Audio Plugins with JUCE

EECS 4462 - Digital Audio

October 19, 2018

YORK UNIVERSITÉ UNIVERSITY

Setup

- Create a new project with the Projucer
- The GUI component as well as the audio parameters for the plugin processor are the same as before
- The main difference is in the processBlock method
- We will now work with the `AudioBuffer<float>&` argument

Important Class: AudioBuffer

- A collection of sample values
- Can be read using pointer arithmetic

```
const float* data =
    buffer.getReadPointer (channel,0);
float s1 = data[5];
```

- Can be written using pointer arithmetic

```
float* data =
   buffer.getWritePointer (channel,0);
float s2 = data[3];
data[7] = s2 + 0.3;
```

A more realistic example

- Double the value of every sample
- This will make the output of the plugin louder

```
int numSamples = buffer.getNumSamples();
for (int i = 0; i < numSamples; ++i)
                data[i] = data[i]*2;
```

- Method getNumSamples() returns the number of samples in the buffer
  - In processBlock, typically the value is 1024

Warning

- From the API of getWritePointer:
- `For speed, this doesn't check whether the channel number or index are out of range, so be careful when using it!`
- Common practice in audio to avoid checks
- It’s up to the programmer to make sure all values will be valid!

Storing audio data

- You can create your own AudioBuffer objects
  - In fact, you will need to do that for A2
- You most likely want to use the constructor

```
AudioBuffer (int channels,
             int samplesToAllocate)
```

- You can change the size of an AudioBuffer with

```
setSize (int channels,
         int samplesToAllocate)
```

Starter code for A2

```
AudioBuffer<float> delayBuffer;

.....

delayBuffer.setSize(2, delayBufferLength);

delayBuffer.clear();

.....

float* delayData =
delayBuffer.getWritePointer(ch);
```

How to set this to 2 sec?