

Interactive Close Reading

Zachary Joseph Risha

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State
University in partial fulfillment of the requirements for the degree of

Master of Arts
In
English

Thomas Gardner, Chair
Quinn Warnick
Anthony Colaianne

May 8, 2017
Blacksburg, VA

Keywords: Poetry, Close Reading, Web Development, Educational Technology

© 2017 Zak Risha

Interactive Close Reading

Zak Risha

Abstract

Over the past two decades, the readership of poetry has declined to the point that the art form is seldom engaged with by the public. I argue that reading poetry requires a skillset that must be learned, practiced, and refined. While close reading is traditionally trained in college classrooms, such spaces cannot reach broad audiences. To address this dearth, I have developed a web app that applies interactive learning strategies, through a series of exercises, to cultivate expert reading practices in novice users. *Close Reading* will guide users through poems by Robert Frost. With each poem, users will progress through exercises grounded in the practices of expert readers. For instance, users will block poems into sections to allow a chunking of the material, slowing down novice reading speeds. Another exercise cognitively models the act of reading by displaying the sequential thoughts of a reader making sense of a work. Furthermore, Socratic questioning will attempt to stimulate an internal dialogue to foster focus and interpretation. These exercises will build on one another and attempt to replicate pedagogical processes observed in the classroom. Performing these pedagogical exercises will provide a resource for developing the skillset necessary for poetry appreciation. This ambitious digital humanities project experiments with a new venue for pedagogy and poetry, promoting an engagement with the public frequently neglected in academic work.

Interactive Close Reading

Zak Risha

General Audience Abstract

Over the past two decades, the readership of poetry has declined to the point that the art form is seldom engaged with by the public. I argue that reading poetry requires a skillset that must be learned, practiced, and refined. While close reading is traditionally trained in college classrooms, such spaces cannot reach broad audiences. To address this dearth, I have developed a web app that applies interactive learning strategies, through a series of exercises, to cultivate expert reading practices in novice users. *Close Reading* will guide users through poems by Robert Frost. With each poem, users will progress through exercises grounded in the practices of expert readers. For instance, users will block poems into sections to allow a chunking of the material, slowing down novice reading speeds. Another exercise cognitively models the act of reading by displaying the sequential thoughts of a reader making sense of a work. Furthermore, Socratic questioning will attempt to stimulate an internal dialogue to foster focus and interpretation. These exercises will build on one another and attempt to replicate pedagogical processes observed in the classroom. Performing these pedagogical exercises will provide a resource for developing the skillset necessary for poetry appreciation. This ambitious digital humanities project experiments with a new venue for pedagogy and poetry, promoting an engagement with the public frequently neglected in academic work.

Table of Contents

Introduction.....1

Section One: **Education 2.0.....3**

Section Two: **Blocking: Adjusting Reading Habits.....6**

Section Three: **Inside the Ivory Tower: Modeling the Cognitive Processes of Close.....9**

Section Four: **Pressure Points: Problematizing a Poem.....14**

Section Five: **Reflection: Reading Inward.....17**

Section Six: **Technical Specifications.....19**

Conclusion.....27

Works Cited.....29

An Introduction

“Poetry is dead.” Crestfallen scholars and cynical writers occasionally mutter this lament. While some envision the crumbling of the Library of Alexandria, the continuous publications of poetry and conferred MFA degrees dampens such a heated claim. But this isn’t as simple as someone crying wolf. While poetry is not dead, it might be fair to say that poetry’s readership is dying. According to the 2012 Survey of Public Participation in the Arts, only 6.7% of Americans have read a poem within the last year (Ingraham, “Poetry Is Going Extinct”). These troubling numbers are down from over 16% in 1992. So who reads poetry? Dana Gioia’s *Can Poetry Matter?* begins, “American poetry now belongs to a subculture. No longer part of the mainstream of artistic and intellectual life, it has become the specialized occupation of a relatively small and isolated group” (1). Despite poetry’s increasing publication rates and conferred MFA degrees, the art form has lost broader cultural significance. While poetry was once covered in newspapers, now these works are seldom widely distributed. Gioia reasons that decades of public and private funding have created a professional class, based primarily in universities, comprised of teachers, graduate students, editors, publishers, and administrators. This group constitutes the current subculture that seems disinterested in addressing the gap between themselves and mainstream readers. Gioia even suggests, “outside the classroom . . . poets and the common reader are no longer on speaking terms” (10). Isn’t it time that poets and scholars begin discussing how to save their craft from irrelevance, rather than fret over publishing a work in a magazine that only fellow poets read? Scholars must seek to reverse the declining popularity of one of our most precious art forms.

While Gioia attributes poetry's decline to cultural aspects stemming from university programs and publication trends, there also exists the significant issue of poetry's difficulty. Even modern poetry, which had higher levels of readership, required a foundational skillset to read and appreciate. Much of the entertainment currently enjoyed doesn't require the commitment and effort that poetry demands, thus new readers are often perplexed that poetry is not something you just consume. Still, difficulty doesn't always discourage people if the end result is perceived to be worth the effort, but how does one tangibly convey the rewards of reading poetry? While demanding and requiring patience, poetry's potential for transformational understanding has significant value. Through poetry, readers discover new ways to think that help cultivate a deep awareness of themselves and the world. These introspective insights gained from reading poetry are as needed as ever but are increasingly difficult to demonstrate to the public.

I argue that poetry's decline stems significantly from a lack of training among readers, resulting in issues of accessibility. Novice readers lack the necessary literacy to interpret poetry, a skillset I'll refer to as "close reading." If readers are unable to correctly interpret lines of verse, then demonstrating the reward or value is nigh impossible. While the difficulty of passing through the beginner's threshold intimidates readers, it is the primary way to reveal poetry's value. The most notable resource for accomplishing this task would be courses in university departments and high school classrooms; the latter could potentially do more harm than good. However, with course requirements for majors so demanding, many do not have the luxury to take an expensive university course to foster a skillset whose benefits are perceived as mysterious. Should one need to be an English major and study under a professor to enjoy this form of art? I argue that there is a large

void for supporting neophytes seeking to engage with the art, one that technology can fill. I believe the greatest purpose of the digital humanities will be a reengagement of the public with the arts.

Education 2.0

The advent of the Internet has revolutionized how humans learn and access information. A primary resource for informal learning, the web is ripe with interactive tools, exhaustive information, and various media to learn a wide variety of skills and hobbies. For instance, programming has become one of the most sought after skills by employers; those talented in this field are actively recruited and paid top salaries. But according to a survey on Stack Overflow, 69.1% of developers claim to be at least partially self-taught and 48% never received a degree in computer science (Krill). Programming is a skillset arguably as difficult as poetry, but instruction in this field has not been limited to classroom education; rather new learners have capitalized on the online medium. No doubt, the financial implications are a huge part of programming's autodidacticism; few people would try to brush up on enjambment for a job interview. But this does not mean poetry cannot learn from programming's prevalence on the web.

In part due to an innate digital nature, the field of programming has developed an extremely sophisticated online community for whoever wants to learn the skillset. There exist lively communities like *Stack Overflow* and countless programming courses on sites like *Udemy*. Beyond the more traditional video-driven courses, websites like *Codecademy* make learning an active, enjoyable experience, walking users through core principles and skills. *Codecademy* utilizes interactive techniques of user interfaces to transform a medium

traditionally thought of as passive. The interface consists on an information panel on the left side of the screen and a text editor for coding on the right (see fig. 1). Users are first given instructions and conceptual information but must also complete an accompanying task that utilizes that knowledge. For instance, the software may provide textbook information explaining JavaScript arrays, but then, users have to program their own array

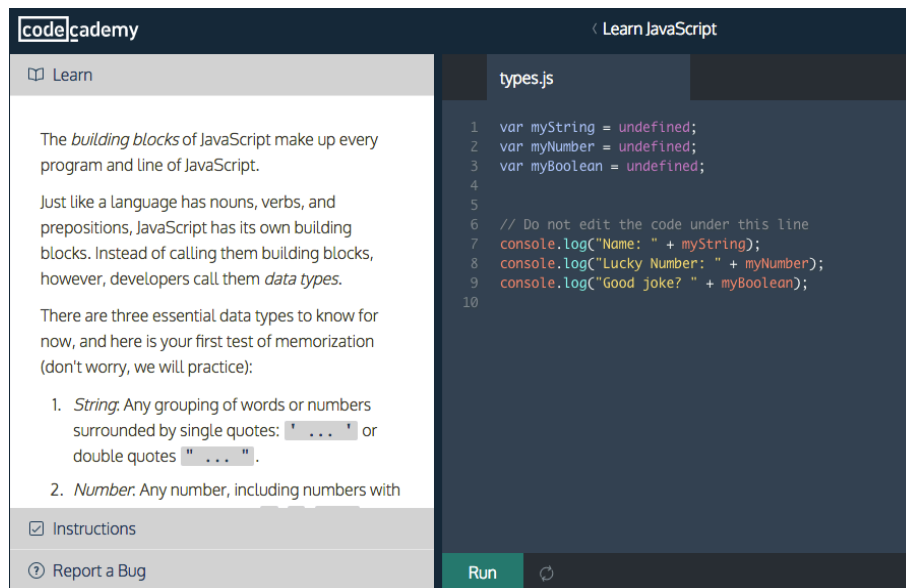


Fig. 1. Codecademy's user interface, "Learn Javascript" Codecademy, Codecademy, n.d, Web, Author's screenshot, 01 May 2017.

to move on. This approach transcends the mundane act of reading and follows with immediate practice, making it a pragmatic and active way to learn how to code. With new capabilities of education technology, we have entered a new era for education, and it's taking place on the web.

While programming has benefited from the web and new education technologies, little has been developed for teaching close reading. Poetry doesn't share such a plethora of online resources as programming and those most commonly used don't attempt to teach a skillset. Rather, the resources out there usually provide another's analysis and summary of a poem, doing little to promote a reader's own interpretation. Poetry has a long history of

being taught in physical classrooms and moving online has not yet been thoroughly explored. Due to decreasing readership, the field needs to experiment with new ways of learning to survive this new era in education and reach broader audiences. My research investigates how to apply interactive, pedagogical techniques similar to those used by software like *Codecademy* to educate users in the art of close reading.

To address this dearth of resources for novices, I have developed an [app](#) using interactive web technologies that will guide users through poems by Robert Frost. With each poem, users will progress through a series of exercises grounded in practices of expert readers. For instance, users will first block poems into sections to allow a chunking of the material, slowing down novice reading speeds. A second exercise cognitively models the act of reading by displaying the sequential thoughts of a reader making sense of a work. Finally, Socratic questioning will aim to stimulate an internal dialogue, fostering focus and interpretation. These exercises build on one another and attempt to replicate pedagogical processes observed in both classroom environments and individual studies. The goals of the software are to train novices, increase readership, demonstrate poetry's value, and promote the art form as a whole.

Developing such a resource comes with steep challenges, for close reading is not a straightforward skillset. Perhaps one of the reasons that poetry has yet to develop online support is the sheer difficulty of developing curriculum for poetry pedagogy. There exist no firm rules or clear answers for how to interpret a poem. In the past, the ambiguity of poetry resulted in poor teaching strategies. Back in 1958, Neville complained that educators became "frustrated when there is nothing tangible to teach, and fall back upon the mechanics of composition as the stuff of poetry. So rhythm, rhyme, stanza forms, names

and definitions of figures of speech, and other techniques of expression are foisted as poetry upon the retreating student” (134). Such a structured and tangible approach overlooks many aspects of close reading, which I argue to be most essential component to understanding and appreciating poetry. Developing curriculum to teach close reading requires a nuanced approach more variable and abstract than the rules, syntax, and functions of programming. Yet, educational technology grows in sophistication and new web capabilities continue to push the envelope for what pedagogical techniques can be used online. Some might oppose online learning for it is frequently done poorly: a video lecture, a couple PowerPoint slides, a reading, and an online quiz. However, the success of interactive websites like *Codecademy* shows how innovation and interactivity can produce a result as, or even more, effective as a traditional classroom. The online medium empowers the average user and offers accessibility to the broader population that poetry is lacking.

Blocking: Adjusting Reading Habits

Before discussing the aspects of close reading, I want to consider the reading practices of the novice user. Reading poetry requires quite a different skillset from traditional reading, complicating an already complex process. To novice readers’ detriment, there often exists an instinctive association with prior reading experience, thus leading students to approach a poem as they would any other piece of writing. Reading speed, often encompassed in fluency, is highly valued in formal education and frequently tested as a component of standardized tests. Formal education conditions students to read as quickly as

comprehension allows. A speed-reading culture has emerged where readers quickly sift through the excess, attempting to speed up knowledge acquisition.

However, slowing down is essential for close reading. Speed is quite opposed to the compact, detailed nature of poetry. With an adjusted speed of reading comes an adjusted attention, allowing readers to pay attention to every noun, adjective, and punctuation mark. Traditional reading doesn't account for the significance of such details. Rather, the focus is more on ideas and concepts and less on linguistic characteristics. But when faced with a poem, a novice may apply the same approach, treating the poem as something to get through and just try to gather information. This habit glosses over many significant details of language, resulting in a very shallow reading that produces confusion or failure. Retraining how one reads serves as an essential first step to close reading.

Close Reading's proposed exercise to help adjust the reading habits of users stems from a literacy strategy pioneered in the 70s and 80s. 'Chunking' was a term used to describe the grouping of individual words into "meaningful linguistic units" (Stevens 126). The theory emerged from studies trying to investigate differences between poor and strong readers. In 1970, Cromer conducted a study by pre-organizing reading into linguistic units, noting gains in comprehension in poor readers (482). Brozo, Schmezler, and Spires concluded, "it may be helpful to prepare instructional material in such a way that meaningful units are obvious to the learner" (444). Stevens noted similar potential: "Perhaps by heeding the need for such organization of verbal input, teachers can improve the reading comprehension of their students" (129). While there isn't consensus on chunking as a cognitive process necessary for reading comprehension, scholars

acknowledge potential pedagogical value in arranging words and content into meaningful units.

While 'chunking' focuses on the more micro level of phrases, *Close Reading's* related concept of 'blocking' breaks a poem into units consisting of multiple phrases that have a connection. Much of poetry has veered away from strict units like stanzas as obvious blocks of attention, calling for a need to break a poem into other meaningful segments. While a way to slow down readers and increase focus, the method also allows users to connect and relate parts of a poem, tying a sequence of blocks together to form interpretations. Blocking a poem forces the user to track shifts in ideas or action that can often be glazed over if one is not actively searching. By trying to block a poem, readers are attuned to these shifts and can better unravel the drama of a poem.

Providing an example of the significance of these shifts, I'll refer to Robert Frost's "Design" as an example. The first block of the poem begins in scene as the speaker finds a white spider, eating a white moth, on a white heal-all. Then, the poem shifts inward to the speaker's mind, as images of death and blight are evoked. The mind then returns to the sight before pondering the set of circumstances that led to the formation of this scene. These four shifts of the mind are necessary to comprehend the poem as a whole, for the last two lines deal with a significant, but subtle, concluding mental move. Frost writes, "What but design of darkness to appall?— / If design govern in a thing so small." These lines can be quickly be glossed over, but if you ask what final shift the mind makes, you can unlock another layer of meaning to the poem. The speaker comes to his dark conclusion, but immediately takes it back. The speaker can't linger on such a troubling thought, and even if he doesn't truly believe that the smallness of the scene cancels out the troubling theory he

has sketched, perhaps the thought allows him to keep moving on. Without the previous blocking of the speaker's mental moves, the real climax goes unnoticed.

“Design” shows that blocking a poem is not a straightforward task. *Close Reading*

first demonstrates the process of blocking to model the strategy for users. *Close Reading* uses a step-by-step user interface (as shown in fig. 2) that targets certain linguistics cues that denote shifts, such as punctuation and subject change. The exercise then blocks a segment of the example poem, providing detailed explanation of

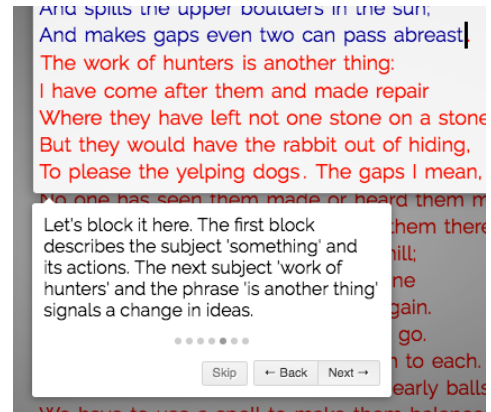


Fig. 2. Step-by-step interface

the rationale. Last, the exercise encourages users to summarize each block in their own words to aid reading comprehension. These interactive techniques help clarify this directive of blocking and aid in training the user. After the tutorial, users are presented with the poem to block themselves. Clicking on any word places a marker and alternates the colors of text to further denote sections. The exercise will lay the groundwork for further interpretation by both supporting a strong literal reading and training readers to notice the moves and shifts of a work.

Inside the Ivory Tower: Modeling the Cognitive Processes of Close Reading

Beyond the more elementary aspect of reading more carefully, there are more intricate, elusive processes of the mind that lead to literary interpretation. While close reading is abstract and difficult to define, scholars have had success at identifying certain processes.

In *Literary Learning: Teaching the English Major*, Linkon argues, “we can identify some of the qualities and common practices involved in expert reading, the cognitive habits that

experts use in reading and analyzing literary texts” (16). When reading, we decode, make connections, notice features of language, tie in personal experience, and relate to other works. Unfortunately, modeling these internal processes frequently doesn’t occur in the classroom. Linkon claims that educators “rarely highlight the thinking processes involved in noticing and interpreting form . . . [presenting] literary interpretation as a form of magic, rather than as a discipline or craft” (31). Linkon points out an educational error that I believe technology can help remedy, illustrating what exactly happens in the literary mind. Cognitive modeling, my second step, is not only an essential resource for replicating close reading, but helps demystify the process for the novice user, reducing anxiety and intimidation.

Close Reading’s educational cognitive model is actually derived from a process used to study reading. Think-aloud (TA) protocol acquires data by having participants verbalize thoughts while performing an action. The method has become popular for retrieving cognitive information stored in short term memory and then utilizing the information for problem solving and analysis. TA’s ability to model mental processes as they develop throughout a task, rather than through a final response via introspection, has made it an ideal approach to study the intricacies of reading. While commonly used in the field of usability, think-aloud has also been applied to study the cognitive processes associated with reading.

Not long after Ericson & Simon’s *Verbal Reports as Data* appeared in 1980 did researchers begin utilizing think-aloud protocol to study reading comprehension. Bereiter & Bird applied the method in hopes of gaining insight for instructional strategies. Educators are often faced with a profound ignorance regarding the many cognitive

processes taking place during reading. All in all, the study concluded that the think-aloud method of expert readers could serve as a source of information for designing instructional reading strategies (Bereiter & Bird 153). Similar to the emergence of process pedagogy in the field of composition, this same emphasis can unlock the mechanisms behind literary thinking.

Building on previous work, scholars began applying think-aloud protocol to the study of literature. Earthman used the technique in combination with reading theories such as Iser's gap-filling, which argues non-explicit texts, such as poetry, "contain gaps or blanks, places where the text is undetermined and where the reader must fill in missing information in order to make sense of the work" (357). For the study, Earthman used think-aloud protocol on two subgroups of participants, one more experienced than the other, in order to compare their reading processes. Gaps in a text that were noted and filled were quantified and analyzed. Earthman's results suggested distinct differences between the experienced readers and novices, suggesting pedagogical practices that cultivate expert processes, like gap-filling, could improve reading (380-81). The study also explicitly revealed how readers handled gaps that are frequent in an art form such as poetry.

Perhaps the most significant study would be Peskin's application of a similar expert/novice approach, this time focusing solely on poetry. The reading of poetry is an even more intricate process as the work is compressed, involving the decoding of figurative language and extended metaphor. Poetry frequently disregards syntax and linearity, putting even greater demands on the reader (Peskin 236). Peskin's study goes in more depth to tease out numerous habits present in the experts but absent in the novices. The investigation concludes, "experts employed productive interpretive strategies (such as

using structure, rhythm, wordplay, and rhyme scheme as cues, scanning to contextualize, looking for meaning at the locus of binary oppositions, and making use of visual representations to highlight structural elements)” (Peskin 256). While many theorists have already described these reading strategies, the study showed that they could be tangibly reproduced and modeled using think-aloud protocol.

Eva-Wood builds on Peskin’s work, focusing specifically on the role emotion played during the meaning-making process of poetry. Eva-Wood argues that emotion has a fundamental role in both interpretation and appreciation, asking participants to not only think aloud but ‘feel aloud’ (183). This last approach adds another cognitive complexity to the study of close reading, revealing another layer to augment interpretation. In constructing a cognitive model, including emotional responses may be useful if presented in a manner that ties it into meaning making. Encouragement of emotional responses might also be crucial to fostering appreciation, helping reverse the downward trending of poetry.

While think-aloud protocol was successfully used to further the study of close reading, the method also can be applied pedagogically. Linkon mentions the “Think-Aloud” as a classroom exercise in which students were asked to “read and begin to interpret an unfamiliar poem and to vocalize their thoughts as they did so” (17). Understandably, this approach was likely adopted for a means of modeling the thought processes associated with close reading in the classroom. Conceptually, however, this think-aloud technique inspired the cognitive modeling exercise for the *Close Reading* software. The intention would be the same as in the classroom, trying to make the process of reading poetry visible to learners.

Close Reading's technique relies entirely on visuals to cognitively model a mock reader working through a text. The exercise seeks to be an imitation of an expert reader's thoughts and responses to certain words and phrases. Graphically, the text of the poem is displayed and words are bolded and colored to represent the eye's current focus. When a

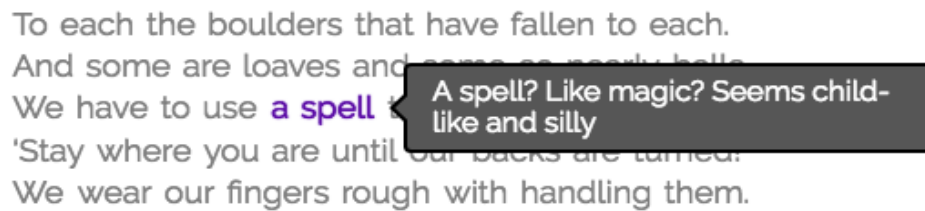


Fig. 3. Think-aloud thought bubble

feature of the poem attracts the mock reader's attention, the text is highlighted and sprouts a tooltip element displaying an expert reader's thought as text (as shown in fig. 3). These thought bubbles allow users to see the inner workings of a reader's mind. This simulates cognition as the brain makes connections, notices features, draws from experience, relates to other texts, develops questions, revises ideas, and arrives at interpretations. This method provides a more visual and tangible example of close reading. However, the focus of this effort is pedagogical, not modeling cognition as accurately as possible.

Despite the didactic emphasis, the exercise still wants to do the reading process justice as to not mislead learners. One of the most important aspects when illustrating close reading involves recursion. Reading poetry doesn't involve one linear path to understanding or even lead to one final complete reading. The process is fluid, constantly shaping and taking form, being refined over and over like a blacksmith tempering steel. One's first interpretation is often either incorrect or severely undeveloped. As Linkon points out: "Literary scholars also read recursively. We move from noticing features to posing questions to making connections, and then back to the text itself, back to what we

know about context, back to theory, again and again” (20). Therefore, the thought bubbles I develop must be recursive and revisionary in this same manner to reveal the nonlinear nature of close reading. The process will also reveal how expert readers are hesitant to draw absolute conclusions and switch between varying approaches. To accomplish this technologically, *Close Reading’s* think aloud has been designed to backtrack and reread previous lines. This technique should at least convey the recursive nature of close reading.

So what does this method look like when applied? How does think aloud aid the teaching of close reading? In Robert Frost’s “Mending Wall,” the think aloud shows how an expert reader notices characteristics of language. The words *elves*, *spell*, *game*, and *mischief* are noted as having a playful connotation that denotes the speaker’s attitude towards walls and the neighbor. The mock reader also examines the extended metaphor of walls that separates the two characters. In addition, the exercise contextualizes the neighbor’s phrase “Good fences make good neighbors” noting that it is a proverb. Last, the think aloud brings to light significant figurative descriptions, pondering why the neighbor “moves in darkness” or looks like a “savage” to the speaker. These thoughts all highlight elements of a poem that trained readers look for and interrogate to aid reading comprehension. Overall, the think aloud tangibly conveys close reading as a craft with processes while also deepening the user’s comprehension of the poem.

Pressure Points: Problematizing a Poem

The modeling I’ve just described provides an example of close reading, but support is also needed as students begin attempting the task on their own. How does one teach the abstract, intangible, unquantifiable aspects of poetry? The art form requires readers to

think critically, but this class of thinking is notoriously challenging to foster in education.

One technique commonly associated with teaching critical thinking is also one of the oldest—Socratic questioning. Socratic questioning attempts to extract understanding from an individual by stimulating and guiding their thinking. The advantage of the technique is that knowledge is generated within the individual rather than transmitted by the teacher, transcending simple memorization. In “Critical Thinking: The Art of Socratic Questioning,” Paul and Elder elaborate on this dynamic: “The goal of critical thinking is to establish an additional level of thinking, a powerful inner voice of reason. Socratic discussion cultivates that inner voice through an explicit focus on self-directed, disciplined questioning” (37).

With this method, educators are going beyond the relaying of information by cultivating the thinking needed for close reading. Readers need a potent inner voice to direct, guide, and question the aspects of poetry, which is why Socratic questioning is well suited to this task.

To accomplish this tangibly, I first argue that poems have pressure points. These are specific phrases or words that educators and readers can ‘press’ to make the poem speak. In literacy, pressure points can be compared to a similar concept known as structural units. A structural unit is a phrase positioned in relation to the theme or main idea of a work, measuring the unit’s essentialness to overall meaning. To provide an example, if a structural unit can be omitted without altering the meaning of the text, then the value is low. To study this theory, Brown and Smiley presented subjects with a story broken down into structural units. Subjects were told that individual units differ in their structural importance to the whole story and were asked to rate their importance on a scale from 1 to 4. While the two younger age groups of third grade and fifth grade seemed incapable of discerning difference, the results concluded “[c]ollege students can distinguish each level of

importance, and seventh graders can separate low, medium, and high levels” (Brown & Smiley 5). In determining the most important structural units, college students had an average 88% accuracy (Brown & Smiley 6). These findings suggest that determining the importance of structural units could be an important component to reading comprehension, improving with age, education, or experience.

Likewise, I argue that poems have similar important units known as pressure points. By flagging these pressure points and then questioning the reader about them, *Close Reading* guides the individual to formulating their own interpretations. As questions frequently drive our own thinking, they can be further used to support novices who may not know the right type of questions to ask or where exactly to look for them. I propose using this technique to help guide learners, presenting questions in a strategic manner to build off previous exercises. These pressure points will seek to highlight the problem(s) the poem is trying to address. Like a typical novel, most poems deal with some sort of conflict that the author is trying to resolve or explore. While the previous ‘think aloud’ is a more passive exercise, pressure points involves informal writing, ensuring an active consideration of the questions and an active analysis of the poem. Pressure points will provide a metaphoric representation to aid users in performing this task of arriving at a statement of the poem’s “problem.”

To accomplish this technologically, *Close Reading* presents the text of the poem with key phrases highlighted in red. Users will then click these pressure points to make a prompt appear on the right margin of the screen. A textbox will be presented to respond to the question, each relating to some problem that the poem is trying to address (shown in fig. 4). Users may respond to as many pressure points as they wish, but must answer to at

least one before progressing. While there is no evaluation mechanism for the writing produced in *Close Reading*, there is significant value in informal writing in assisting one's thinking.



Fig. 4. A pressure point prompt and text box.

To provide an example of the pedagogical benefits, in Frost's "For Once, Then, Something" the speaker tries to perceive past the surface of the well. When finally achieving success, a drop of water falls to "rebuke the too clear water." The associated pressure point asks, "What does the poem suggest about the moments we finally 'see through'? Why is water not supposed to be clear?" The prompt allows the reader to openly explore one of the key problems of the poem—the elusiveness of truth. The pressure point ties this issue to the text via the highlighted phrase and also guides the user with questions to probe the problem. This dual approach helps the reader foster interpretative habits that won't stray too far from a work.

Reflection: Reading Inward

Reading is also personal, informed by our own memories and concerns. Music exemplifies this dynamic as listeners are frequently hearing their own stories sung out by the lyrics of another—almost as if the song were written for them. Rational scholars know better and

temper bias when approaching a work. Poor readers view a work as a mirror, seeing their own lives played out in prose. Good readers understand that works are windows; you may occasionally catch a glimpse of reflection, but truly you should be looking through. But good readers tread this tightrope, reading simultaneously from their own lives and from the text. Therefore, learning to relate the insights and problems of poems to oneself is not just existentially fulfilling, but also improves interpretation.

Scholars have noted this process of relating to our own experience as a part of reading. One of the three models of reading discussed by McCormick, the expressivist model “privileges the reader and the reader’s life experience in the reading process” (13). McCormick somewhat undermines the model and adds the negative connotation of one reading themselves in a work, ignoring the objective text. This thinking, however, is a fallacy, for relating to one’s life experience doesn’t necessarily need to be “privileged” over the text. I argue that reflective practices can augment the reader’s ability to interpret a text if properly restrained. While novices may have a tendency to see themselves in a work of literature, savvy readers can pick up on the communalities of human existence, empathizing to enhance reading.

Close Reading seeks to capitalize on this dynamic by introducing reflective practices and having users apply the concepts found in poems to their own lives. Ideally, the close reading of poetry is a transformative experience that can aid users in resolving inner conflicts and shaping their worldviews. Therefore, the software needs an exercise where students pull away from the text and begin turning inward. The reflection step remains crucial to conveying poetry’s value, the true reward that keeps us straining to comprehend stanza after stanza. The exercise comes last after users have developed a strong reading

grounded in the text of the poem. Novices may muddle their interpretation by tying in their own experiences too early. Practicing reflection may then transfer into user's future reading, strengthening inward connections and fostering interpretations.

To accomplish this tangibly, users will reflect on a prompt that relates to the problems encountered in a poem. For example, Frost's "Mending Wall" deals with the extended metaphor of walls between people and the rigidity of a neighbor. The prompt asks, "Try to think about the problem of the poem. Does it reveal anything in your own life? Can you think of a time where you dealt with walls, rules or barriers? With rigid friends or neighbors?" Asking users to relate the idea to their own lives might trigger memories of a relationship where walls or rules or boundaries were a source of tension. This insight allows one to better perceive elements of the poem, for readers will track the exigence for Frost's ideas. This also unlocks poetry's transformative potential, leading users to possibly rethink approaches to relationships.

Technical Specifications

Close Reading seeks to further the development of educational technology that has begun to take off over the past decade or so. Much of the current instructional design practices rely on course authoring software such as Articulate, Lectora, and Captivate. These software are designed to make course creation accessible to anyone with some patience and moderate tech skills, but overall these types of programs are limited to a spectrum of tactics for presenting information. Innovative websites, like *Codecademy*, have utilized the flexibility of web development to design more sophisticated and unique user interfaces that augment

learning. This new brand of educational technology has the potential to revolutionize the field but has yet to become a pervasive practice.

While plenty of course authoring software has been designed for e-learning, there seems to be an absence of a front-end framework focused on instructional design. One might point to the e-learning framework *Adapt*, but this tool only provides structure to course creation rather than help actualize sophisticated UI (user-interface) development. Designing these new UIs relies on expert knowledge of HTML/CSS and DOM (document object model) manipulation. The document object model is a source of both utility and frustration for creative workarounds are frequently needed. For example, one cannot select an individual word in a paragraph tag using the DOM. Workarounds are common when developing advanced UIs and problems only increase when factoring in browser and device support.

As far as developing these innovative UIs, developers are faced with a myriad of technologies to choose from due to the web's increasing variance. Web developers sometimes suffer from analysis paralysis just trying to pick the right framework or library. There also exists the difficulty of keeping up with an ever-changing field that constantly produces new software. The JavaScript library jQuery has long been a staple for development, providing a wide array of support for creating sophisticated UIs and backed by a strong plugin community. There have been trends of veering away from the user-friendly jQuery due to the advancement of web application program interfaces (APIs), creation of new frameworks, and jQuery's lack of innate structure. In addition, new technologies are continuing to be developed to optimize web based UIs, such as React's virtual DOM. However, the degree of DOM manipulation for this project calls for a Swiss

army knife like jQuery that can tackle some of the most intricate challenges. *Close Reading* utilizes this storied library and implements JavaScript patterns, like the object literal, to address structural concerns.

The Course Shell

Foundationally, *Close Reading* needed a stable and consistent interface to host a series of pedagogical exercises. This need led me to develop the course shell, a web page interface where content could be loaded into two distinct sections via the PHP backend of the software. The primarily technical challenge was designing the shell to always take up the entirety of screen. Overflow of content, instead, was relegated to the two areas rather than the <body> of the page. This led to the creation of two separate <section>s referred to as #info and #course. I achieved consistent section width by using the CSS width property's percentage inputs (30% for #info and 70% for #course). These sections were then floated to appear side-by-size and the overflow property set to hidden to avoid any disruptions caused by content. Inner <div>s were placed in these sections to hold the actual content, utilizing the overflow-y property to produce a scrolling effect. The one technical challenge was making each section also cover the entire height of the page. Neither the CSS percentage or viewport height inputs were effective in setting the correct height property for #course and #info. To address this issue, I wrote a jQuery script to adjust the height properties of each section during the 'ready' and 'resize' events. Section height was calculated by obtaining the 'window' height and subtracting the <header> element's height. Thanks to the script, the course shell consistently took up the entirety of the page, leading to a responsive UI framework.

Intro.js – A step-by-step guide

A UI element was eventually needed to model the process of blocking a poem. Fortunately, web developers have had to tackle similar instructional needs when providing a tour of a website to first time users. Intro.js is a JavaScript plugin that allows developers to create a step-by-step guide that highlights specific elements of a page. I adopted the software to model the process of blocking and also to provide *Close Reading's* own rationale for sectioning the poem. Intro.js did not function properly when applying the script to the scrollable section in #course, rather than the default of the entire <body>. I made edits to the plugin to fix the tooltip layer, a <div> container for step-by-step text content, when no screen element was being highlighted. I also made changes to properly calculate the width of the tooltip layer and fix the positioning by setting the top and left properties to zero. Position errors when using Intro.js with elements inside a scrollable div were corrected by using jQuery's .position() method, rather than the offset property of the element. The .position() method retrieves the top and left properties relative to the parent element, rather than the document. The jQuery .scrolltop() method was also used to factor in scroll position, adding the value to the top property retrieved from .position(). I also used jQuery's .animate() method to make changes to the scrollTop property when an element was not in the viewport, providing smooth transitions between steps. Last, I added a disable feature that stopped the progression of steps when the disabled class was present on the *Prev* and *Next* buttons. jQuery's .hasClass() method was used to determine if the class was present, preventing step progression if the value was true.

Last, I also implemented modifications by using the Intro.js API when initially calling the function. To prevent the error of changing steps too quickly, the `.onafterchange()` method was used to add the disabled class to the 'next' and 'previous' buttons for 2000 milliseconds when a user switched steps. In addition, to add custom scripts throughout the tour, I added a `switch()` statement to take the property input `_currentStep` and run a unique function during a specific step (ex. step 2). This mechanism was used to add CSS animations, such as bouncing and highlighting, around specific words. In addition, I ran a function during a step to sprout a tooltip using jQuery plugin Tooltister. For progressing after completion, the method `.oncomplete` was used to trigger a URL change (`document.location.href = nextPage`). Few modifications were used when providing *Close Reading's* own blocking of the poem. Overall, intro.js proved a valuable plugin to achieving the desired UI behavior, and I was able to fix most errors using jQuery.

blocking.js

I designed several functions to develop the blocking UI for users to section off the poem on their own. Originally encountered when developing the think aloud, one of the initial problems occurred due to the nature of the document object model. Web APIs and jQuery cannot select individual words in an element. Developers have overcome this limitation by wrapping each word in a `` tag. The jQuery plugin Blast.js was designed to perform this function, facilitating typographical manipulation. I wrote two key functions for `blocking.js`: `turnOnBlocking()` and `calculateBlocks()`. The former blasts the words apart and adds two major event handlers to each word: 'click' and 'mouseenter.' Both handlers are designed to add the '.marked' class when triggered and run the `calculateBlocks()` function.

The marked class adds a line after the word, representing someone marking that section off. This class is added when the user hovers over a word (and is removed on 'mouseleave') or clicks a word.

After the class is added, the `calculateBlocks()` function groups words into actual blocks, adding CSS color properties that alternate between even and odd blocks. I first used the jQuery `.each()` method to iterate through every word of the poem. A condition checks to see if the current word has the `.marked` class. If not, the word is added the current `blockid` class. If the word is marked, the `blockid` variable is increased by one to denote the next block. To evaluate parity of the blocks, I applied the modulo operator to the `blockid` variable. An associated 'even' or 'odd' class is then added to the word to style alternating colors. The resulting class list for the first group of words ends up as: `blast.block0.even`. I also developed code to reset classes each time the function was run so that blocks were properly recalculated when new markers added. The jQuery `.removeClass()` method easily removed the 'even' 'odd' classes, but could not select all `blockid` classes that contained a unique variable (`blockid`). To resolve this issue, I passed a function for the `.removeClass()` argument to return matched class names. JavaScript's string `.match()` method was used in correlation with a regular expression to pull all `blockid` names. The `.join()` method combined the returned array before passing it to the `.removeClass()` method, successfully removing all previous block classes to recalculate the new blocks.

thinkaloud.js

The think aloud exercise is the most innovative user experience I developed for *Close Reading*. The code, `thinkaloud.js`, also makes use of the `blast.js` plugin, as well as the plugins

Tooltipster and jquery-timing. The data for the module is embedded in html properties (as shown in fig. 5) to provide easy access to front-end developers and aid in visualizing the deployment. To function, lines of a poem are arranged in `` tags that take a data property for the line order. This lets developers control the sequence, allowing lines to be repeated and done out of order. In each `` tag, developers can add a span tag with the class `.tt` and a title attribute. These properties will denote a tooltip that will launch. In addition, a data property, named `ttorder`, for the tooltip order is included.

```
<li data-lorder="1 5">
  <span class="tt" data-ttorder='3' title="What exactly is this 'something' that makes gaps and
  dislikes walls? Is it nature or just some force?">Something there is</span> that doesn't love a
  wall,</li>
<li data-lorder="2">
```

Fig. 5. HTML markup of thinkaloud.js

The thinkaloud.js function first collects the relevant data from the HTML. An `.each()` method iterates through all `` elements, pulling the line order data, along with the associated `` element, into an array to be used later. To repeat a line, users can include a space with the next numeric order. The data is then `.split()` and included in an object with the associated `` element. This object containing the order and element is then pushed into an array called `lineOrder`.

Next, I used the `.repeat()` and `.until()` methods from jquery-timing to continuously loop through the `lineOrder` array until the highest number (the last line) is found. At each line's iteration, the `colorLines()` function is run to color text and launch tooltips. When the `colorLines()` function becomes triggered, a JavaScript Deferred object is created. JavaScript is a synchronous language, meaning that events occur simultaneously. To stop the code from coloring all lines at the same time, I used the Deferred object with the jquery-timing

.wait() method. The Deferred object is later resolved once the colorLines() function has finished, allowing for the resumed progression of the loop.

To return to colorLines(), at each iteration the line order is pulled from the array's object and compared to the line queue. If matched, all words are selected in the line and iterated through. CSS animations bold and unbold each word and jquery-timing's .wait() method is used to add a delay of 800 milliseconds between words to give the impression of reading. Once all words in the line have been bolded and unbolded, a function is launched to select all tooltip tags and grab the 'ttorder' data. If the tooltip order matches the tooltip queue, the tooltip is launched via the Tooltipster API 'open' method. As soon as the tooltip closes, the Deferred object is resolved and the loop resumes, moving on to the next line to color. Overall, thinkaloud.js remains the most complicated and messy of the code as it was the first exercise I created of the *Close Reading* software. While functional, the structure and cleanliness of the code could be significantly improved.

pressurepoint.js

Unlike thinkaloud.js, pressurepoints.js utilizes a JavaScript pattern to provide structure to the jQuery code. I used the object literal pattern, containing all code as a single object with a series of methods. The .get() method takes a selector as an argument (such as a class name) and collects the embedded HTML data. Pressure points are included in span tags with a data attribute 'note' containing the associated prompt. Once the .get() method is run, a data object is created with the element and attribute. After .get(), the .init() method can be run to initialize the pressure point exercise on the web page. The .init() method attaches an id data attribute and a click event handler to all span tags.

When the 'click' event is triggered, the `openPP()` method retrieves the 'note' data attribute and injects it into the html of the annotation. Then, the method also visually displays the annotation. The graphical elements of the pressure point consist of an annotation `<div>` positioned next to the poem inside of a larger sidebar `<div>`. Initially hidden, this annotation `<div>` is revealed and repositioned by the `.openPP()` method. The top position of the `` tag is then collected and added to the `scrollTop()`. These values, along with the window height, are used to calculate the top position of the annotation so that it appears directly next to the `` tag. The function also loads any associated user annotations that have been previously submitted and stored in the `pressurePoint` object. This is accomplished by using the pressure point's id to store annotations in the `pressurePoint` data property. Overall, `pressurepoint.js` shows the potential of jQuery to retain structure and semantics if proper JavaScript patterns are applied.

Conclusion

Poetry's readership may be in decline, but we can do something about it. The subculture of poetry seems content to continue narrowing into an esoteric path of obscurity. What good is poetry if so few people read it? Why have we allowed the ivory tower to grow taller, rather than try to build stairs for the public? *Close Reading* aims to be a small step in a long journey to reinvigorate the people's interest in poetry, providing the needed resources to bridge the learner's gap and allow users to experience the value of poetry.

To accomplish this, *Close Reading* fosters the necessary skills by progressing users through four pedagogical exercises: blocking, modeling, Socratic questioning, and reflection. Users develop the necessary reading habits that cultivate an attention to the

shifts and linguistic features of a text. Modeling shows users what elements to pay attention to in a work as well as revealing the cognitive processes needed to close read. Socratic questioning provides a form of scaffolding for users to help form interpretations, linking questions directly to passages in the text. Last, reflective habits are encouraged via a writing exercise that applies the lessons to the user's own lives. These pedagogical exercises not only teach users how to read a poem, but also expose them to features that help foster appreciation. This novel approach attempts to train users, viewing the reading of poetry as a skill, rather than a body of knowledge.

Likewise, *Close Reading* also seeks to progress the field of educational technology by applying the tools of the web to develop a sophisticated interactive course. While no framework aimed at developers currently exists for such tasks, the project has the potential to be developed into its own interactive e-learning framework. This framework would have a UI focus and be well suited for text-based learning. *Close Reading* not only takes an experimental approach to the teaching of poetry, but also introduces novel UI concepts. Exercises like the think aloud have applications beyond poetry and could hold even greater pedagogical value. *Close Reading* pushes the direction of several fields, not just poetry, and has the potential to impact educational technology, poetry pedagogy, and the digital humanities.

Works Cited

- Bereiter, Carl, and Marlene Bird. "Use of Thinking Aloud in Identification and Teaching of Reading Comprehension Strategies." *Cognition and instruction* 2.2 (1985): 131-156. Taylor & Francis. Web. 10 Apr. 2017.
- Brown, Ann L., and Sandra S. Smiley. "Rating the importance of structural units of prose passages: A problem of metacognitive development." *Child development* (1977): 1-8. JSTOR. Web. 10 Apr. 2017.
- Brozo, William G., Ronald V. Schmelzer, and Hiller A. Spires. "The beneficial effect of chunking on good readers' comprehension of expository prose." *Journal of Reading* 26.5 (1983): 442-445. JSTOR. Web. 10 Apr. 2017.
- Cromer, Ward. "The Difference Model: A New Explanation for Some Reading Difficulties." *Journal of Educational Psychology* 61.6, Pt.1 (1970): 471-483. APA PsycNET. Web. 10 Apr. 2017
- Earthman, Elise Ann. "Creating the virtual work: Readers' processes in understanding literary texts." *Research in the Teaching of English* (1992): 351-384. JSTOR. Web. 10 Apr. 2017.
- Ericsson, K. Anders, and Herbert A. Simon. "Verbal reports as data." *Psychological review* 87.3 (1980): 215. APA PsycNET. Web. 10 Apr. 2017
- Eva-Wood, Amy L. "Thinking and Feeling Poetry: Exploring Meanings Aloud." *Journal of Educational Psychology* 96.1 (2004): 182. APA PsycNET. Web. 10 Apr. 2017
- Gioia, Dana. *Can Poetry Matter?: Essays on Poetry and Culture*. St. Paul, Minn: Graywolf Press, 1992. Print.

- Ingraham, Christopher. "Poetry Is Going Extinct, Government Data Show." *The Washington Post*. WP Company, 24 Apr. 2015. Web. 10 Apr. 2017.
- Krill, Paul. "Stack Overflow Survey: Nearly Half of Developers Are Self-Taught." *InfoWorld*. IDG Communications, Inc., 10 Apr. 2015. Web. 1 Apr. 2017.
- Linkon, Sherry Lee. *Literary Learning: Teaching the English Major*. Bloomington: Indiana University Press, 2011. Print.
- McCormick, Kathleen. *The Culture of Reading and the Teaching of English*. Manchester University Press, New York; Manchester, 1994. Print.
- Neville, Mark A. "Who Killed Poetry?" *The English Journal* 47.3 (1958): 133–138. JSTOR. Web. 10 Apr. 2017.
- Paul, Richard, and Linda Elder. "Critical Thinking: The Art of Socratic Questioning." *Journal of Developmental Education*, 31.1 (2007): 36–37. JSTOR. Web. 10 Apr. 2017.
- Peskin, Joan. "Constructing meaning when reading poetry: An expert-novice study." *Cognition and Instruction* 16.3 (1998): 235-263. Taylor & Francis. Web. 10 Apr. 2017.
- Risha, Zak. "Close Reading: Learn to Love Poetry." *Close Reading*. Zak Risha, 9 Apr. 2017. Web. 5 May 2017. <<http://zakrisha.closereading.com>>.
- Stevens, Kathleen C. "Chunking material as an aid to reading comprehension." *Journal of Reading* 25.2 (1981): 126-129. JSTOR. Web. 10 Apr. 2017.