Guide to BOOK Chapter 14 Examples:
Steven Yi

Modeling Orchestral Composition

Introduction

Included in this directory are files for the C++ orchestral composition library and an example program demonstrating usage of the library. The following files make up the library and example:

`Note.cpp` – Implementation of the Note class.

`Note.hpp` – Header definition of the Note class.

`Performer.cpp` – Implementation of the Performer class.

`Performer.hpp` – Header defintion of the Performer class.

`PerformerGroup.cpp` – Implementation of the PerformerGroup class.

`PerformerGroup.hpp` – Header defintion of the PerformerGroup class.

`Utilities.cpp` – Utilitiy functions used by Performer and PerformerGroup for conversions betwen Csound PCH notation and a base ten format.

`main.cpp` – Example program demonstrating usage of the programming library.

Also included in this directory is:

`Makefile` – A make buildfile for compiling the example programs and creating output WAV and MP3 files.

`example.orc` – Csound ORC file used in conjunction with the Csound SCO file generated from the example program.

`example.mp3` – A precompiled MP3 of the audio examples generated from the example program.

Requirements

The following programs are required for building the program and musical examples:

g++ – the GNU C++ compiler.
make – cross-platform build utility.
Csound – computer music synthesis system.
lame – (optional) command-line mp3 encoder.

This library and example program has been tested on Linux, Windows XP, and Mac OS X 10.4 operating systems. For Linux, the Fedora 8 distribution was used and g++, make, and lame were installed using the yum package manager. Csound was downloaded in source format and compiled on the computer, though packages may be found in the avalable Fedora yum repositories. For Windows, the MinGW and MSYS tools were installed from http://www.mingw.org which contain g++ and make. For OS X, developer tools were installed for g++ and make. Csound was installed using an installer available through http://www.csounds.com. A compiled version of lame was installed from http://lame.sourceforge.net.

After the required programs are installed, they should all be added to the PATH variable for the operating system so that they may be called from within any directory.

Usage

To compile the program, use the make build tool with *Makefile* by opening up a command-line shell and navigating to the `src` folder of the project.

To compile the example program, type:

```
$ make
```

To create a Csound SCO file from the generated program, type:

```
$ make sco
```

To create a WAV file from the Csound SCO and ORC, type:

```
$ make wav
```

To create an MP3 from the WAV file, type:

```
$ make mp3
```

Each of the above targets is configured in the *Makefile* to depend on the previous step, so calling "`make mp3`" will generate the example program, the Csound SCO file and the WAV before generating the MP3.