

THE ERGODICS OF READING MOO

A Non-Trivial Pursuit

Katherine Parrish

I don't think that Espen Aarseth (1997, 1) intended the opening chapter of *Cybertext* to be comic, yet the notion of "ergodic literature", texts whose traversal requires "non-trivial" effort, never fails to make me smile. I am hard-pressed to think of a truly interesting text that has required less than trivial effort from me as I forged a path through it. All joking aside, I am aware that Aarseth, in his definition, is referring specifically to the physical or extranoematic work required to construct the text. While I still have difficulty maintaining the distinction between the work of my hands and the work of my head, this contribution to the field of cybertextual studies has inspired me to think more generally about the work involved in reading digital texts, and what might be called the Cybertextual work ethic. I hope Aarseth will forgive me if I abuse his term. I am too lazy to invent a new one.

As a teacher working in MOOs, I have witnessed hundreds of people encounter these esoteric virtual environments for the first time. There is nothing trivial about the mental and physical efforts required to begin to orient oneself in MOOs, even to learn such basic gestures as saying "hello", or to be able to locate oneself in relation to one's surroundings. To then move from these somewhat passive engagements to more active, constructive participation requires yet more knowledge, time, patience and effort. Exponentially increase these efforts, and you might learn to program a box of donuts. So, when I contemplate the dearth of truly interesting textual production in MOOs, or the lack of an audience for those pieces which do engage the space in compelling ways, such as John Cayley's *myour darkness*, or Jane Love's *MOO Scream*, I find myself wondering, "Maybe it's just too hard".

When has reading ever been easy? As a teacher in the highschool classroom, yet another virtual environment, I am constantly exhorting my students to work harder at their reading. I often think that most of my work lies in changing the posture of my students from one of passive reception to active engagement. I read to them Jeanette Winterson's words:

The solid presence of art demands from us significant effort, an effort anathema to popular culture. Effort of time, effort of money, effort of study, effort of humility, effort of imagination have each been packed by the artist into the art. Is it so unreasonable to expect a percentage of that from us in return? (1995, 17)

No sooner are the words out of my mouth than I feel that I want to retract them. Can we use terms like the "solid presence of art" when we consider work in digital media, whose presence is more transubstantial than substantial? And what of artists such as Kenny Goldsmith who deliberately attempt a "value-less practice"? In these works, the labour negotiations between artist and consumer have been suspended, the contract broken, and the attendant collateral of guilt need never be paid. I do not wish to essentialize the production and reception of art as Winterson does. After all, idleness is an art in itself, one which I hope to pursue in the most trivial way in the very near future.

Nevertheless, the complexity of the ideas I hear expressed in contemporary poetics seem to yearn for a more complex vehicle, a more complex medium, and these ideas and these media may demand no insignificant effort from readers and writers. For the page, as many have pointed out, has been exhausted. Indeed, some of the most interesting texts in digital media are being written by poets whose relentless search for new expressive forms have taken them deep into the world of the programmer. They have learned JavaScript and Lingo, Flash and Perl. Yet, with a few notable exceptions, these innovative practitioners have not turned their investigative eyes towards MOOs.

MUDs may have a lot to answer for here. Role-playing MUDs, the ancestors of MOOs, are narrative spaces in the most traditional sense, rigidly bound by narrow conventions such as representation, naturalism, linear narrative, consistency in story line and character development. Many MOOs are still haunted by MUD references: administrators bear the title of "wizard", participants are "players". In cybertext theory, MOOs are still primarily characterized as narrative or theatrical environments. And

while I agree with Loss Pequeño Glazier when he says that narrative itself is not the problem, I do think that unless we use a completely different point of departure in our discussions of these environments, we will not see the potential of MOOs as sites of innovative literary practice. (Glazier, 2000)

We need to read MOOs more carefully. Of all new digital literary media, MOOs are the least understood, and the most misrepresented. I have read countless pronouncements about what goes on in MOOs that are simply inaccurate. Often these reports come from those who have not learned the MOO programming language and this inhibits their understanding. This paper does not wish to suggest that one must plummet the depths of MOOs in order to read them at all. A superficial reading can be just as legitimate a reading as any other. But it should not be mistaken for a complete reading – just as an in-depth examination of the relationship of the code to the output of a digital text should not be essentialized as the correct way to read the work. I do strongly suggest that there is much to be gained from such an in-depth reading, and that without this kind of reading and this kind of reader, it is highly unlikely that we will begin to see what the MOO can really do. Those who do possess a fairly fluent grasp of the MOO programming language are not necessarily interested in the literary significance of its operation. I, myself, do not pretend to understand this significance fully. What I present here is the result of some of my efforts to read MOOs. It is by no means comprehensive, and I am quite prepared to find that I have, once again, completely misunderstood my subject. I do hope, however, to frustrate some earlier readings of MOOs which would reduce their complexity, limit their capacity, and render them unattractive to the digital literati who might finally do something interesting with them.

Reading the Player-object

Alison Knowles: George, couldn't we extend the scale further and have a whole city as the contents of a book? Huge pages as the side of an office building, a single page with water pouring down the front in the park, a page out flat as bridge, as door to the city?

George Quasha: I've dreamt of such a city all my life, the house or castle with a vast interior. In the transviromental city one travels to the depths of inside while extending the roads and gardens of

outside. It's all one transrupting surface where anything leads to anything else, governed only by imaginal necessities. The book is the instrument by which we tap in on the reading that produced it. We not only could read it, we live in it. Our limbs leave invisible pollen on the pages for the next readers. (Quasha 1996, 93)

To enter a MOO is to enter a transvirological architecture such as George Quasha envisioned, with one significant difference. The human interactors in a MOO leave very visible traces on its pages. The moment a person connects to the MOO, she is written into the text: several messages are generated to announce the character's entrance, some of which are logged, others merely witnessed by other human interactors or MOO objects. Reading does equal writing the MOO. Like a time-traveller in a teen sci-fi movie, one cannot avoid leaving one's mark on the unfolding story.

The human self in MOO is represented by the player-object, so called because the MOO is an object-oriented programming environment, and everything within its walls is a coded object of some kind. Hence, all subjects on the MOO are, in fact, objects. The player-object, once created, becomes part of that MOO's database, and it is always located somewhere in the MOO, represented as "sleeping", waiting to be animated by its human interactor. No one really leaves the MOO, not entirely. Each player is present, named through the language, the player-object a waiting mask or persona, an iconic representation of the self. In the postmodern age of the simulacra, it might seem that the surface has effaced what lies beneath, cutting our projected, performed selves loose, letting them drift from their origins as far as the signifier is removed from its referent. Yet the cyborg self dances in the interzones. The relationship between authentic self and persona is not binary, but supplementary, co-creative, liminal. This is where the theatrical metaphor of the player-object as mask begins to break down. The dormant player-objects are not static shells. They are an iteration of the performance of the MOO itself, a result of the MOO's text generative activity, subjects in process. Not only so, but these player-objects may possess a degree of agency, independent of their human animators. For the player-object is just that, a piece of code; it may be written and rewritten to perform in many ways. I have written a variety of text-generative processes into the "say" verb, the command that controls speech, onto Salmon, one of my player characters. Through a happy accident, I also wrote a verb that allows any other player to force my character into utter-

ance. The text is generated by random procedures, which undermines my control over what I might say at any given moment. I also have limited control over *when* I might say it: Salmon can be made to speak by another player even when I am not logged on, when I have no immediate control over her actions. One might say that she talks in her sleep.

The self in a MOO is a collaborative creation between human interactors and the algorithms of the MOO. The self in MOO becomes a potential site of creative activity. To program the player-object is to engage in work that most closely resembles the body modifications of Stelarc or Orlan. This is dangerous and exhilarating ground. We court the posthuman – a word that still raises the hackles of cybertext authors and theorists alike.

In last year's edition of the *Cybertext Yearbook*, John Cayley imagined that text generative processes might be used in MOOs to dynamically generate a player's description, and thereby draw attention to the relationship between textual manipulation, textual representation, and identity (2001, 90). I imagine this, and much more. I imagine grotesque couplings of player-objects and generic containers, or player-objects and generic rooms. Such a player could truly say, "I contain multitudes". Maybe such creatures exist. In ten years of MOOing, it seems absurd to me that no one would have attempted to forge such monsters. If they do not yet exist, I believe it is because even experienced MOOers are somewhat leary of the notion of the self as programmable object.

Reading the Stage: MOO as Agent

All the MOO's a stage since all the men and women are player-objects.

(Burk, 1998)

This whimsical epigram reinscribes the division between actor and setting, subject and object, a division that the MOO capriciously frustrates. In my blasphemous vision of hybrid player-room-objects, the men and women could just as easily be the stage, and the MOO itself is just as much a player in this comedy of errors as any human-driven player-object. The MOO itself barely discriminates between human-driven objects and others, casting the props in the starring role. I recently was reading a

MOO transcript of an online session where my character, Salmon, had pointed to my “generic recording device”. The transcript read back: “Salmon points at you”. What kinds of literary experiences are possible when reader, author, and textual object are given the same subjective status?

In *Computers as Theatre*, Brenda Laurel’s influential exploration of computer-human interaction, the computer interface is recognized as the “shared context for actions in which both (person and computer) are agents” (1991, 4). No action on the part of a human player may come to fruition in the MOO unless the MOO hears it, understands it, and responds to it. The dramatic truth of this is revealed if, as a player, you move yourself to the location -1. This is the void of MOO space, a virtual sensory deprivation tank. While in -1, a player cannot hear, cannot speak, cannot move. All of these functions are found in the code for the room-object, not the player-object. Thus, a player’s very creative agency is dependent on the room she is in. The walls have ears, eyes and tongue. These rooms do, themselves, have human authors. If we want trace the path back to first causes, we could say that all player interaction on a MOO has been partially authored by Pavel Curtis, who wrote Lambda MOO and the Lambda Core from which all MOOs are either direct or indirect descendants. Nevertheless, writing the MOO is always a multi-authored, collaborative endeavour. Trying to trace the lineage of certain objects, or the algorithmic genesis of various lines of code is often a futile project. These tangled skeins of code tend to shape the warp and woof of the MOO in ways that become increasingly difficult to predict. Because the authorship is so distributed, the possibility for a poetics of non-intention blooms in the MOO.

Reading the GUI

In recent years, many MOOs have embraced the transition from command-line interfaces (CLI) to graphic user interfaces (GUI), giving users a different set of “windows” into the environment. The most prolific of these interfaces is the enCore/Xpress interface developed by Cynthia Haynes and Jan Rune Holmevik, which allows users to connect to the MOO from their web browser rather than through “raw telnet”, or by having to install one of a variety of MOO telnet clients. With the introduction of the web-based interface, MOOs have now become sites of multi-media expression – a point which is often ignored in the continuing debate over the

superiority of “text-only” MOOs over those with GUIs. For example, Bradley Dilger, contesting the “ideology of ease“ which he says underlies the move from text-based MOO interfaces to GUI, describes the encore interface in these terms:

The enCore/Xpress combination integrates the graphical and text-based interfaces, making it possible for students to move between the two sorts of interaction with a minimum of difficulty. The Xpress interface also offers graphical tools for creating objects in the MOO. I think this is its greatest asset. (Dilger 2000)

Dilger gives no acknowledgement to the varieties of expressive and creative potential afforded by a multimedia environment, and in this, he is not alone. Many critics of GUI MOOs see the interface as only a “cheat”, an attempt to stay on the surface of things, to move away from the experience of the MOO as a coding environment to the AOL experience of “All the fun of the internet without a computer“ (Dilger 2000). Dilger is one of a large body of MOO researchers who use MOOs for teaching in Rhetoric and Composition undergraduate courses. It is perhaps understandable, though lamentable, that this body would privilege the word over the image. If the MOO is perceived exclusively as a narrative environment, I suppose the addition of graphics could be seen as a regressive step towards the “picture book”. And while I have a great fondness for picture books, I have found that GUI MOOs do tend to undermine my own personal sense of the immersive experience of entering a virtual environment, whereas text-only MOOs tend to enhance this sense. However, as James Sempsey indicates in his reflective piece, “When is the MOO too Gooney?”, the disparagement of GUI MOOs and MUDs is widely held among MUD researchers in all fields, not just those who have a vested interest in the textual dynamics of these spaces. Sempsey laments this easy dismissal and urges researchers to take a reasoned critical look at these new kids on the virtual block. (1998)

When one does so, one quickly comes to an understanding that a MOO with an encore/Xpress interface is a beast of even greater complexity than a text-only MOO. The enCore/Xpress interface preserves the command line interface with its split screen presentation. The MOO users’ gaze here moves perpetually from text to image and from image to text. This text generally (though not exclusively) encourages an ekphrastic reading, a transparent gaze where one looks through the MOO to a world created in

the mind's eye. The graphic image discourages this activity, calling attention to itself.¹ In this, the enCore MOO's interface can be read itself as symbolic of the simultaneous radical artifice and naïve transparency characteristic of much contemporary literature. (Lanham 1991)

In Marjorie Perloff's essay, "After Language Poetry", she describes what she terms differential poetry: "poetry that does not exist in a single fixed state but can vary according to the medium of presentation: printed book, cyberspace, installation, or oral rendition". In these varieties of expression, no one iteration is privileged over another; there is no original. I submit that the enCore/Xpress interface not only supports but necessarily performs a differential poetics. This happens at the most basic level because MOOs which use an Xpress interface still allow users to connect to the MOO through a non-graphical telnet client, and many users do so. The telnet MOOer and the GUI MOOer experience dramatically different performances of the MOO text. Moreover, in the Xpress interface, the verbal text is often presented on both the left side of the interface and the right. The dividing line that dissects the MOO window suggests that though these are co-dependent forms, they can be viewed independently.

Perhaps the most dramatic differential readings of texts in the Xpress MOOs occur because of inconsistencies in the integration of the command line and GUI operations that exist in even the most recent implementations of the Xpress interface.² For example, currently, when a player enters a room, a text message announcing the arrival is generated and displayed to all other objects (human driven or otherwise) in that location. However, depending on whether the player moved locations by typing a command in the text-entry box on the left side of the GUI, or by clicking on the hyperlinked exit name in the right side of the interface, the MOO will generate a different text message.

The following message is generated by using a text command to move into a room:

Salmon arrives.

If, instead, Salmon enters the room by clicking on the hyperlinked exit name, this message is displayed:

Salmon has arrived.

Small difference, but hugely significant if one is programming a bot to respond to a player's entrance – a fairly common practice.³ Bots programmed to respond to the text “arrived”, will be mute when anyone “arrives” in the room. When I first noticed this inconsistency, my first impulse was to compensate for it, to mask it, by programming my bots to respond in identical ways to either “arrive message”. I now see this glitch as potentially very interesting, and I hope no one fixes it.

Reading the Core

Most MOOs begin with a core database – a collection of pre-packaged MOO objects – and the server, which reads and compiles these objects. The core objects are arranged in a numerical hierarchy, from #2 to #79 or thereabouts, with the lowest numbers held by the most fundamental objects of the MOO – the root class object, the generic wizard, the generic room, the generic thing, etc. There are a handful of core databases that anyone may download to seed their own MOO. Though each of these cores has its own particular flavour, there really is very little substantial variation from core to core. A \$room is a \$room is a \$room. Since every object in a MOO is a descendant of one of these core objects, one cannot hope to find much that is surprising in the MOO landscapes that are generated from these very generic cores. I wish to suggest that the core objects, those essential building blocks of MOOs from #2 to #79 are themselves just one reading of a text that is gloriously open. This text is contained in the server, which is the heart of the MOO programming language. Why does this language need to generate \$room after \$room after \$room? If one were to alter the basic architecture of MOOs so that the fundamental building block was not a \$room or even anything resembling a Euclidean space, would it still be a MOO? I, for one, am not ready to define what properly constitutes a MOO and what does not. I am ready, however, to see creative programmers take full advantage of the fact that the MOO is an open source project. The core objects may be altered and hacked to suit wildly different visions.

Reading the Language

Before we see genuinely new performances of the MOO, we need readers who are willing and able to read the source text with a critical and creative eye. One such reader is Jason Nolan. In a forthcoming paper called “The Technology of Différance”, Nolan critiques the MOO programming language at the most fundamental level for its privileging of the English declarative statement. Current multi-lingual MOOs provide only a superficial solution to this problem, by translating the help files and commands into a variety of languages and hacking the database so that it can work with the 16-bit ascii characters required for Chinese and Japanese character sets.⁴ Nonetheless, MOO servers still can only parse statements that adhere to a subject-verb-object linguistic structure, and this not only privileges English discourse, but it severely limits the kind of expression that can occur in MOOs. As Nolan reflects, “even more exciting would be the ability to have the MOO parser understand commands in past and future tenses, and in the passive voice”. (Personal correspondence) Nolan is currently embarking on a project that will revise the MOO programming language; in so doing, he shares the desire of poet Erin Mouré:

To try to move the force in language from the noun/verb centre. To de/centralize the force inside the utterance from the noun/verb, say to the preposition. Even for a moment. To break the vertical hold. To empower the preposition to signify and utter motion, the motion of the utterance, and thereby Name. (1998, 94)

Reading the Manual

Even given these constraints, the MOO programming language holds great expressive potential. Nevertheless, while the language parses familiar English constructions, it is still a foreign language. For the MOO author who has no background in object-oriented programming, the Lambda MOO Programming Manual might as well be written in machine code. Very few guides exist for the true “newbie”. Even the best of these guides tend to

gloss over some of the server functions that might be of the greatest interest to the literary practitioner. For example, every MOO has a set of string utilities, a set of built-in commands that operate on strings of text. The command `$string_utils:explode` delimits a given string according to a specified character. The following is a string of words stored in the `.description` property of object #1366 on Project Achieve. In other words, it is that object's description, returned to a player any time that player looks at that object. #1366 is a `$room` object. When a player looks at the room, this is what she sees:

The room is dimly lit, save for a pool of light baptizing the pool table near the back. A few chairs and tables are scattered amidst the debris. The faint echo of music catches your ear: a twangy old guitar playing a sweet sad song.

The following command calls the MOO to delimit this string according to the letter "s":

```
;$string_utils:explode(#1366.description, "s");  
which yields the following result:
```

```
{“The room i”, “dimly lit, “, “ave for a pool of light baptizing the pool  
table near the back. A few chair”, “ and table”, “ are “, “cattered amid”, “t  
the debri”, “. The faint echo of mu”, “ic catche”, “ your ear: a twangy old  
guitar playing a “, “weet ”, “ad “, “ong”. }
```

Again, if people continue to exclusively represent the MOO as a space for traditional narrative experiences, it is highly unlikely that they are going to advertize the existence of the `$string_utils:explode` command. Yet such a command is clearly going to be of interest to the large body of contemporary poets for whom language is the most compelling domain of play. Other string utilities perform a reverse function on the string, strip the string of a given character, search and replace, capitalize, remove punctuation, etc. There is even a built-in command which calls up the alphabet:

```
;$string_utils.alphabet  
“abcdefghijklmnopqrstuvwxy”
```

The MOO programming language has an intimate relationship with words and their constituent parts. It seems to beg for logophilic artists to form a *menage a trois* of literal play. Moreover, all MOO cores contain a range of mathematical and statistical utilities that have yet to be exploited in the ways that artists such as Jim Rosenberg, M.D. Coverley, and Neil Hennessey have explored these functions in other media, to great effect. The MOO can calculate combinations and permutations, manipulate a Fibonacci series, simulate randomness, perform *pi* in the virtual sky.

Is there No Easy Way to Read This?

The ergodic nature of MOOs brings another crashing blow against the still resonating assertions that MOOs are somehow non-hierarchical spaces. Though MOOs suggest a more equitable landscape since text-only MOOs use a technology that is accessible to a twenty-year old computer, the expenditure of time required to become MOO literate constitutes a very real digital divide. This might suggest a bleak future for truly interesting literary work in MOOs. Even if the MOO is able to seduce artists with sufficient time and inclination and perception to take advantage of its virtual potential, who will read these works? Although MOO texts are generally not meant to be ignored, the fact that they very likely will be ignored is not necessarily a cause for lamentation, or a reason to write them off as trivial pursuits. As Nick Piombino notes in, "Cultivate your own Wilderness", an essay in the collection, *The Politics of Poetic Form*, "poetry can germinate and grow quite excellently in the arid desert of practically no response whatsoever". (1990, 232) Charles Bernstein, who edited the volume, echoes Piombino's subsequent assertions that the very existence of the unread poem can indeed transform society. In his words, "I've never been to Alaska, but it makes a difference to me that it's *there*". (1990, 241) I am not convinced that the MOO poetry I yearn for *is* there. And since I am not a poet, I remain a voice crying in this wilderness. I am, admittedly, half-crazed by my visions of the virtual text that is MOO. I will wander here chewing on locusts and honey, waiting. For the harvest is plenty, but the workers are few.

This paper does not wish to suggest that one must plummet the depths of MOOs in order to read them at all. A superficial reading can be just as

legitimate a reading as any other. But it cannot be mistaken for a complete reading – just as an in-depth examination of the relationship of the code to the output of a digital text cannot be essentialized as the correct way to read the work. I will assert, however, that there is much to be gained from such an in-depth reading, and that without this kind of reading and this kind of reader, it is highly unlikely that we will begin to see what the MOO can really do.

NOTES

1. The semiotics of the graphic and textual components of the MOO are far more complex than the reading given here. I explore these in much greater depth in the forthcoming paper, “@chparenting the MOO: notes towards a MOOpoeitics”.
2. I write this just following the release of version 3.02 of enCore/Xpress.
3. It is also interesting when you consider the nature of the representation of time in the MOO, but that’s another paper.
4. Nolan’s *Project Achieve* is one MOO that supports this capability.

REFERENCES

- Aarseth, Espen (1997) *Cybertext: Perspectives on Ergodic Literature*. Baltimore: The Johns Hopkins University Press.
- Bernstein, Charles (1990) "Comedy and the Poetics of Political Form." in Bernstein, Charles (ed.), *The Politics of Poetic Form: Poetry and Public Policy*. New York: Roof.
- Burk, Juli (1998) "The Play's the Thing: Theatricality and the MOO Environment." in Haynes, Cynthia & Holmevik, Jan Rune (eds.), *High Wired: On the Design, Use, and Theory of Educational MOOs*. Ann Arbor: University of Michigan Press.
- Cayley, John. *myour darkness*. pURL MOO. [http://purlmoo.uib.no:8000/@go myour darkness](http://purlmoo.uib.no:8000/@go%20myour%20darkness)
- Curtis, Pavel. *LambdaMOO Programmer's Manual*. [Ftp://ftp.lambda.moo.mud.org/pub/MOO/html/ProgrammersManual_toc.html](ftp://ftp.lambda.moo.mud.org/pub/MOO/html/ProgrammersManual_toc.html)
- Dilger, Bradley. "Interface Interference: America Online, the Ideology of Ease, and MOO Virtual Realities." [Http://web.nwe.ufl.edu/~dilger/s00/interface_interference.html](http://web.nwe.ufl.edu/~dilger/s00/interface_interference.html)
- Glazier, Loss Pequeño (2000) "Between the Academy and the Hard Drive: An E-cology of Innovative Practice." *Open Letter* 10:9, Fall 2000.
- Glazier, Loss Pequeño (2001) *Digital Poetics. The Making of E-Poetries*. Tuscaloosa: The University of Alabama Press.
- Goldsmith, Kenneth. "Uncreativity as a Creative Practice." [Http://www.alienated.net/article.php?sid=5](http://www.alienated.net/article.php?sid=5)
- Lanham, Richard (1989) "The Electronic Word: Literary Study and the Digital Revolution", *New Literary History* 20, 265–290.
- Laurel, Brenda (1991) *Computers as Theatre*. Reading: Addison-Wesley Publishers.
- Love, Jane. *MOO Scream*. LinguaMOO. [http://lingua.utdallas.edu:7000/@go MOOscream](http://lingua.utdallas.edu:7000/@go%20MOOscream)
- Mouré, Erin (1998) *Furious*. Toronto: Anansi Press.
- Perloff, Marjorie. "After Language Poetry: Innovation and its Theoretical Discontents." [Http://www.alienated.net/article.php?sid=62&mode=thread&order=0&thold=0](http://www.alienated.net/article.php?sid=62&mode=thread&order=0&thold=0)

- Piombino, Nick (1990) "Cultivate your Own Wilderness", in Bernstein, Charles (ed.) *The Politics of Poetic Form: Poetry and Public Policy*. New York: Roof.
- Project Achieve*. Toronto: University of Toronto. [Http://achieve.utoronto.ca:2221](http://achieve.utoronto.ca:2221)
- Quasha, George (1996) "Auto-Dialogue on the Transviromental Book: Reflections on the Book of the Bean", in Rothenberg, Jerome & Guss, David (eds.) *The Book, Spiritual Instrument*. New York: Granary Books.
- Sempsey, James (1998) "When is the MOO too Gooley?", *Journal of Virtual Environments* 3:2. [Http://www.brandeis.edu/pubs/jove/HTML/v3/sempey-comment.html](http://www.brandeis.edu/pubs/jove/HTML/v3/sempey-comment.html)
- Winterson, Jeanette (1995) *Art Objects*. Toronto: Alfred A. Knopf.