Programming with Sounds, Part 3

Exercise 6

Let’s return to thinking about volume control. As you discovered, when you increase the volume too much, clipping occurs, which doesn’t sound very good!

A good way to fix this is to first normalize the sound, which brings it to its maximum amplitude. Normalizing a sound makes it as loud as possible without clipping. We typically think of normalized sounds as being at 100% volume.

To normalize a sound, first find the biggest sample, or maximum amplitude, in the sound. Let’s call it maxAmp. Remember that a sound sample whose value is -30,000 is “bigger” than a sound sample of 15,000. (Hint: you might want to use the abs() and max() functions.)

Once you’ve found maxAmp, you need to multiply each sample in the sound by 32767.0 / maxAmp. Why are we multiplying by 32767.0 instead of 32767?

Write a function to find maxAmp. Then write a function to normalize a sound. (Think about whether this function should have a side effect or not; should it change the original sound, or simply return a new sound and leave the original one unchanged.)

Finally, write a function that plays a normalized sound at a given volume, where the volume is between 0 and 100%. (Remember, the normalized sound is 100% volume.)

Check:

☐ proper documentation for each function (name, purpose, parameters, return value, assumptions, side effects)
☐ each function works properly on the sound specified in the exercise
☐ each function works properly with other sounds
☐ all written questions answered