How To Be a Geek

Essays on the Culture of Software

Matthew Fuller
HOW TO BE A GEEK
# CONTENTS

*Acknowledgements* vii  
*Biographies of Co-Authors* ix  

**Introduction** 1  

**Histories**  
1 The Obscure Objects of Object Orientation 15  
   *with Andrew Goffey*  
2 Abstract Urbanism 37  
   *with Graham Harwood*  

**Entities**  
3 Software Studies Methods 55  
4 Big Diff, Granularity, Incoherence and Production in the Github Software Repository 69  
   *with Andrew Goffey, Adrian Mackenzie, Richard Mills and Stuart Sharples*  
5 The Author Field 86  
   *with Nikita Mazurov and Dan McQuillan*  

**Aesthetics**  
6 Always One Bit More: Computing and the Experience of Ambiguity 113
7  Computational Aesthetics  
*with M. Beatrice Fazi*  
132

8  Phrase  
*with Olga Gorjunova*  
155

9  Feral Computing: From Ubiquitous Calculation to Wild Interactions  
*with Sónia Matos*  
167

10  Just Fun Enough To Go Completely Mad About: On Games, Procedures and Amusement  
*Powers*  
191

11  Black Sites and Transparency Layers  
207
ACKNOWLEDGEMENTS

‘The Obscure Objects of Object Orientation’ (with Andrew Goffey) was previously published in Penny Harvey, Eleanor Conlin Casella, Gillian Evans, Hannah Knox, Christine McLean, Elizabeth B. Silva, Nicholas Thoburn and Kath Woodward, eds., The Routledge Companion to Objects and Materials, Routledge, London, 2013, and in a variant form in Zeitschrift für Medienwissenschaft. With thanks to Claus Pias, Elizabeth B. Silva and Nicholas Thoburn.

‘Abstract Urbanism’ (with Graham Harwood) was previously published in Rob Kitchin and Sung Yueh Perng, eds., Code and the City, Routledge, London, 2016, and presented at the Programmable City workshop at Nirsa in Maynooth the previous year. Thanks to all at the Programmable City project.


‘Big Diff, Granularity, Incoherence and Production in the Github Software Repository’ (with Andrew Goffey, Adrian Mackenzie, Richard Mills and Stuart Sharples) was previously published in Ina Blom, Trond Lundemo and Eivind Røssaak, eds., Memory in Motion: Archives, Technology and the Social, Amsterdam University Press, Amsterdam, 2016. It follows research with the co-authors in the ESRC-funded Metacommunities of Code project, led by Adrian Mackenzie.

‘The Author Field’ (with Nikita Mazurov and Dan McQuillan) follows from work on the London Cryptofestival in November 2013. With thanks to Yuk Hui.
HOW TO BE A GEEK


‘Phrase’ (with Olga Goriunova), was previously published in Celia Lury and Nina Wakeford, eds., Inventive Methods, Routledge, London, 2013. With thanks to Olga Goriunova, Celia Lury and Nina Wakeford.

‘Feral Computing: From Ubiquitous Calculation to Wild Interactions’ (with Sónia Matos) was previously published in Fibreculture Journal, Sydney, 2011. With thanks to Sónia Matos and Andrew Murphie.

‘Just Fun Enough To Go Completely Mad About: On Games, Procedures and Amusement’ was previously presented as a paper at the St Petersburg Centre for Media Philosophy; a workshop organized by the ARITHMUS research project at Goldsmiths; and C-Dare at Coventry University. Thanks to Alina Latypova, Evelyn Ruppert, Baki Cakici, Hetty Blades and Scott de la Hunta.

An early version of ‘Black Sites and Transparency Layers’ was given as a talk organized by Robin McKay of Urbanomic at Thomas Dane Gallery, London, March 2015. With thanks to John Gerrard. A later version was presented at the Academy of Fine Arts Nuremburg, with thanks to Kerstin Stakemeier; at the Interface Politics conference, Barcelona, April 2016, with thanks to Jaron Rowan and Pau Alsina; and as an inaugural lecture at Goldsmiths, with thanks to everyone there.

Thanks to Femke Snelting and Sarah Magnan of Open Source Publishing for the cover design, and to John Thompson and George Owers at Polity for an easy and affable collaboration. Many thanks to Fiona Sewell for brilliant diagnoses and surgery of the text.
M. Beatrice Fazi is a Research Fellow at the Humanities Lab, Sussex University, where she works on the cultural theory and philosophy of computing.

Andrew Goffey is co-author, with Matthew Fuller, of Evil Media and translator of books by Félix Guattari and Isabelle Stengers amongst others. He is Associate Professor in Critical Theory and Cultural Studies at the University of Nottingham.

Olga Goriunova is Senior Lecturer in the Department of Media Arts, Royal Holloway, University of London, and is editor and curator of many projects, and author of Art Platforms and Cultural Production on the Internet.

Graham Harwood is a member of the artist group YoHa and Senior Lecturer at Goldsmiths, University of London.

Adrian Mackenzie is Professor of Sociology at Lancaster University and author of numerous books including Wirelessness: Radical Empiricism in Network Cultures.

Sónia Matos is a designer, Lecturer in the Department of Design, University of Edinburgh, and a Research Fellow at the Madeira Interactive Technologies Institute.

Nikita Mazurov is a security and forensics researcher and Postdoctoral Researcher at the Living Archives Project, University of Malmö.
HOW TO BE A GEEK

Dan McQuillan is Lecturer in Creative and Social Computing at Goldsmiths, University of London, and science and technical lead for Science for Change Kosovo.

Richard Mills is a Research Associate at the Psychometrics Centre, University of Cambridge, and a researcher in the statistical analysis of online data.

Stuart Sharples is a Senior Research Associate in the Department of Sociology, University of Lancaster.
The mode of knowing software is not yet established. We are still at a point where a critical language to understand the wider domain of computational culture is only beginning to ferment. There is a lot of writing about software, a lot of writing in and as software, and a lot of conjuration of alphanumeric strings that is done by and as software. Writing on software which works through a critical or speculative mode is not especially different from these things; it inhabits some of the same modes, idioms and intellectual habitats. It works, at one scale or another, with the same logical aggregates. Writing on software is thus at least partially inside software, even when presented as a paper book. The point in this collection of texts is to work into a few places within that condition.

Due to the complexity and variety of entities and processes that produce computational cultures, and the multiplicity of ways in which they can be understood, experienced, put into play, there is a movement between modes of writing here. Some of the texts are musings, almost dazed, gazing into the screen, across a slither of cable, or into the genealogies of logics in an attempt to discern a murmuring between registers. Others are more programmatic essays that attempt to find the intersections between certain combinations of parabola of enquiry and to draw some locations, speeds and movements out from them. Concatenations of numbers drawn from systems are arrayed and compared, alongside ideas about what numbers are.

To write about software cultures, then, is not to attempt to stamp an order on it, but to draw out some conditions that are particular to the way in which computational systems, and the different scales of their articulation, from minor aspects of interface to globally oriented mechanisms, can be brought into resonant communication
HOW TO BE A GEEK

with ideas and questions that extend them beyond the private conversations of technical experts into wider conjugations of ways of asking questions and making problems. Equally, much of the work here operates with the concepts of computer science as fundamentally cultural and political, as something core to contemporary forms of life and thus as open to theoretical cultural exploration as much as architecture, sexuality or economics might be. Indeed, a proposition of this book is that computing is aligned with all three of these, and other factors, in numerous ways and that tracing these conjugations can give us some information about contemporary life that would be unavailable otherwise. This is to say, too, that whilst the work to address software as culture is currently taking many forms, to reduce it solely to the stable sets of categories and objects known to existing disciplines is to miss the historical moment.

Geek

To be a geek is, in one way or another, to be over-enthused, over-informed, over-excited, over-detailed. There is an awkwardness born out of a superfluity of an extraneous kind of desire that becomes a febrile quiver in the face of an interesting problem. To be geeky is to have too much interest in something to the detriment of comportment, code spilling over into a gable, a liveliness found in something that a more reserved protocol would keep under wraps or avoid. To be a geek is to be a bit too public with your enthusiasms, to be slightly unaware in turn that these thrills may, to others, rightly be dull as dust dehydrated with a special process of your own invention. Its mixture of juiciness and dryness, being able to get juiced on dryness, perhaps gets to the core of the problem. Frankly, it’s a ludicrous position to be in – it is after all a bit bewildering to find this stuff so fascinating – but it is one whose perversity puts it in a strange relation of proximity to fundamental dynamics in contemporary life.

Such a condition leads many geeks into precariously powerful positions. Companies founded and staffed by geeks rule the world in many ways. They fill institutions and create commercial entities. One can be both ludicrous and lucrative, a maniac for certain details that remake the cosmos by their syntax, or that found a new grammar of relation between things. Geeks created the internet and fight over its meaning. They govern and subvert governance, or keep it ticking over with regular incremental upgrades. Geeks produce extravagant contraptions that cement their positions in the most comedically venal
INTRODUCTION

ways, but they also make machines with panache that auto-destruct in deserts and car parks and servers; sometimes intentionally. They make games that provide the grounds of individuation for millions, and then find the wealth it occasionally brings depressingly pointless but irresistible.

Given this, geeks may often mute themselves, try and pass as underwhelming. This is probably an adequate survival strategy in many circumstances. An alternative to it is a form of intellectual cosplay: learn to grow and stretch out your membranes to generate new kinds of sensitivity to the present. In the growth of such tender surfaces and depths, become slightly perturbing (as in the Dutch word *gek*, which shares a common etymological root with *geek*, meaning mad, or crazy), develop the capacity to feel the interaction of electromagnetic waves with your anorak of ideas. Work their ordering via logic into concatenations of circuits that then propitiate further ordering. Sense on into their interaction with other forces arrayed as fields and intensities at the scales of the social, the aesthetic, the ideational or economic. Learn to trace, cajole and heighten the fatal and illuminating crossings of wires and desires.

Such an unfolding and an invention of sensitivities means also some reflection on the figure of the geek, as something both powerful and flawed. The geek tragedy improves on the traditional mixture of these two qualities by adding the factor of technology. This variant of the tragedy is there since that of Icarus, who flew too close to the sun and melted his wings. If the traditional reading of Icarus is a warning about the hubris of knowledge in relation to nature and the inviolable space of the gods, technology folds in both violation and knowledge as constitutional factors. Any attempt to ‘White Box’ technology as a simple Good that needs no examination misses this fundamental transformative characteristic. But contemporary technology is not simply an extension of a man – like wings of wood, wax and feather – which would be a purely mechanical effect. (At the very least, the physics of the nineteenth century, which was radically changed by the fields and waves of electromagnetism, should have curtailed that conjecture.) It is one whose characteristics are also profoundly mathematical and logical, bringing the force of abstractions into combination with materialities as diverse as labour, the primitive accumulation economies of extraction (for making metal-hungry devices), and into the complexities of contemporary cultural geopolitics that move from the interpersonal to the international with unaccustomed rapidity. The powers of electronics, and the ability to couple and intensify those with logical descriptions that create processes that traverse the
social, economic and cultural, intensify in turn the powers of fascination and the ability to invent that are characteristic of geeks. We have yet to find ways to think through the full combination of these forces and scales of articulation. As luck, which begins every tragedy, would have it, however, we are in the midst of all this and have at least the capacity to geek out on the problematic. The texts collected here are different kinds of attempts to do just this.

**Collaborators**

But talking of geeks: crucial to many of the texts brought together in this book is that they are collaboratively written. My companions in their writing are a designer, programmers, computer scientists, security researchers, statisticians, social scientists, philosophers, cultural theorists, a reluctant artist, and combinations of these. Such collaborations bring and induce capacities, ideas, combinations and indeed data that would otherwise not come into communication with each other. Key to software studies approaches is that of engaging with texts, ideas, objects and systems from computer science and digital technologies in dialogue with those of cultural theory, digital media cultures, art, hacking and elsewhere. People who embody and enliven this dialogue in different ways are fundamental to this book and provide the corrugation necessary to make these papers stand up.

As a field, software studies has so far successfully remained minor in an era of the consolidation of disciplines – which survive, and strengthen, primarily as buttresses against institutional attrition. This ‘success’ of course is presided over by the irony that, since it emerges to address some of the key features of the last decades, these features are sometimes more or less attended to by other approaches, more disciplinary in nature, that then stand in for a more fundamentally inter- or anti-disciplinary form of knowledge. Here, elements of computational cultures operate as metonyms for what then is magically imaginable as a resolvable whole; or are addressed solely in terms of the categories, tools and formats of such disciplines without taking the risk of their transformation. It is the tendency towards the whole-hearted embrace of the contact zone that characterizes software studies.

This is not a fully accomplished, uncomplicated or universalizable proposition, however. Each of the collaborations here has its own texture rather than a generalized or un-preworked wide-openness. The patterns that each essay brings together create resonances across
distributions of knowledge and skill that create different tensions and confluences. The allocation of terminology or inference, or in the discussions and iterative decisions as to what constitutes a useful fulcrum for levering up and analysing the entity or processes under discussion, provide forces around which the text aggregates. Indeed, in some cases here, the operation of programs and scripts, their relations to systems and data, or the use of forensic techniques, provide another set of collaborators that in turn generate the texts here.

Overview

Collectively, this book presents an assembly of explorations of some of the fundamental infrastructures of contemporary life: computational structures, entities and processes that undergird, found and articulate economies, entertainment and warfare, to name simply their interactions with the ascendant holy trinity. This is to say that software is considered to be fundamentally cultural. Working with the objects and ideas of computer science as part of culture is to say that they take part fully in numerous ways of life; that they are inscribed by ideas and produce feelings and relations through and incidentally to these; and, as a scientific and technically generative set of fields, that it has numerous consistencies and textures within it that are both ‘internally’ significant within those fields, with their own codes, powers and sources of fascination, and, in these more or less abstract or autonomous conditions, able to create alliances with and patterns of excitement, disequilibria and amplification for other aspects of culture, asymmetrically ranging in their iterative acuity in relation to things as disparate as alphabets and institutions, as caresses and contracts. It is the interplay and development of the internal formulations that in many senses excites geeks. But it is their interleaving with these other aspects of software as culture that also changes the position of geeks today. Geeks are not only a ‘people to come’ – inhabiting a world ahead of its realization in ideas or in artefacts, or in selves, by means of what might retrospectively become their precursors – but are also brought into being as entities that inherently entail a kind of spasm of parody, of silliness. This, alongside the way in which the logical foundations of computing render it unstable at different scales, and are therefore often scaffolded, corralled, against this logical unfinishedness at others, is part of what initiates the figure of the geek, and the processes of computing open to their outsides at different scales.
HOW TO BE A GEEK

But these logical foundations are also what allows for a remarkable continuity of form – through the generality of the abstract machine, a metamachine that computing puts into place – that sets up much of the tension and generativity in computing. Between abstraction and control, between replicability and translation, between generality and precision, computing forges and sometimes enforces new conjunctures, tensions and orderings. That is why software is both experiential and political, both iterated through the human yet markedly posthuman, as well as being multiply tied into the political mechanisms that seem only able to offer themselves in terms of a charade of humanism (as its mirror, as its tools, as its better self, as its means of knowledge and communication, as something that will iteratively deduce its secrets and its ailments, and so on). At the same time these political mechanisms curtail the humanism they claim to enhance, enforce seriality, tie it ever more closely into transactional mechanisms that translate to capital as a twinned and increasingly informational abstract machine where the individual, whom it ostensibly lauds, is merely figured as a nexus of contracts and flows of credit units and data. The essays collected in this book are aimed in different ways at prising open this condition.

They are published in thematic clusters: Histories, Entities, Aesthetics, Powers. The Histories section develops accounts of two key objects in computing history, object orientation and agent-based modelling. The chapter entitled ‘The Obscure Objects of Object Orientation’ charts a cultural and critical history of this fundamental form of programming through the development of object-oriented programming in educational psychology and social democratic forms of workplace governance to its present dominance in both the conceptual modelling and the development of software to its position in the global software authoring economy.

‘Abstract Urbanism’ provides a genealogy of key projects and thinking in the domain of urban modelling from the 1960s onwards, developing an account of the way in which it embodied critical and speculative approaches to mathematics, logic and city and social processes, with sometimes dubious and often creative significance. Modelling as a form of thought, as a means of experimentation with parameters and actions, has become one of the most subtle and advantageous computational operations. Its actual and ostensive power authorizes and encourages decisions. Understanding the computational part and the evidential grounding of such decisions, as well as the way they are socially located and conceptually and technically delimited, is crucial in developing the capacity of modelling
for thinking through cities as condensers of social and technical forces.

The Entities section presents ways of addressing analyses of software cultures at different scales and via the different kinds of objects or entities that they entail. It starts with a chapter outlining a brief survey of ‘Software Studies Methods’. Since software is so heterogeneous and so multi-scalar, this chapter is necessarily incomplete as a survey, but it points towards a range of different ways for engagement, analysis and reflection that are being developed. Around these are numerous clusters of researchers, more or less in communication about what come into being as mutual or distinct modes and points of enquiry. Shared amongst them is the sense that, whilst software is not absolutely distinct from other factors, such as, most obviously, hardware, it has sufficient specificity and significance in its own terms to demand enquiry and speculative development. Whilst terms, such as algorithm, code, or infrastructure and others, may move in and out of focus of different kinds of collective attention over time, numerous kinds of continuity are also developed by the field.

The section then goes on to two chapters that describe and analyse two exemplary kinds of object in contemporary software culture. ‘Big Diff’ takes as its object a large-scale distributed software repository, Github. Merging data science techniques with a theoretically grounded analysis, this essay provides both a reading of this major part of contemporary software development infrastructure and what is a hopefully methodologically interesting work in between the two approaches. Github, which is built on the file-management system Git, developed by Linus Torvalds, is particularly interesting as a site of production because it is symptomatic of the way in which some of the conditions of software culture, derived both from computational forms and from cultural tendencies that are manifest as conventions realized in software, drive forms of work, communication and value-creation in ways that are articulated through software’s work on itself. The way in which Git and Github re-order the question of the nature of the archive is particularly symptomatic here. Like much on the internet today, Github is an architecture that centralizes the distributed structure of Git – a file structure that dramatically inverted a social taboo (that against forking, duplicating and varying, code) as a means of establishing a collective resource. In turn, the site also provides the grounds for what this article suggests is a Post-FLOSS (free, libre and open source software) form of code. But software’s work on itself is something that also figures as a subtext of the book as a whole, indicating both the developing complexity and matu-
rity of aspects of software cultures and the degree to which it also becomes something reflexive.

The following chapter, ‘The Author Field’, takes a more microscopic object as its approach, developing a genealogy of a specific piece of metadata, as named in the title. Forensic and counter-forensic literature and techniques are used to examine the modulation between apotheosis and disappearance this term is subjected to by the technical operations of mundane technologies such as word processors. What are the ways in which forms of writing are produced, stabilized, repeated, transduced in such processing – and how does this articulate the figure of the user, author or other such category? In this sense, like some of the other texts gathered here, this chapter develops a line of enquiry also present in a previous book of mine which shares this one’s subtitle, *Behind the Blip: Essays on the Culture of Software*, with which there is a strong sense not only of continuity of topic, but also of the widely expanding nature of the grounds of enquiry.

Shared also with that book is a set of attempts to address the aesthetic dimension of contemporary software cultures. Aesthetics is meant in the broad sense here, of addressing the means by which sensation and perception are involved, at deep levels, with the technical. As a field, software studies draws heavily from art, including the early software art movement, for some of its sensibility. Both are heavily involved in working through the different scales and modes of circulation of software as sites saturated in machinings of various kinds: interpretation which conjoins knowing and transforming, perceiving and doing; interaction with its fusing of experience and technique; hacking with its bringing together of cultural invention and mischief as a mode of subjectivation. Here, art is founded on the capacity of invention that relies on multiplicities of meaning and the absence of any final vindicating authority as the grounds for creation. Reflection on technological experience via technological means as they weave into and interfere with those of other scales is a means of taking them beyond the question of their intents or their purposes, to explore them as conditions of possibility.

The *Aesthetics* section therefore develops accounts of computational aesthetics, understood in the broad sense of involving both perception and composition. It proposes software aesthetics as something with specific idiomatic qualities and as something that mixes with and recomposes other cultural forms, and that in turn may condense through the powers of art at a scale that disrupts other forces of aggregation and conformity. But this is a difficult question, one
that has to be taken as it comes; and with a dose of precision, whilst understanding that the modality of that precision itself interacts with what it works in the midst of.

The first chapter in this section, ‘Always One Bit More’, is a reading of theories of the experience of mathematical calculation, particularly found in the Intuitionist movement, an influence on the work of Alan Turing, as a means of reading computational artefacts and games such as Minecraft. Brouwer, a key theorist of Intuitionism, and, in certain texts, an advocate of mathematics as a mode of cosmic understanding more broadly, argues for calculation as an experiential form of life. This chapter argues that some of the ways in which computer games have been taken up and reworked by players echoes this thick involvement in numerical transitions. Like other texts in this book, one of the underlying arguments is that technical, and mathematical, documents, ideas and movements have notions of culture baked into them. Part of the work of culture is to draw these out and to see how they exceed themselves.

Building on such work, the chapter entitled ‘Computational Aesthetics’ is a somewhat programmatic statement of the impact and significance of the qualities of computation as a set of aesthetic principles. Whilst ineluctably unfinished as a statement of such a vast set of things, the mode of analysis offered in this chapter proposes grounds for critical work with computing in a way that may encourage taking its specificity into consideration. The chapter proposes a subset of characteristic attributes of computing that have particular significance in establishing its aesthetic valences. Setting out the propositions for such a set of traits implies neither the re-description of the contents page of a basic manual of computer science in more lofty terms, nor simply the correlating of existing and established entities in aesthetic theory with those that establish the circuits of computing, but instead seeing what might pass through and change such a set of terms from both computing and aesthetics and in doing so establish their translation and reconstitution.

This specificity also of course involves complications of scales of experience and analysis, and their conjunction with a massive array of movements and dynamics moving across and as them. The following chapter, ‘Phrase’, develops such concerns in relation to the way in which computational forms interweave with other aspects of material life, ranging from dance to voting systems. How, in turn, might we think through these conjunctions as temporary or enduring aggregates that form themselves as, in a term taken from dance, phrases of movement that cross between things as movements of
movement? Such transitions that aggregate then dissipate as a phrasal entity are particularly articulated in technological forms that thrive on conjunction.

Following such an approach, ‘Feral Computing’ elaborates an argument drawing on the question of ubiquitous and distributed computing in relation to the implications of recent developments in cognitive science, specifically those of the enactive, embodied, ecological and extended mind. Aesthetics is founded exactly where one is at this moment, but it also starts in the transitions across the gap between such a position and the movements and reflections that it takes to realize this state as a condition, and that therefore ungrounds itself. As computing moves into distributed forms that draw in spatial characteristics and conditions, and also starts to integrate with things that are more characteristically described in relation to thinking processes, it becomes wilder, begins to leak out of its boxes in ways that are similarly ungrounded. But calculation, experience and thought begin to have spatial characteristics. It is their particular texture as such that they gain the capacity to move from a model of computing as merely a closed mechanism of calculation to one that grounds itself in interaction. It is interactivity as an undergirding drive in the history of computing that, in this text, provides the means for computing to go beyond itself and into a further stage.

The condition of interaction has numerous modes. It also involves some that are more lackadaisical, networks of dawdling, skiving, goofing off, mass enactments of parodies of productivity staged via purposefully pointless protocols. This condition is explored in ‘Just Fun Enough To Go Completely Mad About’, which develops an account of recent debates in software studies by a reflection of the role of algorithms and processes in games. By contrast to the common argument for theorization of games by militantly enthusiastic gamers (whom I admire but can’t keep up with), it adopts a more spaced-out, daydreamy approach, one that takes pleasure in phenomena where logic, procedure and paradox are enacted by myriads of users. The two games discussed in this chapter are Agario and Twitch Plays Pokemon. Both embody, in different ways, an approach to the intercalation of rules, spaces, movement and interaction, amongst other things, that take them at a tangent. The latter game, or more properly, an event within the conjunction of two gaming systems, was remarkable in playing out some of the more recondite pleasures of play. The embrace of the ludicrous, indeed a founding of logical relations under the fullness of its moon, are what define and drive such games.
INTRODUCTION

The pair of texts under the title *Powers* addresses the way in which computational forms establish new grammars and forces of power, but also work to revise and reconstitute prior imaginaries of virtue. Such virtues may be found in conditions not only of communication, of direct manipulation of and access to data, of clarity of design and structure, but also of the reinvention of a broader sense of relation between the ideal user, or subject, and the world at large.

‘Black Sites and Transparency Layers’ analyses the formations of transparency, openness and their manifestation in recent ‘flat design’ interfaces and the changed human–computer interaction (HCI) of the smartphone and tablet, the new architecture of Silicon Valley company HQs, and the relation of these forms to the black sites of surveillance centres, server farms and related places. The chapter proposes that each grammar of transparency set in place by such systems and imaginaries also installs certain conditions of opacity that, whilst not inevitable, are unreflected upon or disavowed. No system of transparency-formation is identical to any other, each bringing its own intentional and incidental plays of shadow and light that are also not immediately parseable into a calculus of black and white. The essay proposes some diagnostic means to find the ways in which contemporary design, at the level of interface, architecture and process, articulates what becomes transparent, to what or to whom and how, and in what ways the pleasures and reasons of transparency are enforced or avowed. At present, Silicon Valley is in a moment of its pomp. The ways, after the revelations of Edward Snowden (that follow on those of the preceding years of the Five Eyes and Echelon systems), in which it inscribes transparency as an official good, in devices ranging from its operating systems to its palaces, are therefore instructive.

Following this, ‘Algorithmic Tumult and the Brilliance of Chelsea Manning’ attempts to map the political-computational condition of the present in relation to the themes of posthumanism, related to the feminist work of Rosi Braidotti, and in particular to the kinds of agency, understanding and skill deployed and embodied by Chelsea Manning. The article proposes that there is a particular conjuncture between two universal systems of equivalence: the Turing Machine – which integrates all symbol-based processes and media, whilst re-composing them in ways that are peculiar to networked and computational digital media – and capital, which providentially allows for the imaginary transduction of all entities, states and processes into numerical and tradeable equivalents. The particular conditions of this conjuncture are primary drivers of contemporary life in which new
means of action and subjectivation are both possible and urgently needed.

Part of the aim of this book then is to show that if computing is going to saturate everything, which seems tendentially, if by no means absolutely, to be one traceable trajectory for its unfolding, then as a field, as a science and as a manifold set of practices of numerous kinds, it can either be thought to reduce everything to its own terms and conditions, or generate a sufficiently complex relation to what it meshes with to recognize that it is, in turn, going to be mutated. One of the fields of such mutations is culture, something not reducible to the mere rubric of ‘digitization’. As a young science with, so far, only a few basic tenets, but with an enormous complexity of things arising from them, computing has much more to go through. This book attempts to map some of the ways this conjugation of culture and computing is underway.