Sample Tweaker

COSC 4P98

4654166

Kevin Delisle

5/4/2012

Overview

This project is a sample tweaker, made in Processing using the Beads and ControlP5 libraries.

Beads allows you to manipulate audio data within the Processing environment. ControlP5 is an open-source library of control objects for Processing to allow for easy creation and use of mouse-based controls.

Sample Tweaker allows you to load a sample, play it back, tweak select variables during playback, and save the recorded buffer to a new wave file.

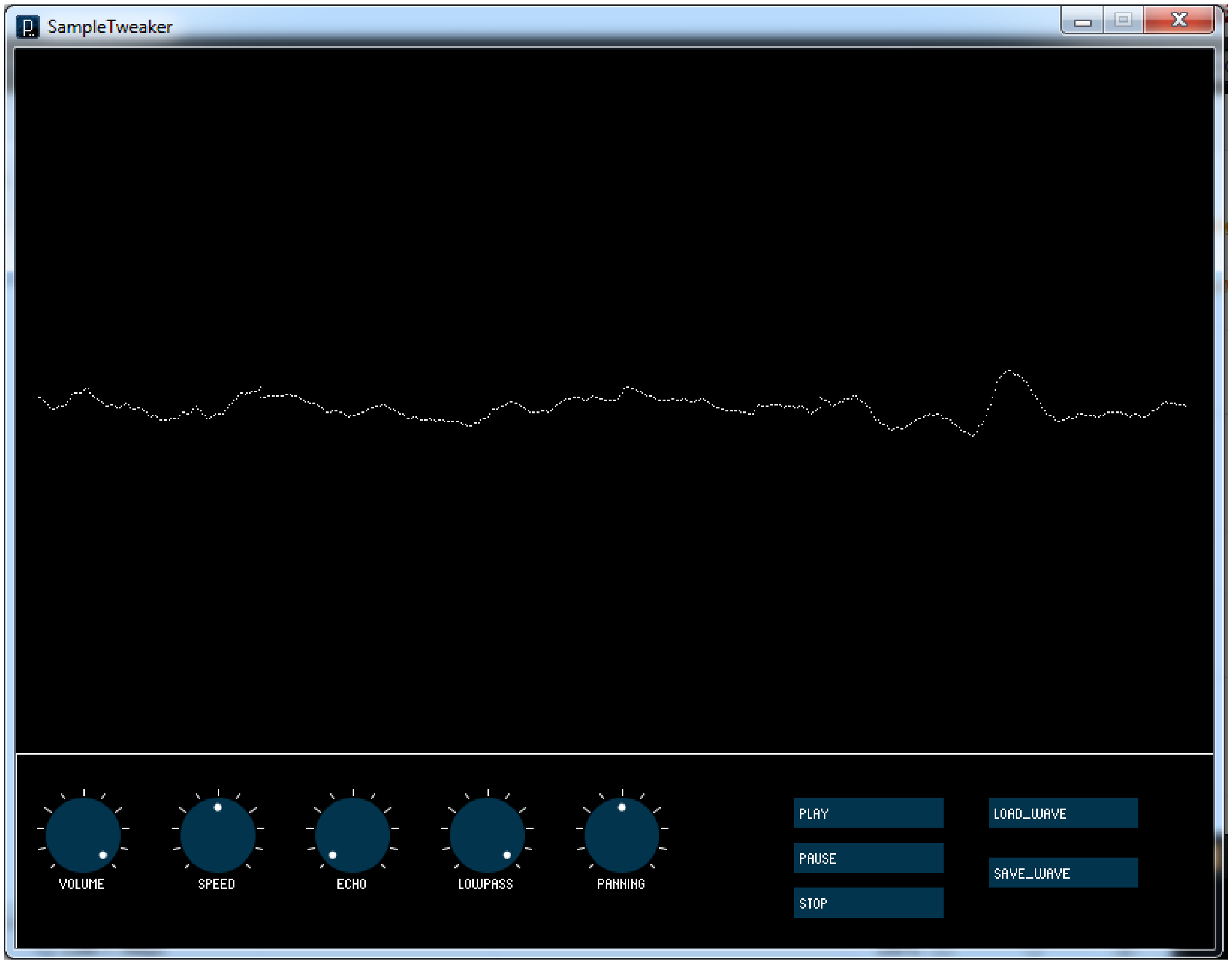

Figure 1 - SampleTweaker

Technical Foundations

General

In Beads, audio is handled through the use of objects referred to as Unit Generators, or “uGens”. All unit generators must be connected to an “AudioContext”, which establishes the base for the manipulation and playback of audio samples. uGens can be fed into AudioContexts so that their calculations can be output as sound, and in addition, uGens can be fed into one another to create multiple effects. For the remainder of this report, when a uGen takes in, or is fed into, another uGen or AudioContext, this will be colloquially referred to as “patching”.

Visualizer

The visualizer rips values from the AudioContext buffer to draw a snapshot of the position of each sample within the playback buffer, and does so repeatedly during the entire playback of the sample (this effect will start and stop with the controls). These buffer values are modified slightly to stretch the wave to fit it to its window properly.

Loading a Sample

Samples are loaded into Beads through the use of Sample objects, and are manipulated and played using SamplePlayer objects. SamplePlayer objects allow us to interface with Samples without having to deal with lower-level manipulation of discrete values.

Playing a Sample

During any playback of a sample, a buffer is setup using the RecordToSample object, another instance of the Sample object, and an AudioFormat object, all of which allow us to capture any output being passed through an AudioContext.

Volume

The volume control is based upon a simple increase in gain, using a Gain object as defined in the Beads library. Each discrete value currently being played from the sample has its value multiplied by a float multiplier (between 0 and 1), captured directly from the volume knob. This effect is applied to the entirety of the sample during playback, modifying the volume. This principle is based upon the design of the .wav format. In the design of this program, the Gain uGen is patched into the uGens that handle delay and panning.

Speed

The speed control directly manipulates the playback rate of the SamplePlayer, but does not synchronize this rate increase or decrease with the recording rate of the buffer. This allows us to alter the frequency of the sample as it plays back, and directly save those changes to the new sample.

Echo

Echo is accomplished using a TapIn and TapOut object. The TapIn object retrieves audio samples as they are being served up by some AudioContext object (specified at instantiation). TapOut has a TapIn object patched into it, as well as a delay (in ms) and pushes the samples out to the patched in AudioContext after the specified delay. In this program, the delay of the TapOut object is tied directly into the output of the Echo knob, which allows real-time addition of echo to the recording buffer.

Lowpass

A lowpass filter was added to this project, not because it was difficult to implement, but rather to reduce clipping of certain samples during playback for better sounding recordings. The Lowpass filter was implemented using the OnePoleFilter object, which simply is patched into an AudioContext and given a maximum allowed frequency. The filter's maximum frequency is then modified by the lowpass knob's function.

Panning

Panning was also added for cleanup of sample balance, (although you can use it to make weird sounding transitions between headphones if preferred) and was implemented using the Panning object, which is patched into an AudioContext and a value between -1 and 1 to represent the balancing between left and right channels, respectively. The Panning knob simply outputs a fraction between these values to modify how much gain is delivered to each channel.

Instructions for Use

Volume – simply click and hold the mouse button on the Volume dial and drag left or right to increase and decrease volume of the sample playback. Note that you can use this to control the intensity of the samples in the recording (allowing you to apply a fade-out effect to the recording).

Speed – click and hold to manipulate the speed; turning the dial to the left of the centrepoint will play the sample back at a fraction of the original sample speed, whereas turning it to the right of the centrepoint will increase playback speed.

Echo – The echo is set to zero by default, but turning the dial will begin to implement a delay effect, up to a maximum of 2000 ms. Turning down the delay to zero again while the delay buffer is playing back will purge the buffer of any excess information, allowing you to restore the playback to an echoless state without having to wait for the buffer to empty (this allows you to implement echo for key parts of the sample if you'd like).

Lowpass – the low pass filter, by default, is turned up to maximum, allowing all frequencies below 20,000 Hz to play at their respective gain. As this knob is turned left, this frequency range will decrease, drastically reducing the gain of frequencies nearer to the cutoff point, and gradually removing frequencies far above the cutoff point. This is useful for samples that are clipping, allowing you to reduce the harsh sound they produce.

Panning – as previously mentioned, panning simply allows you to adjust which stereo channels receive what percentage of the incoming sound. This lets you create the sensation of moving audio, or simply allows you to rebalance the sound of a sample to certain parts of the stereo output.

Load_Wave – Opens up a dialog where you can select a wave file for playback. All of the objects associated with the controls and recording are instantiated at this point and prepared for playback.

Save_Wave – Opens up a dialog where you can save the recorded sample that you modified.

Play – Begins the sample playback, and begins recording sound from the AudioContext.

Pause – Pauses the sample playback and halts the recording for the AudioContext. Pressing Play again will resume playback at the same location and continue recording. Please note that you can use the Pause controls to make changes to the audio controls (and therefore, the recording) in a more precise manner.

Stop – Stops audio playback entirely and stops recording. **Please note that if you press Play again, the current recording will be lost. If you wish to keep the recording you have made, click "Save_Wave".**

Bibliography

Bown, O. (2011, September). *Beads Project*. Retrieved May 1, 2012, from http://www.beadsproject.net/

Bown, O., Crawford, B., & Porter, B. (2011, September). *Beads JavaDocs*. Retrieved May 1, 2012, from http://www.beadsproject.net/doc/

Schlegel, A. (2010). *Control P5 JavaDocs*. Retrieved May 1, 2012, from http://www.sojamo.de/libraries/controlP5/reference/index.html

Schlegel, A. (2010). *processing GUI, controlP5*. Retrieved May 1, 2012, from controlP5: http://www.sojamo.de/libraries/controlP5/