

# Writing-Designing-Programming: The NextText Project

Jason Lewis  
Obx Labs, Concordia University  
Montreal, Quebec, Canada  
jason.lewis@concordia.ca

## ABSTRACT

The Next Text Project is a series of explorations of the way that my collaborators and I create, experience and understand computationally-driven, visually-expressive texts. We have conducted these explorations by creating artistic works while simultaneously developing the new software necessary to realize them. Our methods have been driven by a commitment towards integrating the writing, designing and programming aspects of such work. In this paper I will review the conceptual, aesthetic and technical concerns that motivate the project, briefly present a number of the works produced by the project and make several suggestions as to how future research/creation in this domain might be conducted.

## Keywords

Electronic writing, poetics, materiality, programming, cybertext, electronic art

## 1. INTRODUCTION

“The formal structure of a poem is not something distinct from its meaning but as intimately bound up with the latter as the body is with the soul.” W. H. Auden [1].

The concerns of the NextText project grow out of a long-standing engagement with the question of what a poetry written specifically in and for the computational medium might read, look and behave like. The project is built on ActiveText [2], which developed new software architectures for creating interactive and dynamic texts. ActiveText itself was based upon Dynamic Poetry [3], a theoretical examination of how to articulate the ‘newness’ of new media and how poetry can provide a fruitful ground upon which to establish such an examination, as well as a series of material investigations culminating in a number of ‘dynamic poems’.

Writing, by definition, is a visual medium. Starting with early forms of alphabetic writing and continuing through to the present day, the interplay between the content and the form of writing has been the subject of experimentation in

the different ways of materializing letterforms to impress, enthrall, intimidate and enchant [4].

Poetry in particular serves as our key reference point for expressive form. Poetry is concerned as much with the evocative and aesthetic operations of language as with its literal semantic operations. Writing in a poetic mode requires an economy of interaction between these operations such that they amplify, augment and extend one another. Furthermore, poetry tends to be enframed by readers in a way that calls forth a heightened awareness of how structure directs reading and the interplay between the denotative and the connotative.

While one can look back over two millennia for inspiration regarding visual poetry, [5], in our project we most often turn to twentieth-century writers who engaged the visual presentation of their texts as a central component in the meaning making of those texts and as a necessary strategy for dealing with the complexities of the modern world. These inspirations include the calligrammes created by Apollinaire in pursuit of *simultaneity* [6], the poster manifestos produced by Futurists attempting to confront the socio-technical upheavals of industrialization [7] and the Concrete Poets embrace of early computer typesetting to extend their abilities for capturing the fragmented realities of the mid-century [8].

This context leads us to ask, what happens when we breathe the spirit of material experimentation with textual manifestation embodied by these writers into twenty-first century technology? To help answer that question, we have employed a methodology of writing-designing-programming to guide our own experimentation. These experiments include a core software library on which everything else is built, applications for manipulating letterforms via active behaviors, text-based interactive installations, performances involving real-time speech processing done in collaboration with writers/rappers/poets, and public-space text artworks that uses audience-generated text sent via mobile devices. The goal of this paper is to present a critical look backward at the project as well as provide a foundation for further articulating the advantages

of an integrated approach to writing, designing and programming digital texts.

## 2. MOTIVATION

“Some people think to make a color photograph, you just have to put color film in the camera. The result is not a color photograph.” Harold Allen [4]

The experiments in poetry-based visual language that occurred throughout the twentieth century have proven to be fruitful ground on which to develop (and all too often, discard) ideas about how to best go about the mixing of media that the computational environment so easily allows. The ability of written/printed/pixelated language to move back and forth between something that is read and something that is seen resonates with the bit’s ability to be realized as part of an audio stream, a video sequence or simply an image. Our hope has been that a close engagement with visual language will help us better understand how to harness this malleability to the task of creating digital texts.

We use the term writing-designing-programming to denote the tight coupling between writing a text and determining how it will appear and behave. Over the course of the Dynamic Poetry, ActiveText and NextText projects several guiding principles have emerged, co-evolving with the writing-designing-programming approach. I will now discuss these principles and look at how they drive our approach.

### 2.1 Rhyme and the Ultimate Reasoning Machine

We strive for meaningful applications of computation. Beginning with the Dynamic Poetry project, we have emphasized the fact that simply displaying words onscreen does not make poetry ‘new media’; it only creates another channel of publication. To create poetry that is in some sense fundamentally new media requires thinking and creating not only at the level of word and sentence, but also at the level of the bit and the pixel, manipulated through programming processes. It is in this manner that the material strengths of the digital environment, in distinction to other creative media, can be discovered, understood and applied. Thus, we consider meaningful engagements with text are ones that advance the expressive content of a work, and we use it in conscious distinction to instrumental, navigational or purely aesthetic applications of computation. Given how multivalent the concept of “meaning” is, it is perhaps no surprise that this principle

has proven to be both the most important and the most challenging one to which to adhere.

### 2.2 All Text, All the Time

We maintain a dogmatic focus on text. The NextText creative work is exclusively text-based, with the exception of the occasional integration of both pre-recorded and spoken audio. Such focus requires us to see all of our creative challenges as surmountable with text, and encourages—even forces—us to keep working on such challenges until we have developed a text-centric response. This strategy keeps us working with text when we might otherwise turn to imagery or video or audio to help carry the communicative load.

It has the added benefit of concentrating our technical development efforts within a relatively narrow range. We do not worry about how to integrate images into our architecture, nor are we concerned with handling the intricacies of maintaining synchronization with time-based media.

### 2.3 Reading and Looking and Reading and...

We treat the linguistic, visual and behavioral operations of text with equal intensity. Working with any one of these aspects perturbs the effects of the other two, leading to an incremental and iterative creative process that cycles back and forth between text work, design work and code work.

We find inspiration for this method again in poetry. A poet has a set of strategies that are linguistic (multiple meanings for a word), visual (where to end a line) and aural (rhyme, alliteration, meter) which can be used to shape how a poem is read. We are interested in finding computational strategies that structure the reading of digital poetry in similar ways. One of the key goals in creating the Next Text software has been to provide a common library that makes both the discovery and the creative process easier and more extensible than is the case if constrained to working with either shrink-wrapped applications or custom coding each piece from scratch.

### 2.4 Detonating the ASCII-Pixel Wall

We fight the print-centric nature of most tools for creating digital text. I have argued elsewhere that sustained research on new forms of digital text creation tools peaked with the advent of Postscript and the development of the first page layout applications. [1] This is not to say that those applications, and Postscript successor technologies such as OpenType and Multiple Master Fonts, do not continue to evolve in interesting directions. Rather, it is to say that this

evolution occurs within a framework based in a print mentality that constrains innovation onto a particular path. Coming from a print mentality means that much technical effort has been focused on maintaining fidelity from the screen to the printed page, and designing workflows around a fundamentally static display of text.

If one wants to work with the visual representation of letterforms in a more radical manner, most available tools force one to rasterize the text into an image format susceptible to manipulation by graphical operators. We think of this as the “ASCII-Pixel wall”. On one side are tools that handle text primarily as a stream of ASCII characters. Examples are common programs for word processing, page layout and illustration. The user can edit the text as text, deleting words and inserting letters. On the other side of the wall are tools which allow the user to do complex modifications of appearance, dynamics and interactivity. Examples include Photoshop, AfterEffects and Flash, respectively. These programs handle letterforms as a collection of pixels or group of curves, and neither know nor care that the text has a character aspect. The result is that when the writer wants to move text from the ASCII world to the pixel world, she is forced to abandon the textual aspects of the text in favor of its graphical aspects.

Having grown tired of throwing our writing over that wall, we developed Next Text. Next Text is a software library and set of related applications that approach text with the assumption that the creator will want the text to be spatially and temporally active, that she will want it to interact with itself, the reader, and other data, and that she will want to manipulate it at the same time she is composing it.

### 3. MANIFESTATION

We divide our NextText efforts into four categories based on how a particular project is put to use.

#### 3.1 Production

Production refers to the libraries and applications that we use to implement the creative work.

##### 3.1.1 Library

The NextText library was designed to remove the ASCII-Pixel wall. Text remains text throughout the creation cycle, and as such it can be edited at any time. At the core of the library is an engine for coordinating behaviors among numerous elements, a glyph object that maintains both ASCII and vector information about each glyph, and a set of behaviors. This coordination happens via a data tree that saves text, as it is written, in a hierarchical configuration that allows us to manipulate text at various levels of

granularity, from individual glyphs to words to sentences to paragraphs to entire texts at once. This internal representation allows us to manipulate text visually and behaviorally at the same levels that we use when writing and editing it.

NextText reads OpenType fonts, importing the outline information into a data object for each glyph as it is typed. The data object tracks all of the outline manipulation and location translations done on that glyph. The behaviors that enable the manipulations and translations are additive and interactive, meaning, for example, that a behavior that causes the outline to collapse into a churning mass of lines and curves, added to one that causes the glyph to be attracted to the mouse, will result in a complex behavior where it looks as if the mouse is sucking the glyph through a thin straw. (Figure 1)



**Figure 1. Result of layered behaviors applied to the ‘x’. The mouse is at the tip of right stem descending from the baseline.**

Storing the transformation history of each glyph allows us to continue editing it as text. For instance, if I wanted to change the “x” in FIGURE to an “s”, I can select it, type “s”, and the software will now apply the same history of transformations to the “s” and display it. (Figure 2)



**Figure 2. After editing the text to replace the ‘x’ with an ‘s’. The ‘s’ inherits the transformation history of the ‘x’ it replaced.**

These sorts of operations can be performed on individual letters or on words, groups of words, sentences, groups of sentences, paragraphs or the entire text. The library includes numerous behaviors, allowing the writer to build reasonably complex and fine-grained dynamics into her text. We are constantly developing new behaviors as we undertake new projects.

The main NextText library is implemented in Java. We chose Java over C++ (the language of implementation for ActiveText) due to faster development times, cross-platform adaptability, native integration with the web and accessibility to the growing number of creators who are being introduced to Java via the popularity of the Processing programming environment. In this area we are not dogmatic, as we will at times develop work in other languages depending on requirements. All the development, however, uses the NextText architecture.

### 3.1.2 TextEngine

TextEngine is a middleware application that accepts input from a number of sources (video, audio, and kinetic) in order to drive real-time NextText-based visualizations of spoken text [10]. TextEngine has been developed for real-time environments where a performer wants to visualize her speech and have control over the visualization by modulating her voice, executing gestures with her hands or entire body, or by clapping her hands or pounding on a drum. TextEngine processes the sensor input, analyzes it, and maps it to NextText behaviors per the design of the performer.

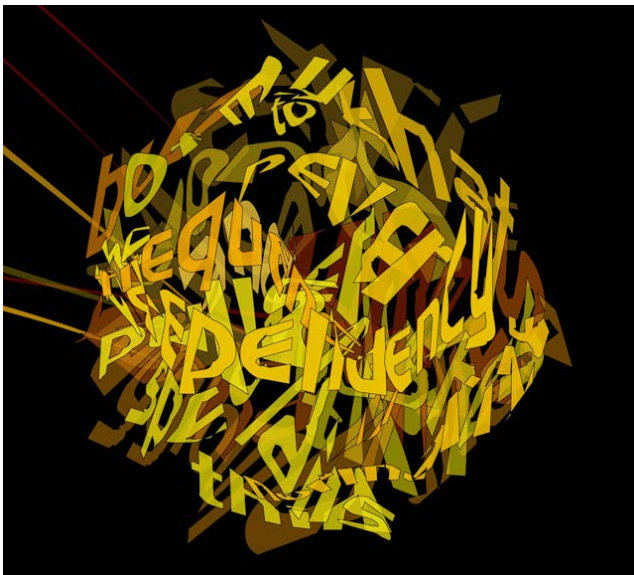


Figure 3. Detail of *Dependency*, created with the Mr. Softie application.

### 3.1.3 Mr. Softie

Mr. Softie is an application focused on manipulating glyph outlines, implemented in C++ due to the performance requirements for supporting interactions between thousands of glyphs simultaneously. We developed Mr. Softie to provide us with an environment optimized for creating high-resolution compositions of super-dense text. The goal was to push hard on the materiality of letterforms within the digital environment, treating them as raw ingredients in a visual composition. (Figure 3)

## 3.2 Poetry

*Still Standing* and *What They Speak When They Speak To Me* represent the ultimate goal of our research/creation trajectory: poetry written explicitly and exclusively for reading via computational interaction.

### 3.2.1 Still Standing

*Still Standing* [11] is poem about searching for a perfect “motionless moment” in which all external input is suspended (excerpt below):

“five chapters of addiction  
for my perpetual commotion  
bring my brain to a stop.  
the inception of sedation  
is needed for the waves to break  
and the spin to reduce  
and the letters to literal.  
the motionless moment hides  
from my sight to seduce.”

The poem is read via an interactive installation. (Figure 4)

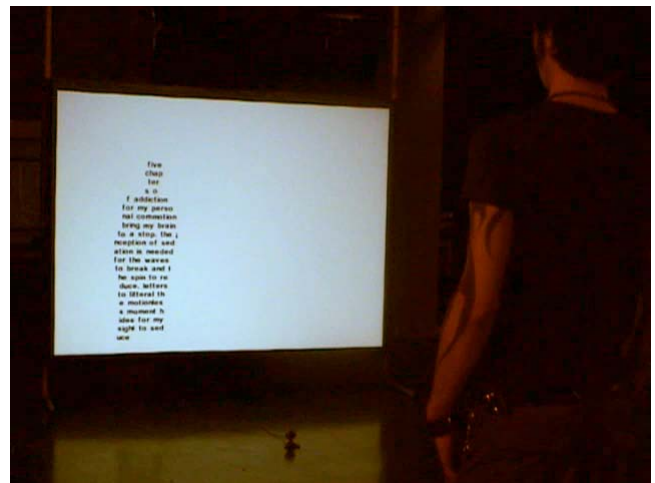


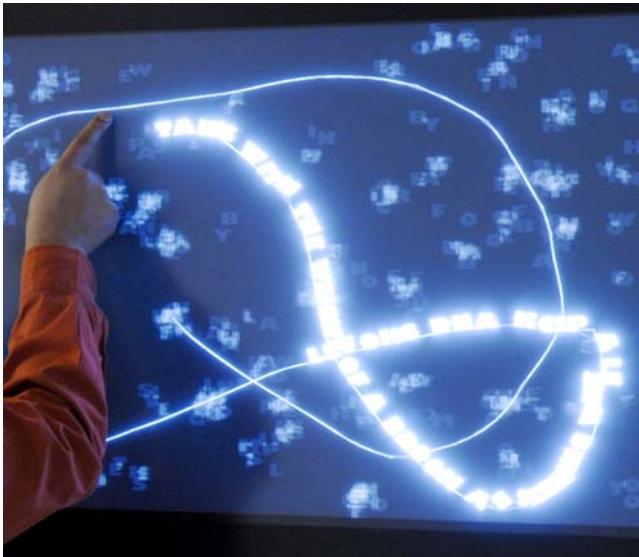
Figure 4. *Still Standing*, after the participant has stood still long for the text to become readable.

When the reader approaches the installation, all she sees is a heap of gently moving individual letters projected on the ground. As she walks in front of the projection, her motion disturbs the letters and they react as if being kicked or pushed by her silhouette. However, if she remains motionless the letters will slowly begin moving up within the silhouette, like water soaking up her body, forming words and sentences as they go. If she remains still long enough the entire text of the work becomes readable.

**3.2.2 What They Speak. When They Speak To Me**  
*What They Speak. When They Speak To Me* is a poem about mistaken identity and language. Through some combination of skin color and skull scaffolding, it happens often that I find myself someplace foreign with enough resemblance to the locally dominant morphology that the natives assume that I am one of them and begin speaking to me in their mother tongue. The text in the piece looks at the sometimes positive, sometimes negative results of this misapprehension (excerpt below):

“...what do these bodies speak  
when we collide in the streets?  
we feast on our own lives as we meet,  
polycultural cannibals wallowing in our own  
monohued blood.”

The work is presented on a 50” plasma touchscreen, which, in its wait state, displays floating masses of translucent white letters spread around the screen. (Figure 5)



**Figure 5. Detail from *What They Speak When They Speak To Me*.**

When the reader touches one of the letters, it fades into full white and, quickly thereafter, all of the letters that belong to

the same line also fade to white. Then, as the reader moves her finger across the screen, the letters assemble themselves behind it one by one until the entirety of the line can be read.

### 3.3 Play

Another pair of works, *Intralocutor* and *Cityspeak*, are systems for creative language play in which the text is generated by the user.

#### 3.3.1 *Intralocutor*

We think of *Intralocutor* as an instrument that allows us to improvise on how the way in which we talk is often as important as what we actually say: participants can modify the appearance of their visualized speech by manipulating the prosodic qualities of their voice. It is in the form of an interactive installation for two participants, facing one another each with their own microphone. (Figure 6)



**Figure 6. Two participants in the *Intralocutor* installation.**

Both of their silhouettes are projected onto opposing screens so that each one of them is looking at both her partner, and at a projection showing both of their silhouettes. As person 1 begins speaking, her speech is recognized by the system. It is then shown coming out of her silhouette and traveling to her partner’s silhouette. The quality of this movement is affected by speaker 1’s volume and cadence. The louder she speaks, the larger the text becomes; the softer she speaks the smaller it becomes. Similarly, the faster she speaks, the more direct path taken from her silhouette to her partner’s silhouette; the slower, the more the text floats between the two silhouettes. The same dynamic occurs as speaker 2 is speaking. (Figure 7)

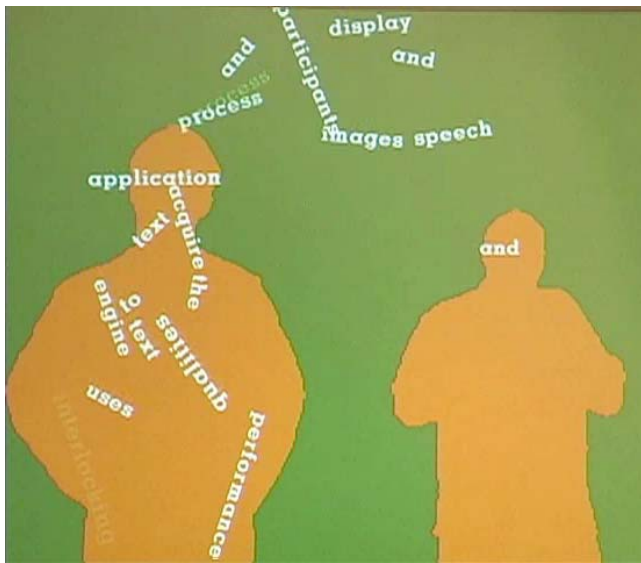


Figure 7. Screenshot from *Intralocutor*.

### 3.3.2 *Cityspeak*

*Cityspeak* was conceived as part the Mobile Digital Commons Network, a collection of artists, academics and activists experimenting with how to use wireless technology to create cultural interventions in public space [12]. We designed *Cityspeak* as a reaction to ever-increasing governmental and corporate control of what used to be considered the public commons, and we designed it in the spirit of rejection of that control such as that practiced by graffitiists. (Figure 8)

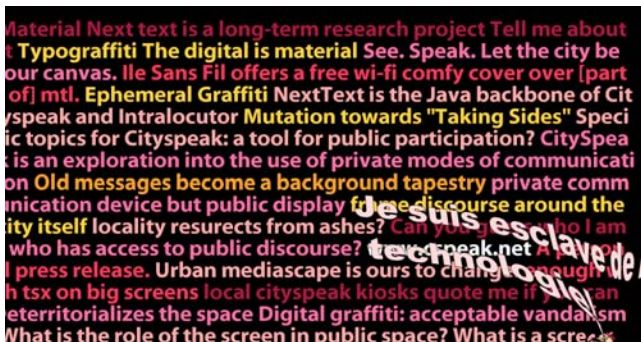


Figure 8. *Cityspeak* screenshot with message transiting from foreground to background.

*Cityspeak* installations take place in spaces with large-scale public displays such as projections or LED screens. Users send text messages to the displays using their cell phones or wireless personal digital assistants or laptops. NextText is used for the text display, and behaviors are used to control the movement of the messages as they migrate from the foreground into the background, becoming part of the constantly scrolling message history.

## 3.4 Performance

Saying Red is the name of a series of text-centric performance pieces we have developed using NextText and TextEngine. The Saying Red performances are collaborations with spoken word artists and rappers who create original compositions that integrate the capabilities of the software.

These collaborations begin with several weeks' 'initialization', during which time we exchange ideas about composition with the outside artists. We also take this time to learn about their writing and performance style, while they become familiar with the technology through regular improvisation sessions and train the voice recognition software on their voice. We then spend several weeks developing the performance, iterating between modifications to the text, the performance and the visuals. To date we have created three prototype performances.

*Taking Sides* is a battle rap, where vocal artist Dwayne Hanley plays both characters facing off, trying to erase each other's presence with words. Each character's text possesses a distinct visual design, allowing the observer to track how he is doing in the battle. The chorus, spoken by both characters, shares a common visual design. (Figure 9) By the end of the performance, the screen has become a solid mass of letters surrounding Hanley's silhouette.



Figure 9. Detail from *Taking Sides*. Red and yellow text represents different sides; white text is the chorus.

*Fugue 8* is a spoken work piece with poet John Stuart and designer Kevin Lo. The texts is composed of original writing as well as material collected by Stuart and Lo from friends, and all deal in one way or another with how the personal interpenetrates the political. The performance is a recitation of the texts, divided into three movements with different design and behavioral qualities that reflect transitions through the two spheres and between the 'found' text and that written by Stuart. (Figure 10)



Figure 10. *Fugue 8*, in the third movement.

*Resolution* was created with Alex Mannarelli. It deals with notions of self-confidence and social pressure, particularly how the latter can undermine or reinforce the former. This piece features an ongoing struggle between Mannarelli and the text, where at times the text is chasing his silhouette and he is moving around the stage to avoid it, and at times he is controlling its movement and directing it away from him. (Figure 11)



Figure 11. Detail from *Resolution*.

## 4. MEDITATION

“No amount of ideological or cultural valuation can transform the propensity of papyrus to deteriorate into gold’s capacity to endure.” Johanna Drucker [13]

What are the propensities and capacities of the digital material? Our explorations suggest several possible answers to this question.

### 4.1 Computation is Semantic

Computation is what a computer does, but we have only recently begun seeing sustained attempts at articulating the different ways we might consider computation as shaping material. Aarseth has argued for the centrality of computational characteristics in any analysis of digitally-based ergodic literature [14]. N. Katherine Hayles, in *Writing Machines*, looks at how we can account for the discontinuities posed by digital material—such as the use of computation—within the continuities sustained through centuries of literary studies of predecessor materials [15]. John Cayley offers up a fine-grained interpretation of ‘codework’ to help disambiguate the different ways the term “code” is used [16]. And Noah Wardrip-Fruin, borrowing from Chris Crawford, proposes a distinction between process-driven and data-driven digital literature as a way to enhance our understanding of the different ways computation may be employed by an author [17].

Our work on the NextText Project leads me to believe that such distinctions belong not only at the level of analysis, but even more so at the levels of creation and tool-making. That is, to make work which possess a reading experience as rich as its interactive experience (and vice-versa) requires that the author be able to engage computation as part of a conscious strategy of meaning making. Such an engagement is both conceptual, in that it requires integrating written, visual and computational components from the moment one starts thinking about a new work, and technical, in that it requires working in an environment that will then allow one to realize those components in parallel and in reflection with one another.

*Still Standing* and *What They Speak When They Speak To Me* are, arguably, the most successful NextText artworks; they began life as a bit of text, a bit of behavior and a bit of computational infrastructure to support the further, mutual development of all three components. The resulting works incorporate texts which cannot be read separately from their computational manifestation, any more than a villanelle can be read separate from its spiraling repetition of end rhymes. *Still Standing* requires the energetic motion when the participant enters to counterpoint the stillness when she ceases to kick the text around; *What They Speak...* requires constant interaction on the part of the reader in order to draw the text out of its chaos.

### 4.2 Speech Recognition Is Hard; This Is Not

Some days I find it miraculous that I can create a video on my desktop by using, all in the same place, Photoshop/Illustrator (image assets), FinalCut (editing) and AfterEffects (compositing). On other days, I find it ridiculous that, despite the underlying representation (binary code, bits of data) being the same, I have to move between all these different applications to get the job done.

And on Sundays I remind myself that the analogue world requires different tools for different kinds of material, despite the underlying implementation (atoms, molecules, etc.) being the same. In short, our tools always embed both a reality and ideology about the propensities and the capacities of the material we use them on. This is not presented as news, just recognition of long-standing critical discourse, and as background to ask why more development effort has not been expended on updating the ways in which our tools handle digital text.

The NextText library does not embody any innovation in the computer science sense. The algorithms are extensions of existing algorithms for manipulating outlines, the architecture borrows heavily from scene graphs in procedural modeling, and the behavior engine is a relatively straightforward application of frameworks for coordinating action between multiple agents. The innovation the library does represent is in rejecting the word processor as the primary tool for writing, and employing data handling techniques from other domains in software engineering to create tools that allow us to manipulate text along several different axes at once. This allows us to examine the materiality of digital literature not only from a theoretical standpoint, but from a practical standpoint as well.

### 4.3 Gesamtkunstwerkzeug

We, of course, are not alone in our attempts to create new techniques for working with digital texts. The Visual Language Workshop at the MIT Media Lab produced an as-yet unrivaled body of technical explorations into how to present digital texts [18][18][20][21]. Creators such as Talan Memmot [22] and Jim Andrews [23] have put standard multimedia software to use in creating meaningful dynamic texts. Giselle Beiguelman's *Poetrica* engages networked technologies to display her poetry interweaved with public contributions [24], while Amy Alexander's *Cyberspaceland* [25] and Eugenio Tisselli's *MIDIPOet* [26] are two significant softwares that give the user the ability to fluidly control the performance of texts in real-time. Yet what we do not see much is collaborations between the tool builders and the text writers that might enable the total 'cyber'-writing experience.

We purposely sought external collaborations for the Saying Red series in order to create example cases of how such collaborations can feed forward into further technical and creative development. However, while the NextText project has made creating our work much easier, the software still requires too much technical effort to use. It requires a reasonably sophisticated knowledge of Java in order to usefully employ the library, and while we have written the

behavior programming interface on the plug-in model, we have no illusions that creating those plug-ins is a task that your average writer-designer can address in an hour. While the technical capacity is there, it is not yet there enough for casual users to fruitfully employ it themselves.

Since we first started writing about and developing around the ASCII-Pixel wall, we have watched with interest as tools were developed from the ground up to work in a dynamic and interactive environment. The most significant of these tools is Macromedia Flash, which uses such a foundation to indeed make creating dynamic and interactive much easier than had previously been the case. [27] It is instructive, however, that Flash still divides its text into 'static' and 'dynamic' types, each with its own set of limitations based on that categorization that effectively recreate the ASCII-pixel wall.

It is also instructive to note that another popular tool for creating dynamic and interactive work on the web, the Processing programming environment, uses as its basic letterform representation the bitmap format first implemented at the Visual Language Workshop in the eighties. [28] This .vlw format was implemented in an era when the only way to get decent real-time performance was to treat everything as a texture mapped onto polygons, and then use a graphics language such as OpenGL to manipulate the polygons. On the one hand, given that the Aesthetics and Computation Group (ACG) out of which Processing sprang was the successor to the VLW, this is not surprising. On the other hand, given the innovative computational typography of ACG's founder, John Maeda [29], not to mention of a number of his students, it is something of a surprise that Processing employs such a graphics-centered means of handling text.

The readiness with which these tools have been taken up by legions of creators who had never done any scripting or programming beforehand is a testament to the skill with which they were crafted. What interesting tools might we see if the same skill was applied to making the total writing tool?

## 5. FUTURE WORK

The future of NextText is two-fold. First, we will be releasing several of our tools over the next year, including Mr. Softie, an authoring environment called Glyphkicker and a library for use in Processing. The vainglorious hope is that these tools will be picked up by others, and both encourage creators to make meaningful work and encourage developers to think twice about how they handle text in their applications.

Second, with the NextText library stable enough to serve as a dependable and flexible platform upon which to create work, it is possible for us to devote proportionately more time to the writing that drew us into this trajectory in the first place. *What They Speak When They Speak To Me* and *Still Standing* are successful in no small part because the technical facilities for creating them were mature enough that they could be relatively quickly prototyped and finalized. Our plan is to now focus on the poetry, as opposed to the production, play or performance aspects of the project, to see if we can create a series of text works that integrate the linguistic, the visual and the computational in a manner that can withstand our own criticism.

## 6. ACKNOWLEDGEMENTS

I would like to acknowledge the Obx Labs members from 2002 to 2007 who were essential to realizing the work described herein. I would also like to acknowledge those who provided financial support for Next Text, including Concordia University, Hexagram, Canadian Heritage, and the Fonds québécois de la recherche sur la société et la culture.

## 7. REFERENCES

- [1] Auden, W.H. *W.H. Auden Reading His Poetry*. Audio cassette. New York : HarperCollins, 1989.
- [2] Lewis, Jason and Alex Weyers. "ActiveText: A Method for Creating Dynamic and Interactive Texts". User Interface Software and Technology Conference, Asheville, North Carolina. (November 1999)
- [3] Lewis, Jason. "Dynamic Poetry: Introductory Remarks to a New Medium." M. Phil Thesis, Royal College of Art, London, 1996.
- [4] Petrucci, Armando. *Public Lettering: Script, Power, and Culture*. Transl. by Linda Lappin. Chicago : University of Chicago Press, 1993.
- [5] Castanyer, Laura Borràs. "Digital Literature and Theoretical Approaches.". *Dichtung Digital*, 2004 (3). [www.brown.edu/Research/dichtung-digital/2004/3/Castanyer/index.htm](http://www.brown.edu/Research/dichtung-digital/2004/3/Castanyer/index.htm) [accessed on 6 June 2007]
- [6] Apollinaire, Guillaume. *Calligrammes*. Transl. by Anne Hyde Greet. Berkeley, CA : University of California Press, 1980.
- [7] Tisdall, Caroline and Angelo Bozzolla. *Futurism*. London, Thames & Hudson, 1977.
- [8] Mary Ellen. *Concrete Poetry : A Word View*. Bloomington , Indiana : Indiana University Press, 1971.
- [9] Smith, Keith A. *Structure of the Visual Book*. New York: The Sigma Foundation, 1992.
- [10] Lewis, Jason and Yannick Assogba. "Taking Sides: Dynamic Text and Hip-Hop Performance." ACM Conference on Multimedia, Santa Barbara, CA. (October 2006)
- [11] Lewis, Jason and Bruno Nadeau. "Inter-inactivity". Digital Arts and Culture Conference, Copenhagen, Denmark. (December 2005)
- [12] Levesque, Maroussia, Jason Lewis and Lucie Belanger. "p2P: Cityspeak's Reconfiguration of Public Media Space". *Wi: Journal of the Mobile Digital Commons Network*. No. 1, vol. 1, Montreal, QC. (Fall 2006)
- [13] Drucker, Johanna. *The Visible Word*. Chicago : University of Chicago Press, 1994.
- [14] Aarseth, Espen J. *Cybertext: Perspectives on Ergodic Literature*. Baltimore, MD: Johns Hopkins University Press, 1997.
- [15] Hayles, N. Katherine. *Writing Machines*. Cambridge, MA: MIT Press, 2002.
- [16] Cayley, John. "Time Code Language: New Media Poetics and Programmed Signification." In *New Media Poetics*, Adalaide Morris and Thomas Swiss, eds. Cambridge, MA: MIT Press, 2006.
- [17] Wardrip-Fruin, Noah. "Expressive Processing: On Process-Intensive Literature and Digital Media.", Ph.D. Thesis, Brown University, Providence, Rhode Island, 2006.
- [18] Rosenberg, Tara. "Prosodic Font: the space between the spoken and the written." Masters thesis, Massachusetts Institute of Technology. Boston, 1998.
- [19] Small, D. "Expressive typography: high quality dynamic and responsive typography in the electronic environment". Masters thesis, Massachusetts Institute of Technology. Boston, 1987.
- [20] Soo, Douglas. "Implementation of a temporal typography system." Masters thesis, Massachusetts Institute of Technology. Boston, 1997.
- [21] Wong, Yin Yin. "Temporal Typography : A Proposal to Enrich Written Express." *Conference Companion, Computer Human Interaction '96*. New York: ACM Press, 1996.
- [22] Memmott, Talan. *Lexia to Perplexia*. In *The Iowa Review*. [http://www.uiowa.edu/~iareview/tirweb/hypermedia/talan\\_memmott/index.html](http://www.uiowa.edu/~iareview/tirweb/hypermedia/talan_memmott/index.html). [accessed February 20, 2007]
- [23] Andrews, Jim. *Arteroids 2.5*. In *poemsthatgo*, no. 14 (Fall).

- <http://www.poemsthatgo.com/gallery/fall2003/arteroids>. [accessed February 19, 2007]
- [24] Beiguelman, Giselle. "Poetrica".  
<http://www.poetrica.net>. [accessed February 15, 2007]
- [25] Alexander, Amy. "Cyberspaceland".  
<http://cyberspaceland.org>. [accessed May 28, 2007]
- [26] Tisselli, Eugenio. "MIDIpoet".  
<http://www.motorhueso.net/midipeng>. [accessed June 5, 2007]
- [27] Macromedia Flash Users Manual. Macromedia, San Francisco, 2006.
- [28] Fry, Ben, Casey Reas, et al. Processing reference.  
<http://www.processing.org/reference>. [accessed February 19, 2007]
- [29] Maeda, John. Creative Code: Aesthetics & Computation. London: Thames & Hudson, 2004.