My Project

Generated by Doxygen 1.9.2

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 WAV Class Reference

Public Member Functions

- bool **loPass** (int32_t max)
- bool **normalize** (int32_t max)
- bool gain (double gain)
- bool loadData (std::string filePath)
- bool writeData (std::string filePath)
- wav_meta **getMetaData** () const
- uint8_t * **getDataBytes** () const

3.1.1 Member Function Documentation

3.1.1.1 gain()

```
bool WAV::gain (
            double gain )
```

Applies a gain effect to the data

Returns

bool if successful

Parameters

gain	decimal value to scale the data

3.1.1.2 loadData()

```
bool WAV::loadData (
            std::string filePath )
```

Loads the meta data into the struct and the data into the 8 bit int array

Parameters

filePath	direct or relative path to the .wav file

3.1.1.3 writeData()

```
bool WAV::writeData (
            std::string filePath )
```

Writes the data from the struct and 8 bit int array to the provided file

Parameters

filePath	direct or relative path to the .wav file to be made or changed

The documentation for this class was generated from the following files:

- /home/kc/Documents/CS-202-Semester-Project-F21/src/wav.h
- /home/kc/Documents/CS-202-Semester-Project-F21/src/wav.cpp

3.2 WAV_HEADER Struct Reference

Public Attributes

- uint8_t **RIFF** [4]
- uint32_t **chunkSize**
- uint8_t **WAVE** [4]
- uint8_t **fmt** [4]
- uint32_t **subchunk1Size**
- uint16_t **audioFormat**
- uint16_t **numOfChan**
- uint32_t **samplesPerSec**
- uint32_t **bytesPerSec**
- uint16_t **blockAlign**
- uint16_t **bitsPerSample**
- uint8_t **subchunk2ID** [4]
- uint32_t **subchunk2Size**

The documentation for this struct was generated from the following file:

- /home/kc/Documents/CS-202-Semester-Project-F21/src/wav.h

Chapter 4

File Documentation

4.1 tools.h

```
1 #ifndef TOOLS_H
2 #define TOOLS_H
3 #include <iostream>
4 #include "wav.h"
8 std::ostream &operator«(std::ostream &output, const WAV &wav);
15 int pow(int base, int exp);
16
17 #endif
```

4.2 wav.h

```
1 #ifndef WAV_H
2 #define WAV_H
3 #include <string>
4 #include <fstream>
5 #include <iostream>
6 typedef struct  WAV_HEADER
7 {
8     uint8_t     RIFF[4];
9     uint32_t    chunkSize;
10     uint8_t     WAVE[4];
11
12     uint8_t     fmt[4];
13     uint32_t    subchunk1Size;
14     uint16_t    audioFormat;
15     uint16_t    numOfChan;
16     uint32_t    samplesPerSec;
17     uint32_t    bytesPerSec;
18     uint16_t    blockAlign;
19     uint16_t    bitsPerSample;
20
21     uint8_t     subchunk2ID[4];
22     uint32_t    subchunk2Size;
23 } wav_meta;
24
25
26 class WAV {
27
28 private:
29     uint8_t* dataBytes;
30     wav_meta metaData;
31     int32_t numSamples;
32     int32_t readSample(int32_t index) const;
33     bool writeSample(int32_t index, int32_t value);
34 public:
35     WAV();
36     bool loPass(int32_t max);
37     bool normalize(int32_t max);
43     bool gain(double gain);
48     bool loadData(std::string filePath);
53     bool writeData(std::string filePath);
54     wav_meta getMetaData() const;
55     uint8_t* getDataBytes() const;
56 };
57
58 #endif
```

Index