible Objected-Oriented Programming Concepts for Blind Students using Java**. It explains some of the physical aspects of sound in a format that is accessible to blind students. It also explains a little about the trigonometric sine and cosine functions and suggests that they can be used to create sound with a computer. An audio file is provided where audio pulses of different frequencies are used to represent points on a graph of a sinusoid.

General background information

By this point in the course, you should have learned enough that you can put Java OOP to work in a substantive way. Normally at this point, my sighted students would begin using OOP to manipulate the pixels and colors in digital images. That is not well suited to teaching OOP to blind students. In this course, which is designed specifically for blind students, you will begin writing substantive programs that deal with something that you are probably very good at -- sound.

For example, you will soon learn how to write a Java OOP program to produce simple sounds such as TonesStereo .

(Note that if you are viewing the HTML version of this module on the Legacy site, you should be able to download this file and the other audio files in this module and play

them using any standard media player that supports audio files of type AU. However, if you are on the OpenStax site, or if you are viewing the PDF version of this module on either site, you may not be able to download the file.)

As you study future modules in this collection, you will learn to write OOP programs to produce more complex sounds such as StereoPingpong .

Ultimately you will learn to write OOP programs that can be used to compose and play various melodies such as the following:

- FourScales
- MaryHadALittleLamb
- Greensleeves

Discussion

What is sound?

An old physics question goes something like this. If the earth were populated solely by humans and a tree fell in the forest with no humans present, would it make a sound?

The answer is "No, it would not make a sound." The explanation follows.

Vibration

Assume that someone strikes the top of a car with their hand. This would cause the metal roof of the car to start vibrating. That vibration would cause the adjacent air molecules to also vibrate causing them to collide with their neighbors. Those molecules would collide with their neighbors, and on and on. We can think of this as a wave of vibrating molecules propagating outward from the car.

This wave of vibrating molecules propagates outward at a speed of approximately one mile every five seconds and decreases in intensity as it propagates out from the source. At some distance from the source, the intensity of the vibrations has decreased to an insignificant level.

Sensing vibration

If you are standing somewhere between the car and the point at which the vibrations are no longer significant, the air molecules in your ear canals will begin to vibrate. This will stimulate nerves that lead from your ear to your brain. Your brain will interpret the stimulation as something that we refer to *sound*.

Therefore, when the tree falls, it causes the air molecules to vibrate. However, if there are no ear canals around, no nerves to be stimulated, and no brain to interpret that stimulation, there is no sound.

Creating sound with a hammer

All we need to do to cause a person to experience *sound* is to cause the air molecules in that person's ear canals to vibrate at a frequency somewhere between 20 Hz *(cycles per second)* and 20,000 Hz, depending on the person's age. Usually as a person ages, the range defined by those upper and lower limits decreases. Thus, a young person may "hear" the high-pitched whine of a power saw while an older person may not experience sound in that situation.

There are many ways that we can cause most humans to experience sound. One way for example, is to strike a bell with a hammer. Another way is to write a program to cause the diaphragm of a computer *speaker* to vibrate. That will be the topic of the next few modules in this course.

Creating sound with a computer

One way to create sound with a computer would be to hold it out and drop it on the floor. However, that is not what we intend to do. Instead, we intend to write computer program that will cause the computer to create sound.

An object of a class

As you will learn in a future module, we can define a class and instantiate an object of that class for use in producing sound with a computer. When used properly in a program, that object will cause signed numeric values stored in an array to be converted to electrical current in the speakers attached to the computer. Positive values will cause the current to flow in one direction and negative values will cause the current to flow in the opposite direction.

Alternating positive and negative values

If we populate that array with alternating groups of positive and negative values, that will cause the current being delivered to the speakers to alternate, switching from one direction to the other. The level of the alternating currents will be roughly proportional to the magnitudes of the alternating positive and negative values. Given sufficient magnitude within a frequency range supported by the speakers, the alternating currents will cause the diaphragms of the speakers to vibrate or move back and forth. This, in turn, will cause the adjacent air molecules to vibrate, which may be sensed as sound by a person within hearing distance of the speakers.

Sine and cosine functions

Populating the array

There are many ways that we can write a program to populate the array with alternating groups of positive and negative numbers. One of the simplest ways is to populate the array with scaled versions of the values produced by the static **sin** function or the static **cos** function of the **Math** class. The **Math** class is contained in the Java Standard Edition class library. A common name for a series of such values is *sinusoid*.

In addition to being one of the easiest ways to populate the array, this is also a way to produce a sound that is very close to being a *pure tone*. By pure tone, I mean a sound that results from something vibrating at a single frequency such as Middle-C *((261.63 Hz)*.

The shape of a sinusoid

If you have taken a math course in trigonometry, you probably know the shape produced by graphing the output from either the **sin** function or the **cos** function. If not, it would be good if you can ask a sighted friend to Google for "sinusoidal function image", download and print one or more of the images and emboss them for you using whatever embossing method you prefer. This will provide you with a tactile representation of a sinusoid.

An audio graph of a sinusoid

In any event, I am going to provide you with a sound file named SinusoidalAudioGraph that provides an audio representation of the graph of a sinusoid.

This audio file contains an 8-second melody consisting of 32 uniformly spaced pulses at different frequencies. The frequencies *(pitches)* of the pulses are centered on middle-C *(261.63 Hz)*. The frequency deviation from middle-C versus time is based on a sinusoidal function with a frequency of 0.5 Hz.

Points on graph

Each pulse represents one point on a graph of the sinusoid. Pulses with frequencies at or above middle-C are delivered to the left speaker. Pulses with frequencies below middle-C are delivered to the right speaker.

The audio output can be thought of as an audio representation of a graph of a sinusoid. Pulses with frequencies above middle-C represent points on the positive lobe of the sinusoid. Increasing pitch represents increasing amplitude on the graph of the sinusoid.

Pulses with frequencies below middle-C can be thought of as representing points on the negative lobe of the sinusoid. In this case, decreasing pitch represents points on the sinusoid that are further from the horizontal axis in the negative direction.

Pulses with a frequency of middle-C can be thought of as representing points on the horizontal axis with a value of zero.

Four complete cycles of the 0.5 Hz sinusoid are represented by the 32 pulses in the 8-second melody.

Hopefully, by listening to this audio file, you can get an idea of the shape of a sinusoid.

A dual purpose

In addition to providing the audio file to help you discern the shape of a sinusoid, I am also providing it as an example of the type of audio files that you will be able to create once you understand the programs in upcoming modules.

Miscellaneous

This section contains a variety of miscellaneous information.

Housekeeping material

- Module name: Jbs2000-What is Sound?
- File: Jbs2000.htm
- Published: 08/25/14

those sales nor does he know who does receive compensation. If you purchase such a book, please be aware that it is a copy of a collection that is freely available on **OpenStax CNX** and that it was made and published without the prior knowledge of Prof. Baldwin.

Affiliation: Prof. Baldwin is a professor of Computer Information Technology at Austin Community College in Austin, TX.

-end-