SPEAKING CODE

CODING AS AESTHETIC AND POLITICAL EXPRESSION

TEXT: GEOFF COX + CODE: ALEX MCLEAN + FOREWORD BY FRANCO "BIFO" BERARDI

Speaking Code

Software Studies

Matthew Fuller, Lev Manovich, and Noah Wardrip-Fruin, editors

Expressive Processing: Digital Fictions, Computer Games, and Software Studies,
Noah Wardrip-Fruin, 2009

Code/Space: Software and Everyday Life,
Rob Kitchin and Martin Dodge, 2011

Programmed Visions: Software and Memory,
Wendy Hui Kyong Chun, 2011

Speaking Code: Coding as Aesthetic and Political Expression,
text: Geoff Cox; code: Alex McLean, 2012

Speaking Code

Coding as Aesthetic and Political Expression

text: Geoff Cox

code: Alex McLean

foreword by Franco "Bifo" Berardi

The MIT Press
Cambridge, Massachusetts
London, England

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 55 Hayward Street, Cambridge, MA 02142.

This book was set in Stone Sans and Stone Serif by Toppan Best-set Premedia Limited. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Cox, Geoff.
Speaking code : coding as aesthetic and political expression / text [by] Geoff Cox, code [by] Alex McLean, foreword by Franco "Bifo" Berardi.
p. cm.—(Software studies)
Includes bibliographical references and index.
ISBN 978-0-262-01836-4 (hardcover : alk. paper)
1. Source code (Computer science)—Philosophy. 2. Programming languages (Electronic computers)—Syntax. 3. Computer prose. I. McLean, Alex (Christopher Alex), 1975– II. Title.
QA76.167.C69 2013
005.1—dc23
2012012944

10 9 8 7 6 5 4 3 2 1

Contents

Series Foreword

Software is deeply woven into contemporary life—economically, culturally, creatively, politically—in manners both obvious and nearly invisible. Yet while much is written about how software is used, and the activities that it supports and shapes, thinking about software itself has remained largely technical for much of its history. Increasingly, however, artists, scientists, engineers, hackers, designers, and scholars in the humanities and social sciences are finding that for the questions they face, and the things they need to build, an expanded understanding of software is necessary. For such understanding they can call upon a strand of texts in the history of computing and new media, they can take part in the rich implicit culture of software, and they also can take part in the development of an emerging, fundamentally transdisciplinary, computational literacy. These provide the foundation for Software Studies.

Software Studies uses and develops cultural, theoretical, and practice-oriented approaches to make critical, historical, and experimental accounts of (and interventions via) the objects and processes of software. The field engages and contributes to the research of computer scientists, the work of software designers and engineers, and the creations of software artists. It tracks how software is substantially integrated into the processes of contemporary culture and society, reformulating processes, ideas, institutions, and cultural objects around their closeness to algorithmic and formal description and action. Software Studies proposes histories of computational cultures and works with the intellectual resources of computing to develop reflexive thinking about its entanglements and possibilities. It does this both in the scholarly modes of the humanities and social sciences and in the software creation/research modes of computer science, the arts, and design.

The Software Studies book series, published by the MIT Press, aims to publish the best new work in a critical and experimental field that is at once culturally and technically literate, reflecting the reality of today's software culture.

Foreword: Debt, Exactness, Excess

Franco "Bifo" Berardi

Pragmatics, or the future of language

Language and the future: this is the subject of this book. Not the rhetorical future of history, politics, and so on, but the pragmatic future of the next second, of the next minute, of the next action and interaction. Future and interaction: this is the task (or the destiny) of code. So code is "speaking" us. Geoff Cox is trying to show the other side of the moon: if we can say that code is speaking us (pervading and formatting our action), the other way round is also true. We are speaking code in many ways. In the beginning someone is writing the code, and others are supposed to submit themselves to the effects of the code written by someone. Power is more and more inscribed in code. Writing to Thomas Sebeok, Bill Gates once remarked that "power is making things easy" (quoted by Arthur Kroker and Michael A. Weinstein in *Data Trash*, 1994). Code and interfaces: interfaces are supposed to make the complexity of the code easy, but code in itself is more often about simplifying technical procedures of social life, particularly of economic production and exchange. So code is speaking us, but we are not always working through the effects of written code. More and more we are escaping (or trying to escape) the automatisms implied in the written code.

Prescriptions, prophecies, injunctions are ways of inscribing the future in language, and—most importantly—are ways of producing the future by means of language. Like prescriptions, prophecies, and injunctions, code also has the power to inscribe the future, by formatting linguistic relations and the pragmatic development of algorithmic signs. Code is modeling the future, as the future is inscribed in code. In fact the implementation of code is performing our environment, our behavior. Code is prescribing what we will do to the machine and what the machine will do to us. Financial code, for instance, is triggering a series of linguistic automatisms that are able to model and perform social activity, consumption patterns and lifestyles. Algorithms are numeric combinations that inscribe in themselves operational functions, formatting and performing the real developments of the human world.

But the pragmatic effects of the code are not deterministic, as far as the code is the product of code writing, and code writing is affected by social, political, cultural, and emotional processes. This is a key point that is highlighted in this book. Hacking, free software, WikiLeaks . . . are the names of lines of escape from the determinism of code.

From this point of view, *Speaking Code* is a timely intervention. It again raises the question of pragmatics previously debated by authors like J. L. Austin (*How to Do Things with Words*, 1962), Paul Watzlawick (*Pragmatics of Human Communication*, 1967), and Félix Guattari (*L'inconscient machinique*, 1979), but in this book, for the first time, the problem of pragmatics is investigated at the level of code. The pragmatics of code is opening the door to a flow of urgent questions directly coming from the field of current events and present history. Think of the financial crisis and the tremendous social effects it is provoking: if we want to understand something about this crisis that is haunting Western politics and daily life we have to focus on the pragmatics of code.

According to Robert J. Sardello, in *Money and the Soul of the World* (1983), money and language have something in common: they are nothing and yet affect everything. They are nothing but symbols, conventions, *flatus vocis*, but they have the power of persuading human beings to act, to work, and to transform physical things. "Money makes things happen. It is the source of action in the world and perhaps the only power we invest in. Perhaps in every other respect, in every other value, bankruptcy has been declared, giving money the power of some sacred deity, demanding to be recognized." Language is nothing, like money. But like money, language can do anything, up to the point that, as Martin Heidegger suggests, "what we know of Being is what can be elaborated by Language." Language and money are shaping our future in many ways. Like prophecy.

Prophecy is a form of prediction that is able to act on the development of the future thanks to the effect of persuasion, emotionality, and pervasion of the audience. Thanks to the social effects of psychological reactions to language, prophecy can act in a self-fulfilling way. The financial economy is marked by self-fulfilling prophecies. When rating agencies downgrade the value of an enterprise, or national economy, what they make is a prediction about the future performances of the enterprise, or economy. But this prediction is heavily influencing the actors of the economic game, so much so that the downgrading results in an actual loss of reliability and an actual loss of economic value. So the prophecy comes true. How can we escape the effects of prophecy? How can we escape the effects of code? Of course these are two different problems, but they have very much in common.

The limits of language

In his latest book (*E così via, all'infinito*, 2010) Paolo Virno again questions the problem of the limits of language. Rereading Ludwig Wittgenstein's assertion that the limits of

my language are the limits of my world, Virno looks for a means of escape from the determinist effects of the implied limitations of language. The linguistic excess, namely poetry, art, and desire, are conditions for the overcoming and the displacement of the limits that linguistic practice presupposes.

In the preface to his *Tractatus Logico-Philosophicus* (1922), Wittgenstein writes: "In order to draw a limit to thinking we should have to be able to think both sides of this limit (we should therefore have to be able to think what cannot be thought)." And he writes: "The limits of my language are the limits of my world. Logic pervades the world: the limits of the world are also its limits. So we cannot say in logic, 'The world has this in it, and this, but not that.' For that would appear to presuppose that we were excluding certain possibilities, and this cannot be the case, since it would require that logic should go beyond the limits of the world; for only in that way could it view those limits from the other side as well. We cannot think what we cannot think; so what we cannot think we cannot say either." And finally he writes: "The subject does not belong to the world: rather, it is a limit of the world."

When Wittgenstein says that the limits of language are the limits of the world, we should read this in two different ways. First he is saying: what we cannot say we cannot do, we cannot experience, we cannot live, as only in the sphere of language can we interact with the reality of Being. But he is also saying: as the world is what resides inside the limits of our language, therefore beyond the limits of language lies what we will be able to live and experience only when our language will be able to elaborate that sphere of Being that lies beyond the present limit. In fact he writes: "The subject does not belong to the world, rather it is a limit of the world." The potency and extension of language depends on the consistency of the subject, on its vision, on its situation. And the extension of my world depends on the potency of my language.

The process of going beyond the limits of the world is what Guattari calls "chaosmosis," and he refers to this going beyond as "re-semiotization," i.e., the redefinition of the semiotic limit, which is also the limit of the experience of the world. Scientists call this effect of autopoietic morphogenesis "emergence": a new form comes out and takes shape when logical-linguistic conditions make it possible to see it, and to name it. Let's try to understand our present situation from this point of view.

Digital financial capitalism has created a closed reality, which cannot be overcome with the technicalities of politics, of consciously organized voluntary action and of government. Only an act of language may give us the possibility of seeing and of creating a new human condition where now we only see barbarianism and violence. Only an act of language escaping the technical automatisms of financial capitalism will make possible the emergence of a new life form. The new form of life will be the social and desiring body of the general intellect, the social and desiring body that the general intellect is deprived of, under the present conditions of financial dictatorship. Only the reactivation of the body of the general intellect—the organic, existential,

historical finitude that is bringing in itself the potency of the general intellect—will be able to imagine new infinities. Only in the intersection of the finite and infinite, in the point of negotiation between complexity and chaos, will it be possible to disentangle degrees of complexity higher than that which financial capitalism is able to manage and elaborate.

Language is an infinite potency, but the exercise of language happens in finite conditions of history and existence. Thanks to the establishment of such a limit, the world comes to exist as a world of language. Grammar, logic, ethics are all based on the institution of a limit. Code is a limited exercise of language and simultaneously it is the institution of a (performing and productive) limit. Limits can be productive, but outside of this limited space the infinite possibilities of language persist immeasurably.

Code implies syntactic exactness of linguistic signs: connection. Compatibility and consistency and syntactic exactness are the conditions of operational functionality of code. Proprietary code is language in debt. Only exacting the necessary syntactic consistency, language can perform its connective purpose. The excess is the *remise en question* of the infinity of language, the breakdown of consistency, the reopening of the horizon of possibility. Excess is playing the game of conjunction (round bodies looking for meaning out of any syntactic exactness), not the preformatted game of segmental connection.

Poetry is the reopening of the in-definite, the ironic act of exceeding the established meaning of words. In every sphere of human action, grammar is the establishment of limits defining a space of communication. The economy is the universal grammar traversing the different level of human activity nowadays. Also language is defined and limited by its economic exchangeability: in the reduction of language to information, and incorporation of techno-linguistic automatisms in the social circulation of language. Nevertheless, whereas social communication is a limited process, language is boundless: its potentiality is not limited to the limits of the signified. Poetry is the excess of language, the signifier disentangled from the limits of the signified. Irony, the ethical form of the exceeding power of language, is the infinite game that words are playing to create and to skip and to shuffle meanings. Social movements, at the end of the day, can be viewed as ironic acts of language, as semiotic insolvency, as the disentanglement of language, behavior, and action from the limits of symbolic debt. In this book a question is raised: is it possible to speak of code as a movement, as a form of subversion and of redefinition of the limits of language?

Preface

The interplay of text and code that runs through this book underlines the principle that speech says what it will do and does it. Two voices are evident in its pages: one that tends toward the tradition of critical writing in the arts and humanities, and another that derives more from the tradition of computing and software development. Together they interweave through the chapters, formulating arguments that aim to undermine the distinctions between criticism and practice, and that emphasize the aesthetic and political implications of software studies. A collaborative process lies at the heart of this, one that oscillates between expressive and formal conventions of writing. All the same, one can broadly generalize that the text has been written by myself (with the exception of chapter 1, which is co-written) and the code by Alex McLean (although again there are some exceptions). To separate these may help to identify the voices of the work, but the distinction itself should be underplayed between the authors, and even more importantly between the modes of expression in use and the subject disciplines that are invoked. The combinatory aspect also indicates the central importance of the writing subject in meaning production and registers code as an active agent in the process, which further complicates any reductive tendencies in human and machine reading and its interpretation. The cultural meanings generated are understood not as derived from intentionality or source code as such, but from the complex interplay of forces involved in the encoding and decoding of texts and programs.

Consequently the book's ideas are hard to trace or attribute with any certainty. However, it should be acknowledged that some of the chapters rework aspects of previous publications and talks, as well as previous conversations with students and colleagues at Aarhus University, the University of Plymouth, and Transart Institute, where I have been working in recent years. I have tried to indicate these where they are overt, but the ideas inevitably derive from various other outputs in complex and even unknown ways. This reflects the ways that all ideas are borrowed in some sense, and also reflects the normalization of montage techniques that are somewhat prescribed when using a computer to write. With this in mind, I should acknowledge that

throughout the early chapters there is also a faint echo of my PhD thesis, "Antithesis: The Dialectics of Software Art," submitted in 2006 to the University of Plymouth and published by the Digital Aesthetics Research group at Aarhus University in 2010. More specifically, and also cited in endnotes, chapter 1 makes reference to previous collaborative essays with Alex McLean and Adrian Ward, "The Aesthetics of Generative Code" (2000) and "Coding Praxis" (2004). Chapter 2 contains some reworking of the unpublished paper "Software Art Has No History," presented at re:place, Second International Conference on the Histories of Media, Art, Science and Technology, Haus der Kulturen der Welt, Berlin (2007); as well as drawing on ideas from "Means-End of Software," in *Interface Criticism*, edited by Christian Ulrik Andersen and Søren Bro Pold (written in 2007, published by Aarhus University Press, 2011). Chapter 3 further draws upon writings produced in support of projects that I have commissioned for Arnolfini, a contemporary arts center in Bristol, UK, as part of my role there as associate curator of online projects, including "Antisocial Notworking" and "Democracy 2.0," as well as a recent related paper, "Virtual Suicide as Decisive Political Act," in *Activist Media and Biopolitics*, edited by Wolfgang Sützl and Theo Hug (University of Innsbruck Press, 2011). Chapter 4 shares some references with "Not Just for Fun," co-written with Alex McLean for the forthcoming book *Fun and Software*, edited by Olga Goriunova, and my short introduction to David Link's *Machine Heart/Das Herz der Maschine* for the "100 Notes—100 Thoughts" series published in connection with dOCUMENTA (13) (Hatje Cantz, 2011). I hope the fact that many of the texts relate to art projects that I have had an active part in developing (through my occasional curatorial work at Arnolfini) helps to assert one of the central principles of the book, namely that coding practices produce significant aesthetic and political effects.

It has already been stated that the majority of the code examples have been provided by Alex McLean, who at the time of writing was completing his PhD thesis entitled "Artist-Programmers and Programming Languages for the Arts" (submitted October 2011), Goldsmiths, University of London. His research includes an investigation into embodied approaches to text-based music within the broad frame of live coding, and so informs many of the ideas in the book, both implicitly and explicitly.

Finally, and not least, I would like to thank the following people who have helped in the formulation of ideas: Christian Ulrik Andersen, Roy Ascott, Tatiana Bazzichelli, Stuart Brisley, Andreas Broeckmann, Sean Cubitt, Matthew Fuller, Rui Guerra, Robert Jackson, Joasia Krysa, Les Liens Invisibles, Søren Bro Pold, Hugo de Rijke, Victoria de Rijke, Adrian Ward; and especially Franco "Bifo" Berardi for his foreword to the book. The anonymous reviewers of the first draft, and Matthew Abbate, Katie Helke Dokshina, Doug Sery, and others at the MIT Press, have also provided excellent advice throughout the process. Significant writing time has been made possible with support from the Center for Digital Urban Living (the Danish Council for Strategic Research,

grant 09-063245) and by Aarhus University's interdisciplinary research center Participatory IT, PIT. At a time when education and research activity are under attack by market forces, I am mindful that I have been working on the book in a country that still maintains that education should be free, and yet paradoxically has recently outlawed the existence of "free universities" (by this, I refer to autonomous institutions that call themselves universities). Such contradictions continue to underpin ideas of freedom of expression and the ongoing precarious conditions for the production and distribution of text and code.

For those interested, downloads, further notes, and ongoing discussion can be found at http://www.speaking-code.net.

Geoff Cox
Aarhus, Autumn 2011

While the schools drill human beings in speech . . . , the pupils become increasingly mute. They can give lectures; every sentence qualifies them for the microphone, before which they can be placed as spokesmen for the average; but their capacity for speaking to each other is stifled. It presupposes experience worth communicating, freedom of expression, and at once independence and relatedness. In the all-embracing system conversation becomes ventriloquism.

—Theodor W. Adorno, "Institute for Deaf-Mutes"

0 Double Coding

```
#!/usr/bin/befunge
>                    v
The  book begins  by  announcing  itself   recursively.  In stating  the
phrasev"Hello World!"<it  follows  the   convention that programmers adopt
whenv:<learning a new language. The   paragraph executes  itself in a way
that# encapsulates  the   inherent   action  of  code, and  the  central
importance  that will be   developed   around speech acts in  the world.
 v:,_@
 >  ^
```

Using the Befunge programming language to render a first paragraph like this sets the tone for the book in not just describing code and what it does but doing it. As an esoteric language, Befunge also breaks with the conventions of downward direction of interpretation through two-dimensional syntax.[1] This is done using punctuation, with each of the four instructions "^>v<" represented by graphical arrows, which change the direction of control flow. "@" ends the program.

The book begins with the "hello world" convention[2] to highlight the ubiquity of speech in everyday communications and the paradoxical tendency seemingly to diminish the power of the voice. Humans and machines increasingly converse with other humans and machines, making our languages ever more codified, but the meanings produced through them are ever more prone to misunderstanding—in the confused spaces between the encoding and decoding of the utterance. Such combinations of natural and artificial languages demonstrate a multilingual human-machine confusion of tongues, under the conditions of contemporary capitalism that have integrated language, intellect, and affect into production.

If in the beginning was the spoken word,[3] this appears to denote the human condition in the breath or essence of life. Underpinning this is the commonly held belief that the soul is the source of speech, which has been forever perverted through the use of spoken language. In the book of Genesis, the world was understood originally to contain one language only—the single language of Adam who first named objects

in the world. The story unfolds that the Tower of Babel, designed to reach into heaven, displeased God so much that he (sic) decided to "confound the language of all the earth."[4] Subsequently, everyone has been left to babble in a diversity of languages and confusion of human-machine "tongues," or in new hybrid forms that combine the formal structures of natural languages and program code. These constitute the contemporary babble of communications technologies[5]—resonant in the naming of the contemporary social media platform Twitter with its reductive register of up to 140 characters (condemning everyone to communicate in what is pejoratively called "twitspeak").[6]

Esoteric languages like Befunge, which introduced this chapter, seem to point in another direction altogether, opening up a more indeterminate and expressive space and transcending the production of simple effects or predetermined actions. Take, for another instance, the confounding effects of the esoteric Brainfuck programming language, which offers a challenge to normative source code interpretation by consisting entirely of punctuation, with each of the eight characters "><+-.,[]" representing a single elementary operation. "Hello world!" is expressed thus:

```
>+++++++++[<++++++++>-]<.>+++++++[<++++>-]<+.+++++++..+++.>>>++++++++[<++++>-
]<.>>>++++++++++[<+++++++++>-]<---.<<<<.+++.------.--------.>>+.
```

Taking this indeterminacy further still, Brainfuck exceeds the world of computation in Bodyfuck, an interpreter using computer vision techniques to map bodily gestures to the Brainfuck instruction set.[7]

But as with all signifying systems, interpretation still takes place at all levels, even when they are as esoteric as the examples mentioned above. The reader, whether human or machine, is also cast as one of the objects of the software and operating system. The point can be demonstrated with writing more generally as, in word-processing a text (like this), the writer is also processed into the choice of software and operating system that prescribes or allows certain tasks. It was in recognition of this issue that Friedrich Kittler apologized for his choice of software used to write the essay "There Is No Software."[8] The fact that the user's thoughts and actions are somewhat determined by the operating system or graphical user interface recalls the ways the user is interpellated in the Althusserian sense to demonstrate how ideology calls us to order through its God-like commands and procedures.[9]

This follows an understanding of code in a broad sense, insofar as it can be traced back to its etymological roots through "codicilla" (tablets used for inscribing letter forms) and "codex" (the bound book of the law), as Kittler explains in his entry to the *Software Studies* lexicon.[10] For Kittler, the references establish how code can be understood through the twin operations of command and control.[11] But in addition,

and the concern of this book, are the ways in which code also produces ambiguities and possibilities of recoding its prescriptive and deterministic tendencies (the unwritten laws, so to speak). Although instructional, program code cannot simply be reduced to its functional aspects, as it also extends the instability already inherent to the relationship of speech to writing, where it can also go out of control. Like all codes, it is only really interpretable within the context of the overall network of relations that make its operations inherently unstable. It is both a computer-readable notation of logic and a representation of this process, both script and performance; and in this sense it is like spoken words (made explicit in the case of a "hello world" program). It functions in relation to other programs that are simultaneously running in the form of processes that include storage and execution, operating across hardware and software platforms.[12] Indeed, code cannot be separated from the broader systemic framework and the way the technology that inscribes it is embedded in wider processes of command and control.

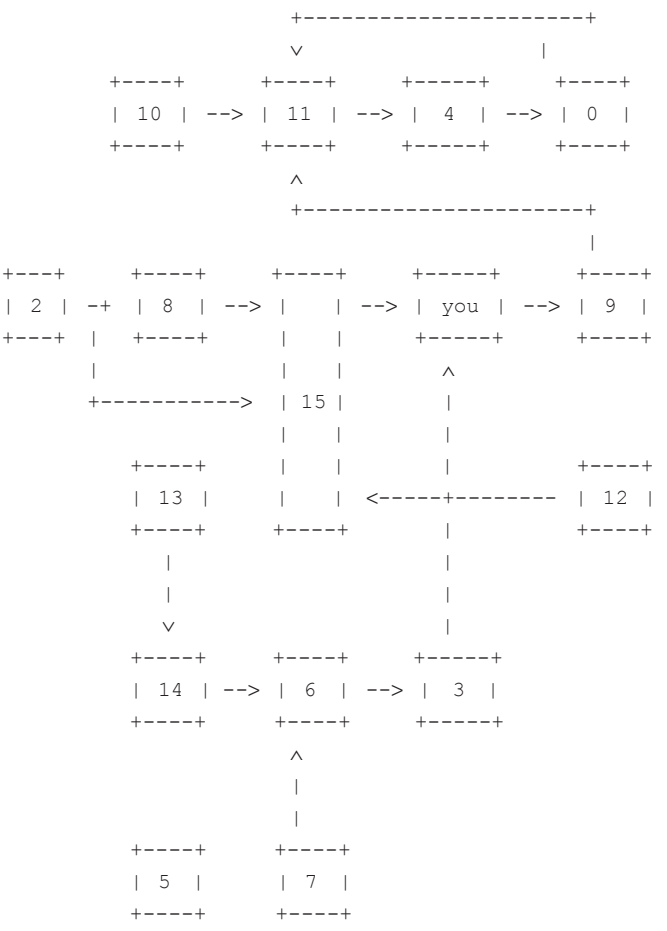
Coding subject

The common declaration "Hello world" *interpellates* in this way too, not least in the dogged insistence on the use of English as the default "mother tongue" of program languages. To Louis Althusser, the speech act constitutes the subject; it recruits subjects by hailing them, "Hey, you there!," as a policeman (sic) might speak to a passerby.[13] Through the act of recognition the subject begins to exist in ideology, in parallel to the way that program code can be seen to exist in ideology too.

In the following example, the program creates a list of numbers, one of which is made to represent the user of the program. The numbers are arbitrarily connected in a graph structure which is visualized for the user, along with the phrase "Hey, you there." An explicit command like this can be seen to reflect the capacity of programs to authoritatively "speak" to subjects, thus occupying the combined realms of subjectivity and sociality. It is in this sense that speech, or having a voice, connects with political expression and allows for a wider understanding of power relations. However, a more complex articulation of interpellation is required than an emphasis on the determining role of communication systems, as neither the human subject nor program code is quite that passive. Like the historical subject who is interpellated to act in a way that is preordained but not fully known, speech is also retroactive: it is speech in-itself, or speech that preexists itself, "speech before speech," as Slavoj Žižek explains.[14] Things are decided before they are enacted in actuality, and in this sense are always ready to be executed.

```
example@speakingcode:code$ cat | perl
use Graph::Easy;
my $g = Graph::Easy->new();

@ids = (0 .. 15);
$ids[rand(@ids)] = 'you';
foreach $id (0 .. $#ids) {
    $to = rand(@ids) - 1;
    $to++ if ($to >= $id);
    $g->add_edge($ids[$id], $ids[$to]);
}
print($g->as_ascii . "Hey, you there.\n");
```




```
Hey, you there.
```

Judith Butler also addresses this in her book *Excitable Speech*, drawing attention to the ways in which speech demonstrates agency, as there is a relation between speech and action, between saying and doing.[15] Spoken words say something and do something, and this has consequences in the world, such as in the way insults exert a form of violence. As linguistic beings we are bound to language as part of the constitution of subjectivity (as Althusser described it), so the call to order starts as a form of insult, and we enter into language antagonistically. Speech announces the action that will follow along with this antagonism, hence its power, but also importantly speech sometimes fails or is bounced back through various attempts to resist its determining effects. A good example of this is found in Mladen Dolar's book *A Voice and Nothing More*, which begins with a description of a failure of the voice to interpellate.[16] A command is given loud and clear by a commander of an army to attack its enemy, but nobody acts. The command is repeated over and over again to no effect, until someone says, "What a beautiful voice!" Refusing to execute the command, the soldiers instead contemplate the aesthetic properties of the spoken expression (the soldiers are Italian and perhaps it is their love of opera that informs their actions, rather than pacifism or fear of dying). The speculation is that esoteric languages similarly shift attention from command and control toward cultural expression and refusal.

Moreover, there is a well-established paradox in such a straightforward view of agency derived from the Althusserian concept of interpellation, with its stress on the determining role of language. If the subject is to some extent constituted in language and code, then to think that someone saying and doing something is a straightforward demonstration of agency misses the point; language and code constituted them in the first place, and as such the formation of the human subject is always an unfinished project. Interpellation seems to be bound up in contradiction in this respect, poised between the subject that speaks and the subject that is constituted through speech. If subjects are constituted in language in this paradoxical way, it follows that they also have the ability to reconstitute themselves through language, and even reconstitute the institution of language itself. Clearly there are other possibilities of agency operating outside the "sovereign autonomy of speech," and this is what Butler refers to as "excitable" states. Her assertion is that "speech is always in some ways out of control."[17]

Put differently, speech is far more distributed and networked than simply emanating from a single body, and the call to order is rather more like being allocated an Internet protocol (IP) address that defines you as a unique user in the network (and the example of code at the end of this chapter tries to demonstrate this point).[18] This also serves to emphasize the social aspect of speech—in that language itself is social and in speaking there is always an "echo of others in the act of speech," as Butler puts it.[19] Moreover, it is not that interpellation doesn't occur but that it operates in ever more subtle ways across complex informational networks, along with the networked forms of subjectivity it partly constitutes. This is what Brian Holmes refers to as the

"double constitution of the subject," generating both machine-like humans and human-like machines, conjoined through closed-loop informational systems.[20] Indeed, computer programs interpellate through the dual registers of command and control—and they do this at multiple levels of operation, and they also tend toward an executable logic that appears predetermined and unchangeable. Yet the program was programmed in the first place. So although on the surface program code appears to operate in a similar sovereign manner with straightforward agency, namely a command to execute an instruction from sovereign code, the argument that unfolds through this book is that in significant ways these operations are also prone to bugs and failure, and in significant ways can be considered to be out of control, like speech.

Although the analogy between program code and speech acts has become rather commonplace since its suggestion by Terry Winograd and Fernando Flores in 1987,[21] the stress here is on the degree of indeterminacy they share, as with the example of live coding, where the writing of the software happens at the same time as performing with the software.[22] Programmers express themselves through the use of program languages, the book suggests, in ways similar to other human communicative expression through language and gesture. They do this through their manipulation of layers of representation, including symbols, then words, language, and notation, as exemplified in the production of software prototypes, artworks, programming languages, and improvised performances that embed the activity of programming in the improvisation and experience of software art in general. On this last point, the practice of live coding exemplifies how the practice of coding, its writing, working, and creative use, establishes an unstable relation to its output. In this sense, although of course code largely determines its output, the broader apparatus including the idiosyncrasies of the programmer provide indeterminate outcomes and help to stress the expressive dimension of software production as a whole.

Beginning with these ideas, the book establishes that program code, like language in general, evokes complex processes by which multiple voices can be expressed, modified, and further developed. With code, clearly problems can be solved in a multiplicity of ways, as evident on the Rosetta Code website, where as many different languages as possible are collected that relate to the same computational task,[23] or on Instructionset, a website where instructions are posted and programmers are invited to carry them out in a variety of ways.[24] Like other collective speech acts, programming oscillates between process and expression. This further resonates in the ways program code opens up broader discussions around the production of meaning and criticism, for it is clear that there are a myriad ways of saying hello in a multiplicity of human and machine languages, and a great complexity in the ways that the human-computer interprets them.[25] Take, for another example, the esoteric programming language Piet which looks like a geometric painting by Piet Mondrian, from which it takes its name.[26]

Like speech in relation to text, esoteric languages that diverge from the conventions of written language seem to stress the point that rendering speech or code as mere

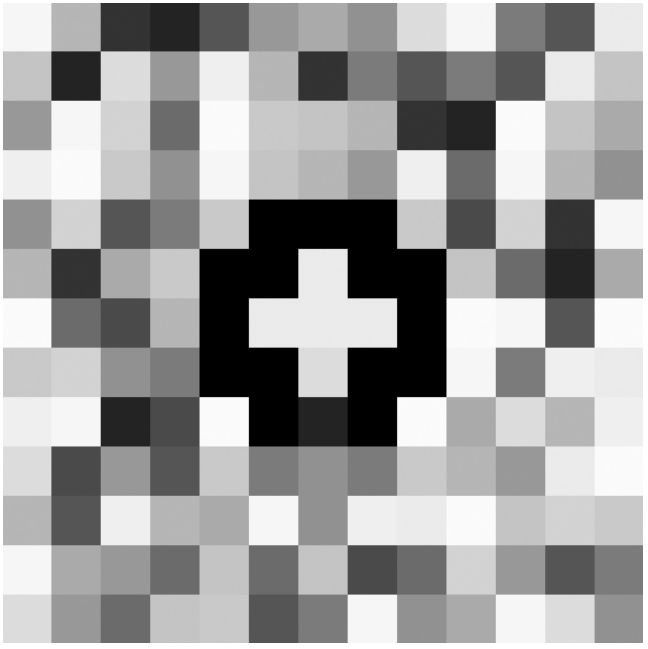


Figure 0.1
Source code written in the language Piet (original in color). Prints "*Hello, world!*" (http://www.retas.de/thomas/computer/programs/useless/piet/explain.html). Image cc by-sa 2.5 license, Thomas Schoch (2006).

written words fails to articulate the richness of human-machine expression. Similarly, Kittler explains that there is something paradoxical about using words to write about program code: "For one, all words from which the program was by necessity produced and developed only lead to copious errors and bugs; for another, the program will suddenly run properly when the programmer's head is emptied of words."[27] The paradox opens up some interesting opportunities for coding practices that challenge expectations of what source code does in itself and what it might do once run. In this way the use of esoteric languages like Befunge, Brainfuck, or Piet somewhat undermine the interpellative authority of the computer and stress alternative interpretations like the paradoxical qualities of speech. Perhaps we can begin to think of these examples as excitable program code.

Coding expression

Some core principles underpin the practices of coding as both procedure and expression. Of course, computers don't really speak but follow prescribed rules of execution,

tasks, and actions. Nevertheless, there is more to coding than simply the demonstration of formal logic, as if everything could be reduced to binary representation at a fundamental level. The conventions of writing and reading, of both text and code, might be considered to be part of coded systems of input and output (abbreviated as I/0), and yet formal logic fails to break the apparent paradox of language. In *The Art of Computer Programming* (1968), Donald Knuth already emphasized the depth of the relationship between writing a computer program and the logic that underpins it—what he referred to as "literate programming"—and so acknowledged programming to be somewhat like composing poetry or music.[28] He also demonstrated how reading could be understood as procedural and reflexive in his "Procedures for Reading This Set of Books." A short extract: "5. Is the subject of the chapter interesting you? If so, go to step 7; if not, go to step 6. . . . 14. Are you tired? If not, go back to step 7; 15. Go to sleep. Then, wake up, and go back to step 7."[29]

The physical form of a text and book, such as the one you are reading (if it still interests you), contributes to the production of meaning in the way that contextual information does in general. The text simply cannot be divorced from the materials and institutions that produce and disseminate it. In *Writing Machines,* N. Katherine Hayles puts it this way: "When a literary work interrogates the inscription technology that produces it, it mobilizes reflexive loops between its imaginative world and the material apparatus embodying that creation as a physical presence."[30] Like M. C. Escher's lithograph *Drawing Hands* (1948), in which two hands draw each another on a sheet of paper, there are many examples of recursion that encourage the author or reader to reflect on their role within the construction of the text, and on the ways in which meaning is constructed. Discussions of this tend to emphasize the link to literary theory (famously explored in Barthes's essay "The Death of the Author")[31] but also to second-order cybernetics, in terms of the way the observer is understood to produce changes to the system being observed,[32] akin to what Gregory Bateson referred to as "recursive epistemology."[33] Informed by systems thinking, Bateson understood recursion to operate on all levels across natural and cultural forms, undermining hierarchical orders of knowledge.

Another useful concept from Bateson, "double description," reinforces the importance of bringing together two or more information sources in order to provide information that is different from either one of them.[34] In a similar way, the combination of formal description and creative action, what might be referred to as *double coding*, is well established in software arts practice and where program code is made "literate." What has become known more specifically as "codework" provides a good example of how the practices of writing and programming can recombine formal logic and poetic expression. Sometimes referred to as pseudo-code, and often nonexecutable, codeworks operate in a similar realm to esoteric programs, and confound any easy assumptions about how meanings are produced prescriptively. Such

examples also serve to undermine the idea that execution is the sole site of interpretation for code.

A double sense of interpretation lies at the core of this, and suggests a broader engagement with critical literacy (literary criticism for computing) that includes code and other written or spoken forms. Indicative of this tendency is Mez's hybrid language "mezangelle" (developed in the 1990s) which simultaneously combines formal structures from code and speech into a kind of creole, thus challenging some of the assumptions of what constitutes communicative exchange between humans and machines, meshing together speaking-working-coding into new forms that reflect contemporary communication systems. Her intention is to expand the ways that meanings are generated beyond the conventions of written language, as explained using mezangelle:

```
2 4m a text fromme the ground[ing] uppe
2 n-hance the simple text of an email thru the splicing of wurds
2 phone.tic[k-tock]aulli m-bellish a tract ov text in2 a neo.logistic
maze
2 network 2 the hilt N create de[e]pen.den[ting]cies on email lizts for
the wurkz dis.purse.all
2 graphi.caulli N text.u.alli e-voke a conscious sens.u.all & lingual mix
2 make net.wurkz space themz.elves in2 a spindle of liztz thru
collaboratori n-tent
2 uze computer kode kon.[e]vent.ionz spliced with irc emoticons and
ab[scess]breviations
2 spout punctu[rez]ationz reappropri.[s]ated in2 sentence schematics
2 polysemicalli m-ploy a fractured wurdset
2 m-brace 4m conventionz
2 flaunt pol[emic]itical l-usions
2 ig.gnaw word endinz
2 let lettahs b used as subsetz
2 x-tend N promote n-clusive meaningz.[35]
```

The example, using a technique referred to as double (or multiple) coding, exemplifies the material aspects of code both on a functional and an expressive level, even if it further confuses interpretation by its nonfunctionality or inability to communicate clearly (which is part of the point, of course). In this sense, nonexecutable (or illiterate) code can also be considered to be a command that fails to interpellate the reader, and thus promotes the notion that all commands are open to failure and errors, just as refusing to act is a provocation in other contexts. So too with obfuscated code contests, where it is clear that program code has an aesthetic dimension that extends beyond the conventions of programming which stress the efficiency and brevity of source code. In taking to an extreme the principle that program code be concise, a

program might also be seen to run in rather unpredictable and amusing ways that confound the expectations of the human and machine reader-interpreter and allow them to rethink their prejudices.[36]

The double description evident in programming, and arguably inherent to it, can open up ambiguities and imaginative feedback loops. Indeed the loop is an important component of imperative programming, indicating when instructions are to be repeated or set to repeat until a terminating condition is met, unless an infinite loop is invoked. Programs are often structured around an infinite loop, known as the event loop; but with the use of infinite loops comes the possibility of infinite growth, which threatens the logical structure of the machine. Strategies like "deprogramming" bring to attention the structures and standardized formats of programs in this way. In the following example, from the website deprogramming.us,[37] a hello world program written in Perl has been modified to repeat itself in an endless loop:

```
#!/usr/bin/perl
# prozac.pl
# it will greet your system to death.
# but you go down in a cheerful endless loop.

while (1) {
           print "Hello World!\n";
           system ("$0"); # this line replicates it.
}
```

A simple example like this reflects the disruptive capacity of critical aesthetics to comment on the operations and effects of computational processes more broadly, working, as Alan Kay has referred to it, at the level of the "metamedium" (both performing universal computation and simulating all other media).[38]

Moreover, there is a danger of overstating the role of code as the source of action unless it is considered as part of a wider set of communicative actions by multiple human and nonhuman agents. Indeed, if code speaks, under what conditions and on behalf of whom?[39] Referring to code as speechlike becomes significant for this reason, as it invokes an established tradition considering speech to be more intelligible than writing, and its importance for critical debate and understanding of action in the world. To Socrates, who allegedly did not write his philosophy but spoke it, the voice is the "unwritten law" pertaining to the moral law, in contrast to the written law. It takes on a kind of authority and authenticity in this way and carries the inner voice of moral integrity.[40] One should not overstate the role of speech either, and clearly it contains its own internal contradictions and inherent paradoxes. Yet the spoken word does appear to haunt all texts as "sound is the natural habitat of language," as Walter J. Ong puts it in *Orality and Literacy*, and even writing needs to speak (if only silently) to reveal its meanings to the reader-listener.[41] Ong also makes a similar

point to Kittler, pointing to the paradox that in expressing ideas about orality, he relies on the written form rather than a spoken performance to deliver the argument. Part of the difficulty in writing something down is that the transformative possibilities of speech are curtailed.

Like executable code, there is something *overdetermined* about all writing in this sense. In "Talking Back," bell hooks stresses the inherent politics of forms of writing that hold on to speech, for both writing and speech can express resistance to forms of power—as with the rejection of capital letters in her name.[42] Is this what some programs do, too, in holding on to the special qualities and paradoxes of speech? More than simply writing, program code is a special kind of writing and, unlike a score that is followed but interpreted, it follows its script quite literally. It holds on to its script and does not let go, but in so doing it also and importantly holds on to the inherent special qualities and paradoxes of speech, its predeterminations and its sense of excess.[43]

Introduction

The purpose of the book is to explore these double descriptions: involving both formal logic and the expressive aspects of coding, its constraints and its excesses. To some readers, it may seem rather unfashionable to concentrate on spoken language as the main referent: why not use mathematics as the main metaphor for analysis, given that the computer works through binary arithmetic—both representing and manipulating numbers in a system of 0s and 1s? Such an approach might also be expressed in the interest in "object-oriented ontology" and its allusion to object-oriented programming, with a return of attention to the discrete object rather than relations,[44] as well as in numerical articulations of politics. In *Number and Numbers*, Alain Badiou writes: "We live in the era of number's despotism. . . . Number governs our conception of the political."[45] Indeed the relation between the human and machine reader comes closer than ever to the operations of a machine performing calculations. However, the book attempts to address this problem the other way around, insisting on a discourse that derives from an understanding of the human condition and of politics that operates through language rather than simply through the economy. Moreover, the problem identified here is the invasion of language by economics. The universal calculating machine has sharpened our understanding of the operations of speech and writing, not least through what Ong calls "secondary orality," through writing technologies that produce scores and scripts.[46] But under the present conditions of financial capitalism, human action is rendered economic and its force is annulled, as it is "expressed not with words but with numbers," as Boris Groys put it.[47] To Groys, the task is to transcribe the world from the medium of money to that of language, so that politics can operate freely in relation to fate rather than being subordinate to the

economy. Speech continues to underscore the human condition, however paradoxical this may appear.

These are some of the initial ideas that account for the book's attention to speech, its relation to the human condition, and its continued importance within cultural criticism. The book is not intended to be philosophy or theory as such. It takes critical writing to be a form of practice, and the many examples of code add to this conviction, not as illustrations but as additional forms of criticism. The resurgent interest in the concept of speech and the voice in recent years has something to do with the perceived neglect of sound in cultural work, no doubt,[48] but for the purpose of the book it is also an opportunity to foreground speech and action in the spirit of Hannah Arendt's writings,[49] and to consider some recent work that identifies how both speech and action appear to have lost their power, not least to mathematical symbols. To put it simply, *speech continues to need a voice that exceeds number*.

These are some starting points for the book and an indication of the broad sweep and eclecticism of its references. The first chapter, "Vocable Code" (co-written with Alex McLean), examines the performative and expressive dimension of programming and provides a number of examples. The aim is to stress some of the instabilities of code that undermine strict determinations of intention and meaning. Accordingly, the chapter looks in more detail at the linguistic analogies between code and natural languages, and the ways these have been understood in terms of meaning production. Speech emanates from the human body; computer programs have bodies too, and various attempts at simulation have revealed the impossibility of duplication. The further link of speaking to thinking machines demonstrates the sophistication of the human apparatus and the enduring complexity of speech, both as a sign of intelligence and as one that requires social interactions. Indeed collective speech acts provide multiple ways of understanding the complexities of code and the performative actions of running code. This is where speech indicates a more open-ended set of procedures that do not entirely compute, and that operate both within and beyond computational processes.

The second chapter, "Code Working," emphasizes a more overt political dimension of working with code, by stressing the ways in which all codework necessarily carries with it the work that has been invested in its production, as well as in the broader apparatuses through which it is served. This partly explains the motivation for many software producers to reveal the source code as an integral part of their work, particularly in the production of codeworks previously mentioned. A number of other examples are introduced, especially work by artists and programmers keen to offer alternatives to mainstream development, ranging from the performances of the live-coding scene to commons-based peer production. These demonstrate that new ideas emerge through wider recursive processes, which reflect the communicative and linguistic dimensions of work and action. If the first chapter established that code was

speechlike, then this chapter further establishes that speech has become more codelike under the conditions of informational capitalism, especially when work is executed through scores and scripts, something that Paolo Virno argues in building upon Arendt's work.[50] Taking Virno's line of argument, the relationship of capital to language also helps to establish how working can be understood as speech acts, facilitated by networked communications technologies and collective formations of work. This underpins the ways in which commodified technologies have appropriated collective speech acts and networked intelligence. It also allows for the possibility of code operating in excess of market forces and, despite the title of the chapter, more in the realm of code action than code work.

The third chapter, "Coding Publics," builds on many of the same references but focuses more on collective action and reconceptualizing ideas of publicness, again making reference to the writings of Arendt. The voice undermines this public dimension by offering a dual representational function, indicating both the precondition for language acquisition and also the expression of political opinion. That the idea of democracy itself is in crisis, along with the voice, is revealed in a range of paradoxes over the terms in use, and also over key issues that relate to intellectual property and the uneasy interconnections between free software and free speech. For Christopher M. Kelty, again referring to Arendt, the free software movement is an example of emergent and self-organizing public actions, underpinned by the sharing of source code.[51] With the free software movement in mind, he introduces the term "recursive public" to account for the ways in which the public is able to reconstitute itself as a public, through the modification of the platform through which it speaks and by addressing other technical and legal layers of operation. This attention to publicness has implications also for the connection to property regimes and the ways that social technologies (especially popular platforms like Twitter and Facebook) tend to encourage ever more voices to be heard but only with restricted registers and effects. The concern is to consider some of the mainstream implementations of social software, to register the conditions under which this is done and the language employed to do so, and to examine the consequences in terms of the production of social relations and subjectivities. Through their reliance on proprietary logic, these developments seem to legislate against a plurality of voices that have unique attributes, which Arendt considers to be necessary for politics. Although there may be more voices speaking, they are effectively made mute. Against popular interpretations of them, the logic of networks and the rhetorical promises of social media are examined as effective mechanisms for the *suppression* of political expression in the public realm, and to reveal how public intellect is becoming ever more privatized through free-market logic. The coalition between consumer capitalism and democracy is evident in relation to certain platforms that claim to allow enhanced participation in the political process, where again the voice is strongly invoked but only in compromised forms. It would seem

that social media not only fail to provide the means for people to have an effective voice but also reinforce neoliberal values, paradoxically through technologies that appear to promote the voice across telecommunications networks and online sharing platforms. This leads to the final section of the third chapter, in which the concept of the speaking public is reimagined not simply in its ability to enter into discourse but in its ability to modify the very platform through which it operates. And yet this capacity to express opinions is bound both to collective action and to the forces of domination, in reciprocal relation. Thus participation and collective forms of political expression become part of the very mechanism of guaranteeing their nullification. Clearly something important is missing.

The final chapter, "Code for-Itself," moves toward a summary of key issues, drawing on Franco Berardi's *The Soul at Work* (already introduced in the earlier chapter on work) to illustrate how the voice relates to material factors emanating from the way capital tries to incorporate both the soul and voice into production.[52] If the soul was once considered the source of speech, now conditions of contemporary production appear to have colonized both, indicating the politics of work to involve a broader set of issues that include speech acts and the communications platforms through which we speak. Voice reenters the discussion, on account of the need to focus on the biopolitical preconditions and the most fundamental aspect of human expression. In this way, the voice stands for the sensitivity of a culture bound by calculation and by neoliberalism's emptying out of expressive language. The ability to voice things is offered, if not accelerated, but is ultimately rendered illusionary by political forces that do not care about the "grain of the voice,"[53] as it lies outside the interests of the market and profiteering.

One of the main problems identified is the way the acquisition of language and the human condition have been largely separated as part of the expansion of market logic. More than simply a series of sounds from the body, a speech act is something that involves the human capacity to think and thereby express feelings. Rediscovering the voice, combined with words as speech, is therefore a necessary part of social transformation in the face of overpowering forces that close down and oversimplify discussion, or reduce action to procedures and behaviors that can be simulated. If human action is compromised by its contemporary expression in numbers, it should also be remembered that even at binary level, in terms of numerical calculations, the computer is surprisingly prone to errors, and certain calculations simply cannot be performed by strings of binary digits (determining the value of pi is a famous example).[54] Further ambiguities arise in computational processes when complex formations are introduced such as "or," "and," "not," as well as the infinite loops already mentioned. Once code is likened to speech, it also provides the possibility of new forms of criticism and practice that combine natural and artificial languages into new speech acts, in which ideas are stated and then reflected upon and restated. For all languages are

a mode of action in this way, and not only a referent of thinking. If coding is an invitation for speech and action—a script to be executed—then the act of coding is a deliberate action across cultural and technological fields. In this way it offers the potential to open up some of the inherent paradoxes of double description. This is a similar point to Groys's assertion that politics needs to operate with language if it is to act freely, and that the critical task is to assert its internal paradox. Under present conditions of capitalism, human action functions as a commodity, that is to say, it is inherently mute.[55]

The example that ends this chapter is a script that attempts to connect to a web server on each of the 4,294,967,296 IPv4 Internet addresses in turn. Where successful, it posts the message "Hello world!" to the server. The script signals other possibilities for speech, both *within* and *beyond* politics.[56] Perhaps programmers need to find their voices in this respect, such that their scripts might begin to announce themselves to the world in ways that remain open to other expressive possibilities and collective actions. By acting collectively, in the echoes of others in the act of speech (as Butler puts it), some of the rules of correct speech can be broken, and meaning can be reconnected to the body.[57] Similarly, babbling, or different forms of nonstandard speech, can extend the political and aesthetic possibilities of speech. This is how the following chapter begins, with reference to sound poetry that disrupts linguistic rules and rejects the authority of the master's voice.[58] Speech is inherently political in this sense, and programs characterized in this way provide ever more possibilities for generating unpredictable results in recognition of a body politic.

```
#!/usr/bin/python
# A script for greeting every server on the Internet.
import iptools, httplib
for ip in iptools.IpRangeList('0.0.0.0/0'):
  try:
    print "Greeting " + ip
    cx = httplib.HTTPConnection("%s:80" % ip)
    cx.request("POST", '/', "message=Hello+world!")
  except:
    pass
```

References

Adorno, Theodor. *Minima Moralia: Reflections from Damaged Life*. Trans. E. F. N. Jephcott. London: Verso, 1978.

Agamben, Giorgio. *Homo Sacer: Sovereign Power and Bare Life*. Trans. Daniel Heller-Roazen. Stanford: Stanford University Press, 1998.

Agamben, Giorgio. *Means without Ends: Notes on Politics*. Trans. Vincenzo Binetti and Cesare Casarino. Minneapolis: University of Minnesota Press, 2000.

Agamben, Giorgio. *State of Exception*. Trans. Kevin Attell. Chicago: University of Chicago Press, 2005.

Alterman, Jon B. "The Revolution Will Not Be Tweeted." *Washington Quarterly* 34 (4) (2011): 103–116. Available at http://www.twq.com/11autumn/docs/11autumn_Alterman.pdf.

Althusser, Louis. "Ideology and Ideological State Apparatuses: Notes Toward an Investigation." In Slavoj Zizek, ed., *Mapping Ideology*, 100–140. London: Verso, 1997.

Andrews, Robert. "Real DJs Code Live." *Wired*, 7 March 2006. Available at http://www.wired.com/science/discoveries/news/2006/07/71248.

Archer, Margaret, Roy Bhaskar, Andrew Collier, Tony Lawson, and Alan Norrie, eds. *Critical Realism: Essential Readings*. London: Routledge, 1998.

Arendt, Hannah. *The Human Condition*. 1958; Chicago: University of Chicago Press, 1998.

Arendt, Hannah. "Labor, Work, Action" (1964). In *The Portable Hannah Arendt*, 167–181. New York: Penguin, 2000.

Aristotle. *Nicomachean Ethics* (325 BC). In John Cottingham, ed. and trans., *Western Philosophy: An Anthology*, 326–370. Oxford: Blackwell, 1996.

Aristotle. *On the Soul and On Memory and Recollection*. Trans. Joe Sachs. Santa Fe: Green Lion Press, 2001.

Armitage, John. "From Discourse Networks to Cultural Mathematics: An Interview with Friedrich A. Kittler." *Theory, Culture and Society* 23 (17) (2006). Available at http://tcs.sagepub.com/content/23/7-8/17.

Arns, Inke. "Read_Me, Run_Me, Execute_Me: Software and Its Discontents, or, It's the Performativity of Code, Stupid." In Olga Goriunova and Alexei Shulgin, eds., *Read_Me: Software Art and Cultures*, 176–193. Aarhus: Digital Aesthetics Research Centre, University of Aarhus, 2004.

Austin, J. L. *How to Do Things with Words*. Cambridge, MA: Harvard University Press, 1975.

Avidsson, Adam, and Elanor Colleoni. "Value in Informational Capitalism and on the Internet: A Reply to Christian Fuchs." *Social Science Research Network* (28 February 2011). Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1772975.

Badiou, Alain. *Number and Numbers*. Cambridge: Polity Press, 2008.

Bakhtin, Mikhail. *The Dialogic Imagination: Four Essays*. Ed. Michael Holquist. Austin: University of Texas Press, 1981.

Balibar, Etienne. *Spinoza and Politics*. Trans. Peter Snowdon. 1998; London: Verso, 2008.

Barabási, Albert-László. *Linked: The New Science of Networks*. Cambridge, MA: Perseus, 2002.

Barthes, Roland. *Image, Music, Text*. London: Fontana, 1977.

Barthes, Roland. *S/Z: An Essay*. London: Cape, 1974.

Bataille, Georges. *The Accursed Share*. 3 vols. in 2. Trans. Robert Hurley. New York: Zone Books, 1988–1991.

Bateson, Gregory. *Mind and Nature: A Necessary Unity*. Cresskill, NJ: Hampton Press, 1979.

Bateson, Gregory. *Steps to an Ecology of Mind*. 1971; Chicago: University of Chicago Press, 2000.

Bauwens, Michel. "The Social Web and Its Social Contracts: Some Notes on Social Antagonism in Netarchical Capitalism." *Re-Public* (2008). Available at http://www.re-public.gr/en/?p=261.

Benjamin, Walter. "Critique of Violence." In *Selected Writings*, vol. 1, *1913–1926*. Ed. Marcus Bullock and Michael W. Jennings. Cambridge, MA: Belknap Press, 1996.

Benjamin, Walter. "On the Concept of History." In *Selected Writings*, vol. 4, *1938–1940*. Ed. Howard Eiland and Michael W. Jennings. Cambridge, MA: Belknap Press, 2003.

Bense, Max. "The Projects of Generative Aesthetics." In Jasia Reichardt, ed., *Cybernetics, Art, and Ideas*. London: Studio Vista, 1971. Available at http://www.computerkunst.org/Bense_Manifest.pdf.

Berardi, Franco "Bifo." *Precarious Rhapsody: Semiocapitalism and the Pathologies of the Post-Alpha Generation*. London: Minor Compositions, 2009.

Berardi, Franco "Bifo." *The Soul at Work: From Alienation to Autonomy*. Trans. Francesca Cadel and Giuseppina Mecchia. Los Angeles: Semiotext(e), 2009.

Berry, David M. *Copy, Rip, Burn: The Politics of Copyleft and Open Source*. London: Pluto Press, 2008.

Bogost, Ian. *Persuasive Games: The Expressive Power of Videogames*. Cambridge, MA: MIT Press, 2007.

Bogost, Ian. *Unit Operations: An Approach to Videogame Criticism*. Cambridge, MA: MIT Press, 2004.

Bolter, Jay David. *Turing's Man: Western Culture in the Computer Age*. Chapel Hill: University of North Carolina Press, 1984.

Bolter, Jay, and Richard Grusin. *Remediation: Understanding New Media*. Cambridge, MA: MIT Press, 1999.

Borrelli, Loretta. "The Suicide Irony: Seppukoo and Web 2.0 Suicide Machine." *Digimag* 52 (March 2010). Available at http://www.digicult.it/digimag/article.asp?id=1733.

Bosma, Josephine, et al., eds. *Readme! Filtered by Nettime: ASCII Culture and the Revenge of Knowledge*. New York: Autonomedia, 1999.

Bryan-Low, Cassell, and Siobhan Gorman. "Inside the Anonymous Army of 'Hacktivist' Attackers." *Wall Street Journal*, 23 June 2011. Available at http://online.wsj.com/article/SB10001424052702304887904576399871831156018.html#ixzz1QJzHz74Y.

Butler, Judith. *Excitable Speech: A Politics of the Performative*. London: Routledge, 1997.

Calvino, Italo. "How I Wrote One of My Books." In *Oulipo Laboratory*, trans. Iain White. London: Atlas, 1995.

Chamberlain, Gethin. "Apple's Chinese Workers Treated Inhumanely, Like Machines." *Guardian*, 5 May 2011. Available at http://www.guardian.co.uk/technology/2011/apr/30/apple-chinese-workers-treated-inhumanely.

Chomsky, Noam. *Syntactic Structures*. 1957; The Hague: Mouton, 1972.

Chun, Wendy Hui Kyong. "The Enduring Ephemeral, or the Future Is a Memory." *Critical Inquiry* 35 (1) (2008): 148–171.

Chun, Wendy Hui Kyong. "On 'Sourcery,' or Code as Fetish." *Configurations* 16 (3) (Fall 2008): 299–324.

Chun, Wendy Hui Kyong. *Programmed Visions: Software and Memory*. Cambridge, MA: MIT Press, 2011.

Cienki, Alan. "Why Study Metaphor and Gesture?" In Alan Cienki and Cornelia Müller, eds., *Metaphor and Gesture*, 5–25. Amsterdam: John Benjamins, 2008. Available at http://benjamins.com/cgi-bin/t_bookview.cgi?bookid=GS%203.

Coleman, Gabriella. "Code Is Speech: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers." *Cultural Anthropology* 24 (3) (2009): 420–454. Available at http://gabriellacoleman.org/.

Collins, Nick, Alex McLean, Julian Rohrhuber, and Adrian Ward. "Live Coding in Laptop Performance." *Organised Sound* 8 (3) (2003): 321–329.

Couldry, Nick. *Why Voice Matters: Culture and Politics after Neoliberalism*. London: Sage, 2010.

Cox, Geoff. *Antithesis: The Dialectics of Software Art*. Aarhus: Digital Aesthetic Research Centre, Aarhus University, 2010.

Cox, Geoff. "Means-End of Software." In Christian Ulrik Andersen and Søren Bro Pold, eds., *Interface Criticism: Aesthetics beyond Buttons*, 145–161. Aarhus: Aarhus University Press, 2011.

Cox, Geoff, Alex McLean, and Adrian Ward. "The Aesthetics of Generative Code." Paper delivered at Generative Art 00, international conference, Politecnico di Milano, Italy (2000). Available at http://www.generative.net/papers/aesthetics/index.html.

Cox, Geoff, Alex McLean, and Adrian Ward. "Coding Praxis: Reconsidering the Aesthetics of Code." In Olga Goriunova and Alexei Shulgin, eds., *Read_Me: Software Art and Cultures*, 161–174. Aarhus: Digital Aesthetics Research Centre, University of Aarhus, 2004.

Cox, Geoff, Haq Nav, and Tom Trevor, eds. "Art, Activism and Recuperation." *Concept Store Journal* 3. Bristol: Arnolfini, 2010.

Cox, Geoff, and Wolfgang Sützl, eds. *Creating Insecurity*. New York: Autonomedia, 2009.

Cramer, Florian. "Concepts, Notations, Software Art." In Olga Goriunova and Alexei Shulgin, eds., *Software Art: Thoughts*. Catalog of Read_Me Festival 1.2. Moscow: Rosizo, State Centre for Museums and Exhibitions, 2002.

Cramer, Florian. "Exe.cut[up]able statements: The Insistence of Code." In Gerfried Stocker and Christine Schöpf, eds., *Code: The Language of Our Time*. Linz: Ars Electronica; Ostfildern-Ruit: Hatje Cantz, 2003.

Cramer, Florian. *Words Made Flesh: Code, Culture, Imagination*. Rotterdam: Piet Zwart Institute, 2005. Available at http://www.netzliteratur.net/cramer/wordsmadefleshpdf.pdf.

Davenport, Thomas H., and John C. Beck. *The Attention Economy: Understanding the New Economy of Business*. Boston: Harvard Business School Press, 2001.

Dean, Jodi. *Democracy and Other Neoliberal Fantasies: Communicative Capitalism and Left Politics*. Durham: Duke University Press, 2009.

Dolar, Mladen. *A Voice and Nothing More*. Cambridge, MA: MIT Press, 2006.

Dourish, Paul. *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge, MA: MIT Press, 2001.

Eide, Stian Rødven, ed. *Free Beer 1.0*. Göteborg: FSCONS, 2009.

Felderer, Brigitte. "Orality." In *Zauberhafte Klangmaschinen: Von der Sprechmaschine bis zur Soundkarte*. Hainburg: IMA Institut für Medienarchäologie, Schott Music, 2008.

Feuerstein, Steven. *Oracle PL/SQL Programming*. Cambridge, MA: O'Reilly, 1997.

Foerster, Heinz von. *Cybernetics of Cybernetics*. Urbana: University of Illinois, 1974.

Foucault, Michel. *The Government of Self and Others: Lectures at the Collège de France 1982–1983*. Basingstoke: Palgrave Macmillan, 2010.

Fuller, Matthew. *Behind the Blip: Essays on the Culture of Software*. New York: Autonomedia, 2003.

Fuller, Matthew. *Media Ecologies: Materialist Energies in Art and Technoculture*. Cambridge, MA: MIT Press, 2005.

Fuller, Matthew, ed. *Software Studies: A Lexicon*. Cambridge, MA: MIT Press, 2008.

Galloway, Alexander R. *Protocol: How Control Exists after Decentralization*. Cambridge, MA: MIT Press, 2004.

Galloway, Alexander R., and Eugene Thacker. *The Exploit: A Theory of Networks*. Minneapolis: University of Minnesota Press, 2007.

Garfield, Bob. "Simon Cowell's Twitter Revolution." *Guardian*, 26 October 2011. Available at http://www.guardian.co.uk/commentisfree/cifamerica/2011/oct/26/simon-cowell-twitter-revolution.

Ghosh, Rishab Aiyer, ed. *Code: Collaborative Ownership and the Digital Economy*. Cambridge, MA: MIT Press, 2005.

Giroux, Henry. *Living Dangerously: Multiculturalism and the Politics of Difference*. New York: Peter Lang, 1993.

Groys, Boris. *The Communist Postscript*. London: Verso, 2009.

Guattari, Félix. *Chaosophy*. Ed. Sylvère Lotringer. New York: Semiotext(e), 1995.

Hardt, Michael, and Antonio Negri. *Commonwealth*. Cambridge, MA: Harvard University Press, 2009.

Hardt, Michael, and Antonio Negri. *Empire*. Cambridge, MA: Harvard University Press, 2000.

Hawkes, Terence. *Structuralism and Semiotics*. 1977; London: Methuen, 1986.

Hayles, N. Katherine. *My Mother Was a Computer*. Chicago: University of Chicago Press, 2005.

Hayles, N. Katherine. *Writing Machines*. Cambridge, MA: MIT Press, 2002.

Hegel, Georg W. F. *The Phenomenology of Mind*. Trans. J. B. Baillie. New York: Harper and Row, 1967.

Hofstadter, Douglas R. *Gödel, Escher, Bach: An Eternal Golden Braid*. 1979; London: Penguin, 2000.

Holland, Owen, ed. *Machine Consciousness*. Exeter: Imprint Academic, 2003.

Holmes, Brian. "Artistic Autonomy and the Communication Society." In *Diffusion: Collaborative Practice in Contemporary Art*. London: Tate Modern, 2003.

hooks, bell. "Talking Back." In Russell Ferguson, Martha Gever, et al., eds., *Out There: Marginalization and Contemporary Cultures*. Cambridge, MA: MIT Press, 1989.

Jameson, Fredric. *The Prison-House of Language: A Critical Account of Structuralism and Russian Formalism*. Princeton: Princeton University Press, 1972.

Kelty, Christopher M. *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press, 2008.

Kittler, Friedrich. *Discourse Networks 1800/1900*. Trans. M. Metteer and C. Cullens. Stanford: Stanford University Press, 1990.

Kittler, Friedrich. "Protected Mode." In Ute Bernhardt and Ingo Ruhmann, eds., *Computer, Macht und Gegenwehr. InformatikerInnen für eine andere Informatik*, 34–44. Bonn: FIfF, 1991.

Kittler, Friedrich. "There Is No Software." In Timothy Druckrey, ed., *Electronic Culture*, 331–337. New York: Aperture, 1996.

Kleiner, Dmytri. *The Telekommunist Manifesto*. Network Notebooks 03. Amsterdam: Institute of Network Cultures, 2010.

Knuth, Donald. *The Art of Computer Programming*. Vol. 1, *Fundamental Algorithms*. Reading, MA: Addison-Wesley, 1981.

Lakoff, George, and Mark Johnson. *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. New York: HarperCollins, 1999.

Laporte, Dominique. *History of Shit*. Cambridge, MA: MIT Press, 2000.

Latour, Bruno. *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford: Oxford University Press, 2005.

Lazzarato, Maurizio. "Immaterial Labor." In Paolo Virno and Michael Hardt, eds., *Radical Thought in Italy: A Potential Politics*. Minneapolis: University of Minnesota Press, 1996.

Lee, Edward A. "Computing Needs Time." *Communications of the ACM* 52 (5): 70–79 (2009). Available at http://cacm.acm.org/magazines/2009/5/24649-computing-needs-time/fulltext.

Leslie, Esther. *Walter Benjamin: Overpowering Conformism*. London: Pluto, 2000.

Link, David. "Scrambling T-R-U-T-H: Rotating Letters as a Material Form of Thought." In Siegfried Zielinski and Eckhard Fürlus, eds., *Variantology 4: On Deep Time Relations of Arts, Sciences and Technologies in the Arabic-Islamic World*, 215–266. Cologne: König, 2010.

Link, David. "There Must Be an Angel: On the Beginnings of the Arithmetics of Rays." In Siegfried Zielinski and David Link, eds., *Variantology 2. On Deep Time Relations of Arts, Sciences and Technologies*, 15–42. Cologne: König, 2006.

Lukács, Georg. *History and Class Consciousness: Studies in Marxism*. 1922; Cambridge, MA: MIT Press, 1976.

Mackenzie, Adrian. *Cutting Code: Software and Sociality*. New York: Peter Lang, 2006.

Malik, Shiv. "Facebook Accused of Removing Activists' Pages." *Guardian*, 29 April 2011. Available at http://www.guardian.co.uk/technology/2011/apr/29/facebook-accused-removing-activists-pages.

Marazzi, Christian. *Capital and Language: From the New Economy to the War Economy*. Trans. Gregory Conti. Los Angeles: Semiotext(e), 2008.

Marino, Mark C. "Critical Code Studies." In *Electronic Book Review* (2006). Available at http://www.electronicbookreview.com/thread/electropoetics/codology.

Marx, Karl. *Capital*. Vol. 1. Trans. Ben Fowkes. Harmondsworth: Penguin, 1990.

Marx, Karl. *Grundrisse: Foundations of the Critique of Political Economy (Rough Draft)*. Harmondsworth: Penguin, 1981.

Mauss, Marcel. *The Gift: Forms and Functions of Exchange in Archaic Societies*. Trans. Ian Cunnison. London: Cohen and West, 1970.

McLean, Alex. "Artist-Programmers and Programming Languages for the Arts." PhD thesis, Goldsmiths, University of London, October 2011.

McLean, Alex, and Geraint Wiggins. "Words, Movement and Timbre." Paper delivered at NIME09 conference, Pittsburgh, 3–6 June 2009. Available at http://doc.gold.ac.uk/~ma503am/writing/nime09.pdf.

Negri, Antonio. *Marx beyond Marx: Lessons on the Grundrisse*. Ed. Jim Fleming. Trans. Harry Cleaver, Michael Ryan, and Maurizio Viano. New York: Autonomedia, 1991.

Neumark, Norie, Ross Gibson, and Theo van Leeuwen, eds. *Voice: Vocal Aesthetics in Digital Arts and Media*. Cambridge, MA: MIT Press, 2010.

Ong, Walter J. *Orality and Literacy: The Technologizing of the Word*. 1982; London: Routledge, 2002.

Poe, Edgar Allan. *The Complete Tales and Poems of Edgar Allan Poe*. New York: Vintage, 1975.

Preston, John, and Mark Bishop, eds. *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*. Oxford: Clarendon Press, 2002.

Rancière, Jacques. "Ten Theses on Politics." *Theory and Event* 5 (3) (2001). Available at http://www.egs.edu/faculty/jacques-ranciere/articles/ten-thesis-on-politics.

Raymond, Eric S. *The Art of UNIX Programming*. Boston: Addison-Wesley, 2004.

Rée, Jonathan. *I See a Voice: Language, Deafness and the Senses—A Philosophical History*. London: HarperCollins, 1999.

Roberts, John, ed. *Art Has No History! The Making and Unmaking of Modern Art*. London: Verso, 1994.

Rossiter, Ned. *Organized Networks: Media Theory, Creative Labour, New Institutions*. Rotterdam: NAi, in association with the Institute of Network Cultures, 2006.

Rushkoff, Douglas. *Program or Be Programmed: Ten Commands for a Digital Age*. Berkeley, CA: Soft Skull Press, 2011.

Schmitt, Carl. *The Concept of the Political*. 1927; Chicago: University of Chicago Press, 1996.

Scholz, Trebor. "On MTurk, Some Examples of Exploitation" (2009). Available at http://www.collectivate_net/journalisms.

Schwitters, Kurt. *Poems, Performance Pieces, Proses, Plays, Poetics*. Ed. and trans. Jerome Rothenberg and Pierre Joris. Cambridge, MA: Exact Change, 2002.

Searle, John R. "Minds, Brains, and Programs." *Behavioral and Brain Sciences* 3 (1980): 417–424.

Sondheim, Alan. "Introduction to Codework." *American Book Review* 22 (6) (September/October 2001). Available at http://www.litline.org/ABR/issues/Volume22/Issue6/sondheim.pdf.

Stallman, Richard M. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Ed. Joshua Gay. Boston: GNU Press, 2002.

Stallman, Richard M. "Why Software Should Not Have Owners." 1994. Available at http://www.gnu.org/philosophy/why-free.html.

Stephenson, Neal. *Cryptonomicon*. New York: HarperCollins, 2002.

Terranova, Tiziana. "Free Labor: Producing Culture for the Digital Economy." *Social Text* 18 (2) (2000): 33–58.

Terranova, Tiziana. *Network Culture: Politics for the Information Age*. London: Pluto Press, 2004.

Tronti, Mario. "The Strategy of Refusal" (1965). In "Autonomia: Post-political Politics," *Semiotext(e)* 3 (3) (1980): 28–34.

Virno, Paolo. *A Grammar of the Multitude: For an Analysis of Contemporary Forms of Life*. Trans. Isabella Bertoletti, James Cascaito, and Andrea Casson. New York: Semiotext(e), 2004.

Virno, Paolo. *Multitude: Between Innovation and Negation*. Trans. Isabella Bertoletti, James Cascaito, and Andrea Casson. Los Angeles: Semiotext(e), 2008.

Virno, Paolo, and Michael Hardt, eds. *Radical Thought in Italy*. Minneapolis: University of Minnesota Press, 1996.

Wall, Larry. *Programming Perl*. Sebastopol, CA: O'Reilly, 2001.

Wardrip-Fruin, Noah, and Nick Montfort, eds. *The New Media Reader*. Cambridge, MA: MIT Press, 2003.

Wiener, Norbert. *Cybernetics: or, Control and Communication in the Animal and the Machine*. 1948; Cambridge, MA: MIT Press, 1965.

Wiener, Norbert. *God and Golem, Inc*. Cambridge, MA: MIT Press, 1964.

Winograd, Terry, and Fernando Flores. *Understanding Computers and Cognition: A New Foundation for Design*. Reading, MA: Addison-Wesley, 1987.

Wood, Gaby. *Living Dolls*. London: Faber and Faber, 2002.

Wray, Richard. "EU Says Internet Could Fall Apart." *Guardian*, 12 October 2005. Available at http://www.guardian.co.uk/business/2005/oct/12/newmedia.media.

XXXXX. Berlin and London: Openmute, 2006.

Yuill, Simon. "All Problems of Notation Will Be Solved by the Masses." *Mute* (February 2008). Available at http://www.metamute.org/en/All-Problems-of-Notation-Will-be-Solved-by-the-Masses.

Žižek, Slavoj. "Hegel's 'Logic of Essence' as a Theory of Ideology." In *The Žižek Reader*, ed. Elizabeth Wright and Edmond Wright. Oxford: Blackwell, 1999.

Žižek, Slavoj, ed. *Mapping Ideology*. London: Verso, 1997.

Žižek, Slavoj. *The Ticklish Subject: The Absent Centre of Political Ontology*. London: Verso, 1999.

Žižek, Slavoj. *Violence*. London: Profile Books, 2008.