

Table of Contents

Bang: Release the Robot Legions of Terror 1

Bang: Release the Robot Legions of Terror

By Andy James Farnell

Interest in the rapidly developing visual programming language Pure Data, the "crack cocaine" of multimedia software, is growing. In this review of *bang*, the book about Pure Data published as part of the International Pd Convention held in Graz 2005, Andy Farnell momentarily puts down the pipe to focus on the phenomenon and the scene around it

This book is attractive. Like finding a wallet lying in the park it requests your attention. With only the word *bang* written in large black letters on a red cover, you are simply compelled to open the book, if only to make sure it's safe. What first appeared to be an attempt on my life by the clown mafia, turned out to be a pleasant 170 pages of intelligent and challenging discussion on multimedia software.

Visually engaging, with lots of graphs, diagrams, odd lines of code and equations scattered throughout the text, it's full of useful information but not hard work. Colour is reserved for photos of the many contributors at various events and gigs, a nice reality anchor on an otherwise sky high journey through the possibilities of using a general purpose real-time data processing tool to make arbitrary user definable applications. And why would you want to do that?

Â Â Â [IMAGE]

Well if you're interested in computers, particularly making music, animations or interactive programs and gadgets then you need a tool to do it with. Of all the ways to accomplish your goals a visual programming language is the most accessible and expedient. On the face of it a visual programming language seems like just another application, one where you plug little boxes together, but it has the ability to transform itself into any other new application you might want. This is empowering, because instead of using off the shelf DJ and VJ applications you just build your own. One such visual language which is very easy to use, and is the subject of this book, goes by the name Pure Data, or just Pd. For the uninitiated it is the crack cocaine of multimedia development software. Sure, everybody dabbles in a little Cubase or 3DMax from time to time, but this is the stuff for the shock ads with before and after photos. It's addictive because it's productive. In just a few minutes you can achieve rewarding results.

So, what is Pure Data? Is Pure Data for me? Where can I get it? Who else is using it, and where can I find out more and get support?

Pure Data is an open ended signal processing tool with which you can build almost any musical or visual application. For those who have reached the creative limits of software like Cubase, Reaktor and Reason it is the next step. It is free open source software, which means you can legally download it for no cost, modify and republish derivatives, and use it safe in the knowledge that it comes without hidden malware, digital restrictions management (DRM) or crippled functionality like copy protection code that will damage your computer system. Versions are available for Mac OSX, Microsoft Windows and Linux, and there are development versions of a lighter Pda for hand-held and mobile devices, so you can turn your iPod into a drum machine or mp3 mixer. The original Pure Data is still actively developed by it's original author Miller Puckette at the Center for Research in Computing and the Arts. You can download the official latest build from <http://crca.ucsd.edu/~msp/software.html>, though there are spicier versions with add-ons such as the Desire-Data interface and pd-extended maintained by Hans Chistoph Steiner.

Another attraction is that it's a real-time environment, which means you see or hear changes right away. Some programmers become so adept at working with running code they are able to perform with it as *live coding*, building and modifying applications on the hoof in front of an audience. But it's more than a DJ tool, people on Pd tend to do unusual things, like build internet radio stations that broadcast from their iPod or automate their cat feeder so it can recognise a hungry meow sound. A dream for artists is integrating media and Pd and is possibly the most open technological playground for techno-artists you will find. Surreal ideas like interactive art installations that read your facial expressions with a camera and have the paintings smile or frown back at you, or musical games built with dance mats and beam sensors become possible. It's a very enabling development software.

[IMAGE]

Oddly, each section of the book is started by a random page from the Pd manual, so you find yourself picking up snippets of wisdom along the way. But *Bang* is not a manual, nor is it a narrative of multimedia development, or an analysis of current art/tech culture. It is accidentally all of these things while being a fairly disparate collection of articles, explanations and commentary by Pd practitioners and developers. Together they create a strong image of the art and technology avant garde, a culture where experimentalism extends to defining the tools and the media itself, and where the result extends into the reality of home and community in a way that mass media signals simply cannot.

A few of the sections are quite weighty. I was transported back to computer science undergrad classes on generative grammars to grapple with one piece, but most parts are a light and fun read and would be enjoyable to anybody with a basic appreciation of music and video software. Some O level physics is needed to enjoy one discussion of physical modelling. But rather than going into the dreary equations of general physmo principles it's an eloquently written and exciting overview of a specific application, that of getting DJ faders to respond in a nicer way to movement. Pd isn't just about music and video signals, it's about all kinds of signals that fall within your computers bandwidth, so no problem building that hydroponic system for your tomatoes using a Pd program, or implementing the ultimate model railway control system. Sadly there were few words in this book on physical computing, certainly not enough for my liking. The richness of the physical interface of Pd is one of its enormous strengths, there are so many input output paths easily available for you to rig up joysticks and game-controllers, and with USB and network drivers for PIC I/O boards you can control your robot legions of terror.

Other parts of the book provide interesting historical background material on the evolution of computer music languages and environments. Exploring the development of early languages and synthesisers we see how flexibility and usability have changed over the years. Going back over the work of the computer music pioneers and looking at old systems like MusicV and Csound sets a context for Pd. Open source development philosophy is discussed from time to time too. I think it is important to remember just how far we have come with user definable software and the free open source movement. Before you had a tough choice. There was the world of expensive or unlicensed applications, defined somewhat by their lack of interoperability, and each requiring an investment of time to learn its interface. Or there was learning a real language like C or Java and how to program the hard way. Pd is a pleasant middle ground for the rapid development of one of a kind or prototype programs and with it comes a *can do* attitude to connecting anything to anything else and quickly plugging together new tools for any fresh idea. Want a video clip sequencer with a midi drum pad controller? No problem, just place a few of the right objects on the screen, wire em up and off you go. Well, that's the theory. In practice it's almost that easy, which is amazing, but Pd is a program in development itself and faces many challenges to making this process work perfectly. The book explains some of this development process and makes no secrets of the research issues like storage, persistence, representation and other challenges in developing, and working with Pd.

We are never far from the fact that Pd is about DSP, which is another word for hard sums and programming. Throughout the text the scientific background of Pd is ever present and we are reminded that it is also a powerful tool for serious real-time data processing, rapid prototyping and experimentation. But there are entirely artistic discussions of Pd, such as one on synaesthesia, being spot on the mark identifying Pd a most powerful environment for integrating audio and visual data. Although the search for synaesthetic experience may be misguided the project bodes well for exploring new visual metaphors, things for making new interfacing gizmos and finding fresh visualisations of data to represent complex dynamics.

The book, and Pd itself are suitable for casual geek and serious hacker alike. If code is your work then Pd gives the mind a way to unwind and program in a fun way again, but if you know almost nothing about programming or mathematics you can still enjoy Pd as an artistic tool. According to some descriptions, the mix of philosophy, computer science, psychology and culture in *bang* leaves it contradictory and provoking more questions than it answers. I don't think it is, or at least, I think the answers are implied between the lines of this collection. It's fragmented perhaps, but none of the pieces are at odds with the others, they are just parts of a very interesting bigger picture of software and art evolution.

Pure Data is rapidly becoming the platform of choice to teach digital arts, particularly on quality courses that deal with the essential principles rather than simply demonstrating the use of a set applications. Although adoption by mainstream media production houses is still slow, Pure Data is creating an enthusiastic buzz across the cutting edge of the technological arts underground and users include producers, bands, radio stations, theatres and games developers to mention only a few. *bang* is the first publication that examines this phenomenon and tries to make it accessible to the average reader. While this book will not serve as an instruction manual on how to use Pure Data, you can find a great abundance of documentation online and within the help pages of Pure Data itself

bang

**Pure Data 1. International PD-Convention Graz,
Frank Zimmer (ed.), Wolke, 2006, ISBN-10: 3-936000-37-9**

Andy Farnell, <<http://obiwannabe.co.uk/>> is a computer scientist with 15 or so years experience with digital sound working as a programmer, producer or developer with audio-visual software