

Leonardo FloresUniversity of Puerto Rico
San Juan

DHTML Dances: The Making of an E-Poet

Abstract: Canadian poet, programmer, and artist Jim Andrews lived and worked in Seattle 1997–2000, participating in the exuberant economic and technological growth known as the dot-com era. Andrews’ DHTML poems engage the materiality of Web technologies from this moment in computational history and were instrumental in his formation as a poet. This article performs media-specific analyses on *Enigma n* and *Seattle Drift* to contextualize and demonstrate Andrews’ evolution into an e-poet.

Key words: Jim Andrews, DHTML, e-poetry

Canadian poet, programmer, and artist Jim Andrews lived and worked in Seattle 1997–2000, participating in the exuberant economic and technological growth known as the dot-com era. This was a vibrant period of creative growth for Andrews, who developed his website *Vispo.com*¹ from a simple online gallery into an authoritative site in which he still publishes his work. More importantly, his artistic practice shifted from creating static visual poetry for print and/or Web publication to programming kinetic and interactive works that offered tools for his audience to participate in the creative process.

It is necessary to reconstruct some of the technical history of the Web during late 1990s because Andrews’ poetic works relied on and responded to the computational environments developed and standardized during that time. As a programmer for dot-com companies, and for his own creative interest, Andrews learned to produce works in Dynamic HTML (DHTML) – a combination of Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and the Document Object Model (DOM). This was made possible by the publishing of JavaScript and DOM standards by the World Wide Web Consortium (W3C) in 1997. “The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document”.² In a nutshell, the DOM presents a hierarchical representation of an HTML document in which every element is treated as an object in the browser’s memory which can be manipulated using JavaScript. This functionality did not arise without its complications, primarily the fierce competition for dominance of the browser market popularly known as the browser wars.

¹ J. Andrews, *Vispo: Langu(im)age*, <http://vispo.com>, 1996 (access: 15.06.2014).

² *HTML DOM Introduction*, <http://www.w3ctutorial.com/html-dom/dom-intro>, 2011 (access: 15.06.2014).

The browser wars began over DHTML. (...) Microsoft and Netscape each wanted its version of DHTML to become the industry standard. (...) While the W3C was drafting a recommendation for DHTML, versions 4 of Internet Explorer and Navigator each added a number of proprietary DHTML elements that were completely incompatible with the other browser. As a result, when working with advanced DHTML techniques such as animation, a programmer had to write a different set of HTML code for each browser. Unfortunately for Netscape, the W3C adopted as the formal standard the version of DHTML found in version 4 of Internet Explorer, which prompted many loyal Netscape followers to defect to Microsoft.³

The complicated development environment caused by the browser wars led many Web developers to shift to more standardized proprietary tools, such as Macromedia (now Adobe) Director and Flash, particularly as they became optimized for the Web and viable as authoring systems. This period saw the rise of Flash as the multimedia development tool of choice and the waning of the older and more powerful Director, in part due to its higher bandwidth requirements. Other older technologies, such as VRML and animated GIFs allowed embedding small objects into an HTML document that a user could manipulate, but none were as generative for Andrews as DHTML.

But what is DHTML? It allows people to make Web documents that change in appearance and function quickly. More generally, it turns documents into programs. When we look at documents on the Web, we see text and graphics and controls and so forth. But upon understanding the basics of DHTML, we begin to see the neath text, what's unseen but present in the source code and begin to reconceptualize the document as a collection of objects with properties that can change as the reader reads. The objects can also respond to changes in other objects or initiate changes in other objects. And changes can be caused either by the underlying logic of the neath text without the reader's intervention or be caused by the reader's responses to the visible manifestation of the document.⁴

The practice of "turning documents into programs" is a significant step in his development as an e-poet because, up to that point, Andrews used programs to create documents, but now the product and process became conflated. We can track this shift with two key works: *Seattle Drift* (1997) and *Enigma n* (1998). These DHTML poems were the testing ground in which he found his voice as writer of electronic poems, imbuing his writing with textual behaviors.

This article's methodology is combines new and traditional critical practices, what I have elsewhere referred to as close reading 2.0. The basic idea is to pay close attention to a works' text including media-specific analysis and code readings. N. Katherine Hayles is a key reference point for media-specific analysis, from her 2002 book *Writing Machines*, her essay *Print is Flat, Code is Deep: The Importance of Media-Specific Analysis* to her recently edited book *Comparative Textual Media*, she has been instrumental in focusing scholarly attention on the material conditions under which texts are produced. On the same vein, perspectives such as Platform Studies, Software Studies, and Critical Code Studies have drawn

³ D. Gosselin, *JavaScript: The Web Technologies Series*, "Cengage Learning" 2010, p. 4.

⁴ J. Andrews, *Infoanimism*, <http://www.vispo.com/animisms/enigman/EnigmanInfoanimism.html>, 1998 (access: 15.06.2014).

attention to the computational conditions for the production of texts. My own contribution has been a typology of textual behaviors observable in digital media, establishing a critical vocabulary used in my dissertation⁵ (from which this article is adapted), my I ♥ E-Poetry scholarly blogging project, and described in a recent article, titled *Digital Textuality and its Behaviors*.

It is a brief list of characteristics which I have observed in e-poetry, along with some basic subcategories, that should describe a wide range of behaviors programmable into electronic texts. One could think of this as a taxonomy, which can be used to tag different textual behaviors within an electronic text.

- Static texts are the default we're used to in print – they are texts that do not move or change on the screen.
- Scheduled texts may reveal themselves over time, which may be linear or looped; they may force a rate of reading by disappearing or scrolling; they may also trigger events over a programmed or random schedule.
- Kinetic texts move on the screen: this motion may be looped or linear, random, programmed, or responding to cues from the reader.
- Responsive texts take advantage of the computers' interface devices (most commonly the mouse and keyboard) to create a feedback loop between the reader and the text.
- Mutable texts involve programmed or random changes or may be generated on the fly.
- Aural texts have a sound component: verbal, musical, or simply noise.

These categories are not by themselves unique to electronic media, nor are they mutually exclusive. They are often found in combination and in some cases they are inseparable—aural and kinetic texts are always scheduled, for instance.⁶

Because electronic literature, such as that of Jim Andrews, explores the materiality of a moment in computational history, it is important to pay close attention to how he deploys his poetic strategies in meaningful ways. This article will analyze the more minimalist poem *Enigma n* and then focus on *Seattle Drift*, which while earlier, offers a more complex poetic engagement of digital media.

A Meaningful *Enigma*

Jim Andrews' second DHTML poem, *Enigma n*,⁷ consists of a series of operations one can run on the word “meaning”. After an epigraph by Phyllis Webb, “The world is round. / It moves in circles”, the reader is presented with the word “meaning” (in green) placed in the center of a black screen, and the words “prod”, “stir”, and “tame” (in gray) on the top left corner right after the title of the poem, which is also in green.

⁵ L. Flores, “Typing the Dancing Signifier: Jim Andrews’(Vis) Poetics”. Dissertation submitted to the Faculty of the Graduate School of the University of Maryland, College Park, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2010.

⁶ L. Flores, *Digital Textuality and its Behaviors*, “Journal of Comparative Literature and Aesthetics” 2013, vol. 36: 1–2, p. 81–103.

⁷ J. Andrews, *Enigma n*, <http://vispo.com/animisms/enigman/>, 1998 (access: 15.06.2014).

As one clicks on the options in the menu bar (which is the function of the gray words), the text is set in motion, stopped, and changes its appearance, while new options appear after exploring the options offered. Once the options “About” and “Run Away” appear in the menu bar, the reader has the opportunity to end the poem and go either to the essay *Info-animism* or to *ADVEXP: xes*, a work by Ted Warnell that opens with:

>> The meaning of anything
>> is open to vast interpretation.
>> But that does not mean
>> X means anything
>> you want it to mean.
>> What do you mean?⁸

The version I am describing is the most recent one: the one updated in 2004 by Marco Niemi and Jim Andrews to be more compatible with W3C standards and therefore with more browsers. When written and published in 1998 there were two distinct versions: one for Netscape Navigator 4 and one for Internet Explorer 4, and the gatekeeper was a page titled “meaning.html” that detected which browser one was using and sent one to the appropriate link. If one was using Safari, Opera or some other browser, one would get an error message and wouldn’t be allowed to read the poem.⁹ The Internet Explorer version was basically the same as the current version, even though it ran a little differently in terms of animation speed and other minor ways. The Netscape Navigator version was different in several ways:

- Its menu text reads slightly differently: “Prod meaning Stir meaning Tame meaning” instead of the single word “Prod Stir Tame” in the Internet Explorer version.
- It lacks the options “1/0”, “Color”, and “Discombobulate” which are present (and supported) in the Internet Explorer version.
- It had a “drag” function, which allowed the reader to drag the letters of the text. This was not present (or supported?) by the Internet Explorer version and has been discontinued in the current version.

The 2004 code revision collapsed the dual versions to include a single version that works on most browsers. An interesting detail is that this version’s source code alludes to a disabled (“detracted from focus” in Andrews’ words) “follow function”, in which the letters would follow the pointer in the window. This,¹⁰ along with the drag function, shows a conscious decision on Jim Andrews’ part to focus the texts’ responsiveness (and therefore the reader’s interaction) on the toolbar on top of the window, making it more like an application and less like a document. It also makes the text more like a toy that follows instructions rather than one that can be manipulated directly or that responds directly to the reader’s symbolic presence in the browser window. In other words, we can make the letters dance

⁸ T. Warnell, *ADVEXP: xes*, <http://www.warnell.com/advexp/advxes.htm>, 1998 (access: 15.06.2014).

⁹ The message read “Your browser does not seem to understand DHTML. Netscape 4 or IE 4 should work if you have a PC. I’m not sure which Mac browsers support DHTML”.

¹⁰ Source code for *Enigma n*.

for us, give instructions for them do dance differently or stop dancing, we can even take our pointer and join them in the dance.

If we were to apply this statement to the meaning of the words “meaning” and “enigma” perhaps this poem is talking about the ways we interact with the language we read, whether on the page or screen. The meaning of something may be enigmatic, and we can turn words around, play with them, put them in motion or stop them, change their appearance in diverse ways, but we cannot directly manipulate them: they will always be just beyond our grasp, always something of an enigma.

I will explore this idea further through a detailed description of e-poem as a first generation electronic object. The paradigm that inspired Jim Andrews to write this work is that of the document functioning as an application, a point he discusses in the companion essay *Infoanimism*. According to Andrews, this paradigm draws attention to the neath text, because it allows us to “conceive of an onscreen word as being an object with its own properties and behaviors”.¹¹ These properties and behaviors can usually be observed and determined from the displayed text, but given the potential for different interpretations of the programming codes by different browsers and different platforms, particularly when dealing with client side programming, it is useful to look under the hood (so to speak) to see how it is put together.

Enigma n is a perfect example of this. I had read *Enigma n* many times with Firefox (my browser of choice), and played abundantly with its options. When I started examining the source code, I realized that I was missing a component of the text, so I opened the document in Internet Explorer and saw what I had been missing: mousing over different words or parts of the document displays additional texts in the status bar. Here is a breakdown of the words/objects that trigger the display of these neath texts, the conditions in which they appear, and what effect they have upon the central word in the text (“meaning”) when clicked on:

As must be obvious by now, this document functions as an object-oriented computer program that runs on the browser to treat all of its displayed elements as objects, each of which has potentially variable characteristics, behaviors, and conditions under which these characteristics and behaviors can change. The document “enigman.htm” is created and operates under the visual metaphor of the application, by creating its own menu bar and employing the browser’s status bar. Each of the gray words in the menu bar (with the exception of “About” and “Run Away”) somehow affects the appearance and behavior of the letters of the word ‘meaning’ in the center of the screen. I specify letters because each one is an object that can be affected differently with each mouse click, as is the case with option 3 in discombobulate. The variables affected for each letter are the following: size, color, movement, rotation (further diversified by variables for axis, direction, radius, speed, and overlap). With clear labels for providing input to some of these variables, Jim Andrews keeps the tone playful rather than intimidating, framing the work as a “toy for poets and philosophers from the age of four up”.¹²

¹¹ J. Andrews, *Infoanimism*, op. cit.

¹² J. Andrews, *Enigma n*, op. cit.

Table 1. Responsive Objects in *Enigma n*

Object responsive to mouseover and/or conditions for display	Text displayed in status bar	Behavior of „meaning” upon mouse click
blank spaces and the word “meaning”, when starting the poem	“by Jim Andrews, 1998”	None
blank spaces and the word “meaning”, after mouseover of any of the words in the toolbar	“It is the world that you love, after all, is it not?”	None
“Enigma n” (in green letters on menu bar)	“Enigma n home.”	None. Link returns reader to introductory page.
“Prod” (gray, green while activated – applies to all subsequent words)	“Meaning prod. Prod meaning.”	Each letter begins to rotate on a different axis, direction, radius, and speed.
“Stir”	“Stir meaning. Meaning stir.”	“the letters have a common center [for rotation]”
“Tame”	“Repeated tamings collapse meaning within itself.”	“the letters have a common point of intersection”
during mouseover immediately after mouse click over gray words in menu bar, until you mouse away	“Meaning is yours to discover and create”	None
“Spell” (appears after first 3 gray words are clicked on)	Spell meaning out. Spell for literalists.	Spells out “meaning” in the center of screen.
“0/1” (appears after “spell” and another gray word are clicked on)	“Freeze/thaw meaning.”	Stops and starts motion and size change of letters (in case of “discombobulate”. Does not affect color change.
“Colour” (appears after “0/1” is clicked twice in a row)	“Colour meaning.”	Changes letter colors.
“Discombobulate” (appears after “colour” is clicked)	“Resize meaning.”	Changes discombobulate mode: “no font size change occurs” “the font size changes the same for all letters” “the letters change in size differently”
“Speed” (appears several menu clicks after “Discombobulate”)	“Adjust speed of meaning.”	Provides chart on left hand side of screen with 30 speed settings.
“About” (appears, along with “Run away”, after speed has been adjusted)	“About meaning.”	Links to “About Enigma n” page.
“Run away”	“Dialog with Ted Warnell”	Links to <i>ADVEXP: xes</i> by Ted Warnell.

Source: L. Flores, “Typing the Dancing Signifier: Jim Andrews’(Vis) Poetics”. Dissertation submitted to the Faculty of the Graduate School of the University of Maryland, College Park, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2010.

So how is *Enigma n* poetic and philosophical? The poetic approach is clearly Lettriste, atomizing the word “meaning” into its component parts, letters, charging each one with individual color, sizes and behaviors, and giving the reader control to set them in motion. The title *Enigma n* is an anagram of “meaning” and between the title and the word, he raises the question that the e-poem is designed to provide some answers to. What is the meaning of a word, when its letters are scrambled, spinning on different axes and directions, changing colors and sizes? Do those letters mean the same thing, or do new meanings emerge from these motions? Is the meaning an enigma, with n number of possible interpretations? All these questions are the kind that Lettrisme wants us to ask of language. Another poetic tradition that informs *Enigma n* is Concrete poetry, particularly if one uses the “1/0” option to pause the twirling letters to form a constellation of letters, in the style of Eugen Gomringer, for instance.

For a reader to read “meaning” here she would have to break several „rules” of reading, particularly those that deal with sequence. But some words do emerge without too much effort: “amen” and either “nig” or the reverse, “gin”, which might make more sense. How does this lead us into insight or discovery? Is this an intended combination, attributable to Jim Andrews? The answer to that last question is both yes and no. Jim Andrews created a machine made of words with the potential to arrange the component letters of “meaning” into many permutations, and he did it with that purpose, as suggested by the anagrammatic title. However, by granting the readers control over when to read – whether it’s during full speed animation, a slowed down version, or a completely paused still image as shown above – Jim Andrews has placed responsibility over the interpretation of the text on the reader. If something doesn’t make sense, they can stir, prod, or carry out any number of operations before pausing and reading again, and again, until they find an arrangement that seems readable and therefore meaningful. In a way, this resembles a Rorschach blot test, in which whatever the viewer sees in the random ink blots, is more of an indicator of their own state of mind than of any inherent meaning in the random marks themselves. And that is the point of this poem: meaning is an enigma, to be resolved by the reader through manipulation and play with the linguistic signs of the word “meaning”.

By creating the letters as objects, imbuing them with behavior and variable physical characteristics, and creating a playful interface for the readers to manipulate them, Jim Andrews has said what he needed to say about the meaning of language in an electronic, programmable environment: what the letters of a word may spell out in any given moment is shaped as much by the way they “hang out together” as by what the reader needs or wants them to spell. The great British poet W.H. Auden once said that he would give less chance of success to a young writer who said he had something to say than he would to a writer who said that he liked to watch the way words hang around together. DHTML allows writers to make documents in which words hang around together and interact with each other and with the reader and possibly with other documents and readers on the Web in ways that can be relevant to what Auden said but in radically different ways than he had in mind.¹³ *Enigma n* is designed to lead letters to hang out together in different ways – lined up in unchanging sequences readable from left

¹³ J. Andrews, *Infoanimism*, op. cit.

to right to form clear words, splayed on the surface of the screen for us to try to combine them into words, or spinning and changing colors and sizes in ways that are practically unreadable as words. It is up to the reader to make sense of these letters, or even transcend the need for letters to mean something, simply taking pleasure in their playful dance.

Drifting from the Scene

This piece was written, in part, in response to the questions about text and poetry that the new medium prompts in me. I wanted the text of *Seattle Drift* to initially look like plain ordinary text so that the contrast is more apparent. The questions that I have about text and poetry prompted by the new medium are similar in fashion to those prompted by abstract art about art and representation. Both prompt, rather than raise the questions directly.¹⁴

Seattle Drift (1997) is the most popular poem from this DHTML period in Andrews' career, and an important expression of his poetics. It was published at a time when e-poetry was starting to grow as a Web-based art form, yet it doesn't seem as dated as other contemporary e-texts. Its simplicity of design and directness of its conceit are deceptively simple: this e-poem enacts a critique of current and historical poetry scenes in order to create a space for a new e-poetry scene. This is also a piece that speaks well to contemporary audiences, particularly when being introduced to the concept of electronic poetry, because it takes as a starting point a traditional notion of poetry (verse) and leads the reader into the new scene of electronic poetry, with a concrete poetry transition to smooth the way.

At a glance, this e-poem looks very much like a contemporary free verse poem: divided into unrhymed, unmetred lines, and a straightforward text. A central poetic device at work here is personification of the text itself, which is the voice of the poem. This self-aware voice of this poem has found expression in other works by Andrews written during this period, such as the *Pop-up Poems*.¹⁵ The self-referential tone draws attention to the poem's form, leading the reader to look at the words rather than just reading them, an important aspect of Andrews' visual poetics. The only elements that would be out of place in a print poem are the red words found above the poem's title, which are options for the reader to activate with a mouse click, evoking the familiar computer interface of the menu bar. The presence of these input cues and the personified text's request to the reader are the keys to this poem's conceit and to its behaviors.

Stylistically, the piece is similar to the pop-up poems (though not in behavior) in that the text talks about itself. I like this approach because it focuses attention on the questions and also allows me to develop character. The character is the text itself, and the character commenting on its own nature and behavior, though embodying that nature and behavior also, beyond it but within it, like the rest of us.¹⁶

¹⁴ J. Andrews, *Seattle Drift*, source code, <http://vispo.com/animisms/SeattleDriftEnglish.html>, 1997 (access: 15.06.2014).

¹⁵ J. Andrews, *Pop-Up Poems*, <http://www.vispo.com/popups/popups.htm>, 1996–1998 (access: 15.06.2014).

¹⁶ J. Andrews, *Seattle Drift*, *op. cit.*

The speaker of *Seattle Drift* characterizes itself as “a bad text” that “used to be a poem”. It has apparently ceased to be a poem because it “drifted from the scene”, yet what we see is clearly a poem. Why would it make such a contradictory statement? Instead of providing reasons or further explanation, the speaker shifts to a request: to “do” it, reiterating that it is all it wants. Perhaps once the reader “does” the text, such explanations will no longer be necessary. Perhaps the text itself derives some pleasure from being “done”. In this context the three phrases in red gain more meaning: “do the text”, “stop the text”, and “discipline the text”. The connotations of the words “do me” and “discipline the text” combined with the texts’ statement that it is “bad” create a slightly sadomasochistic scenario, involving the reader in the punishment/pleasure of the text. This is emphasized when the pointer becomes a hand when it is placed over the red phrases. If the cursor is the symbolic presence of the reader in the electronic texts, is the extended index finger in that hand a representation of the reader’s hand, about to touch the text? Symbolism aside, this is an indication that the phrase is a hotspot, or to use less suggestive language, an input cue.

Initially, the only input cue that produces a response is the first one, “do the text” because the other ones depend upon this one to have a noticeable effect. Upon activating it with a mouse click, the words begin to drift across the screen in a random pattern that slowly leads the words to the edges of the screen and out of sight. Once the text is in motion, one can “stop the text” or “discipline the text”, which brings it back to the original formatting and static behavior. The responsive behavior of the text is very simple, requiring minimal input, yet important because it triggers three other textual behaviors: kinesis, mutability, and scheduling. The text is static without the reader setting it in motion by “doing the text”. The responsive element needs to be activated by the reader for him or her to be able to read the e-poem properly, or at least fully. The mutability becomes clear upon subsequent “disciplining” and re-“doing” of the text, because the drifting is subtly different every time. The use of time (or scheduling) is linear and infinite: the text can potentially drift for as long as the browser and computer are turned on, which raises questions about the limits of the “scene” the poem refers to.

The poem’s motion, key words, and spaces are essential to unlock this e-poem. It may not be surprising that the words drift across the screen when activated, given the poem’s title. And yet, this movement brings attention to the use of the word ‘drift’ in the title and the text of the poem. One of the primary uses of the word is to describe the motion of the words – random, uncontrolled, slow – but it can also mean moving away, perhaps without a clear direction. The poem states that it used to be a poem “but drifted from the scene”, establishing a causal connection between its drifting and it no longer being a poem. The word “scene”, can be read to represent a theatrical stage. The space in the window that one can see without needing to scroll, and the words of the poem are like theater performers, who eventually exit the stage after a performance (or do they drift out of the scene?). We cannot read the words we can’t see, though we can chase them with the scroll bars. Another meaning of scene is that of poetry scenes, that is performance spaces, traditions, audiences: the cultural milieu in which poetry is written, published, performed, and received. The poem’s strategies (including rhetoric, behaviors, and potential readings) stem from the latter meaning of scene and will inform this analysis.

What we see at the beginning of *Seattle Drift* is easily identifiable as a poem, at least in its formatting. It is written in free verse that reads somewhat like a William Carlos Williams, Marianne Moore, or a Charles Bernstein poem, in the sense that it is prosaic in its use of language, unadorned by figurative language, figures of speech, rhyme, meter, rhythm, or other devices that we would readily identify as poetic, with the exception of the personification of the poem itself. The line breaks coincide with the ends of sentences or independent clauses, so the lines are all closed, that is, each contains a complete idea. So, while it is not a very “poemy poem”, as Jim Andrews describes more traditional poems, we can identify it as a poem primarily because it is cut into lines, which makes it verse. We can say it belongs to the general scene of poetry, focusing it further as belonging to a poetic scene associated with Language Poetry, which often subverts the conventions of poetry by using prosaic language that has been cut into lines.

To “do the text” is to allow the poem to drift away from that scene of poetry that we can define as verse, or at least prose cut into verse. As the drifting begins, the words lose their line formatting, as they drift in different directions, at times over each other, yet maintaining their horizontal alignment, so they are still readable. The original linguistic text breaks down as new ones emerge from the newly reconfigured words, something that goes on continuously until the reader clicks on “stop the text”. At that point the text becomes static, its linguistic text stabilized, and the reader can read the newly combined text... or should I say poem? A screen capture after a few seconds of drift should provide some insight (see figure):



Is this text still a poem? Certainly. It looks like a poem by Stephane Mallarmé, Charles Olson, Susan Howe, Eugen Gomringer or any poet belonging to the Concrete Poetry scene – and these are just a few examples of poetic traditions that use spatial arrangement in ways beyond the traditional formatting of verse. We can give this new linguistic text even more attention we gave to the original one and get insightful material to inform our interpretation, because it has become less familiar and perhaps more poetic language. For instance, the opening lines can be read as “text I’m SEATTLE I Drift” after which the reader must make decisions on what direction to read and how to cluster words in ways that make sense (or produce pleasurable nonsense). This drifted text retains its personified sense of self, as seen in phrases like “I DRIFT”, “Poem want I you bad”, “be used” and “do a the me”. It also retains some of the language of desire to be “done”, that is, to be cut loose from what ties it to specific scenes and drift away into other poetry scenes.

And yet the personified text claims that it is no longer a poem, because it “drifted from the scene”. Perhaps it drifted from the traditionally-formatted-poetry scene to join the spatially-composed-poetry scene, but will it drift out of that too? Given enough time, all the words will drift from the screen, leaving only the initial input cues. At what point does it stop being a poem? At what point does it cease to be a poem, but is it still a text? When there is only one word? If we consider “the scene” to mean poetry traditions and/or venues, then the poem is questioning the scene itself, and the personified text is being rebellious. If we read “the scene” as a theatrical metaphor, then the text is being self-conscious about its materiality, and refers to its literal departure from the figurative stage of the screen. Does an empty page mean there is no text in it? Certainly.

Does the same apply to an empty screen? Not in the case of *Seattle Drift*, because as long as the program is running, the text is drifting, and it can drift indefinitely because the animation operates on a linear and endless schedule. The farther the words drift from their point of origin, the larger the electronic space of the poem gets. And since the browser window has horizontal and vertical scroll bars that allow one to move the window to wherever there may be text to read, one could literally allow the poem to drift to a truly monstrous size, rendering it horrendously meaningless. Or perhaps, after years of running continuously under ideal conditions, programmer-critics would have to create new tools to navigate the enormous black spaces of the poem in order to find words to read, becoming astronomers of this language constellation of a size to give Eugen Gomringer nightmares.

The constellation is the simplest possible kind of configuration in poetry which has for its basic unit the word, it encloses a group of words as if it were drawing stars together to form a cluster.

The constellation is an arrangement, and at the same time a play-area of fixed dimensions.

The constellation is ordered by the poet. He determines the play-area, the field or force and suggests its possibilities. the reader, the new reader, grasps the idea of play, and joins in.

In the constellation something is brought into the world. It is a reality in itself and not a poem about something or other. The constellation is an invitation.¹⁷

¹⁷ E. Gomringer, *From Line to Constellation*, [in:] M.E. Solt (ed.), *Concrete Poetry: A World View*, Indiana University Press, Bloomington 1970.

When Eugen Gomringer wrote this essay in 1954, the dimensions of a “play area” had to be fixed, whether he was working with the page or a landscape, as British Concrete poet Ian Hamilton Finlay became famous for. This notion is echoed by Charles Olson in his manifesto *Projective Verse* when he describes a poetics of “composition by field”. In all of these cases, these poets are taking advantage of the space they have at their disposal, which pretty much meant the page for all of them. Jim Andrews also takes advantage of the space available to him, but in a potentially more extreme way because the electronic page (or landscape) are mathematically infinite, and limited only by a computer’s storage capacity and processing capability, than by any actual physical dimensions.

It is at this point, when we start to engage the poem’s materiality as an electronic object that the poem has reached its true scene: the e-poetry scene. This is a scene of possibilities and experimentation, of play and invitations for readers to embark on a journey that challenges their traditional training as readers, their assumptions about language and how it should be configured, their preconceptions about what poetry is and should be. The e-poetry scene in 1997 was even smaller than it is today, yet there was an enthusiastic energy that came from being among the first to boldly go where no poet has gone before, even if some of those paths had been blazed conceptually in other media. How far readers would be willing to follow on this path is secondary to the need to explore it, and the invitation is always there.

Keeping the reader always in mind, however, Jim Andrews does provide the option to “discipline the text” back into “the scene” – back into traditional poetic form and into the metaphorical stage readers are accustomed to. But this act runs counter to the poem’s desire to drift and be in its own scene, to explore cyberspace (or at least RAM space) beyond our ability to follow and be free from tradition, convention, interpretation. And yet is an obedient text: it responds faithfully to the three commands we are given as options. Any controls on the text come from the author, who created it (along with its programmed behavior), and the reader, who has the power set it in motion, stop it, and ‘discipline’ it at his or her own convenience.

Let’s not get lost in the personification, however. The words have no control over their motion: they drift as Jim Andrews programmed them to. The programming is based on randomly generated numbers and a fairly simple positional equation that sends them slowly moving towards the edges of the screen and beyond. This element is important because Jim Andrews has relinquished some control over the text over to the computer’s ability to generate random numbers, something that occurs during the production performance on the reader’s side. This assures that the text the readers are exposed to will always be different, once they “do the text”, but within certain parameters, as multiple close readings will reveal clusters of words drifting at different speeds.

Thus the literal drift is the agent for metaphorical and semantic drift of this e-poem. This deceptively simple conceit may initially seem gimmicky, yet it is essential to the poem because it reconfigures the meanings of key words. Drift is also an important aspect in Andrews’ poetics, finding literal expression in many of his works, such as *Arteroids*, *A Pen*, and conceptual expression in works like *dbCinema* and *Aleph Null*. To drift is to move with-

out controlling one's direction, surrendering to whichever forces have influence over one's movement, such as wind and current in the case of a boat that isn't using its engines, sails, oars, or rudder to guide it. The geological term "continental drift" also comes to mind, particularly in the context of Seattle and the Pacific Northwest region, since its tall mountains and volcanoes are the result of an ocean tectonic plate drifting against the North American continental plate. A telling notion of drifting is the Situationist practice of the *dérive*, which directly translated means "drift", but which they describe as:

Dérives involve playful-constructive behavior and awareness of psychogeographical effects, and are thus quite different from the classic notions of journey or stroll. [...] from a *dérive* point of view, cities have psychogeographical contours, with constant currents, fixed points and vortexes that strongly discourage entry into or exit from certain zones. But the *dérive* includes both this letting-go and its necessary contradiction: the domination of psychogeographical variations by the knowledge and calculation of their possibilities. In this latter regard, ecological science, despite the narrow social space to which it limits itself, provides psychogeography with abundant data.¹⁸

This notion of the *dérive* can illuminate Andrews' use of drift in this poem and in his other works, whether he was aware of this theory or not, because it points out several aspects of drifting that are in tune with his own use of it. For instance, both *Enigma n* and *Seattle Drift* showcase Andrews' "playful-constructive" side of giving up control in order to discover insights, through the use of mutable and responsive textual behaviors. The random aspects of the texts are not without parameters – his texts, much like cities, contain psychogeographical contours, with constant currents, fixed points' and other means by which both the reader and the text can be guided in their drift. *Seattle Drift* has several such parameters: from the menu bar on the top left corner of the screen (seducing its readers to activate, stop, and reset the text), to the Vispo logo marking the lower right hand edge of the original window (framing the psychogeographical "scene" in which the poem begins), to the programming currents that guide the letters as they drift in their pseudo-random directions away from their original position. Andrews carefully crafts the ecology of his spaces, designing the psychogeography of his interfaces, so that he can then "let go" of his control over both "langwidgets" and readers and let them be drawn to what they will.

Understanding the role of textual behaviors in *Seattle Drift* – stasis, responsiveness, kinesis, mutability and scheduling – allows us to interpret the poem more completely than would be possible through a purely textual analysis, because each one helps shape the reader's experience. While I believe that textual analysis is at the heart of any good interpretation of a literary work, I am convinced that it is insufficient for the interpretation of an e-poem such as this, even though it is still a primary tool. The main reason is that since all the possible variations are practically impossible to be read, partly due to the mutability of the drift, partly due to the difficulty of starting and stopping the words at exactly the same time. A close reading of the text requires access to a relatively stable text, with the possibility of rereading it. *Seattle Drift* offers no such possibility, aside from the initial text. Perhaps in

¹⁸ G. Debord, *Theory of the Derive*, [in:] K. Knabb (ed.), *Situationist International Anthology*, Bureau of Public Secrets 2006, p. 50–54.

drifting away from the scene of traditional poetry, this text has also drifted away from the scene of traditional literary criticism, and requires new approaches to supplement or replace the critical tools of the past.

Concluding Remarks

In the 17 years since writing his first DHTML poems, the critical tradition has developed new tools for the analysis of electronic literature, such as Platform Studies, Software Studies, and Critical Code Studies (CCS). Paying attention to the source code, the software it is developed in and the platform in which it runs optimally yields insight on a work's creative process and functionality. The analyses offered of both *Enigma n* and *Seattle Drift* are informed by a functional analysis of the source code, revealing how they were partly inspired by the logic of DHTML. The study did not touch on all aspects of the source code, however. For example, both poems contain short essays about the work in the source code documentation, offering insights into Andrews' poetics and strategies. The analysis is also informed by an understanding of contemporary browser software and how they served as platforms for the execution of the works. These perspectives allow readers to examine the works as first generation digital objects, paying close attention to meaningful details beyond what we see on the screen. Jim Andrews might have produced very different works, had he been faced by a timeline interface, or had he been working with bitmapped images, vector graphics, or sprites.

In the 17 years since writing his first DHTML poems, Jim Andrews has continued to explore his artistic and poetic vision by creating works that operate like software. In 2000, he shifted to Macromedia Director as his authoring software, which he used to create works like *Nio* (2000), *Arteroids* (2001–2004), *dbCinema* (2004), and the unfinished visual music project *Jig-Sound* which was rendered inoperable when Adobe purchased Macromedia and changed the audio engine in Director 11. Prompted by the rapid obsolescence of Director, Andrews retooled to some of Adobe's more open authoring software, such as AIR and Flex. More importantly, he returned to open source programming with HTML5 to produce *Aleph Null* in 2012. His most recent work is titled *Teleporter*, which uses HTML5 and Google Earth Street View data to direct a browser to random locations around the world that are documented with 360 degree photography. He has published both iOS and Android app versions of *Teleporter*.

References

- Andrews J., *Enigma n*, <http://vispo.com/animisms/enigman/>, 1998 (access: 15.06.2014).
Andrews J., *Infoanimism*, <http://www.vispo.com/animisms/enigman/EnigmanInfoanimism.html>, 1998 (access: 15.06.2014).
Andrews J., *Pop-Up Poems*, <http://www.vispo.com/popups/popups.htm>, 1996–1998 (access: 15.06.2014).

- Andrews J., *Seattle Drift*, source code, <http://vispo.com/animisms/SeattleDriftEnglish.html>, 1997 (access: 15.06.2014).
- Debord G., *Theory of the Derive*, [in:] K. Knabb (ed.), *Situationist International Anthology*, Bureau of Public Secrets 2006.
- Flores L., "Typing the Dancing Signifier: Jim Andrews' (Vis) Poetics". Dissertation submitted to the Faculty of the Graduate School of the University of Maryland, College Park, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2010.
- Flores L., *Digital Textuality and its Behaviors*, "Journal of Comparative Literature and Aesthetics" 2013, vol. 36: 1–2, p. 81–103.
- Gomringer E., *From Line to Constellation*, [in:] M.E. Solt (ed.), *Concrete Poetry: A World View*, Indiana University Press, Bloomington 1970.
- Gosselin, D., *JavaScript: The Web Technologies Series*, Cengage Learning 2010.
- Hayles N.K., *Print Is Flat, Code Is Deep: The Importance of Media-Specific Analysis*, "Poetics Today" 25.1, 2004, 25.1, p. 67–90.
- Hayles N.K., *Writing Machines*, MIT Press 2002.
- Hayles N.K., Pressman J. (eds.), *Comparative Textual Media. Transforming the Humanities in the Postprint Era*, University of Minnesota Press 2014.
- Kirschenbaum M.G., *Materiality and Matter and Stuff: What Electronic Texts are Made of*, "Electronic Book Review" 2001, <http://www.electronicbookreview.com/thread/electropoetics/sited>.
- Marino M.C., *Critical Code Studies*, "Electronic Book Review" 2006, <http://www.electronicbookreview.com/thread/electropoetics/codology>.
- Warnell, *ADVEXP: xes*, <http://www.warnell.com/advexp/advxes.htm>, 1998 (access: 15.06.2014).