A Simple 1980s Story Machine with Lessons for Narrative and Computing*

Nick Montfort[1,2]

[1] *Massachusetts Institute of Technology, 77 Massachusetts Ave Room 14E-316 02139 Cambridge, Mass., United States*
[2] *Center for Digital Narrative, University of Bergen, Langesgaten 1-3 NO-5020 Bergen, Norway*

Abstract

There are computational models of narrative found “in the wild,” ones developed for popular use rather than research purposes. *Story Machine,* commercial educational software from the early 1980s, is very simple as a narrative model, but nevertheless shows how a model of narrative can (1) be exposed to users, (2) embody concepts of the world and social norms, and (3) be fun. The multiple versions for different home computer platforms make detailed analysis complicated in some surprising ways. Because *Story Machine* is oddly compelling, I have started to collaborate on further study of the system by describing the different versions in greater detail and reconstructing an abstraction of this software. I have also engaged with *Story Machine* through artistic practice.

Keywords

story generation, narrative models, human computer interface, educational software, platform[1]

1. Introduction

I introduce and discuss a computational model of narrative that was created for popular use rather than as a means for researching aspects of narrative, the creative writing process, or cognitive science. *Story Machine*[1,2,3,4,5,6] is commercial educational software from the early 1980s. It models narrative in an extremely simple way, but I argue that it nevertheless makes interesting assertions, providing sometimes provocative answers to the following questions:

1. How can a model of narrative be exposed to users?
2. How can models of narrative embody concepts of the world and social norms?
3. What makes narratives, and play with narrative on a computer, fun?

The lessons that *Story Machine* offers are not universal answers to these questions: The software is not revealing *every* way users can access narrative models, for instance. Rather, the main lesson here is that even a very simple model of narrative can address important questions such as these. My position, then, is that a computational model of narrative can be simple[2] but still embody meaningful responses to important questions.

Beyond this, I note complexities that arise in trying to understand and analyze *Story Machine*. The main one is that there is no “there there”: different versions of the program for different home computer platforms had slightly different models exposed in different interfaces. I also report on (1) the very preliminary stage of a project that will eventually include creating systematic miniatures (“nubs”) of *Story Machine* and many other systems, and (2) a completed artistic project that has provided some insight and from which more can be learned. Finally, at the end of the discussion, I mention other directions that could be pursued, some of which I may take in collaboration with other researchers and several of which are general to the study of existing computational narrative systems.

****8th *International Workshop on Computational Models of Narrative: May 28–30, 2025, Geneva*

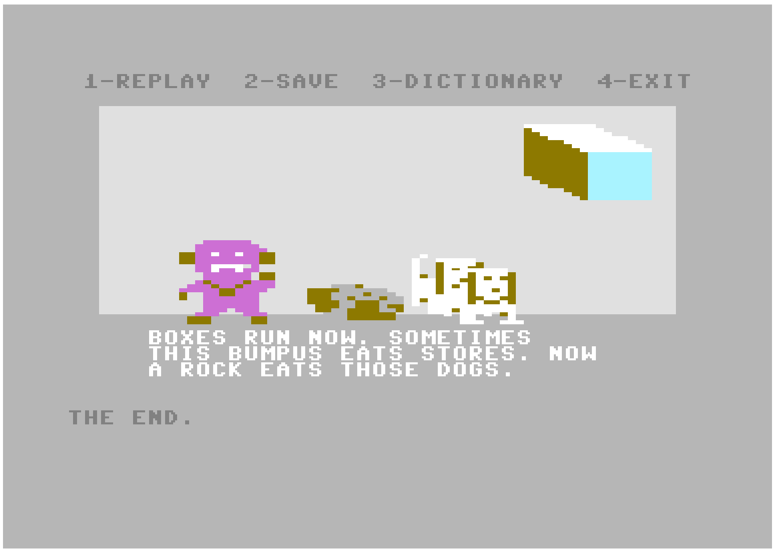
[1] ✉ nickm@nickm.com 0000-0001-7558-5160

[2] This paper isn’t an argument about whether *Story Machine* is truly a minimal computational model of narrative, or a discussion of what the simplest possible model would be. However interesting a topic, that’s a different and much more theoretical one. It suffices to note that the model here is exceptionally simple.

2. Narrative Modeling for Kids, on Home Computers

For the purposes of our discussion, *Story Machine* is a model of narrative in which grammatical sentences, using a fixed vocabulary, strictly correspond to valid narratives in a simulated story world. There is just a single page (or screen) of vocabulary, and the grammatical rules are simple. On a different single screen, sprite graphics and animations appear to illustrate texts. Musical leitmotifs are also associated with each entity in the story world. If a user types the text “An apple dances to a fence,” for instance, a red apple will wiggle toward a white fence (both of which are the same size) and a phrase from the *William Tell Overture* will play afterwards. If the apple and fence have not been previously mentioned, they will appear as the words are typed.

At least, this is how it happens in several versions, including those for DOS[3], TI-99/4A[6], and Commodore 64[5]. In other versions, the user wouldn’t be presented with a red apple.[3] *Story Machine,* developed by DesignWare Incorporated, is typical of educational and game software for home computers in the early 1980s in that it is really several pieces of software for different platforms, obviously related but also distinct in particulars. The software was released for three platforms at the end of 1982 and for three others in 1983.[4] In fact, this diversity of *Story Machines* was not a special feature of home computer software released across platforms. Even research systems such as ELIZA were assemblages of different operating systems, libraries, code bases, and specific scripts (such as DOCTOR, in this case) at different points in time as they were being developed by their original authors [9]. ELIZA has also been re-implemented many times over the years, with an early LISP version often mistakenly thought to be original. To really discuss a software system, whether it is ELIZA or *Story Machine,* we need to be quite specific, indicating, for instance, “Weizenbaum’s ELIZA as represented in the 1966 ACM paper” or (assuming there is only one) “the Commodore 64 *Story Machine.*”



```
BOXES RUN NOW. SOMETIMES
THIS BUMPUS EATS STORES. NOW
A ROCK EATS THOSE DOGS.

THE END.
```

Figure 1: Screen capture from a Commodore 64 version of *Story Machine* running in emulation, in VICE[10], from the “WATCH A STORY” mode. The bumpus has eaten the stores, which appeared on the screen but disappeared to indicate that they have been eaten. Although the system generated “A rock eats those dogs,” the dogs blinked and remained on the screen.

[3] Although all of these computers had color graphics, the Atari 800[2] and VIC-20[4] versions of the program had black and white story illustrations, while the Apple II[1] version used orange to approximate red because of that system’s hi-res pallette.

[4] To make things more complex, the software was released on different media for some of the same platforms: a cartridge and 5.25" disk for Atari 8-bit computers, for example, and a cassette tape and cartridge for the Commodore 64. Releases on different media may have different programs on them. There are also various cracked[7] versions of the software[8] in addition to the official ones.

Some of the *Story Machine* versions even had entire modes of operation that were absent in other versions. Notably, three of them had story generators, containing a mode in which narratives are produced by randomly selecting valid words, with the sentences that resulted being animated. The earlier Apple II, Atari 8-bit, and VIC-20 programs lacked this ability.

A truly comprehensive study of *Story Machine* would detail the programs for all six platforms. While this matter of software variants and versions ranging across platforms, media, and time has been addressed by platform studies to some extent, it remains undertheorized. One way to approach it is from the academic field of bibliography[11,12]. This field of study has dealt with variant texts and different copies, printings, and editions of books for more than a century. If one is looking for a core narrative model of *Story Machine,* it is important to see where these versions overlap. Perhaps the first version is "best"? But earlier versions for more limited computer hardware may not reflect the designer's complete vision. Those studying systems today from different perspectives may have different ideas about what is best and what is core to the software.

While I plan to detail more about the six *Story Machine* programs in collaboration with others, for now, I will choose to focus on one of the more capable implementations, that for the Commodore 64, aka C64[5]. For my purposes, I have drawn upon a close examination of this version of the software — which I ran both on hardware and in emulation — along with consultation of the workings of other versions and inspection of paratexts of different sorts[13] including boxes, manuals, and reference cards.[5]

3. Claims of Educational Value

The Spinnaker manual emphasizes reading skills and learning English syntax:

> The STORY MACHINE emphasizes the rules of grammar, and encourages kids to write clear, correct sentences. It also develops their sight vocabulary — the ability to recognize words without sounding them out — as they write stories.
> STORY MACHINE combines learning with creativity, fun and excitement.

The TI manual, written by "Staff members of Texas Instruments Instructional Communications," emphasizes different benefits, suggesting that the software will lure children to become creative writers and to acquire computer skills:

> Story Machine offers your child an exciting and creative introduction to the world of writing. The activities develop essential writing skills by helping your child write sentences using modifiers, nouns, pronouns, verbs, and prepositions. Colorful pictures animate your child's sentences and entice him or her to write a story. As your child writes, he or she also learns the fundamentals of using a computer.

Did the different authors imagine different "audiences" of parents? This might be overthinking the situation. Perhaps *Story Machine* didn't originate with principled educational goals that could easily be expressed. Maybe it was just meant to be a fun narrative "toy" for young computer users. In that case, any justifications of educational benefits might have been added later.

4. Three Questions *Story Machine* Addresses

In this section I describe how *Story Machine* answers the three questions listed earlier. This program does not make a definitive, universal answer to any of them. It offers its own perspective.

[5] Specifically, here I am only considering peritexts, elements that are not part of the core software but were sold with it.

4.1. How Can a Narrative Model Be Exposed to Users?

The C64 version has five menu options: (1) Watch a story, (2) Take turns [with the computer] writing a story, (3) Write your own story, (4) Watch a story you've saved before, and (5) Make up names for characters. The first four of these present a screen with a region devoted to multimedia representation of the storyworld and area below it where the text of the story is either displayed automatically or can be typed, and a menu with numeric options at the top. Figure 1 presents an example. Option (4) requires users to have saved stories previously, which can be done within modes (2) and (3). When a user employs modes (2) and (3), minimal information about grammatical errors is provided. For instance, typing THE BOY RUN and then pressing the spacebar, instead of the "S" key, causes the system to inform the user about the disagreement in number.

The system (again, restricting the discussion to the C64 version) can therefore serve either as a story generator or a means of checking whether typed input complies with grammatical, thus narrative, rules. At a high level, the narrative model remains the same through different interfaces.

4.2. How Can a Narrative Model Conceptualize the World and Social Norms?

Any entity represented by a noun can act, whether or not it is human or animal. And, anything can be a direct or indirect object. This makes the concept of the world fundamentally *animistic.* The C64 and Atari manual's front cover features a scene which depicting many curious events that emphasize this perspective, one of which is an anthropomorphic tree eating a house. The actual sprite graphics for entities other than the boy, girl, dog, and cat also feature limbs and other anthropomorphic features.

Story Machine presents a radically flat vision of entities in the world, as would later be theorized in Object Oriented Ontology[14,15,16] and in Jane Bennett's *Vibrant Matter.*[17] It offers a strong contrast to the great chain of being, introduced as a major organizing principle of Christian thought during the middle ages. The great chain of being holds that entities with more properties are better than those with fewer: plants, which grow and develop, are better than rocks; animals, which can learn have behaviors, are better than plants; humans, who have language and foresight, are better than animals; and of course God who has all properties is better than everything. George Lakoff and Mark Turner argue that this metaphorical chain still plays a tremendous role in organizing our Western understanding of the world[18]. This way of thinking is not pervasive throughout the world or even the West, however. One can find Zen Bhuddists and adherents to indigenous beliefs who do not hold with it at all. It seems that *Story Machine* doesn't, either, as the narrative model which lacks any hierarchy of this sort and holds that everything is equal — almost.

As far as initial investigations have determined, there is a single exception to this non-hierarchical idea of the world. The actions corresponding to "eat" and "zot" (a nonsense verb) are destructive in most cases: performing these actions removes the sprite corresponding to the direct object from the screen. However, the boy, girl, cat, and dog blink and then remain on the screen if something eats or zots them, as if they had merely been nibbled upon, and the word "Ouch!" appears. The Atari and Commodore manual explains: "It is not nice to hurt other people. The STORY MACHINE won't let you act out a story that hurts any human or animal characters." Actually, human and animal characters, it seems, can be hurt ("OUCH!" appears on the screen) but don't really sustain injuries and are certainly not eradicated — an exceptional case.[6]

Story Machine is otherwise a purely syntactical system, making this one intrusion of semantics, and the way it ruptures an otherwise animistic universe, particularly interesting. However, it's almost certainly the case that this exception is due to social mores rather than philosophical ideas: an overly protective concept of what children's media should represent. Fairy tales are full of killing and dismemberment, after all, and there are plenty of extremely violent cartoons for children, which in American culture include *Looney Tunes, Tom and Jerry,* and *Ren & Stimpy.*

[6] Well, almost. The given/new list doesn't always reflect what entities have been eaten or zotted and thus removed from the storyworld. So after a rock has been eaten by some plural subject (e.g., "THE TREES EAT THE ROCK"), it can be revived with a pronoun that can't possibly refer to anything else (e.g., "IT JUMPS"). This seems to be a bug.

On the other hand, the radical flattening that pervades *Story Machine* went against some social norms of the United States from the early 1980s. An amusing one is that “The cat kisses the dog” is an accepted and potentially generated sentence, no matter that these two animals are traditionally opponents. Beyond that, homosocial (perhaps homosexual?) affection is just as permissible as heterosexual affection in the narrative models that include the verb “kiss,” as the Commodore 64 version does: “The boy kisses the boys,” “The boy kisses the girls,” “The girl kisses the boys,” and “The girl kisses the girls” are all accepted as input.

This is a popular analogue of how *Mexica* — a research system that tells stories about the pre-Hispanic inhabitants of the valley of Mexico[19,20] — allows for homosexual emotional attachments to be as likely as heterosexual ones. Homosexuality did seem to be acknowledged among Mexicas (aka Aztecs). Early texts suggest it was not suppressed (as happened after colonialism) but met with certain types of cultural disapproval.[21] There is evidence, for instance, that passive male partners practiced transvestism and been the object of some mockery.[22] In *Mexica,* however, a Jaguar Knight can fall in love with an Eagle Knight and a Princess can fall in love with a Lady as easily as any other pair of characters can, which isn’t consistent with the little known about the culture.

4.3. What Makes Narratives (and Narrative Play) Fun?

The multimedia representations enacted by the program and the illustrations on paratexts both strongly suggest that the animistic nature of *Story Machine* is a significant part of the fun. It’s not just the “colorful pictures” that the TI manual mentions — although this is certainly an important aspect — but also the wacky range of actors and actions that can entice young users to write stories. Even with a purely grammatical model, things would not be as strange and fun if *all* the available actors in the world were, for instance, humans, anthropomorphic animals, and humanoid robots. *Story Machine* would more strongly resemble other children’s literature. The model we have here is fun, in large part, because of the vocabulary, which allows for reference to rocks and flowers along with stores, houses, and fences, all of which can serve as actors as well as scenery.

Verbs, too, are more fun than they otherwise might be. Not only are there verbs of motion; “eat” allows actors to remove other entities from the story world — unless those entities are human or animal. And, “kiss” allows children to enact stories with affectionate actions. Then there’s the nonsense verb “zot,” indicating some sort of destructive action. It’s the counterpart among the verbs to the noun “bumpus,” designating a monster. These words could be justified as a way to teach children that there are open class words (in this case, verbs and nouns) and closed class words. This could hint to young children that language change most often doesn’t involve adding new prepositions, for example, but rather new verbs (e.g., “doxx” and “blog”) and nouns (e.g., “hater” and “blog,” which is also a noun). However, that’s not what the TI manual says about them. It says they “add to your child’s enjoyment.”

5. A “Miniature” and a Tiny Creative Work Inspired by *Story Machine*

5.1. A Nub with Story-Generating Essentials

As an early part of a planned research project, Narrative Nubs,[7] I developed what Roger Schank would have called a “miniature,”[23] a simplified version of *Story Machine* that encompasses the core functionality of its story generator. For my purposes, I was only interested in how *Story Machine* could put sentences together as a textual story generator. I was also interested in writing

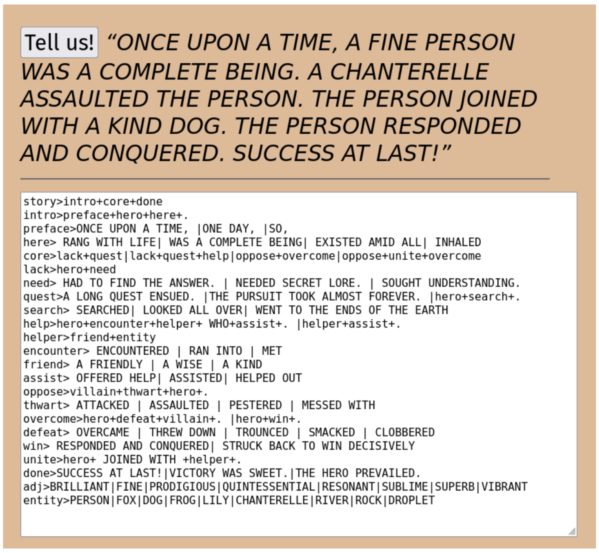
[7] Narrative Nubs will be a substantial set of simplified implementations of story generators, almost all of them research systems originating in the 20th Century. The different nubs will express the core concepts of these systems, will be coded in a consistent way, and will share libraries when appropriate, facilitating teaching, research, and even use by writers and artists. Because *Story Generator* is extremely simple, it will not use any libraries; other nubs will use ones for Conceptual Dependency (CD), Hierarchical Task Network (HTN) planning, and elaborate story grammars. Despite the importance of *Story Machine*’s multimedia aspects in the early 1980s, the essence of this narrative model can be expressed textually within this project, because all the other story generators that we will use as points of comparison will be text only.

code that would be concise, clear, and in a programming language that was widely used in teaching and research — I chose Python.

A preliminary version of this nub, not yet released, is less than than 150 lines of code and incorporates fairly complex aspects of sentence and story generation. For instance, there is a list of givens, entities referred to in the discourse who still exist in the story world. If the most recently referenced entity of a certain gender or number ends up being referred to again, the system uses a pronoun. The nub also implements the basic aspects of *Story Machine*'s grammar: verbs of motion and transitive verbs, for instance. The nub also has "hurtful" verbs ("eat" and "zot") remove non-human, non-animal entities from the story world and givens list but leave the humans and animals intact.

While there are further plans for this nub and the Narrative Nubs project overall, the process of creating a nub does, by itself, have benefits in terms of better understanding the system that is being re-implemented. In order to abstract the model in a clean and consistent way, decisions need to be made about what aspects of the narrative model are essential.

5.2. "Gram's Fairy Tales," Inspired by *Story Machine*



"ONCE UPON A TIME, A FINE PERSON WAS A COMPLETE BEING. A CHANTERELLE ASSAULTED THE PERSON. THE PERSON JOINED WITH A KIND DOG. THE PERSON RESPONDED AND CONQUERED. SUCCESS AT LAST!"

```
story>intro+core+done
intro>preface+hero+here+.
preface>ONCE UPON A TIME, |ONE DAY, |SO,
here> RANG WITH LIFE| WAS A COMPLETE BEING| EXISTED AMID ALL| INHALED
core>lack+quest|lack+quest+help|oppose+overcome|oppose+unite+overcome
lack>hero+need
need> HAD TO FIND THE ANSWER. | NEEDED SECRET LORE. | SOUGHT UNDERSTANDING.
quest>A LONG QUEST ENSUED. |THE PURSUIT TOOK ALMOST FOREVER. |hero+search+.
search> SEARCHED| LOOKED ALL OVER| WENT TO THE ENDS OF THE EARTH
help>hero+encounter+helper+ WHO+assist+. |helper+assist+.
helper>friend+entity
encounter> ENCOUNTERED | RAN INTO | MET
friend> A FRIENDLY | A WISE | A KIND
assist> OFFERED HELP| ASSISTED| HELPED OUT
oppose>villain+thwart+hero+.
thwart> ATTACKED | ASSAULTED | PESTERED | MESSED WITH
overcome>hero+defeat+villain+. |hero+win+.
defeat> OVERCAME | THREW DOWN | TROUNCED | SMACKED | CLOBBERED
win> RESPONDED AND CONQUERED| STRUCK BACK TO WIN DECISIVELY
unite>hero+ JOINED WITH +helper+.
done>SUCCESS AT LAST!|VICTORY WAS SWEET.|THE HERO PREVAILED.
adj>BRILLIANT|FINE|PRODIGIOUS|QUINTESSENTIAL|RESONANT|SUBLIME|SUPERB|VIBRANT
entity>PERSON|FOX|DOG|FROG|LILY|CHANTERELLE|RIVER|ROCK|DROPLET
```

Figure 2: "Gram's Fairy Tales" with a generated story at the top and the default grammar in a text area below it. The grammar can be edited directly in the browser.

A compact creative work of mine, "Gram's Fairy Tales,"[24] was inspired by the flat ontology of *Story Machine* and the extremely simple story grammar implemented by Joseph Grimes in the early 1960s.[25,26] Grimes's system suggested that there should be some very basic types of stories produced, with heroes and sometimes villains and/or helpers. *Story Machine* was what inspired me to allow any entity (including, for instance, rocks and rivers) to serve as hero, villain, or helper.

"Gram's Fairy Tales" is a very tiny HTML5 system (only 2KB) that allows users to generate stories and to also define their own grammars of story. It was published as part of *Taper #11: Tools.* Grammars by Jhave, Kyle Booten, and Kavi Duvvoori were released when this literary journal issue came out[27]. All of the works published in *Taper* adhere to a 2KB size restriction.[28] Although other grammar writers can develop their own generators using this system, or platform, or tool, this tiny creative work seems to enforce its own animistic worldview. Only if others continue to use it will we gather more evidence on this point.

For now, however, there are two insights about *Story Machine* that arise from the "Gram's Fairy Tales" project. First, the simplicity of *Story Machine,* which maps grammar and vocabulary directly to narrative possibilities, can be combined with another simple model of narrative, one that allows for a beginning, middle, and end where a hero sets out on a quest, overcomes adversity, and triumphs. Second, “Gram’s Fairy Tales” makes it clear that *Story Machine* has a philosophy or perhaps even ideology that is enforced by its narrative model. It may be a refreshing one — as I think it is — but it is a model that enforces a flat, non-hierarchical world just as other models enforce hierarchy.

6. Future Directions

There are many other ways to research *Story Machine*, extending past the details of software differences across different platforms. These include researching in print-based and online libraries and archives, consulting the writings of and online interview[29] with Jim Schuyler (founder of Design Ware and designer and original programmer of *Story Machine*), conducting new interviews with Schuyler, seeking the source code, and reverse-engineering the programs to ensure that all aspects of their workings have been accounted for.

It’s possible to consider that the constellation of software known as *Story Machine* functions in a unified way to project a particular concept of narrative, despite the differences between versions. A challenge for developing a version 1.0 nub of *Story Machine* will be suitably representing this single, high-level concept while remaining sensitive to the differences that platform-specific versions of the software introduced. Within the larger Narrative Nubs project, the *Story Machine* nub will have to work alongside the others to appropriately show one of the many historical approaches to the computational modeling of narrative.

Acknowledgments

Then-undergraduate research assistants Daniel Villagran and Eva Goldie discussed *Story Machine* with me in the MIT Trope Tank. While they aren’t responsible for any errors in this paper or for the arguments I make here, they thought about this software extensively, even developing their own Python nubs of *Story Machine* independently of mine so that we could compare features. Their work was made possible by the UROP program at MIT. Madison Landry did additional late-breaking research on *Story* Machine which informed my revision of this paper. She and Villagran, along with the anonymous reviewers, offered comments that helped me revise. I appreciate others who conversed about and interacted with *Story Machine*, including colleagues at the Center for Digital Narrative. This research was partially funded by the Research Council of Norway Centres of Excellence project number 332643, the Center for Digital Narrative.

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] Design Ware, Story machine, Spinnaker Software, Educational software, Apple II. 1982.

[2] Design Ware, Story machine, Spinnaker Software, Educational software, Atari 800. 1982.

[3] Design Ware, Story machine, Spinnaker Software, Educational software, DOS. 1982.

[4] Design Ware, Story machine, Spinnaker Software, Educational software, Commodore VIC-20. 1983.

[5] Design Ware, Story machine, Spinnaker Software, Educational software, Commodore 64. 1983.

[6] Design Ware, Story machine, Spinnaker Software and Texas Instruments, Educational software, TI 99/4A. 1983.

[7] G. J. Albert, New Scenes, new markets: the global expansion of the cracking scene, late 1980s to early 1990s, Wider Screen 23:2/3 (2020). https://www.zora.uzh.ch/id/eprint/203800/1/albert--new_scenes_new_markets.pdf.

[8] Design Ware, Story machine, Spinnaker Software, Cracked by 4am, rev. 1 of crack. Educational software for Apple II. 1982. https://archive.org/details/StoryMachine4amCrack.

[9] R. Lane, A. Hay, A. Schwarz, D. M. Berry & J. Shrager. ELIZA Reanimated: The world's first chatbot restored on the world's first time sharing system. (2025) *arXiv:2501.06707v1.*

[10] VICE (The Versatile Commodore Emulator) v3.6.1. Ubuntu version.

[11] R. B. McKerrow, An introduction to bibliography for literary students, Oxford at the Clarendon Press, 1928.

[12] C. Fernandez-Vara, N. Montfort, Videogame editions for play and study, MIT Trope Tank Technical Report TROPE-13-02. https://tropetank.com/reports/TROPE-13-02.pdf.

[13] G. Genette, Paratexts: Threshholds of interpretation, trans. J. E. Lewin, Cambridge Univ. Press, 1997.

[14] G. Harman, Object-oriented ontology: A new theory of everything. Penguin UK, London, 2018.

[15] T. Morton, Here comes everything: The promise of object-oriented ontology. Qui Parle: Critical Humanities and Social Sciences 19.2 (2011) 163-190. doi.org/10.5250/quiparle.19.2.0163 .

[16] A. Cole, The call of things: A critique of object-oriented ontologies. Minnesota Review 2013.80 (2013): 106-118. doi.org/10.1215/00265667-2018414.

[17] J. Bennett, Vibrant matter: A political ecology of things. Duke University Press, Durham, North Carolina, 2020.

[18] G. Lakoff, M. Turner, More than cool reason: A field guild to poetic metaphor. Univ. of Chicago Press, 2009.

[19] R. Pérez y Pérez, M. Sharples, MEXICA: A computer model of a cognitive account of creative writing. Journal of Experimental & Theoretical Artificial Intelligence 13.2 (2001): 119-139. doi.org/10.1080/09528130010029820.

[20] N. Montfort, R. Pérez y Pérez, Computational models for understanding narrative, Revista de Comunicação e Linguagens 53 (2023) 97-117. https://revistas.fcsh.unl.pt/rcl/issue/view/74/45.

[21] G. Kimball, Aztec homosexuality, Journal of Homosexuality 26:1 (1993) 7-24. doi.org/10.1300/J082v26n01_02

[22] P. Sigal, Queer Nahuatl: Sahagún's faggots and sodomites, lesbians and hermaphrodites, Ethnohistory 54.1 (2007): 9-34.

[23] R. C. Schank, C. K. Riesbeck (Eds.), Inside computer understanding: Five programs plus miniatures, Psychology Press, New York, 1981.

[24] N. Montfort, Gram's fairy tales, Taper 11 (Spring 2024), https://taper.badquar.to/12/grams_fairy_tales.html.

[25] Exploring the fascinating world of language. Business Machines 46 (May 1963) 11–12.

[26] J. Ryan, Grimes' fairy tales: A 1960s story generator, Interactive Storytelling: 10th International Conference on Interactive Digital Storytelling, ICIDS 2017 Funchal, Madeira, Portugal, November 14–17, 2017, Proceedings, 89-103, doi.org/10.1007/978-3-319-71027-3_8.

[27] N. Montfort, Gram's Fairy Tales: Manual & Grimoire, Post Position (Montfort's blog), 2024-05-01, https://nickm.com/post/2024/05/grams-fairy-tales-manual-grimoire/.

[28] Taper. https://taper.badquar.to.

[29] K. Savetz, Jim Schuyler, founder of DesignWare — interview, Video, 2017-10-15, https://archive.org/details/jim-schuyler-founder-of-designware.