

CS 1316 – Homework 4 – Sampled Sounds

Due: Friday June 12th 2009 before 6pm

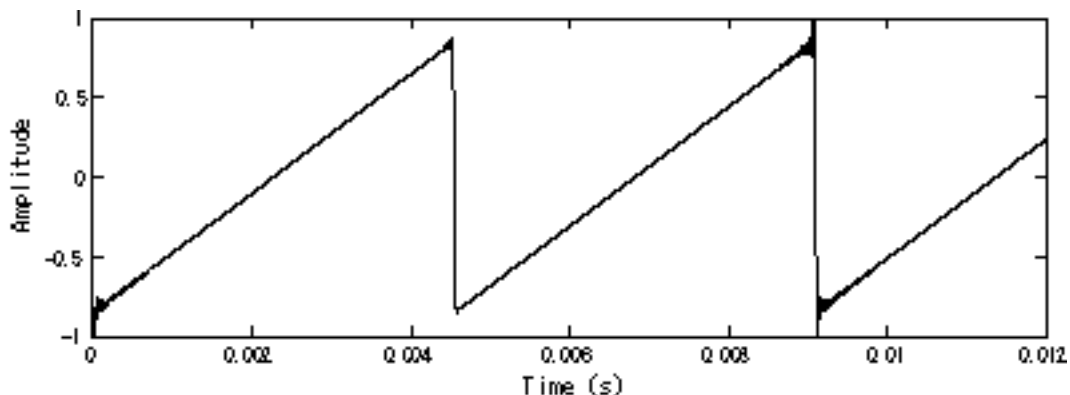
This is a pair programming problem! You are expected to work with the person you have been paired with in class, and you are both responsible for submitting the exact same code to T-Square.

You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others. For pair programming assignments, you and your partner should turn in identical assignments. Collaboration at a reasonable level will not result in substantially similar code.

Homework 4 – Sounds

For this homework, you will be adding one static method and two object methods to the Sound class (You will need to edit the Sound.java file in java-sources, and when you are done, you will need to submit the Sound.java file. Make sure you add a comment with your name and collaboration statement to the top of the file). The static method will produce a sawtooth wave while the object methods will carry out sound manipulations.

Sawtooth



Your sawtooth method should return a Sound object and take in a double for the time (in seconds) and an integer for the wavelength (in number of samples). Here is a syntax example:

```
Sound saw = Sound.sawtooth(1.5, 400);
```

In this example, you should create a sound that was 1.5 seconds long and had a peak every 400 samples. Note that we use 44,100 samples per second. This means that the sound would start at -32000 and go up to 32000 over the first 400 samples, and then drop back down to -32000 immediately. The sawtooth wave should go from -32000 to 32000 in a linear manner, using uniform steps.

For further details see: http://en.wikipedia.org/wiki/Sawtooth_wave

Inverse

When called on a Sound object, your inverse method should return the same sound with an inverted phase. Here is a syntax example (note `snd` is an existing Sound object):

```
Sound inv = snd.inverse();
```

The inverse should behave such that silence is produced when equally mixing a sound and its inverse. Example:

```
Sound mix = snd.mix(inv,0.5);
```

For further details see: http://en.wikipedia.org/wiki/Sound_cancellation

Linear Extension

When called on a Sound object, your extend method should return a new sound with twice the length and lowered one octave. Here is a syntax example (note `snd` is an existing Sound object):

```
Sound ext = snd.extend();
```

You must use linear interpolation to fill in the gaps in the sound (i.e. your resulting sound will have one less than twice the number of samples in the original). Make sure to preserve the sample rate.

Grading Criteria

Sawtooth

- Header correct 10pt
- Produces a sound of the correct length 10pt
- Sound is a sawtooth wave 20pt

Inverse

- Header correct 5pt
- Produces a sound of the correct length 5pt
- Sound is the inverse of the source 10pt

Extend

- Header correct 5pt
- Produces a sound of twice the length of the source 15pt
- Sound is one octave lower than the source 20pt

Total: 100pts