Sound - Minim

Minim provides a library of classes that work with sound files. You can play sound files and record sounds saving them to sound files.

This is a very brief look at sound using the classes in Mimim and is not intended to be exhaustive. It is presented here for those of you who might want to explore sound and possibly use it in your projects.

We will start with a simple program that plays sound and then look at one that records sound.

Playing Sound

First you have to get the sound files. There are places on the web that provide free sound files. Even though they are free, there are copyright rules each site requires and you must make every effort to abide by those rules. One site Jim used for downloading mp3 file is SoundJay at:

http://www.soundjay.com/

To play a sound there are a series of steps that you have to follow:

1. import the library:

   ```
   import ddf.minim.*;
   ```

   In a Processing program go to the sketch menu tab at the top. Scroll down to import Library and select the minim option. This will add something like the following to your program:

   ```
   import ddf.minim.*;
   import ddf.minim.signals.*;
   import ddf.minim.analysis.*;
   import ddf.minim.effects.*;
   ```

   This tells Processing that you will be using "stuff" in this library. The * indicates that you might use anything in the library. The * is called a wildcard.

2. Next, we have to declare object references for two different classes:

   ```
   Minim m;
   AudioPlayer s1, s2, s3, s4, s5;
   ```

   These two classes work together. The AudioPlayer objects, s1, s2, s3, s4, and s5 actually reference the sound files we want to play.

3. We have to new the Mimim object first because we use it to load the sound files and assign the AudioPlayer objects to reference them:

   ```
   m = new Minim(this);
   ```

   The word (this) refers to the program that is being executed – in other words, this program.

4. The five AudioPlayer objects are null. We do not new them but we assign them the references returned by the function loadFile( ) in the Mimim class.

```
s1 = m.loadFile("t1.wav", 1024);
s2 = m.loadFile("t2.wav", 1024);
s3 = m.loadFile("t3.wav", 1024);
s4= m.loadFile("t4.wav", 1024);
s5 = m.loadFile("t5.wav", 1024);
```

The parameters are the names of the sound files (they must be in a data folder that is in the same folder that has the .pde file) [1].

5. To play the sound one time we use the play function:

```
s1.play( );
```

Executed like this, the sound can be played only one time.

6. To replay the sound you have two choices – you can rewind the sound or load it again – there may be advantages to one way over the other but the documentation is not clear. The functions are:

```
s1.play( );
s1.rewind( );
```

or

```
s2.play( );
s2 = m.loadFile("t2.wav");
```

7. You can loop a sound infinitely or for a set number of times. This should play the sound three times :

```
s4.loop( 3 );
```

and this should play it infinitely:

```
s4.loop( );
```

8. The last thing we must do is to close everything down neatly and cleanly so the program does not crash and the files are not corrupted. This is usually done in a stop( ) function that Processing calls for us (just like **setup( )** and **draw( )**). We close things in the reverse order that we created them starting with the **AudioPlayer** objects and ending with *the* **Minim** object*:*

[1] If you are going to use sound in a project, you have to copy the data folder into the applet folder before you transfer the applet folder to your www directory.

```
void stop( )
{
  // always close audio I/O classes
    s1.close();
    s2.close();
    s3.close();
    s4.close();
    s5.close();
 // always stop your Minim object
    m.stop();
    super.stop();
}
```

The `super.stop();` is how we have to call the stop function of the parent class. **Super** refers to the parent' class. Since the parent class is above the child class in the tree, it is in a superior position – hence the use of **super** to call a function in the parent class.

Refer to the **SoundPlaying** folder in the class code Set 21 to see how this code works.

You may have to increase the memory allocated to Processing to get your program to run. This is done by clicking on the Processing tab and selecting the preferences option. There is a line for increasing the maximum memory – this should be a power of 2 – try 2048. You should restart Processing.

Keep the sound files as small as possible. However, you can still run out of memory if you have too many sound files or they are too large.

Recording Sound

You can record your own sounds using Minim but the program will not work in an applet. The sound files you record will work in your program and in an application if you choose the export Application option under the file menu…and they will be royalty free.

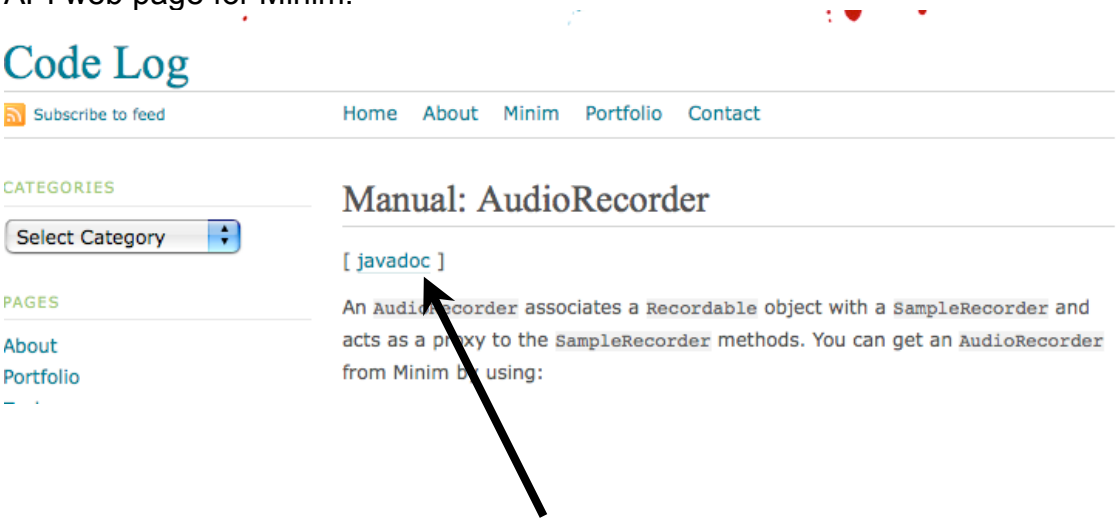
To record sound, code similar to the following must be in your program:

1. Import the library:
   ```
   import ddf.minim.*;
   ```
2. Create three object references:
   ```
   Minim minim;
   AudioInput in;
   AudioRecorder recorder;
   ```

3. New them or assign the returned references:
```
minim = new Minim(this);
in = minim.getLineIn(Minim.STEREO, 512);
recorder = minim.createRecorder(in, "s.wav", true);
```
4. To begin recording you the **beginRecord( )** function:
```
recorder.beginRecord();
```
5. To end recording, use the **endRecord( )** function:
```
recorder.endRecord();
```
6. When the recording is finished save the sound file:
```
recorder.save();
```
7. **Finally, close things down to insure the files are not corrupted:**
```
in.close();
minim.stop();
super.stop( );
```

To see how this code works, refer to the code in the **SoundRecording** folder in the class code set 21

This is only a brief introduction to sound. It is not intended to be thorough, deep or exhaustive. It is here only to tease you and lure you even closer to the event horizon. To see more about the Minim library, click on the javadoc link on the API web page for Minim:



This is a very technical API page but the parts you need to know are very straightforward.
Just remember
- fields are variables
- methods are functions

and you should be ok. . . you should be ok. . . you should be ok. . . you should be ok. . .