TransCoder and The Soft Queer Body
A Critical Inquiry

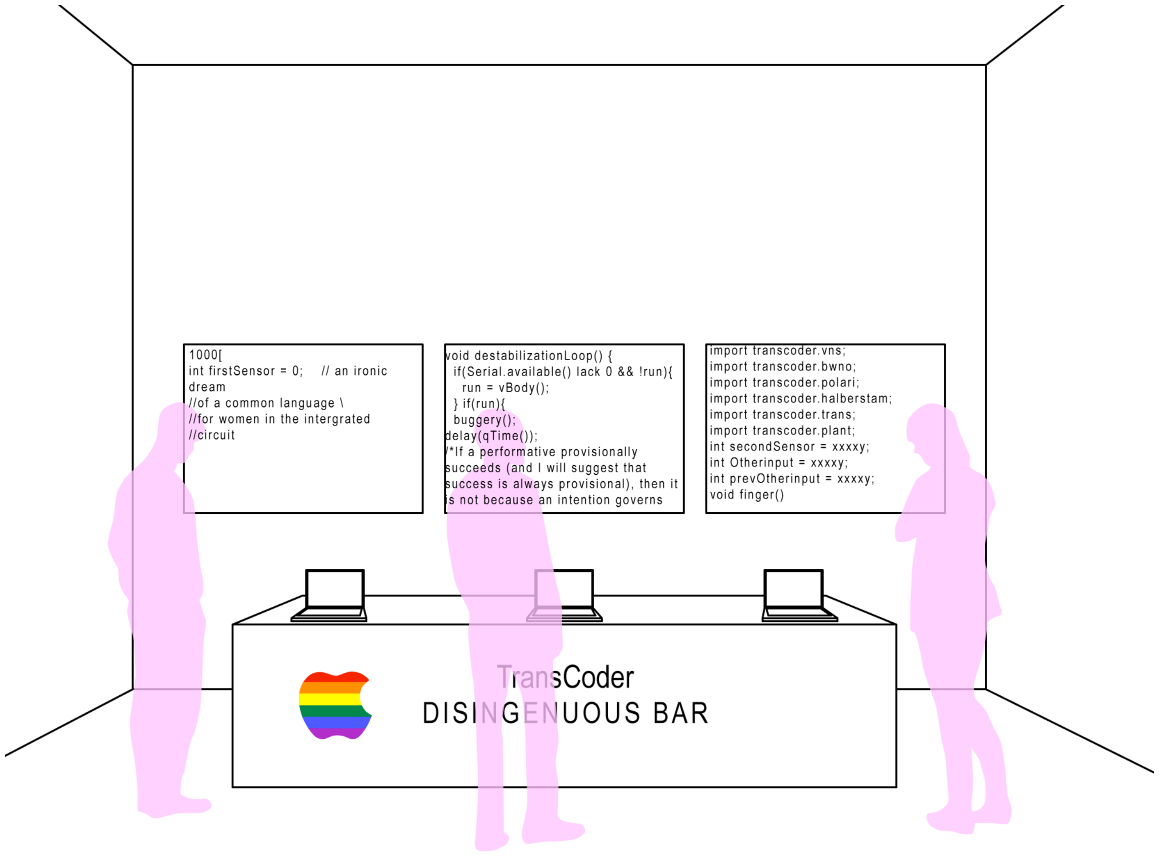
(forthcoming from A Minima *new media journal)*

(Due to recent surveillance of Lorenz Tunny—the creator of TransCoder, the author of this essay has chosen to remain anonymous.)

TransCoder: Mutating Soft Bodies

For homogeneity does not produce images: it produces the visual, otherwise put, "looped information." –Nicolas Bourriaud, *Relational Aesthetics*

According to Friedrich Kittler, language is moving beyond humans toward machines. –Wendy Hui Kyong Chun, *Control and Freedom: Power and Paranoia in the Age of Fiber Optics*



```
1000[
int firstSensor = 0;    // an ironic
dream
//of a common language \
//for women in the intergrated
//circuit
```

```
void destabilizationLoop() {
  if(Serial.available() lack 0 && !run){
    run = vBody();
  } if(run){
  buggery();
delay(qTime());
/*If a performative provisionally
succeeds (and I will suggest that
success is always provisional), then it
is not because an intention governs
```

```
import transcoder.vns;
import transcoder.bwno;
import transcoder.polari;
import transcoder.halberstam;
import transcoder.trans;
import transcoder.plant;
int secondSensor = xxxxy;
int Otherinput = xxxxy;
int prevOtherinput = xxxxy;
void finger()
```

TransCoder – Disingenuous Bar Installation

<u>Code, Gender, Ideology, Performativity</u>

Tunny's *TransCoder* project creates two reactions: riotous laughter or extreme confusion (and in this case, the two are not mutually exclusive). While *TransCoder* heavily relies on a comprehensive understanding of computer code and queer theory, for all its esotericism, the piece incites enjoyment from the enlightened and confused—whether by the "spectacle" (how I hesitate to use this word) of performative installation or the engagement it initiates with users.

TransCoder is a play on transgender and Lev Manovich's 5th principle of new media – transcoding. Manovich writes, "to 'transcode' something is to translate it into another format."[1] Within computing and new media, Manovich identifies a "cultural layer" and a "computer layer" affecting each other: "we can say that they are being composited together. . . . Cultural categories and concepts are substituted, on the level of meaning and/or language, by new ones that derive from the computer's ontology, epistemology, and pragmatics."[2] Defining transcoding as a "process of cultural reconceptualization" is one way, according to Manovich, to start thinking about the transition "from media theory to software theory" as a method for the analysis of new media forms and objects, looking beyond visual representation to underlying structures and logics.[3]

Tunny has designed *TransCoder* as an open source software application that interrogates how computer code operates within the artist's interests of performativity, gender, ideology, and queerness. Users conceptually and practically engage in an exploratory dialogue aimed to question the structure, logic, and semantic meaning at the basis of constructing computer code (an arduous task, make no doubt). Specifically, as a queer software application, *TransCoder* is devoted to rupturing the heteronormative superstructure that has infiltrated coding and software historically, discursively, and culturally. Tunny strives for a complete shattering of code's ontology. Viewing *TransCoder* as a "language" battle between seemingly disjunctive fields of discourse (computing and queer theory), Tunny desperately wants to sever ontological and epistemological ties to unquestioned technologies, to interrupt a flow of circulation

[1] Lev Manovich. *The Language of New Media* (Cambridge, MA: MIT Press, 2001), 47.
[2] Ibid., 46 – 47.
[3] Ibid., 47 – 48.

between heteronormative culture, coding, and visual interface. *TransCoder* stretches out to the sublime of destruction—this desired ontological rupture of functionality, designed to initiate a conceptual reassessment beyond the technical. Yet, the *TransCoder* software inherently contradicts Tunny's goal, as it is built with the "language" being critiqued.

At the *TransCoder* website, visitors are offered links to download the application and learn about the program, a full list of its coding libraries with explanations, a contribute section to further develop the program with external libraries, a message board, as well as an exhibition site for users to share work. *TransCoder* is *programmed* to transcode between Manovich's cultural layer and computer layer. The cultural layer of queerness, which, according to Judith Halberstam, consists of "subcultural practices, alternative methods of alliance, forms of transgender embodiment, and those forms of representation dedicated to capturing these willfully eccentric modes of being . . . [including a] nonnormative logics,"[4] acts upon and mutates mutually with the computer layer of algorithms, binary logic, data structures, code, software, and digitization. The application requires code to be imported from any programming environment. Once loaded into *TransCoder*, one begins to disassemble, reassemble, subvert, fuck with, and mutate the code using *TransCoder*'s libraries. Recognizing Wendy Chun's critique of transcoding as erasing "the computation necessary for computers to run,"[5] *TransCoder* does seem to be only aware of the transcoding process. To Tunny's credit, I will grant that this work is well aware that software is more than "translation," but *TransCoder* wishes to specifically focus upon the act of transcoding in order to explore how the culturally queer maps onto coding and structures of software.

[4] Halberstam, *Queer Time*, 1, 6.
[5] Wendy Hui Kyong Chun. "On Software, or the Persistence of Visual Knowledge" *Grey Room 18* (Winter 2005), 46.

When *TransCoder* is performed and installed, its effectiveness resides in Tunny's ability to cast an atmosphere of business cool "marred" by queer aesthetic. Playing upon The Apple Store's Genius Bar, Tunny has constructed a Disingenuous Bar identical in design: a white bar, wooden stools, even a team of "geniuses" in "company" T-shirts! However, the Disingenuous Bar's team of technocratic queers reassuringly makes no promises about computer "geniuses" offering false solutions to ideological—not technical—problems; rather, they guide users to explore and interact with *TransCoder*, helping import, mutate, and export code. For those uneasy about approaching the bar (and at art exhibits, so many people would rather observe than engage), projectors are connected to each computer, allowing others to see the processes and possibilities of *TransCoder*.

Instantly noticeable is the Disingenuous Bar's appropriation of Apple Computers' old rainbow apple for a logo. This symbol becomes problematic and incredibly paradoxical. While Tunny's use of the apple implies numerous readings, the apple brands the project in a confusing way. Did Apple actually fund this project? Is Tunny supporting Apple and using the logo because rainbows have become a symbol of GLBT culture? Of course, with Tunny, everything comes back to Alan Turing. Again relying on historical positioning, *TransCoder's* apple is used as an homage to Turing.[6] If *TransCoder* is attempting to return the bitten rainbow apple to the forefront as a visual reminder of the permanent mark Turing's death has left on relations of gender, sexuality, and technology, the meaning of the apple still remains blurry. A symbol of Turing or not—a visual site of

[6] From an interview with Tunny: "the transformation of Turing's death by biting into a poisoned apple into the logo of a company whose technological innovation is driven solely by corporate capitalism—not to mention controlled by a white heterosexual male—could be nothing more than exploitation."

struggle for queer freedom within technological infrastructure or not—the apple is unquestionably owned by Apple. No matter how carefully Tunny has conceptualized the apple as a symbol for thinking through alternatives to obtain freedom, the apple will always be read first and foremost as the apple of Steve Jobs. Judith Halberstam correctly acknowledges the apple as moving beyond a Tree of Knowledge to the "fruit of a technological dream."[7] The fruit of Tunny's dream is ultimately claimed by the global dream of corporate capitalism.

Without question, gender and sexuality operate as strong components of technological design. Gemma Shusterman has described sex in design as "an integral part of our commodity culture . . . reflect[ing] the complexities of capitalism," exposing how people desire each other through product—"desiring product."[8] Speedy sports cars are designed for heterosexual men to express their masculinity; pink cell phones become a normative feminized communications device. (As Shusterman points out, a man would not buy a hot pink cell phone unless he is actively trying to subvert his normatively assigned gender role.) However, looking underneath the "hood" of technology, traces of gender and sexuality become opaque and blurry. (Here, Tunny wants to look under the "hood" of Apple's logo, the Genius Bar, and computer code at the same time!) While Adrian MacKenzie notes the performativity of white, heterosexual male culture at play within the coding structures of Linux,[9] research into the transcodings between queer culture and computer code are lacking.

[7] Halberstam. "Automating Gender," 445.
[8] Shusterman. *PASS*, 51.
[9] See Adrian MacKenzie's "The Performativity of Code: Software and Cultures of Circulation" in *Theory, Culture and Society* (2005).

There has been much debate regarding the process of transcoding at the level of computer code: from Mark B. N. Hansen referring to digital code as "referenceless"[10] to Friedrich Kittler claiming software does not even exist[11] to cyberfeminist Sadie Plant finding cultural representations of gender and sexuality all the way down to the machine code of zeros and ones.[12] Alexander Galloway's explanation of code as a paradox posits these skewed interpretations into a somewhat stable middle: "software," he writes, "is both scriptural and executable. . . . software is both language and machine. . . . To see code as subjectively performative or enunciative is to anthropomorphize it, to project it onto the rubric of psychology, rather than to understand it through its own logic of 'calculation' and 'command.' [Yet, even as] representation as mathematical recoding, not as any socially or culturally significant process of figuration . . . at the end of the day what emerges is exactly that."[13] Galloway's acknowledgement of code constantly returning to its cultural formations recognizes the psychology continuously pouring into coding constructions. Kittler also acknowledges a grounding in natural language: "there would be no software if computer systems were not surrounded any longer by an environment of everyday languages."[14] Presenting itself primarily as a machinic language, code maintains the luxury of alluding critique as "natural" languages of speech and writing. However, as Manovich and Galloway exhibit with their concepts of transcoding and the paradox of code, software and code must be rigorously interrogated

---

[10] Mark B. N. Hansen. *Bodies in Code: Interfaces with Digital Media* (New York: Routledge, 2006), 37.

[11] See Friedrich Kittler's "There is No Software." ctheory.net (18 October 1995), http://www.ctheory.net/articles.aspx?id=74 (accessed on 1 April 2007).

[12] See Sadie Plant's *Zeros and Ones: Digital Women and the New Technoculture* (Fourth Estate: London, 1998).

[13] Alexander R. Galloway. "Language Wants to Be Overlooked: On Software and Ideology." *Journal of Visual Culture* Vol 5 No 3 (December 2006), 325 – 229.

[14] Kittler. "There is No Software," 3.

beyond the level of the machine—to the cultural—to obtain a comprehensive understanding of the power structures code uses to inscribe technically, discursively, historically.

The power and pervasiveness of code has incited an urgency into untangling its technical and cultural structures, and Tunny is embracing this moment of interrogation. If code is potentially replacing human language and becoming "the lingua franca of . . . all physical reality,"[15] then it must be assessed through culture *and* computing, alongside speech and writing. Indeed, Katherine Hayles writes "language alone is no longer the distinctive characteristic of technologically developed societies; rather, it is language plus code."[16] With computer code as the first truly "executable" language for a machine that conveys "meaning into action,"[17] it assumes the ability to move beyond Austin's speech act theory and physically alter whatever it communicates with.

Tunny undoubtedly sees computing as a field marked by the "murder" and mutilation of Alan Turing and as a field that has gone too long without adequate interrogation into how its cultural environment affects its technical environment, and in turn, how its heteronormatively "embodied" technical infrastructure affects the whole of its users' interactions and cultural representations. The *TransCoder* libraries lie at the heart of this matter. With contemporary queer communities forming, at least in part upon a philosophy of the theoretically queer, as a discursive tactic, *TransCoder* offers libraries rooted in specific theories of queerness. In as much of an esoteric, self-referential, and privately humorous methodology as the "rhetoric" that constructs the forms, regulations,

[15] N. Katherine Hayles. *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005), 15.
[16] Ibid., 16.
[17] Alexander R. Galloway. *Protocol: How Control Exists After Decentralization* (Cambridge, MA: MIT Press, 2004), 165 – 166.

and "identities" of coding methodologies, *TransCoder* uses the rhetoric of its people to ignite a "struggle" (Tunny would consider it nothing less) of mutual mutation, fusion, connectivity, and recombinant linguistics. Libraries include: 1) Haraway's Taxonomies for a Genderless Future, 2) Sadie Plant's 0 as 1 (Fuck Lacan), 3) Fantabuloso Discursivity (Anti-Language for Queer Liberation), 4) Halberstam's Technotopic Topologies, 5) Executable Speech Acts (Queering Speech, Writing, and Code), 6) Fisting as Friendship (Foucauldian De-Categorization), 7) Butler's Destabilization Loop (Citing the Other), 8) Cyborgian Non-Essentialist Posthumanism, 9) Trans Cut-Ups, 10) Planes of Queer Consistency | Bodies with New Organs, 11) VNSMatrixized GenderCode Fuck, and 12) mySoftQueerWare (customizable). The libraries are immensely populated with countless coding functions, operators, and variables. In fact, *TransCoder*'s libraries are so richly developed, they appear to be more comprehensive than commercial languages like Java, ActionScript, or Lingo.

The discourse associated with *TransCoder*'s libraries have mutated from a recognizable form of writing to a hybridized version of coding. Tunny could be immediately attacked for essentializing queer theory into the digital logic under scrutiny; yet, this would be reductive. Rather, Tunny allows queer theory to mutate and transpose in order to infect computer code's structure, logic, and "language." Tunny views queer theory as viral, accommodating itself in whatever form necessary. To restrict queer theory to the language it is originally written in would be the truly essentialist action (or non-action). If one were to consider *TransCoder* a battle against the mythology of language, Barthes provides the war cry: "the best weapon against myth is perhaps to mythify it in its turn, and to produce an *artificial myth* . . . Since myth robs language of something, why not rob myth?"[18]

TransCoder literalizes Galloway and Thacker's statement, "Today, to write theory means writing code."[19] Users work through libraries to reconstruct the structure, logic, and semantic of imported computer code to form new queer readings, considerations, and articulations of computer code's many ontologies and epistemologies. Based on results during the installation's opening (posted on the Exhibit link of the *TransCoder* website), imported code can morph to endless choices of queer non-essentialism: from Boolean statements transferring to a multitude of states beyond and between true or false, loops fluctuating wildly and unpredictably, if / then logic dissolving into if / if / if / if ad infinitum, small comments between pieces of code becoming digital manifestos for queer empowerment, the "logic" of queer discourse undermining control operators, variables stripped of heterosexist terminologies, to coding structures resembling passages from Butler, Haraway, or Irigaray rather than C++ or Java. In the end, everything is transitory and unstable—always becoming.

To take an example from *TransCoder*'s website, combining Butler's Destabilization Loop (Citing the Other) library with myQueerWare (customizable) mixes random operators and the destabilizationLoop() function to break apart and disassemble continuously iterating processes that bind. A for loop

```
for(int i=0; i<40; i=i+1) {
line(30, i, 80, i);}
```

can change to

```
destabilizationLoop() {
```

[18] Roland Barthes. *Mythologies* (New York: Hill and Wang, 1972), 135.
[19] Galloway and Thacker. *The Exploit*, 129.

```
for(int i= != < <= > >=0; i is anyNumber; i=i+anyNumber) {
line(i, i, i, i);}
}
```

Deleuze defined the codes of our time as “passwords”—control passwords that indicate “whether access to some information should be allowed or denied.”[20] If the “terrorism of code”[21] is always secured by our nation and conflated with personal freedom, what happens “when freedom is conflated with security”?[22] As Wendy Chun writes, “freedom is no longer free.”[23] Upon examining software, which comprise of coding structures, Chun describes its functionality as an ideology of imaginary relations: software “offer us an imaginary relationship to our hardware.”[24] Based on Althusser’s notion of ideology as representing “the imaginary relationship of individuals to their real conditions of existence . . . [and as a material that] always exists in an apparatus, and its practice, or practices,”[25] software, inscribed by code, offers users a set of imaginary situations formulated by concealment of code and simulated visual metaphors of “user-friendliness.”[26] This user friendliness—commonly “referred to as *your* preferences”—offers “‘choices’ [that] limit the visible and invisible, the imaginable and the unimaginable.”[27] As a result, Deleuze notes that “the digital language of code . . [makes] individuals become ‘dividuals.’”[28] Alan Liu points out that “the ‘user friendly’ face of

---

[20] Gilles Deleuze. “Postscript. On Control Societies.” *Negotiations* (New York: Columbia University Press, 1995),180.
[21] Jean Baudrillard. “Requiem for the Media.” *The New Media Reader* (Cambridge, MA: MIT Press, 2003), 285.
[22] Chun. *Control and Freedom*, vii.
[23] Ibid.
[24] Chun. “On Software,” 43.
[25] Althusser, Louis. “Ideology and Ideological State Apparatuses.” *Lenin and Philosophy and Other Essays* (New York: Monthly Reviews Press, 2001), 109, 112.
[26] Chun. *Control and Freedom*, 20
[27] Chun. “On Software,” 43.
[28] Deleuze. “Postscript,” 180.

information . . . [technology is] strangely cold,"[29]generating a "remoteness" between user and computer that becomes frozen over by a technological "cool."[30] Alexander Galloway echoes this remoteness with his explanation of obfuscation in software as "what you see is not what you get . . . code is never viewed as is."[31] Back to Chun: "In order to understand control-freedom, we need to insist on the failures and the actual operations of technology."[32] How is the corporate cool of technological infrastructure and security ideologically limiting, exploiting, and erasing users of computing technology? (This is one of those colossal questions Tunny is asking with *TransCoder*.) Considering that software is rooted in "a gendered system of command and control,"[33] how is the heteronormative infrastructure of technological innovation affecting queer users? Are computing technologies more user friendly to heteronormative *dividuals*? Down to the level of code, how is queerness executed, performed, and interpellated by technologies? Could it be, as Friedrich Kittler has suggested, "we simply do not know what our writing does"[34] anymore, or, as Chun states, "we do not and cannot fully understand nor control computation"?[35]

For code to execute, it requires "reflection,"[36] which consists of defining the "complete syntactic and semantic rules of a computer language . . . before the real 'language' takes place," that is, in advance of interpreting, parsing, or executing the

[29] Liu. *The Laws of Cool*, 76.
[30] Ibid., 76.
[31] Galloway. "Language," 325.
[32] Chun. *Control and Freedom*, 9.
[33] Chun. "On Software," 27.
[34] Kittler. "There is No Software," 2.
[35] Chun. "On Software," 44.
[36] Galloway. "Language," 322.

code.[37] Without initially establishing the laws that the computer code must follow, upon any type of execution, the code will fail to cite pre-established rules—these "unforeseen articulations . . are essentially dismissed out of hand as errors or 'exceptions.'"[38] Citation plays a critical role in implementing technological performativities of gender. As one continuously experiments with *TransCoder*'s libraries, the incredibly vague goal of the program is revealed as citational rupture within performatives of computer code. It is worth quoting Judith Butler at length on the operation of citationality within performative structures to fully illustrate the citational process (and to give Tunny thorough theoretical backing):

> If the power of discourse to produce that which it names is linked with the question of performativity, then the performative is one domain in which power acts *as* discourse. Importantly, however, there is no power, construed as a subject, that acts, but only, to repeat an earlier phrase, a reiterated acting that *is* power in its persistence and instability. This is less an "act," singular and deliberate, than a nexus of power and discourse that repeats or mimes the discursive gestures of power. Hence, the judge who authorizes and installs the situation he names invariably *cites* the law that he applies, and it is the power of this citation that gives the performative its binding or conferring power. And though it may appear that the binding power of his words is derived from the force of his will or from a prior authority, the opposite is more true: it is *through* the citation of the law that the figure of the judge's "will" is produced and that the "priority" of textual authority is established. Indeed, it is through the invocation of convention that the speech act of the judge derives its binding power; that binding power is to be found neither in the subject of the judge nor in his will, but in the citational legacy by which a contemporary "act" emerges in the context of a chain of binding conventions. . . . If a performative provisionally succeeds (and I will suggest that "success" is always provisional), then it is not because an intention governs the action of speech, but only because that action echoes prior actions, and *accumulates the force of authority through the repetition or citation of a prior, authoritative set of practices*. What this means, then, is that a performative "works" to the extent that *it draws on and covers over* the constitutive conventions by which it is mobilized. In this sense, no term or statement

[37] Ibid., 322, 325.
[38] Ibid.

> can function performatively without the accumulating and dissimulating historicity of force.[39]

Wendy Chun observes: "to emerge as a language or text, software and the 'languages' on which it relies had to become iterable."[40] Code becomes a "collective agency in the process of constituting itself . . . [moving in a] cultural life of . . . circulation,"[41] continuously a performative citing sets of prior authorizations. As a nexus of power, code circulates through previously established structures of control historically, socially, and technically. Code installs this power not through execution but its programmed reflection: not how variables and establishing commands are set but how the structure, logic, and semantics of code (the very form code takes) cite an invisible technological superstructure that is formulated by a politics of gender, sexual, racial, and class bias. The form a programming language takes—its functions, operators, structures, variables—suggests a language developed only on technological and mathematical logic, functionality, and practicality. Yet, a programming language uses these seemingly objectified decisions based on "scientific rationale" to *cover over* the conventions that it continuously cites. These conventions are grounded in corporate capitalism, heteronormativity, militaristic innovation, and governmental surveillance and control. Quoting Chun: "software—something theoretically (if not practically) iterable, repeatable, reusable, no matter who wrote it or what machine it was destined for. Programming languages inscribe the absence of the programmer and the machine in its so-called writing."[42] This absence is filled by conventions driving code's citational

[39] Judith Butler. *Bodies That Matter: On the Discursive Limits of "Sex"* (New York: Routledge, 1993), 225 – 227.
[40] Chun. "On Software," 30.
[41] MacKenzie. "The Performativity of Code," 73.
[42] Chun. "On Software," 30.

performatives. If software is both "ideology *and* ideology critique . . .a concealing and a means of revealing,"[43] as Chun suggests, then *TransCoder* is a program operating on both levels: as ideology, concealing the internal structure that runs its application,[44] and as ideology critique, by revealing coding structures to mutate as a form of critical engagement.

Adrian MacKenzie has noted that the "cultural life of code in circulation"[45] operates as a powerful performative. Referring back to Gemma Shusterman's writings on sex in design, it is easy to see how citational performatives play out through use of a speedy sports car or pink cell phone to lock normative gender assignments to repetitive powers that bind. Put simply, the continuous use of these technologies interpellates a subject's identity into a heterosexual matrix. However, on the level of code, usually invisible and running on potentially never-ending loops, interpellation operates abstractly. The historical, cultural, discursive, citational legacy of code—that which gives code its performative force—interpellates users (or dividuals) "as concrete subjects."[46] Whether or not you drive a speedy sports car to pump up your masculinity or use a pink cell phone to accessorize your femininity, as you interact with contemporary technology, an inaudible "'Hey, you there!'"[47] resounds within executions of code . This paradoxically silent hailing binds you as a subject to code's repetitive loops, citing a legacy of heterosexist power and control that subject you—interpellate you—as a subject bound within its structure as an ideological state apparatus. The computer, that apparatus,

[43] Ibid., 44.
[44] Although its source code is available to view, Tunny fails to address this meta level of ideology at work throughout the functional code running the *TransCoder* program.
[45] Ibid., 73.
[46] Althusser. "Ideology," 117.
[47] Ibid., 118.

simulates its ideology and repetitive citations of power within visual constructs of user friendly and cool—software. Importantly, you are not required to acknowledge this interpellation like the hailing performed by the policeman; rather, you acknowledge as you interact and use technology. In this circulatory system of citation and code, queerness—never "programmed" into the infrastructure—has historically been dismissed "as errors or 'exceptions.'"[48] The paradox of Tunny's work resides in *TransCoder*'s dual use of programming: while queerness *is* programmed conceptually into the project, the actual code running is as normative as normative can be (give or take a few comments). Yet, Tunny is careful to define *TransCoder* as primarily a thought experiment, exploring how the microphysics of code's power interpellate bodies into soft(ware) bodies of subordination. *TransCoder*, Tunny states, produces "The Soft Queer Body,"[49] outside all subordination. Arguably not evasive of subordination, The Soft Queer Body is a nexus of networked, posthuman queer resistance.

The Soft Queer Body

Tunny's notion of The Soft Queer Body resists against interpellation. Straight or gay, to identity as either implies that at some point a person was identified as gay or straight and self-identified by saying to themselves, "'He/she/it must mean me.'"[50] Yet, queerness remains too dynamic for interpellation to always succeed in hailing a queer subject. If we are in what Deleuze has described as control societies, "bodies are consonant with more distributed modes of individuation that enables their infinite variation."[51] The Soft Queer Body—a hacked network of software, code, technology, and

[48] Galloway. "Language," 322.
[49] Tunny. Personal Interview. 11 May 2007.
[50] Samuel Delany. *Times Square Red, Times Square Blue* (New York: New York University Press, 2001), 191.
[51] Galloway and Thacker. *The Exploit*, 47.

flesh—does not fall slave to "masterdiscourses"[52] of normative control but resists ideology "by any means necessary."[53] The Soft Queer Body—"anti-aesthetic," "anti-scientific," "contaminated," "techno," and both hard and soft—exists as a posthumanism dedicated to generating "multiple viabilities"[54] for queer freedom. As a multiplicitous assemblage—a desiring machine—on fields of consistency, The Soft Queer Body infects the matrix of heteronormativity, producing social interstices—subcultural places, heterotopias[55] (think the Disingenuous Bar), and new technologies of becoming. The Radical Software Group have touched upon this notion of interstitial technology with *Notes for a Liberated Computer Language*,[56] in which the group creates conceptual coding structures to incite thought outside of current technical means of coding and computing.

I will argue that *TransCoder* interacts with code as a body: a soft body, a body politic, a posthuman body, a body without organs on planes of consistency. It is clear by the play on transcoding and transgender that mutation is at the site of and extends beyond the biological body. The code that *TransCoder* mutates becomes a queer body, a trans body, a body with new organs, a cyborg body, a viral body. Working with mutated code, one can feel its desire to rapidly spread and infect. *TransCoder* has built into its application an export feature that bundles altered code into various file formats and provides "Network Infections" for distributing mutated code on the Internet. Upon selecting a Network Infection option, *TransCoder* automatically sends out queered code

---

[52] Judith Halberstam and Ira Livingston. "Introduction." *Posthuman Bodies* (Bloomington: Indiana University Press, 1995), 2.
[53] Ibid., 15.
[54] Ibid., 1, 18.
[55] See Michel Foucault's "Of Other Spaces." http://foucault.info (accessed on 12 April 2007).
[56] Project located at http://r-s-g.org/LCL/ (accessed on 15 March 2007).

to specified email accounts, blogs, and posting sites to individuals, businesses, institutions, governing bodies, and other appropriate venues for distribution. For example, after using cyberfeminist libraries Sadie Plant's 0 as 1 (Fuck Lacan) and VNSMatrixized GenderCode Fuck, I sent mutant code through the network infection Big Daddy Mainframe, which sent my code to various email contacts at high tech companies with white heterosexual males as C.E.O.s, such as Microsoft, Apple, Dell, and Gateway.

TransCoder wants to extend beyond the individual user's experience; it struggles to move past the affective experience of the normative body to bodies within technological networks. How can code exist, spread, mutate, and populate through technological networks? How will computing systems attempt to read *TransCoder's* code? How will programs fail when this code is imported into an application for use?[57] The act of infecting—of sending out—seems to be crucial. Results are not necessarily important; rather, it is the act of releasing *TransCoder* code out into the networked world, allowing it to spider through unwelcomed spaces / places and poison infrastructure with traces of queer techno viral bodies. It is as if *TransCoder* has generated a digital AIDS virus and sent it out into the world as a new queer weapon of hybridized warfare. Here, we arrive at a sublime of technological queer destruction; yet, *TransCoder* seems to reach this point in concept and thought only. For what is really at stake—and note the cynicism—is receiving illegible code like so much spam and promptly deleting it.

New technological codes, like *TransCoder*, *do* form queer "anti-languages" as tools for new subjectivities, connections, and disruptions within languages subordinated

[57] Tunny already knows of several incidents of this!

by heterosexist citational performatives. Paul Baker, writing on Polari,[58] describes anti-languages as "a means by which an alternative social structure (or reality) could be constructed. [Anti-languages are] generated by anti-societies and in their simplest forms are partially relexicalised languages, consisting of the same grammar but a different vocabulary in areas central to the activities of subcultures."[59] Baker even defines anti-languages as a "sociolinguistic *coding* orientation."[60] If in the past queer communities have used subcultural structures like anti-languages to develop their identities, now, computer code *with* language has become re-structured to create a contemporary anti-language for queer cyborgian identities. Cyberfeminist group VNS Matrix use this anti-language in their manifesto when they write, "we speak in tongues . . . terminators of the moral code,"[61] and again in their "Bitch Mutant Manifesto": "eat code and die . . . SUCK MY CODE."[62] VNS Matrix have created their own hybridized, technological, posthuman anti-language with no concern for bourgeois morale or "proper" uses of language. When they incite readers to "eat code and die," the statement exhibits possible effects that normative code can have on the queer body—heteronormative constructs conceptually and physically kill queerness. Yet, the same proclamation can also be a provocative alluding to "SUCK MY CODE"—a command of queer anti-language that oozes with "abjection"[63] and perverse sexuality, refusing to disentangle culturally queer viral infections from "purely technological" code. "Eat code and die" is an open call for a

[58] Polari is a homosexually coded speaking "language" used in the UK by queer communities between 1940 – 1980.
[59] Paul Baker. *Polari: The Lost Language of Gay Men* (London: Routledge, 2002), 13.
[60] Ibid., 13.
[61] VNS Matrix. "Cyberfeminist Manifesto."
[62] VNS Matrix. "Bitch Mutant Manifesto." http://www.aec.at/meme/symp/contrib/vns.html (accessed on 20 April 2007).
[63] VNS Matrix. "Cyberfeminist Manifesto."

queer virus to contaminate all heteronormative languages. With *TransCoder* and VNS Matrix, code is a language *trick* to automate perverse possibilities.

The question that Tunny's work consistently asks—what is queer technology?—(while offering no specific answers) has returned us to Halberstam's concept of mutual mutation in queer technotopias: the criterion for a queer technology has become the modes in which queer bodies interpret, intercept, and infect technologies while technologies continuously interpret, intercept, and infect queer bodies.