Natalie Berkman
Princeton University, Center for Digital Humanities
Summary of Accomplishments

Digital Oulipo:
Programming Potential Literature

1. Narrative Description

The explicit goal of this project was to produce five digital annexes to accompany my dissertation, and I am happy to report that I have completed four of the five that I originally proposed (I will explain why I did not complete the fifth below). I hoped that these programmed editions of Oulipian texts or procedures would help my reader understand the underlying mathematical principles, offering a ludic introduction to what many believe to be a dry, abstract, and difficult discipline. However, carrying out a digital humanities project on the Oulipo — itself a quasi-academic research group that both analyzes and creates — resulted in a hybrid experiment with multiple, varied results and insights, which helped me both understand the history and development of Oulipian aesthetics as well as understand their texts better. While my annexes are functional and amusing to play with, the *real* product has been the important insights I have gained from the experience.

In a sense, the product was a pretext in order to take the idea of the Oulipo and in a quasi-Oulipian fashion, think about formalizing it on computers. From learning to program in Python (even though Python is not the language early Oulipians used in their computer endeavors), I have gained a greater understanding of early Oulipian techniques, why the Oulipo abandoned computer programming relatively quickly and relegated this sort of activity to the ALAMO, and a greater appreciation for how exactly to use digital humanities tools to explore new avenues in these texts that were created with algorithmic procedures in mind. I have learned that digital humanities can be a powerful tool when used properly. As with Oulipian computer-texts that lost their interest when programmed onto a computer, digital tools are best when properly understood and critically implemented, at least with regards to Oulipian work. Throughout my work, I was forced to approach these texts in a new way, critically designing my own programs to "read" in a way that would be faithful to Oulipian principles.

Annex 1: Set Theory
I abandoned the idea of making an annex for my first chapter, as the Oulipo never invented any formal constraints that were explicitly set theoretic. Creating an appropriate annex would have required me to engage in formal Oulipian work myself, which was not in the scope of my project. Understanding that this annex was out of the scope of this project was paradoxically helpful to me in my own research, as I was able to refine my understanding of the Oulipo's quasi set theoretical constraints, which will greatly contribute to my rewriting of Chapter 1 of my dissertation.

Annex 2: Algebra (S+7)
This digital annex was the last one I completed, consisting of a modest S+7 generator that operates using nltk in Python. Since this is such a canonical Oulipian technique that is often mentioned but very rarely analyzed, my hope was to provide the reader of my chapter 2 with a program to generate his/her own S+7's. Through my programming and research, it became clear that three elements are at play in the S+7 that determine its result: the base text, the dictionary, and the number of the offset. My program therefore allows the reader to choose a base text from a set of excerpts from recognizable English works: *The Declaration of Independence*, *Genesis*, *The Raven*, *A Tale of Two Cities*, and *Moby Dick*; a specific dictionary generated from specific authors or texts (all the nouns from the Bible, for instance), and an offset of his/her choice.

Devising this program has forced me to consider the S+7 from a critical perspective, rather than a purely inventorial one. While I had initially included the S+7 as a basic example of an arithmetical Oulipian technique, I now see it as a first attempt in a long line of Oulipian research on productive substitutions. These valuable insights, while not necessarily unique to digital humanities work, were facilitated and brought to light as a direct result of this project. The collaborative nature of the project is one reason: discussions with Cliff Wulfman enabled both of us to understand this procedure simultaneously as inspired by early computers and algorithmic methods. The specific tools I used make up a second reason: studying and implementing natural language processing with nltk forces one to consider language differently, which is akin to the Oulipian mindset that spawned these early procedures.

Annex 3: Combinatorics (Raymond Queneau's *Cent mille milliards de poèmes*)
I began my project with the *Cent mille milliards de poèmes* (1961) annex with two main goals: first, to use this text as a practical introduction to computer programming, allowing me to understand the Oulipo's struggle with this text's computer legacy[1]; and also to use the creative enterprise of designing new methods to produce pseudo-random sonnets in order to grapple with the Oulipian notion of chance myself.[2]

To transform Queneau's work, I first had to visualize the original 10 sonnets in computer science terms. A sonnet, by definition, is an array of fourteen verses that can each be attributed a numbered index. Once I had this data structure, I wrote a program that would generate a pseudo-random sonnet from a 14-character key (the digits of this key would each be between 0 and 9, indicating which of the original 10 sonnets to take the verse from). Finally came the creative part: I had to decide how to produce "pseudo-random" sonnets in a way that would remain faithful to the Oulipian notions of chance. In the end, I came up with three different ways:

1. **Age:** The reader inputs his/her birthday and the program uses the digits in the birthday to generate the key. The calculations I subject the digits of the birthday to in order to produce the key are invisible to the reader, and have nothing to do with the reader's age. My next goal is therefore to tweak it to calculate exactly how old the reader is (in minutes) at the time the poem is generated and use that number to generate a poem.
2. **Location:** The reader is directed to a website which gives him/her a latitude and longitude value which he/she then inputs into my program. These numbers are manipulated to create the key. In this way, the reader can generate sonnets from not only his current location, but any location of which he/she has the latitude/longitude. The implications of this type of pseudo-random poem are certainly in the spirit of the Oulipo, but once again, the actual calculations to which I subject the data to generate the key are too divorced from the method to be considered truly Oulipian.
3. **UUID:** A UUID is generated from the user's computer, which will produce a new sonnet each time (given that the particular UUID that was enough characters and only digits) is dependent on the time.

While functional, my annex is not intellectually stimulating to a reader. That said, perhaps *programming* this programmatic text is one viable way to read it. Queneau claims at the outset that his little poetry volume is utterly impossible to read in a human lifetime, but also claims that *only* a computer can appreciate the poems. Clearly, to have any chance at "reading" the volume, one must read it as a machine. Through programming, I have gained a more thorough

[1] The result of this goal is detailed and analyzed in chapter 4 of my dissertation.
[2] The insights I gained from this creative work can be found in my analysis of the *Cent mille milliards de poèmes* in my chapter 3, which corresponds with this annex.

understanding of this text and its implications, but I also understand how Queneau's choice of extremely specific subject matter for each of his original poems is the key factor in the hilarity of the potential poems. Each verse of Queneau's original poems is essentially *tagged* with vocabulary linking it to its specific subject. A verse from the first poem that is imbued with an Argentinian vocabulary, for instance, cannot possibly be interchanged with a verse about the Parthenon without producing a destabilizing effect. And while certain combinations of these verses may indeed be funnier than others or produce unexpected effects, it is precisely the content of the original poems (that is to say, what would be invisible to a computer reading the text) that produces the surrealist effect of reading the potential poems. While my methods for generating pseudo-random poetry given Queneau's originals may not be satisfying, I can use my program to analyze the text further. My next goal is therefore to take my program and replace Queneau's poems with Shakespearean sonnets that all have more or less the same theme. I'm curious to see if poems with similar thematic material are as funny when subjected to such a machine.

Annex 4: Algorithms (Raymond Queneau's *Un conte à votre façon*)
At this point in the project, the *Cent mille milliards de poèmes* annex had taught me some programming basics, but Cliff suggested that I try to find a way to become better initiated with programming. I worked independently through *Learn Python the Hard Way*[3], and at the end of the program, the author had an example where you could make a program that allowed a reader to move through a "Map" with a pre-established number of scenes. For my first semi-autonomous programming, I adapted that program to make my own digital *Un conte à votre façon* (1967). While I was proud of myself for writing a program, I was still not satisfied. I wanted my program to generate a possible graph, just as Queneau himself had done by hand.

Cliff introduced me to graphviz, an open source graph (network) visualization project that is freely available for Python. With this graph theoretical program in my Python arsenal, I was able to make my own graph of *Un conte à votre façon.* I integrated the two programs, giving my original program the structure of a graph so that it would be able to generate a graph with graphviz at the end. My program now has classes of nodes and edges that correspond with Queneau's prompts and choices. With this data structure, the program enters the graph at the first node and, depending on the choice of the user, proceeds through the story, printing out the selected nodes and the corresponding edges. The program records the path the reader takes and at the end, prints out a graph with the reader's path filled in green, to represent the little peas. While my program does not take into account the full potential of the intersection of graph theory and literature as proposed by Queneau's text, I am very pleased with how it functions.

Annex 5: Geometry (Italo Calvino's *Le città invisibili*)
This annex was took the form of a Tinderbox hypertext, based on the geometric representation of the table of contents of Italo Calvino's *Le città invisibili* (1972). While productive in its own way, this annex is fundamentally different from the other three, given its creation using outside software and its purpose as a research and note-taking tool, rather than as an interactive, digital edition of the text itself. Indeed, I am still using the TinderBox hypertext in to organize the mathematical and theoretic influences of Calvino, which is greatly informing my Chapter 5, regardless of its status as an official annex. While designing the structure of the hypertext was useful to me as I write on the larger implications of this geometric design and its manifestation in the table of contents, making this TinderBox hypertext available to others would be a project in and of itself (and most likely impossible considering copyright laws).

[3] Zed Shaw, *Learn Python the Hard Way*. (http://learnpythonthehardway.org)

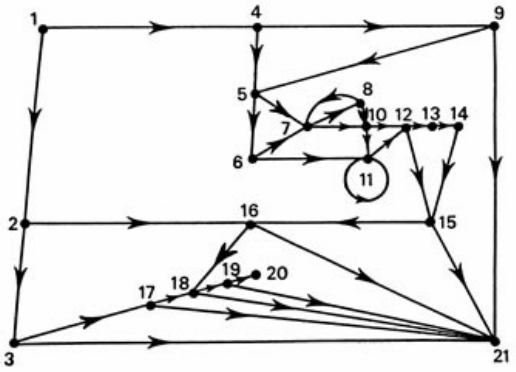
Natalie Berkman
Princeton University, Center for Digital Humanities
Summary of Accomplishments

2. Updated Project Blurb

Through this project, I designed pedagogical, interactive digital versions of various experimental texts that I discuss in my dissertation on mathematics in the OuLiPo (Workshop of Potential Literature, founded in Paris in 1960). The three annexes I created correspond with my second, third, and fourth chapters of my dissertation, complementing the critical analysis. The first deals with algebra and operations on basic units of language, focusing on the S+7 method, which replaces every noun (substantif) with the one that comes seven entries later in a dictionary. My program allows the reader to experiment with the method itself, carrying out the method using specific dictionaries generated from the collected works of certain authors. The second annex is an electronic edition of Queneau's famous *Cent mille milliards de poèmes* (1961) that allows the reader to generate pseudo-random poems from the 10^{14} proposed by Queneau, based on his/her age, location, and computer. The third and final annex programs Queneau's interactive, choose-your-own-adventure story, *Un conte à votre façon* (1967). Through this example, the reader can experiment with graph theory and learn about the mathematical possibilities of this simple, yet elastic tale, as my program generates a particularly interesting graph corresponding to the various paths available in the story and the specific one the reader has taken.

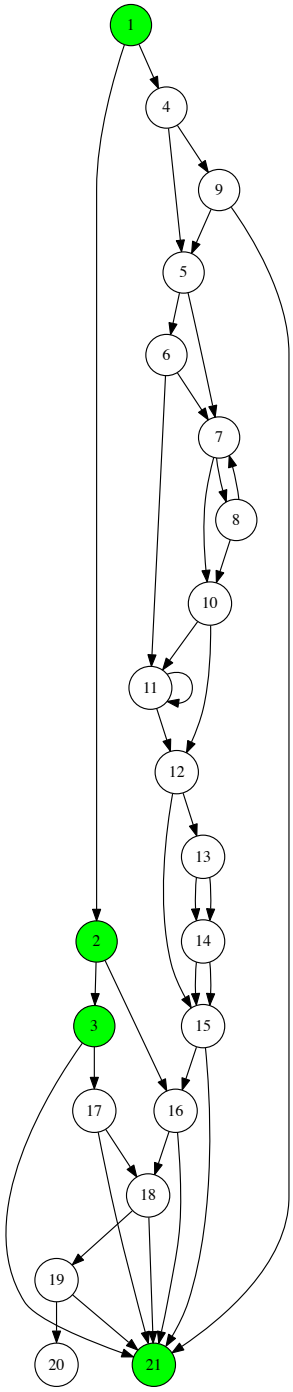
Oulipian texts challenge traditional notions of reading. The reader no longer passively absorbs the text written by another, but must play and construct by himself if he wishes to get anywhere at all. The topic of my dissertation examines the various mathematical aspects in play in Oulipian production and answers this fundamental question: how does the addition of mathematical thought change the reading experience? By offering the reader of my dissertation with electronic versions of Oulipian texts, that reader can learn firsthand how such texts disrupt any traditional linear mode of reading, and also enjoy a ludic introduction to various mathematical principles. Through the act of designing these electronic editions myself, I have gained a better understanding of these texts and how they operate, as well as a practical introduction to digital humanities and computer programming.

Images:

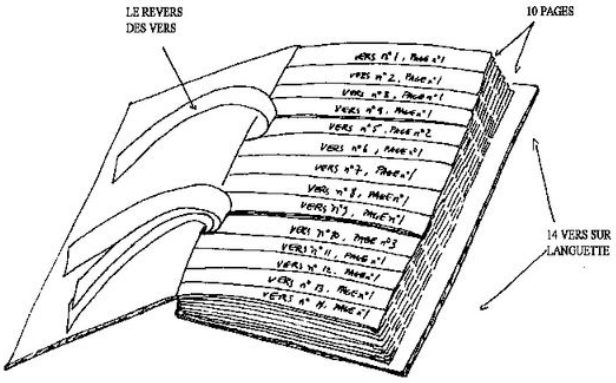


Description: Graph of Raymond Queneau's *Un conte à votre façon*. This is the graph that the author designed himself, but not the only possible visual representation of his story's possibilities.

Image Source: http://www.infolipo.org/ambroise/cours/immediat/images/conte_graph.jpg



Description: This is an example of a graph produced by my *Un conte à votre façon* program. Mathematically equivalent to Queneau's own graph, mine was produced by graphviz and thanks to its linear layout, is a bit more readable (but perhaps less aesthetically pleasing) than Queneau's.



Description: This is a visual representation of how Raymond Queneau's *Cent mille milliards de poèmes* functions as a physical volume.

Image source: http://openset.nl/blog/wp-content/uploads/2012/08/Prevert.jpg

3. Budget
$250: Tinderbox license
$1,008.63: Flight to participate in the "Poetics of the Algorithm" conference at the Université de Liège, Belgium, where I presented this work.
$807.36: Other conference-related expenses, including hotel, gas, and tolls

4. Data Generated

Link to GitHub repository: https://github.com/Princeton-CDH/digital-oulipo