## Guide to DVD Chapter 16 Examples: *John ffitch*

# A MIDI-based Algorithmic Composition Library

## Building

The library and example program a provided with a Makefile that should work on any platform; it does not depend on any libraries other that ANSI Standard C.

Issue the command:

```
$ make
```

and that will build the library `midilib.a` and the example programs `miditest, midimono and midiex`; it also creates a program `midiex1` that is designed for user modification.

## Using the Library

The library provides the foillowing functions.

```
int new_score(void)
```

This function should be called as the first operation as it initialises internal state. It will destroy any previous score. Only one MIDI score can be created at one time. The function always returns zero.

```
int add_note(int pitch, int volume, int instr,
             int when, int length)
```

Every time this function is called a note at the MIDI pitch ([0-127] with 60 representing Middle C) and amplitide in range 0-127 for the instrument numbered is added to the score at the time when (in milliseconds). The note lasts `length` milliseconds when a NoteOff event is created. Internally the score is sorted into order.

```
int save_score(char *name)
```

When sufficient notes have been created a call to `save_score` will create a MIDI file whose name is provided as the `string`. It should be noted that the score is not lost, and so, more notes

can be added, or a different file can be created. It is `new_score` that removes the internal score. These functions are all declared in the header file `midilib.h`

## The miditest Program

The *miditest* program uses the MIDI library to create a MIDI file consisting of a C major scale starting on middle-C (MIDI NN 60). Each successive note is one second long.

## The midimono Program

The *midimono* program is the pre-cursor to the *midiex* programs. It uses the MIDI library to create a stochastic composition for a **single** instrument (piano) that closely converges on the sequence C-A-G-E. The pitches are created by a random number generator and have a chance distribution scheme. Details are in the text.

## The midiex Program

The *midiex* program uses the MIDI library to create a stochastic composition for four instruments (a wind quartet) that closely converges on the sequence C-A-G-E. The pitches are created by a random number generator and have a chance distribution scheme. Details are in the text.

## The midiex1 Program

The *midiex1* program is built from the source `midiexamp.c` and it is essentially the same program as described in the text, except that: different instruments are used; a different starting point is given for the pseudo random number generator; and there are different weights. In the source code you will find comments indicating places where the program could be changed or generalized. Try it. Then try changing it and running `make` again. If you create a masterpiece, claim it as yours!