The Authorial Burden

Joey Jones

University of Southampton, Southampton
j.d.jones@soton.ac.uk

Abstract. Abstract: There are limits that emerge out of the interactive nature of interactive digital narrative that make authoring it challenging. These limits include exponential branching, where branches in the narrative increase the amount of content needed to be written progressively throughout the work; combinatorial explosion, where increasing combinations of possible game states makes writing additional content complex; as well as programming scope problems that are seen in any digital project, wherein the range of features or game interactions that could be implemented is infinite but development time finite. These limits place on the authors of interactive digital narrative an authorial burden, increasing the amount of content needed to be written, states managed or features programmed. There are multiple strategies for tackling the burden, from reducing or reusing content, to decontextualising and generating content.

1 Introduction

Why is writing interactive digital narrative challenging? Any creative endeavour has its difficulties, many of which arise out of their respective mediums. In oil painting, the painter must mix the paints and this is a different challenge to the requirement of film-makers to control the lighting in their shots. Interactive narrative similarly has its own creative challenges. Allowing interaction — especially choices which lead to alternative content — can require a a great deal of additional content to be created. This content can be increasingly complex to author. This has been referred to as the authorial burden.[1] This isn't meant to be pejorative: the act of writing doesn't need to feel burdensome. Rather, the authorial burden refers to the workload cost of authoring incurred by making interactive design decisions.

Content can be slow to write because there is a great deal of it to produce, or it can be slow to write because each new bit of content is complex to author: Garbe refers to these limits as the "authoring wall" and the "complexity ceiling".[18] Authoring interactive digital narrative is not just writing, but rather a range of complimentary activities including planning, visualisation, structuring, testing and so forth. [24] If content is difficult to visualise or test, then it will be slower to write and have a greater concomitant burden.

[1] The first appearance of this term appears to be by Mateas and Sterne in 2002 when discussing the authoring of their interactive play Façade.[27] though the problem clearly predates the term.

1.1 Authoring Wall

If you have a branching story with two equal length paths, then you'll need to write twice as much; if you have a story with content displayed differently based on combinations of different game states, then adding more states will exponentially increase the content required; if you give the player a great deal of freedom of action, i.e. if the scope if particularly large, then a greater amount of content will need to be written.

The simplest structure of interactive narrative is the branching story, in which choices split the plot and the separate paths do not merge. This tree structure is sometimes known as the broomstick (especially with regards to endings)[7] or the Time Cave (after the Choose-Your-Own-Adventure novel of the same name that had this structure)[1]. Time Caves most quickly run into the problem of exponential branching. Without merging branches, any kind of branching work faces the limit of exponential writing. To date, the longest complete branching work which has no merging is Girth Loinhammer's Exponential Adventure [40]. This is only ten passages long, with each complete playthrough having 9 choices with exactly two options each. This leads to 512 separate endings. If the author, Damon Wakes, were to write one extra choice for every possible playthrough, he would have to write 1024 additional passages. Another choice on top of that would be 2048 additional passages. With this simple structure, the authorial burden doubles every time the average game length is increased by one choice. The limit on how long a narrative of this type can be is quickly reached.

1.2 Complexity Ceiling

The authorial burden can be large because of the amount of content that needs to be written, but it also can be large because the content that needs to be authored is difficult to write due to complexity. These are two sides of the same coin, with any reduction in complexity allowing more content to be written. Classic authoring problems from literature, like avoiding writing oneself into a corner, or maintaining continuity in a long work are compounded in an interactive piece if there are multiple possible plot-states true for any given scene, or if certain choices must be disallowed to ensure continuity later on in the story. For instance, a character dying can incur an authoring burden for future scenes in which that character could appear, requiring extensive additional writing, but the act of making sure that any given scene isn't incurring an undue burden is itself a complicating factor that can slow down authorship.

The simple branching structure has only one game state at any given time. A common way to manage choice in interactive fiction is through tracking multiple states. This allows any choice to have an effect on branching the story without requiring that every choice immediately creates a hard fork. However, works that rely on combinations of states run into a different problem of unmanageable combination sets. When each new state added can be combined with a number of existing states and new content needs to be written for this combination, the author eventually runs into a limit of complexity that they can handle.[19]

Related ideas to the complexity ceiling can be seen in software creation more broadly. **Software entropy** is the idea that as software grows it is more likely to be modified, as it becomes modified it becomes more complex.[4] Design choices for any given iteration incurs a **technical debt** for future iterations, increasing the amount of future programming labour required.[2] Programmatic approaches such as dynamic programming and machine learning face the **curse of dimensionality**: the rapid expansion of the difficulty of programming problems as variables or dimensions increase.[13] The strategies software engineers use such as refactoring, modularising, or loose coupling[17] can overlap with the techniques of narrative designers.

1.3 Forms

Interactive narrative can take many different forms which have different authorial burdens. Long-form branching narrative, whether text-based, filmed or animated, where the player selects from choices, will be authored differently to classic text adventure with a parser where players type commands. These are different to agent-based simulationist works where a narrative emerges from interaction with virtual actors; or database fiction where a player explores some body of content in order to make their own connections; or games like walking sims where the player explores an 3D environment. Some narratives require tactical decision making from the player or puzzle solving, others have purely kinetic interaction. Nevertheless, the underlying interactive nature of these various forms generates a need for authors to write additional content which, if not properly managed, can expand exponentially. As such, many of the broad approaches for managing the authorial burden apply to multiple different forms of interactive narrative, and many of these different forms can themselves be understood not just as aesthetic decisions but also as means by which the constraints associated with cumulative branching can be overcome.

1.4 Strategies for Unburdening

Strategies can be deployed to reduce or re-use content, decouple or generate segments of content, or otherwise improve the process of authorship. Many of these approaches will be given more extensive treatment elsewhere in this volume, so this will be something of a whistle-stop tour.

Reducing— Authors can seek to cut the authorial burden by reducing the amount of possible content such as by limiting scope or merging branches.

Reusing— Existing content, such as backgrounds, animations, even scenes, can be repurposed. This can occur either by repeating the same representation (e.g. having the same scene occur in two branches) or using the same assets, text snippets, animations etc. to represent different things (e.g. using the same library of responses for different non-player characters).

Decoupling— New content can be harder to write the more it needs to be continuous with existing content. Making sections more context-independent is a way to reduce the writing complexity, allowing more to be written.

Generating— Content can be procedurally generated allowing a great deal of possible novel combinations.

Embracing— Rather than changing the form of an interactive narrative itself, there are strategies which seek to improve the ability of authors to write them. These approaches embrace the authorial burden.

These broad meta-strategies can be seen instantiated in the various approaches to tackling the authorial burden. It should be noted that there are many other reasons for making design choices beyond their effect on the total workload. Managing authorial labour is only one of several reasons an author might use a specific structure of interaction or set of tools. Different plot types, such as the epic plot-based or the dramatic character-based form may fit better or worse with different modes of interaction. [35] Narrative structures expressed through choices offered to the player, or actions possible, are also poetic gameplay devices for evoking specific emotions in the player.[9] Rather that, given the untenability of compounded branching, any story with alternative content will necessarily be authored with some strategy for managing the authorial burden. The strategies aren't mutually exclusive, and the categories can overlap. Reusing content is at the same time reducing possible content and so forth.

2 Reducing

The most straightforward way that a digital project can involve a great deal of writing is by having a large scope, indeed, one way of conceiving of the authorial burden is as under-managed scope. All completed projects necessarily limit scope to some extent: decisions always have to be made about what not to include. Scope limitation comes in different forms. In interactive narrative, scope is limited along at least these two axes: objects and verbs.

2.1 Limiting Objects

Implementing a large range of objects in a digital work (locations, characters, items, musical scores etc.) incurs a burden of content creation for all things implemented, and a typical way of reducing scope is to simply cut planned or possible content. With works especially in multimedia (animation, acting, illustration etc.), keeping the range of assets used manageable can be important for keeping costs down and not just for time reasons. This has been called the "economics of building".[31] Diagetic constraints are often employed, grounding the limitation of elements in the fiction: this is why there is a plethora of closed houses, isolated islands and small casts in interactive narrative.

2.2 Limiting Verbs

In an interactive work, there can be a range of actions that a player is able to undertake. This can range from choosing between links to click, all the way up to simulating a virtual environment. The range of verbs allowed then might range from one, to many hundred (in the case of some parser=based games). For these games where the player might type anything, some authors have chosen to supply a very curtailed list of accepted words, but then implement bespoke responses for every valid combination. CEJ Pacian employs this approach in several his works and calls the approach “shallow but broad”[30]: in Superliminal Vagrant Twin, the player can travel to a large range of planets, allowing a great sense of openness, but this is acheived by tightly limiting the possibility space on each planet to less than a dozen options in any given place.

2.3 Primacy of Text or Dialogue

Assets range considerably in cost and development time. Next to animation, live filming, illustration, and so forth, text is comparatively cheap. Works can contain a great deal of text in descriptions, dialogue and so forth in a highly scalable way, in that adding more lines of text does not typically require new capabilities. Interactive fiction as a genre is almost synonymous with text based games.

Similarly, lines of dialogue don’t necessarily require unique animations or assets (though in some productions they may be voiced), and so by centring the exchange of words as the primary experience, the authors save on having to produce and program animation that would be required in a more physically expressed experience. This is a common strategy in computer roleplaying games which can have tens of thousands of lines of dialogue. [38]

2.4 Existing Tools

Creating bespoke engines is programming labour which can have a huge time cost, and so using existing narrative engines, programming languages, visualising tools etc. can save a lot of possible work, with the compromise that what can be made will be limited to what is possible within those tools. For example, the walking sim Dear Esther was originally written as a mod for the Half-Life 2 Source Engine. This allowed the game to be made using a ready-to-use existing tool that the developers were familiar with, although until its remake, this limited what could be achieved both in the structure of the island and in the visual art style.

2.5 Virtuality

Abstracting away realist elements is a common scope-management technique. Players of games in general, including narrative games, will accept a great deal of artificiality which would break the suspension of disbelief in film or literature; the player can cultivate a playing stance which separates the abstractions and gamelike elements from their understanding of the work’s narrative.[23] For example, choice-based conversations will often happen outside of real-time without breaking immersion.

2.6 Gauntlet

A classic solution to writing reasonably lengthy stories with choices but without compounded branching is the gauntlet structure.[1] In the gauntlet there is one core path with many short endings off of this path: in gamebooks this would often be represented by premature deaths. This structure only suits certain kinds of stories, though it still be can be seen in contemporary works such as in large parts of the interactive film Bandersnatch.[6] A variant of the gauntlet is the 'friendly gauntlet' in which there are no dead-ends: all side branches fold back onto the main path.

3 Reusing

It is often economical to reuse the same element (a scene, a piece of art, a sign-off in a conversation etc.). This reaches its limit if it becomes prohibitively difficult to write content using existing elements or to cohere with pre-decided plot points.

3.1 Merging Branches

Branching and then merging is the most fundamental strategy for managing interactive narrative. Across a whole work, it's been known as "branch and bottleneck" structure, for the way paths can branch out in a story and then return at bottlenecks.[1] Hargood and Crawford have separately referred to this as a foldback [21][12], in which the branches of the story bend back on themselves. Most other hypertext patterns are versions of this at different scales.

There are different extents to which you can merge, forming a spectrum of interactivity with various subsequent authorial burdens. This can be conceived of as a way to reduce possible content by sending the player back to a central trunk of content.

Empty Choice The purest merge is to offer a choice and redirect the player to the same subsequent content regardless of their pick, with no states tracked. Here the choice might be functioning as a pacing device or to offer the player a different sense of the story or protagonist. This has little concomitant burden beyond the labour involved in conceiving of the non-choices. Fendt et al refer to this technique as Illusory Agency.[15] Such a strategy may genuinely save labour, but players (especially on replay) will likely see through the device, especially if consistently or overwhelmingly employed.

Recognised Choice Commonly there will be some small divergence of outcome, some differing text or dialogue, say. This creates some writing burden, but the burden isn't compounded as the story continues the same regardless of what is chosen.

State changes Alternatively, a choice can have different outcomes that are tracked by the narrative. A relationship might change, or an item may be gained, etc. These states can then be used later in the narrative without the requirement to branch immediately. Choice of Games, a publisher of interactive novels, refers to this as 'delayed branching'.[14]

Parallel branches A more authorially taxing form of splitting and merging branches is having substantial branches run in parallel, which then might merge at choke-points within the story. Ashwell refers to this structure as the Quest,[1] and it can be seen in works like 80 Days where multiple different parallel paths are possible for traversing the narrative at any given time.

3.2 Loops

Loops are a way of re-using the same content or set of choice offered. The player is returned to the same scene or location after making a choice and is presented with either the same or an overlapping set of choices. By re-using the choice-point but allowing the player to see different branches on subsequent returns, this structure more efficiently uses content. Large narrative games like Fallen London make extensive use of this structure, with most content capable of being re-experienced. Time-travel stories such as Elsinore are based around this core idea, allowing larges swathes of the game to be re-experienced, often with a separate player-agenda on replays, allowing for continual agency despite repeated content.

3.3 Hub Nodes

The hub node, is one use of a loop. It is a node of an interactive story which can be returned to repeatedly in a cycle pattern until some condition is met to move forward [5]. At a hub, a list of choices are offered. After exploring one of these choices (which may have its own sub-choices), the player returns to the hub. Each time they return, used choices may be removed and new choices may be added. This is often the way conversations systems are implemented. Clusters of hub nodes can be used to implement a location structure in a choice-based work that is more commonly seen in parser fiction (see for instance, 16 Ways To Kill A Vampire At McDonalds [11], or With Those We Love Alive [8]). The hub node is an effective way of structuring narrative segments in which there are multiple smaller scenes that could coherently be experienced in any order.

3.4 World Modelling

Many of the strategies discussed are based on a choice branching structure. Moving away from this going further than the loop is implementing a world model of persistent locations, objects, and characters. Text adventure games are defined by their use of the world model. As models become more richly implemented, they create their own authorial burden (for instance, in implementing different verbs and accounting for various combinations of objects). Still, modelling a persistent set of locations and objects is very common in fully illustrated works, as creating artwork for a location is costly, so it often makes sense to re-use places as much as possible (this is a common feature of narrative adventure games, such as Monkey Island, where the same locations and characters are returned to repeatedly).

3.5 Cumulative Variables

A straightforward way of lowering complexity in tracking narrative states is to use cumulative variables instead of, say, multiple Boolean values. For example, the interactive space opera, Mass Effect, tracks how 'Renegade' the protagonist is. Instead of checking a list of every possible time they acted in a renegade way, the player's Renegade score accumulates at such occasions, and when relevant the single variable is checked. This principle is used extensively in the works of Choice of Games.[22]

4 Decoupling

Decoupling is a way of reusing content, by allowing scenes to appear regardless of previous content. The more self-contained segments of content are, the easier new sections can be written without having to write lots of variations depending on the world-state.

4.1 Storylets

The storylet structure is a clear example of decoupling. The term was coined by the writers of the massive multi-million word browser-based text game, Fallen London to refer to the chunks of content that can be experienced in many different possible orders. When these storylets are displayed based on conditional triggers, this is referred to as Quality-Based Narrative.[36] Storylets in Fallen London work on a principle that has been referred to as the ‘fires in the desert’ approach: the writers create self-contained chunks of story (the well-lit ‘fires’), leaving it up to the reader to infer the linkage between these chunks (the dark ‘desert’ between these bright spots of story).[16] This strong context independence asks more from the reader than in traditional storytelling where events have much clearer causal links.

4.2 Modularisation

Modularising interactive narrative is to separate it into relatively self-contained segments. This is the same concept as 'levels' in videogames more broadly. To take an example: most long-form interactive novels published by Choice of

Games[2] adhere somewhat to their stylebook, a set of guidelines for content and structure. The content guidelines ensure that content is in line with the values of the publishers (inclusive choices, no hate speech etc.), the structural guidelines are a set of "best practices for game management". One of the guidelines is that works should be comprised of approximately ten vignettes that occur in sequential order. This ensures that the games are of sufficient length to meet their expected standards, but it also manages branching. It ensures that on a macro-scale across the whole interactive novel, no matter how many branches there are internally in a chapter, and no matter how any of them may end, they each must lead on to the next chapter in sequence. This modular design prevents the stories from sprawling out to an unmanageable number of parallel branches. Essentially this approach is a more formal application of the "branch and bottleneck" structure, but in other works, modules might be even more independent.

5 Generating

Generating content combines the approaches of reusing and modularising content, most often allowing the same elements to be recombined in a wide variety of contexts. Combining elements can happen at various different scales. It can happen at the level of the individual collection of pixels (as in procedural animation) but also at the level of the word and letter. Tools like Tracery allow word lists to be recombined to create a huge variety of grammatical sentences according to some simple iterative rules.[10]

5.1 Procedural Generation

Procedural Generation is a strategy for creating content where there are many possible states instead of hand authoring unique state combinations. Sufficient content is required to populate generative lists and a testing is needed to make sure the output is of sufficient quality. As such, the narrative designer Cat Manning once quipped that procedural generation means "generating twice the content in twice the time".[3]

5.2 Supplementary Generation

Reed has presented a system for reducing the authorial burden of writing by procedurally generating satellite sentences which are there for pacing and establishing context.[32] Ryan has demonstrated recombinant conversation generation for creating filler conversations[34]. Neither of these approaches necessarily produce content as satisfactory as bespoke-authored content, but they could be used to supplement such content (for example, by giving variations of greetings and goodbyes and other formulaic conversation elements).

[2] Such as Choice of Robots, Crème de la Crème, Trials of the Thief-Taker and so on.

[3] See https://twitter.com/catacalypto/status/1470893540964134913

5.3 Natural Language Processing

Natural language processing is a set of tools for parsing user input. Rather than create bespoke responses to every possible useful input (as would be common in normal parser fiction), this approach seeks to dynamically 'understand' user input to decide what content should be shown next.

This technique is demonstrated by the experimental interactive play Façade which has a parser which takes the player's written input and interprets it into various viable responses.[26] The workable responses depend on the current story beat being played through, so the same player input can be interpreted in different ways throughout the story (e.g. if the player continually says the word 'no' what they are saying it in response to and how the other characters might respond will be will vary considerably throughout the narrative). For any given beat, there are about eight different interpretations the parser will channel the player's response into. These can further the current story beat or trigger a new beat. In theory, this approach could negate much of the need for hand-authoring parser interpretations, but in practise the authors ended up creating many ad hoc phrases for the parser to understand for specific beats.

5.4 Simulated Agents

While not necessarily a way of reducing labor, creating simulated agents pushes the authorship into a different domain, that of writing patterns of behaviour[3]. With the right tools, this approach can allow the creation of a great deal of novel situations that emerge out of the simulation.

Works like Prom Week involve simulating characters, with a narrative coming through the player's interaction with these characters and their simulated behaviour amongst themselves [28]. Martens and Iqbal have made Villanelle, a story-engine for the creation of these kinds of narratives [25]. Among commercial works, agents can be seen in highly procedural games like Dwarf Fortress, and more tightly authored experiences like Elsinore.

6 Embracing

Rather than seek to design it away, Stern has argued that we ought embrace the combinatorial explosion.[39] The authoring wall is only high in comparison to the capacity of the authors of an interactive narrative. Embracing the burden is the final strategy.

6.1 More Hours

One common, if controversial, method of overcoming the authorial burden is alloting more time to it. This can either mean taking longer to create the work, or it can mean crunch: packing more working hours into a short period of time, typically before product launch.

6.2 More Writers

One paradigm of interactive narrative is the solo-authored piece, product of a single vision. But the other paradigm is that which is common in videogame companies, to have several writers, often in large teams. While there are diminishing marginal returns with such an approach, if you have more writers, under the right conditions more content can be written.

6.3 Developing Craft Knowledge

Access and education for authors has been highlighted as an important area for developing interactive narrative; [37] being able to think and write interactively is a skill that writers of other mediums don't automatically possess. The more acquainted authors are with common interactive patterns, and familiar they are with their tools, the more ambitious projects they will be able to complete. It is perhaps no surprise that the most popular narrative engines, such as Inform7, Twine and Ink all have extensive documentation.

6.4 Developing better tools

It has been argued that improving the "user experience" of authoring tools can help authors attain greater competency with tools and unlock deeper affordances,[33] as well minimising interruptions to authoring flow[20]. This approach is taken up elsewhere in this volume.

Tool creation itself can be unproductive if the tools are never utilised or recreate the affordances of existing engines. However, for some story-structures, the right kinds of visualisation and testing tools can have a huge effect. Emily Short described such an "author-friendly toolset for writing Versu stories... that sped up content production by a factor of at least ten and meant that we could produce much bigger, longer stories than previously released."[29]

7 Conclusion

The most basic form of interactive narrative, the branching story, creates on of the largest burdens, sharply curtailing the possible length of such a story. As greater levels of state tracking and content structuring is introduced, the complexity of writing new content increases. Ways of ameliorating this twin burden of volume of content and difficult of production can in themselves create new challenges, requiring different skill-sets from authors. Strategies don't necessarily alleviate work, so much as change the form it takes.

When faced with an ambitious idea, the author of an interactive work has many paths they can take to realising it. They are likely to pare back the scope to something that feels manageable; they may employ structures like merging and loops to keep branching under control; they could make use of more advanced forms, using salience to decide what to display, or procedurally generating a large number of variations. They might take their time, improve their skills, or work with others to see the idea through.

The authorial burden then isn't a hard limit on what can be achieved, but a malleable border that shifts. It's the point at which an author is willing to compromise between their vision and what tools, time and their own powers allow them to achieve.

References

1. Ashwell, S.K.: Standard Patterns in Choice-Based Games (Jan 2015), https://heterogenoustasks.wordpress.com/2015/01/26/standard-patterns-in-choice-based-games/
2. Avgeriou, P., Kruchten, P.: Managing Technical Debt in Software Engineering. Dagstuhl Reports (2016). https://doi.org/10.4230/DAGREP.6.4.110
3. Badler, N.I., Reich, B.D., Webber, B.L.: Towards personalities for animated agents with reactive and planning behaviors. In: Trappl, R., Petta, P. (eds.) Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents, pp. 43–57. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (1997). https://doi.org/10.1007/BFb0030569, https://doi.org/10.1007/BFb0030569
4. Bakota, T., Hegedűs, P., Ladányi, G., Körtvélyesi, P., Ferenc, R., Gyimóthy, T.: A cost model based on software maintainability. In: 2012 28th IEEE (ICSM). pp. 316–325 (Sep 2012). https://doi.org/10.1109/ICSM.2012.6405288, iSSN: 1063-6773
5. Bernstein, M.: Patterns of hypertext. In: 9th ACM Hypertext and Hypermedia. pp. 21–29. HYPERTEXT '98, Association for Computing Machinery, New York, NY, USA (May 1998). https://doi.org/10.1145/276627.276630
6. Brooker, C., Slade, D.: Bandersnatch (2018), https://ifdb.org/viewgame?id=dh7kphu7nmjt1ia5
7. Bruckman, A.: The Combinatorics of Storytelling: "Mystery Train Interactive". The MIT Laboratory (Apr 1990)
8. Charity Heartscape, P., Neotenomie, B.: With Those We Love Alive (2014), https://ifdb.org/viewgame?id=445d989vuwlh4cvz
9. Chew, E.C., Mitchell, A.: Bringing Art to Life: Examining Poetic Gameplay Devices in Interactive Life Stories. Games and Culture p. 1555412019853372 (Jun 2019). https://doi.org/10.1177/1555412019853372, https://doi.org/10.1177/1555412019853372, publisher: SAGE Publications
10. Compton, K., Kybartas, B., Mateas, M.: Tracery: An Author-Focused Generative Text Tool. In: Schoenau-Fog, H., Bruni, L.E., Louchart, S., Baceviciute, S. (eds.) Interactive Storytelling. pp. 154–161. Lecture Notes in Computer Science, Springer International Publishing, Cham (2015)
11. Corfman, A.: 16 Ways to Kill a Vampire at McDonalds (2016), https://ifdb.org/viewgame?id=s8oklhvdqoo5dv4l
12. Crawford, C.: Flawed Methods for Interactive Storytelling | Interactive Storytelling Tools for Writers (Sep 2020), http://www.erasmatazz.com/library/the-journal-of-computer/jcgd-volume-7/flawed-methods-for-interact.html
13. Dobrovsky, A., Borghoff, U.M., Hofmann, M.: Applying and Augmenting Deep Reinforcement Learning in Serious Games through Interaction. Periodica Polytechnica Electrical Engineering and Computer Science **61**(2), 198–208 (May 2017). https://doi.org/10.3311/PPee.10313, https://pp.bme.hu/eecs/article/view/10313, number: 2

14. Fabulich, D.: By the Numbers: How to Write a Long Interactive Novel That Doesn't Suck (Jul 2011), https://www.choiceofgames.com/2011/07/by-the-numbers-how-to-write-a-long-interactive-novel-that-doesnt-suck/
15. Fendt, M.W., Harrison, B., Ware, S.G., Cardona-Rivera, R.E., Roberts, D.L.: Achieving the Illusion of Agency. In: Oyarzun, D., Peinado, F., Young, R.M., Elizalde, A., Méndez, G. (eds.) Interactive Storytelling. pp. 114–125. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2012)
16. Games, F.: Echo Bazaar Narrative Structures, part three (Mar 2010), https://www.failbettergames.com/echo-bazaar-narrative-structures-part-three/
17. Gamma, E., Helm, R., Johnson, R., Vlissides, J., Booch, G.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, Reading, Mass, 1st edition edn. (Nov 1994)
18. Garbe, J.: Increasing Authorial Leverage in Generative Narrative Systems. Ph.D. thesis, UC Santa Cruz (2020), https://escholarship.org/uc/item/4dq8w2g9
19. Garbe, J., Kreminski, M., Samuel, B., Wardrip-Fruin, N., Mateas, M.: StoryAssembler: an engine for generating dynamic choice-driven narratives. In: 14th International Conference on the Foundations of Digital Games. pp. 1–10. FDG '19, Association for Computing Machinery, New York, NY, USA (Aug 2019). https://doi.org/10.1145/3337722.3337732
20. Green, D., Hargood, C., Charles, F.: Contemporary Issues in Interactive Storytelling Authoring Systems. In: Interactive Storytelling. pp. 501–513. Lecture Notes in Computer Science, Springer International Publishing, Cham (2018)
21. Hargood, C., Hunt, V., Weal, M.J., Millard, D.E.: Patterns of Sculptural Hypertext in Location Based Narratives. In: Proceedings of the 27th ACM Conference on Hypertext and Social Media. pp. 61–70. HT '16, Association for Computing Machinery, New York, NY, USA (Jul 2016), https://doi.org/10.1145/2914586.2914595
22. Hill, J.S.: A Taxonomy of Choices: Axes of Choice (2017), https://www.choiceofgames.com/2017/12/a-taxonomy-of-choices-axes-of-choice/
23. Karhulahti, V.M.: Suspending Virtual Disbelief: A Perspective on Narrative Coherence. In: Interactive Storytelling, vol. 7648, pp. 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
24. Kitromili, S., Jordan, J., Millard, D.E.: What is Hypertext Authoring? In: Proceedings of the 30th ACM Conference on Hypertext and Social Media. pp. 55–59. HT '19, Association for Computing Machinery, New York, NY, USA (Sep 2019). https://doi.org/10.1145/3342220.3343653
25. Martens, C., Iqbal, O.: Villanelle: An Authoring Tool for Autonomous Characters in Interactive Fiction. In: Cardona-Rivera, R.E., Sullivan, A., Young, R.M. (eds.) Interactive Storytelling. pp. 290–303. Lecture Notes in Computer Science, Springer International Publishing, Cham (2019)
26. Mateas, M., Mateas, M., Stern, A.: Façade: An Experiment in Building a Fully-Realized Interactive Drama (2003), http://130.203.136.95/viewdoc/summary;jsessionid=72514A39F6B5D5F3EC79C74A537960DA?doi=10.1.1.14.6176
27. Mateas, M., Stern, A.: Architecture, Authorial Idioms and Early Observations of the Interactive Drama Façade (2002)
28. McCoy, J., Treanor, M., Samuel, B., Mateas, M., Wardrip-Fruin, N.: Prom Week: social physics as gameplay. In: Proceedings of the 6th International Conference on Foundations of Digital Games. pp. 319–321. FDG '11, Association for Computing Machinery, New York, NY, USA (Jun 2011). https://doi.org/10.1145/2159365.2159425, https://doi.org/10.1145/2159365.2159425

29. Nutt, C.: The end of Versu: Emily Short looks back (Mar 2014), https://www.gamedeveloper.com/business/the-end-of-versu-emily-short-looks-back, section: business
30. Pacian, C.: Superluminal Vagrant Twin (2016), https://ifdb.org/viewgame?id=5xzoz5wimz4xxha
31. Pinchbeck, D.: Dear Esther: An Interactive Ghost Story Built Using the Source Engine. In: Spierling, U., Szilas, N. (eds.) Interactive Storytelling. pp. 51–54. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2008)
32. Reed, A.A.: Sharing Authoring with Algorithms: Procedural Generation of Satellite Sentences in Text-based Interactive Stories. In: Proceedings of the The third workshop on Procedural Content Generation in Games. pp. 1–4. PCG'12, Association for Computing Machinery, New York, NY, USA (May 2012). https://doi.org/10.1145/2538528.2538540, https://doi.org/10.1145/2538528.2538540
33. Revi, A.T., Millard, D.E., Middleton, S.E.: A Systematic Analysis of User Experience Dimensions for Interactive Digital Narratives. In: Bosser, A.G., Millard, D.E., Hargood, C. (eds.) Interactive Storytelling. pp. 58–74. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020)
34. Ryan, J.O., Barackman, C., Kontje, N., Owen-Milner, T., Walker, M.A., Mateas, M., Wardrip-Fruin, N.: Combinatorial Dialogue Authoring. In: Mitchell, A., Fernández-Vara, C., Thue, D. (eds.) Interactive Storytelling. pp. 13–24. Lecture Notes in Computer Science, Springer International Publishing, Cham (2014)
35. Ryan, M.L.: Interactive Narrative, Plot Types, and Interpersonal Relations. In: Spierling, U., Szilas, N. (eds.) Interactive Storytelling. pp. 6–13. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2008)
36. Short, A.E.: Beyond Branching: Quality-Based, Salience-Based, and Waypoint Narrative Structures (Apr 2016), https://emshort.blog/2016/04/12/beyond-branching-quality-based-and-salience-based-narrative-structures/
37. Spierling, U., Szilas, N.: Authoring Issues beyond Tools. In: Iurgel, I.A., Zagalo, N., Petta, P. (eds.) Interactive Storytelling. pp. 50–61. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2009)
38. Staff, G.: Star Wars: The Old Republic revealed (2008), https://www.gamespot.com/articles/star-wars-the-old-republic-revealed/1100-6199726/
39. Stern, A.: Embracing the Combinatorial Explosion: A Brief Prescription for Interactive Story R&D. In: Spierling, U., Szilas, N. (eds.) Interactive Storytelling. pp. 1–5. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2008)
40. Wakes, D.: Girth Loinhammer's Most Exponential Adventure (Jul 2017), https://damonwakes.wordpress.com/books/girth-loinhammers-most-exponential-adventure/