Loadbang

Programming Electronic Music in Pd

Johannes Kreidler

wolke

Second Edition 2013
© Johannes Kreidler
All rights reserved by the publisher Wolke Verlag, Hofheim
Translation: Mark Barden
Cover design: Friedwalt Donner and pd-graz
Typesetting: michon, Hofheim
Printing: Fuldaer Verlagsanstalt
ISBN 978-3-95593-055-4

Table of Contents

Preface

This book is the result of my experience of teaching electronic music. Through the teaching process, I became familiar with the most common stumbling blocks students encounter—especially when the student's native language is not the language in which lessons are conducted.

Pd (Pure Data) is a professional, high-performance programming language for electronic sound processing. It is *open source*, i.e. available for free on the Internet. One disadvantage of this is that Pd is only discussed in certain institutions or Internet forums. The complicated technical terminology usually found there is enormously difficult for beginners to understand. This book will help first-time users to clear those first few hurdles when learning Pd.

Pd's main designer, Miller Puckette, is also writing a book about the theory and technology of electronic music processing with Pd. Surely there is no better teacher of a programming language than the person who designed it; his primarily scientific approach certainly does cover all the material in a thorough, systematic fashion. However, his method of teaching can be difficult to comprehend. My pedagogical experience has been that Puckette's text demands a large amount of mathematical, computer science, and terminological knowledge from its readers.

This book is designed for self-study, principally for composers. It begins with explanations of basic programming and acoustic principles before gradually building up to the most advanced electronic music processing techniques. Some knowledge of physics is assumed and explanations of basic physics concepts have been intentionally omitted. My book's teaching approach is focused primarily on hearing, which I regard as a faster and more enjoyable way to absorb new concepts than through abstract formulas. In terms of mathematics, I explain only what is absolutely necessary to comprehend a given processing concept. I explain the various techniques from a compositional perspective, rather than attempting a computer science-, math-, or physics-based discussion of processing phenomena or structures. Therefore, the decisions and comments I have made are purely subjective and are open to debate.

This book would not have been possible without the support of Prof. Mathias Spahlinger, the expert supervision of Prof. Orm Finnendahl, suggestions and patches from the Pd community, the manuscript editing and DocBook-XML coding efforts of Esther Kochte. I would also like to thank Mark Barden for the English translation and the Musikhochschule Freiburg and the state of Baden-Württemberg for financing the project. This book will hopefully increase interest in electronic music, thereby indirectly enriching the aesthetic discourse of New Music.

Johannes Kreidler, January 2008

Introduction to this book's methodology

The following material begins with basic computer knowledge. The first steps are therefore described in meticulous detail.

Pd can run on different platforms (like Linux, OS X, or Windows) and this book is not platform-specific. Problems relating to the operating system will not be discussed, as they are simply beyond the scope of this tutorial (and it is also quite likely that changes—updates, bug fixes, etc.—will occur in the near future). It is therefore assumed that Pd has been correctly installed and has been integrated with the hardware environment (consult an Internet forum to resolve any of these sorts of problems, e.g. "Pd-list").

How to use this book: Each lesson is comprised of a theory part, a practice part and an appendix, as well as individual aspects that are explained in greater detail at the end of each section. This in-depth information is aimed at advanced users and is not essential to acquire a basic working knowledge of Pd. I recommend working through the whole book without consulting these additional details first, then going back to learn them later.

Now and again, some fundamental concepts of acoustics are discussed. The exercises contain not only specific compositional questions, but also applications that are useful for musicians' everyday needs—e.g. tools like the metronome or tuning device. In this respect, the tutorial could be used by interpreters as well as composers.

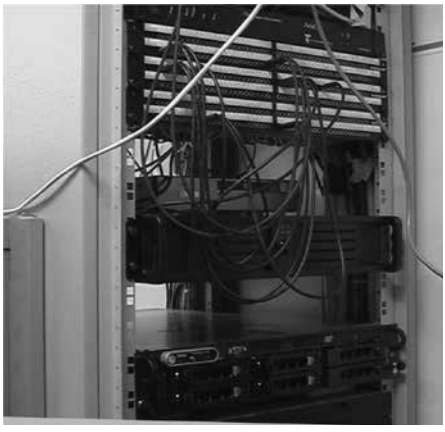
Chapter 1
Introduction to Pd

1.1 General remarks

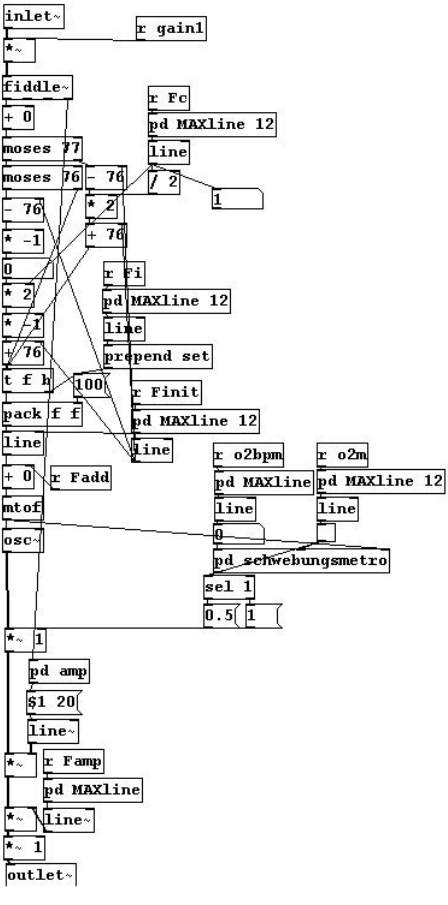
Pd (Pure Data) is a programming language for electronic music. Creating music on a computer is technically referred to as DSP (digital signal processing). "Digital" means that information is represented by digits—computers, as you may know, work only with numbers. "Signal" is the technical term for a special mode of computer operation that deals with sound. "Processing" refers to functions executed by the computer.

Pd was initiated by American software engineer Miller Puckette, who previous co-developed the well known and similarly structured software Max/Msp. Pd is not commercial software; i.e., it was not developed by a corporation and is not for sale. Instead, it is "open source": its source code can be viewed by anyone. This source code is also not the (patented) property of a corporation, but is rather freely available to all. This also means that, provided sufficient knowledge, anyone can change the program. Today, many other programmers, musicians, acoustic engineers, and composers have joined Miller Puckette to continue Pd's development. As a result of this, there is no final, definitive version of Pd; the program is under constant development. In addition to the huge advantage of free availability on the Internet, it is also "democratically" expanded and optimized on a professional level. One drawback to this is that a detailed operating manual for users who lack programming experience has not existed until now. In contrast to a corporation, which has a monetary interest in ensuring that first-time users can easily operate new software, the open source movement lacks such a driving force to make itself accessible. This book is an attempt to fill that gap.

In precise terms, Pd is a "real-time graphical programming environment for audio processing". Traditionally, programmers work with text-based programming languages. They create what is called "code", which is processed by a computer to produce a result. To carry out its programming functions, Pd uses visual objects that the user places and alters on the screen. These visual objects—small boxes that can be connected to each other—are a throwback to analogue studios that were used to produce electronic music before the advent of computers: various devices—today symbolized by our little boxes—are connected to each other using lines that—like cables—symbolize physical connections between the boxes. (Due to this type of connection, Pd is referred to as a datastream-oriented programming language.)



An analog studio—devices are connected with cables.



Pd boxes are connected to each other.

One major advantage of Pd is the aspect of „real-time“. This means that, in contrast to most programming environments where a text is first entered that must be separately processed by the computer before obtaining a result, changes in Pd can be made during performance. Like on a classical instrument, the user hears the result instantaneously and can change it immediately. This makes Pd especially well suited for use in live performance.

Pd has become much more than a programming language for electronic music. Since users across the globe can participate in the project, there are user-programmed modules for what are called „externals“: video, Internet connection, joystick integration, etc. Whole libraries of these modules even exist („external libraries“). Some of these externals have been integrated into the regular version of Pd.

1.2 Installing and setting up Pd

Readers of this book should have Pd installed on their computer so they are able to try out the processes described. Without this simultaneous practical experience, this tutorial will be difficult to understand.

First you need a computer with at least 128 MB main memory, a 500 MHz processor and ca. 500 MB hard disk space (these are the absolute minimum requirements!). Pd works with the following operating systems: Linux, OS X, and Windows.

Then you need to download the newest version of Pd-extended from the Internet. Enter "Pd-extended" into an Internet search engine. Since the address for the download portal may change in the future, no link to the site will be provided here. Pd-extended is a version of the original software (also called "Pd vanilla") that has been expanded with numerous libraries. Most of the exercises described here work with the original version of Pd, but not all of them. The extra objects in Pd-extended make the program much more practical in general. This tutorial assumes Pd-extended version 0.39 or higher.

Once Pd has been installed, we open it from the directory Pd/bin/. A window appears. This is the main control center, so to speak. Here you can test whether Pd is functioning properly: In the main menu, click on **Media → Test Audio and MIDI**. Under "TEST SIGNAL", click first on the box next to "-40", then on the box next to "-20". You should hear a sine tone coming out of the computer's loudspeaker (A4). If you do not, then you need to adjust your hardware settings (under **Media → Audio settings**). More information regarding problems that arise at this stage cannot be given here. For help resolving any problems, please consult the "Pd-list", a forum of Pd users on the Internet. If a microphone is connected, the digits in at least the leftmost two boxes under "AUDIO INPUT" should change in response to sound picked up by the microphone. As long as the test tone is working, you can work with the program without a microphone. (By Chapter 3 at the latest, however, you will sometimes need a microphone.)