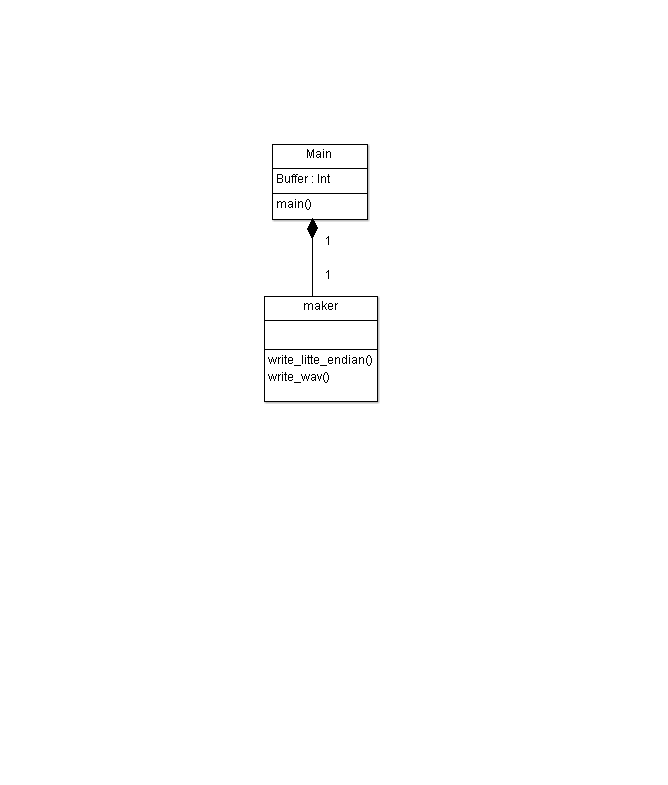
Project: WAV maker

Author : Kevin Karplus

Actual Code Lines: 73

//It’s not a simulation, but I though the interaction with WAV files could be interesting.

Files: main.c , make\_wav.c , make\_wav.h

UML

Main.c

#include <stdio.h>

#include <stdlib.h>

/\* test\_make\_wav.c

\* Fri Jun 18 17:13:19 PDT 2010 Kevin Karplus

\* Test program for the write\_wav function in make\_wav.c

\*/

#include <math.h>

#include "make\_wav.h"

#define S\_RATE (44100)

#define BUF\_SIZE (S\_RATE\*2) /\* 2 second buffer \*/

int buffer[BUF\_SIZE];

int main(int argc, char \* argv)

{

int i;

float t;

float amplitude = 32000;

float freq\_Hz = 440;

float phase=0;

float freq\_radians\_per\_sample = freq\_Hz\*2\*M\_PI/S\_RATE;

/\* fill buffer with a sine wave \*/

for (i=0; i<BUF\_SIZE; i++)

{

phase += freq\_radians\_per\_sample;

buffer[i] = (int)(amplitude \* sin(phase));

}

write\_wav("test.wav", BUF\_SIZE, buffer, S\_RATE);

return 0;

}

Make\_wav.c

/\* make\_wav.c

\* Creates a WAV file from an array of ints.

\* Output is monophonic, signed 16-bit samples

\* copyright

\* Fri Jun 18 16:36:23 PDT 2010 Kevin Karplus

\* Creative Commons license Attribution-NonCommercial

\* http://creativecommons.org/licenses/by-nc/3.0/

\*/

#include <stdio.h>

#include <assert.h>

#include "make\_wav.h"

void write\_little\_endian(unsigned int word, int num\_bytes, FILE \*wav\_file)

{

unsigned buf;

while(num\_bytes>0)

{ buf = word & 0xff;

fwrite(&buf, 1,1, wav\_file);

num\_bytes--;

word >>= 8;

}

}

/\* information about the WAV file format from

http://ccrma.stanford.edu/courses/422/projects/WaveFormat/

\*/

void write\_wav(char \* filename, unsigned long num\_samples, short int \* data, int s\_rate)

{

FILE\* wav\_file;

unsigned int sample\_rate;

unsigned int num\_channels;

unsigned int bytes\_per\_sample;

unsigned int byte\_rate;

unsigned long i; /\* counter for samples \*/

num\_channels = 1; /\* monoaural \*/

bytes\_per\_sample = 2;

if (s\_rate<=0) sample\_rate = 44100;

else sample\_rate = (unsigned int) s\_rate;

byte\_rate = sample\_rate\*num\_channels\*bytes\_per\_sample;

wav\_file = fopen(filename, "w");

assert(wav\_file); /\* make sure it opened \*/

/\* write RIFF header \*/

fwrite("RIFF", 1, 4, wav\_file);

write\_little\_endian(36 + bytes\_per\_sample\* num\_samples\*num\_channels, 4, wav\_file);

fwrite("WAVE", 1, 4, wav\_file);

/\* write fmt subchunk \*/

fwrite("fmt ", 1, 4, wav\_file);

write\_little\_endian(16, 4, wav\_file); /\* SubChunk1Size is 16 \*/

write\_little\_endian(1, 2, wav\_file); /\* PCM is format 1 \*/

write\_little\_endian(num\_channels, 2, wav\_file);

write\_little\_endian(sample\_rate, 4, wav\_file);

write\_little\_endian(byte\_rate, 4, wav\_file);

write\_little\_endian(num\_channels\*bytes\_per\_sample, 2, wav\_file); /\* block align \*/

write\_little\_endian(8\*bytes\_per\_sample, 2, wav\_file); /\* bits/sample \*/

/\* write data subchunk \*/

fwrite("data", 1, 4, wav\_file);

write\_little\_endian(bytes\_per\_sample\* num\_samples\*num\_channels, 4, wav\_file);

for (i=0; i< num\_samples; i++)

{ write\_little\_endian((unsigned int)(data[i]),bytes\_per\_sample, wav\_file);

}

fclose(wav\_file);

}

Make\_wav.h

/\* make\_wav.h

\* Fri Jun 18 17:06:02 PDT 2010 Kevin Karplus

\*/

#ifndef MAKE\_WAV\_H

#define MAKE\_WAV\_H

void write\_wav(char \* filename, unsigned long num\_samples, short int \* data, int s\_rate);

/\* open a file named filename, write signed 16-bit values as a

monoaural WAV file at the specified sampling rate

and close the file

\*/

#endif