

Preface

This book is the culmination of many years of work and development. At the Csound Conference in Hannover, 2011, there was a general agreement among the community that a new book on the many new features of the language was a necessity. For one reason or another, the opportunity of putting together an edited collection of chapters covering different aspects of the software never materialised. As Csound 5 gave way to Csound 6, and the system started expanding into various platforms, mobile, web, and embedded, the need for a publication centred on the system itself became even more evident. This book aims to fill this space, building on the already rich literature in Computer Music, and adding to previous efforts, which covered earlier versions of Csound in good detail.

A major aspect of this book is that it was written by a combination of system developers/maintainers, Lazzarini, Yi and ffitch, and power users/developers, Brandtsegg, Heintz and McCurdy, who together have a deep knowledge of the internal and external aspects of Csound. All the major elements of the system are covered in breadth and detail. This book is intended for both novice and advanced users of the system, as well as developers, composers, researchers, sound designers, and digital artists, who have an interest in computer music software and its applications. In particular, it can be used wholly or in sections, by students and lecturers in music technology programmes.

To ensure its longevity and continued relevance, the book does not cover platform and host-specific issues. In particular, it does not dedicate space to showing how to download and install the software, or how to use a particular feature of a currently existing graphical frontend. As Csound is a programming library that can be embedded in a variety of ways, there are numerous programs employing it, some of which might not be as long-lasting as others. Wherever relevant and appropriate, we will be considering the terminal (command-line interface) frontend, which is the only host that is guaranteed to always be present in a Csound installation. In any case, the operation topics left out of this book are duly covered in many online resources (more details below), which are in fact the most appropriate vehicles for them.

The book is organised in five parts: Introduction; Language; Interaction; Instrument Development; and Composition Case Studies. The two chapters in the Introduction are designed to give some context to the reader. The first one localises Csound in the history of Computer Music, discussing in some detail a number of its predecessors, and introducing some principles that will be used throughout the book. This is followed by a chapter covering key elements of computer music software, as applied to Csound. As it navigates through these, it moves from general considerations to more specific ones that are central to the operation of the system. It concludes with an overview of the Csound application programming interface (API), which is an important aspect of the system for software development. However, the focus of this work is firmly centred on the system itself, rather than embedding or application programming. The next parts of the book delve deeper into the details of using the software as a sound and music computing system.

The second part is dedicated to the Csound language. It aims to be a concise guide to programming, covering basic and advanced elements, in an up-to-date way. It begins with a chapter that covers the ground level of the language, at the end of which the reader should have a good grasp of how to code simple instruments, and use the system to make music. It moves on to discuss advanced data types, which have been introduced in the later versions of the system, and provide new functionality to the language. Core issues of programming, such as branching and loops, as well as scheduling and recursion are covered in the third chapter. This is followed by an overview of instrument graphs and connections. The fifth chapter explores the concept of user-defined opcodes, and how these can be used to extend the language.

The topic of control and interaction is covered in the third part of the book. The first chapter looks at the standard numeric score, and discusses the various types of functionality that it offers to users. The reader is then guided through the MIDI implementation in Csound, which is both simple to use, and very flexible and powerful. The facilities for network control, via the Open Sound Control protocol and other means, is the topic of the third chapter in this part. A chapter covering scripting with a separate general-purpose programming language complements this part. In this, we explore the possibilities of using Python externally, via the Csound API, to control and interact with the system.

The fourth part of this book explores specific topics in instrument development. We look at various types of classic synthesis techniques in its first chapter, from subtractive to distortion and additive methods. The following one examines key time-domain processing elements, studying fixed and variable delay lines and their applications, different types of filtering, and sound localisation. The third chapter introduces sound transformation techniques in the frequency domain, which are a particularly powerful feature of Csound. The more recent areas of granular synthesis and physical models are featured in the two remaining chapters of this part.

The final section of the book is dedicated to composition applications. An interesting aspect of almost all of the developers and contributors to Csound is that they are composers. Although the system has applications that go beyond the usual electronic music composition uses, there is always significant interest from old and

new users in its potential as a music composition tool. The six chapters in this section explore the authors' individual approaches to using the software in this context. These case studies allow us to have a glimpse of the wide variety of uses that the system can have.

All the code used as examples in this book is freely available for downloading, pulling, and forking from the Csound GitHub site (<http://csound.github.io>), where readers will also find the latest versions of the software sources, links to the release packages for various platforms, the Csound Reference Manual, and many other resources. In particular, we would like to mention the Csound FLOSS Manual, which is a community effort led by two of the authors of this book, covering a number of practical and platform/frontend-specific aspects of Csound operation that are beyond the scope of this book. The Csound Journal, a periodic online publication co-edited by another one of the book authors, is also an excellent resource, with articles tackling different elements of the system.

It is also important to mention here that the essential companion to this book, and to the use of Csound, is the Reference Manual. It is maintained to contain a complete documentation of all aspects of the system, in a concise, but also precise, manner. It is important for all users to get acquainted with its layout, and how to find the required information in it. The manual represents a triumph of the collaborative effort of the Csound community, and it contains a wealth of knowledge about the system that is quite remarkable.

The development and maintenance of Csound has been led by a small team of people, including three of the book authors, plus Michael Gogins, who made numerous contributions and has kept the Windows platform versions up to date in a very diligent way, and Andrés Cabrera, who also authored a widely used cross-platform IDE for Csound (CsoundQt). As Free software, Csound is fully open for users to play with it, fork it, copy it and of course, add to it. Over its thirty-odd years of existence, it has benefitted from contributions by developers spread all over the world, too many to be listed here (but duly acknowledged in the source code and manual).

At some point, users will realise that Csound can crash. It should not, but it does. The developers are always looking out for flaws, bugs and unprotected areas of the system. We have minimised the occurrence of segmentation faults, but as a programming system that is very flexible and produces 'real', compiled, working programs, it is vulnerable to these. No matter how closely we look at the system, there will always be the chance of some small opportunity for perverse code to be used, which will bring the system down. The development team has introduced a number of safeguards and a very thorough testing program to keep the software well maintained, bug free, and defended from misuse.

An important part of this is the issue tracking system, which is at present handled by the GitHub project. This is a very useful tool for us to keep an eye on problems that have arisen, and the user community is our first line of defence, using and pushing the software to its limits. Reporting bugs, and also asking for features to be implemented, is a good way to help strengthen Csound for all users. The developers work to a reasonably tight schedule, trying to address the issues as they are reported. Once these are fixed, the new code is made available in the develop branch of the

source code revision system (`git`). They become part of the following software release, which happens quite regularly (three or four times yearly).

We hope that the present book is a helpful introduction to the system for new users, and a valuable companion to the ones already acquainted with it.

Maynooth, Rochester, Bath,
Hannover, Trondheim, and Berlin
December 2015

Victor Lazzarini
Steven Yi
John ffitch
Joachim Heintz
Øyvind Brandtsegg
Iain McCurdy

Csound

A Sound and Music Computing System

Lazarini, V.; Yi, S.; Fitch, J.; Heintz, J.; Brandtsegg, Ø.;

McCurdy, I.

2016, XXX, 516 p. 125 illus., 12 illus. in color.,

Hardcover

ISBN: 978-3-319-45368-2