Contents

1 Getting started with SuperCollider 1
1.1 About SuperCollider 1
1.2 SC overview 3
1.3 Installation and use 6
1.4 Objectives, references, typographical conventions 9

2 Programming in SC 13
2.1 Programming languages 13
2.2 Minima objectalia 16
2.3 Objects in SC 19
2.4 Methods and messages 26
2.5 The methods of type `post` and `dump` 33
2.6 Numbers 36
2.7 Conclusions 39

3 Syntax: basic elements 41
3.1 Brackets 41
3.2 Expressions 42
3.3 Comments 43
3.4 Strings 44
3.5 Variables 45
3.6 Symbols 49
3.7 Errors 51
3.8 Functions 52
3.9 Classes, messages/methods and keywords 57
3.10 A graphic example 59
3.11 Control Structures 64
3.12 Yet another GUI example 67
3.13 Conclusions 71

4 Synthesis, I: Fundamentals of Signal Processing 73
4.1 A few hundred words on acoustics 73
4.2 Analog vs. digital 76
4.3 Synthesis algorithms 81
4.4 Methods of `Signal` 89
4.5 Other signals and other algorithms 93
4.6 Still on signal processing 105
4.7 Control signals 109
4.8 Conclusions 117

5 SC architecture and the server 119
5.1 Client vs. Server 119
5.2 Ontology of the server as an audio synthesis plant 123
5.3 The server 132
5.4 SynthDefs 134
5.5 UGens and UGen graphs 138
5.6 Synths and Groups 144
5.7 A theremin 149
5.8 An example of real-time synthesis and control 151
5.9 Expressiveness of the language: algorithms 154
5.10 Expressiveness of the language: abbreviations 156
5.11 Conclusions 160

6 Control 161
6.1 Envelopes 161
6.2 Generalizing envelopes 167
6.3 Sinusoids & sinusoids 174
6.4 Pseudo-random signals 180
6.5 Busses 184
6.6 Procedural structure of SynthDef 197
6.7 Multichannel Expansion 201
6.8 Conclusions 208

Introduction to SuperCollider

7 Organized sound: scheduling 211
7.1 Server-side, 1: through UGens 211
7.2 Server side, 2: Demand UGen 217
7.3 Language-side: Clocks and routines 221
7.4 Clocks 226
7.5 Synthesizers vs. events 229
7.6 Graphic interlude: drawings and animations 236
7.7 Routines vs. Tasks 239
7.8 Patterns 243
7.9 Events and Event patterns 252
7.10 Conclusions 260

8 Synthesis, II: introduction to basic real-time techniques 261
8.1 Oscillators and tables 261
8.1.1 Synthesis by sampling 269
8.1.2 Resampling and interpolation 271
8.2 Direct generation 275
8.2.1 Synthesis by fixed waveform 276
8.2.2 Modulation 277
8.2.3 Ring and Amplitude modulation 278
8.2.4 Ring modulation as a processing technique 280
8.2.5 Frequency modulation 285
8.2.6 C:M ratio 288
8.2.7 Waveshaping 292
8.3 Spectral modelling 297
8.3.1 Additive synthesis 297
8.3.2 Subtractive synthesis 305
8.3.3 Analysis and resynthesis: Phase vocoder 311
8.4 Physical Modeling 319
8.5 Time-based methods 327
8.5.1 Granular synthesis 328
8.5.2 Techniques based on the direct generation of the waveform 331
8.6 Conclusions 335

9 Communication 337
9.1 From server to client: use of control buses 337
9.2 From server to client: use of OSC messages 341
9.3 OSC to and from other applications 346
9.4 The MIDI protocol 349
9.5 Reading and writing: File 353
9.6 Pipe 359
9.7 SerialPort 362
9.8 Conclusions 365

Foreword

My first encounter with SuperCollider dates back to 2002, thanks to my friend Hairy Vogel, one of its first (and bold) users. At that time, I remember being deeply impressed by the sound quality, the seamless integration between audio synthesis and algorithmic composition, the native real-time operating mode. So, I decided to buy an Apple computer, as SuperCollider 2 was working on MacOS9 systems only, while SuperCollider 3, that had just become open source, was still an unstable release, porting to Linux had just begun (as far as I can remember) and there were no plans at all for a Windows version...But in 2002 I was lacking some basic skills to be productive on SuperCollider, so I left it momentarily, to get back to it in 2005. In 2008 I switched to SuperCollider in my class for the Multimedia and Arts program (DAMS) at the University of Turin. I was not able to find a basic, yet comprehensive resource to use for teaching. Comprehensive of what? This is of course the main point. My class is a basic introduction to computer music, so I was in need of reference for both some basic notions in DSP *and* SuperCollider. Still, I was confident that the latter was the apt tool to approach the former. Yet, introducing SC is to also introduce computer languages in general and certain notions concerning computer science *tout court* (e.g. the client/network architecture). That is why I decided to write this book the scope of which is, consequently, too broad to offer completeness in any of each subjects. From a very personal perspective, I have to confess that I have tried to write the book that I wished to have while starting with SuperCollider. My hope is that someone might be in need of it as I was.

Originally written in Italian in 2008 in an electronic only version, the book has been substantially re-edited and updated to account for the SC 3.7 version in 2015, and then, revised and translated into English in 2016 to be published in a printed version thanks to Logos Verlag. It has been translated into English by myself and Marinos Koutsomichalis, thanks also to the support of the SuperCollider community. A crucial help on language and content revision has been provided by Joe Higham and Joshua Parmenter. I am deeply thankful to Marinos, Joe and Josh for their work.

SuperCollider has a strong, supporting and passionate community. Without it, my journey with SC probably would have ended very soon. So, I am profoundly indebted to all its members. Finally, a special “grazie” to James McCartney to have his ideas supercollide in his work.

Ciriè, Turin, May 27, 2016