



Working Backwards: Creating a Character Backstory Generation System Using Idealized Creative Writing Outputs

Creating a Character Backstory Generation System Using Idealized Creative Writing Outputs

Lynda Clark

InGAME, University of Dundee; University of Edinburgh
lynda.clark@ed.ac.uk

Divij Sood

InGAME, Abertay University
d.sood@abertay.ac.uk

ABSTRACT

This paper is a reflection on the process of designing and developing a character backstory generation system prototype following literary analysis of idealized textual outputs. An overview of previous generative systems (both academic and commercial) and some of the design priorities associated with these systems is described in order to set a context for the project. The design process is then described, with particular focus on the creation of the idealized outputs and their purpose. Finally, the learning outcomes following initial generative texts created by the prototype engine are shared, weighing the pros and cons of both the design approach, the resulting generator and its outputs.

CCS CONCEPTS

• **Applied computing – Arts and Humanities;**

KEYWORDS

Text generation, narrative design, creative writing, literary analysis

ACM Reference Format:

Lynda Clark and Divij Sood. 2022. Working Backwards: Creating a Character Backstory Generation System Using Idealized Creative Writing Outputs: Creating a Character Backstory Generation System Using Idealized Creative Writing Outputs. In *FDG '22: Proceedings of the 17th International Conference on the Foundations of Digital Games (FDG '22), September 05–08, 2022, Athens, Greece*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3555858.3555867>

1 BACKGROUND

While AI-powered generative systems are constantly advancing, and GPT3, the latest iteration of OpenAI's Generative Pre-Trained Transformer continues to improve in generating coherent textual outputs, the areas where such systems excel tend to be limited by the suitability of data sets available. In addition, creative and abstract tasks remain difficult for pre-trained AI systems to achieve due to issues with coherence, repetition, and contradiction [6]. Therefore, for many creators aiming to generate narrative content, at least some degree of pre-authored text is desirable. [3, 13]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

FDG '22, September 05–08, 2022, Athens, Greece

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9795-7/22/09...\$15.00
<https://doi.org/10.1145/3555858.3555867>

However, as Ryan et. al observe [8, 16, 17], it is all too easy for generative systems to require enormous amounts of authored content in order to produce varied, inventive outputs. Striking a balance between these two extremes is challenging, and therefore requires attention to system and output structure alongside consideration of predecessors and current exemplars. This paper is a reflection on the process of developing a narrative generation system prototype following literary analysis of idealized textual outputs. An overview of previous generative systems (both academic and commercial) and some of the design priorities associated with these systems is described in order to set a context for the project. The design process is then described, with particular focus on the creation of the idealized outputs and their purpose. Finally, the learning outcomes following initial generative outputs created by the prototype engine are shared, weighing the strengths and weaknesses of both the design approach and the resulting generator.

1.1 AI vs Algorithmic Generation

Defining 'Artificial Intelligence' can be problematic as it has increasingly become something of a buzzword, applied fuzzily to a wide variety of systems and technologies [19]. This paper deals with systems which might be considered more algorithmic in nature and require on-going human authorship and narrative design, rather than Artificial Intelligence (in this paper, AI is defined as technologies and systems with a machine-learning component which seek to reduce or eliminate human-authoring). This paper focuses primarily on algorithmic systems as opposed to AI because the project in question is writer-driven and is therefore more interested in balancing authored and generated content (as seen in, for example, *Wildermith* [20]) and providing authors with a tool and process for generation than it is in the technical challenges associated with pure content generation (as seen in, for example, *AI Dungeon* [12]). However, it is of course acknowledged that regardless of the technology employed, any generational system is 'still largely dependent on the information input to it by the human author' [11] whether such input is narrative or computer science-related, and therefore examples of many types of generation inform the discussion.

1.2 Some Approaches to Story Generation Systems

There are various ways in which story generation systems are approached, described and categorized. These approaches usually relate to either the degree and type of generation vs pre-authored content, or the author's role within the system. In their comprehensive survey of types of story generators, Kybartas and Bidarra focus on automatic generation vs manual authoring. Broadly speaking,

they consider these degrees of generation in relation to the creation of plot (events) and space (constituent parts of a story such as characters, locations, props and the initial world state). Essentially, Kybartas and Bidarra propose that all story generation systems may range from manual to generated in terms of both plot and space. Plots may be manual, structured, follow a template, be constrained (automated with some authoring and adjustment to improve output), or fully automated (examples range from Propp and various systems based on his work on plot structure, to Cavazza et al's EmoEmma which uses the setting and themes of *Madam Bovary* as a starting point for plot generation). Similarly, spaces may be manual, modified, simulated, constrained or fully automated (Here examples include the GameForge space generation system, which 'takes a story as input and outputs a valid space' and *CarSim*, which generates a 3D construction of a space from car crash data). Kybartas and Bidarra also observe thematic approaches such as conflict-led (e.g. soap opera plot generators) and social interaction-led models (e.g. *Prom Week*). Naturally, these are not mutually exclusive with regard to the other categorizations. Kybartas and Bidarra acknowledge that where story systems are highly generative and emergent, they may 'form the character actions into *some form of coherent plot*' (our emphasis) rather than a more traditional narrative. [11]

Bailey divides story generation systems somewhat similarly, referring to story models (equivalent to Kybartas and Bidarra's plot), world models (equivalent to space), and the additional category 'author models' – those which attempt to simulate the actions taken by an author. As mentioned above, Kybartas and Bidarra's position is that the 'human' element cannot be completely removed from any system, and therefore some degree of authorship is implied regardless of model. While Bailey acknowledges the variation between and within these categories, he does not attempt to distinguish such variation further. Instead, he argues for an alternative 'reader model' which centers 'storiness', or 'the effect of a story on its reader'. Bailey recognizes that authors are of course also readers, suggesting that what is important is to 'address the story itself as an artefact, rather than concentrating wholly on the cognitive aspects of the human authorial process'. [5] In other words, the effects of the authors process are more important than the way they are achieved. This is an observation also noted by Crawford when considering *The Sims* as a system where '[perceiving] a story is all that matters'. [8]

Sharing elements of both these approaches, Aylett and Louchart make the distinction between plot-driven (or top-down) and character-driven (or bottom-up) generation but incorporate the idea of authorial goal-driven systems within their plot-driven category. They suggest that their own emergent 'character-driven' approach, while lacking the application of a global plot seen in most plot or author-driven systems, is actually closer to an author's genuine methodology, as it involves generating story 'bottom-up by standing in the shoes of the story characters' [4].

Finally, Jonathan Lessard et al describe an approach to generation which privileges 'author-friendliness'. 'Author-friendly' tools are described as those which: 'should afford authors to (relatively) painlessly: (1) enter new content; (2) predict runtime behaviour; (3) identify, and (4) fix errors'. Rather than being an 'author model', this methodology could perhaps be considered a 'writerly model' as it acknowledges that writers moving into this sphere may be

"naive" developers (not specialists of computational linguistics)' and focuses on giving them new forms of expression rather than attempting to mimic existing processes. [13]

While obviously all these approaches are different in terms of their goals, focus and level of automation, what they all share is the need to consider the role of the author within the system. Whether this is in terms of the degree to which content is pre-authored, how the narrative is ultimately conveyed to the reader, or how the author is able to make use of the system in order to produce content, it seems defining author contribution to the system is an essential and unavoidable component of narrative generation system design. It is also clear that there is a requirement for imposing some form of 'storiness' [5] or 'coherence' [10, 11] on the output in some way, although as Crawford observes above with *The Sims*, in character-driven systems, players are likely to 'imagine stories based on behaviour' even when there is no explicit story [8].

1.3 Influential Exemplars

There were three key influences for the team in terms of existing story generation: *Dwarf Fortress* [1], *Wilderness* [20] and *Sid Meier's Pirates!* [9] While *Dwarf Fortress* is not a story generation system per se, its use of 'bottom-up' simulation results in emergent narratives [17]. Players are introduced into a living world where 'settlements are raised, trade routes are formed, and wars are waged'. Creator Tarn Adams suggests four key design principles which were used to create *Dwarf Fortress*, but which he believes can be generalized for any type of simulation: avoiding overplanning and instead using iterative design, ensuring a clear understanding of the interplay between simulation elements, avoiding overcomplication wherever possible, and taking inspiration from real-world models [2].

Another recent example, *Wilderness*, uses 'alternating layers of authored and system-driven content' to create its narratives. *Wilderness* takes a 'top-down' approach, with an authored plot into which procedurally generated characters, events and encounters are slotted. Character dialogue is pre-authored, but certain lines are only deployed according to the character's personality, which is emergent. However, creator Nate Austin acknowledges that this results in a number of limitations to the kind of stories that might be told. There are no particularly unique or 'iconic' characters, as technically any character could be cast in any role. Once the dynamics within the party are set, they will not change. (For example, characters who have a friendship cannot then initiate a romance or become rivals). This means that there can be no betrayals or anti-heroes, as this would result in inconsistent behaviours and relationships [3].

Published some thirty years before *Dwarf Fortress* and *Wilderness*, *Sid Meier's Pirates!* shares several elements with both. Somewhat like *Wilderness*, *Pirates!* has a set framework of possible events (e.g. courting noblewomen, dueling, searching for lost family members) but these can be freely chosen by the player and occur within the simulated world, which, like *Dwarf Fortress* includes its own constantly developing faction rivalries and events (although far fewer and simpler) and despite creating stories, is not specifically a narrative generator. When discussing the impetus behind designing *Pirates!* in his recent autobiography, Meier explains: 'Adventuring didn't have to mean blindly groping for a set path. It could mean

making up your own story, being in charge of your fate just like a pirate would be.' The desire to leave behind a linear 'set path' and instead embrace 'making up your own story' seems a call towards emergent narrative, even if that term was not widely employed at the time. Meier also suggests that using a pirate theme brought with it a kind of 'cultural shorthand' which 'allowed the player to fill in the environment without realising they were doing it' [15]. This observation simultaneously suggests an additional benefit to Adams' remarks about the use of real-world models, while also addressing Bailey's concern regarding the effect of the system on the player and Crawford's observation of the player's role in story generation.

The overall theoretical influence on the project was Ryan's *Curating Simulated Story Worlds*, which takes the underlying principle of treating emergent narratives as events which have happened, and then curating the flow of information into something more readable [16]. He specifically explores systems that simulate storyworlds and allow narrative to emerge as a result of character activity. His other work explores how knowledge, as a key component of human storytelling, can be included in such narrative generation systems. Ryan mentions his project *Islanders* as a precursor to his larger projects. This was a simpler simulation of characters' lives at sea and offered another influence for this project due to the similar focus on a short-term preliminary exploration of character-based simulation and the pirate-based theme chosen. [18]

2 PROJECT OVERVIEW

This project was a collaboration between developers from independent games studio EvoRevo, and research and development fellows from InGAME (Innovation in Games and Media Enterprise). EvoRevo's developers brought extensive knowledge of content generation systems, although their previous work focused more on world simulation than narrative. InGAME's fellows were a narrative design and writing specialist and an expert in procedural generation and procedural system design. However, for all collaborators this particular combination of story and procedural generation was a new area.

2.1 R&D Approach

The project ran in a cycle of synchronous and asynchronous discussions and planning sessions, followed by asynchronous development with regular communication. An initial co-design workshop held over Zoom established the theme, structural outline and desired core experience of the generative system prototype. This was followed by a research phase where collaborators undertook individual research tasks relating to their own agreed area of work. Research was a mixture of 'traditional' (e.g. reading of relevant papers and books) and practice-based (e.g. playing exemplars and writing, designing and making small-scale test pieces and annotated idealized outputs). A second workshop reviewed the created materials, selected a preferred idealized output and expanded the structural outline into a more detailed plan. The plan was then expanded asynchronously with reference to the selected idealized output and its accompanying analysis and then signed off, allowing work to commence on the prototype generator. Collaborators

worked remotely and asynchronously with regular brief catchups to ensure alignment.

2.2 Aims and Objectives

The central goal of the project was to generate character backstories for non-player characters. Some generated stories should be interesting (it was accepted that some might be more mundane) but all should be narratively coherent (e.g. no contradictory information as in *AI Dungeon*) and avoid excessive repetition in both structure and content. It was determined that while there would be an initial element of authorial labour, ultimately this generator would be a time-saver, allowing games writers to focus more on player-character or quest narrative. It was also important for the system to be as 'author-friendly' as possible, and allow input and editing to the text outputs without knowledge of code.

Some elements were considered out of scope, as this was a short form (3 month) project. It was agreed that the project would result in only an early-stage prototype generator which had the potential to be plugged into a larger game – there would be no actual gameplay or interactive elements and the output would not be a commercial-ready product, merely a proof of concept of the generation system itself.

2.3 Risks and Challenges

The key risks and challenges associated with the project were not unique, but two were particularly keenly felt due to the rapid turnaround required for the project, while the third remained a question until additional research had been undertaken. Primary concerns were that the problem would be too large to solve, and that system outputs would be flawed to the point of being unusable (particularly given the project timeline). Previous examples such as *Wildermuth* [20] and Crawford's various projects [8] indicate that generating interesting character backstories may simply require a large amount of writing alongside multiple layered generation systems. However, it was agreed that through focusing on working backwards from idealized outputs, a process for developing a solution could still be produced, even if the prototype generator itself was of limited usefulness. This line of thinking also helped allay concerns about actual outputs – even if they failed to live up to the idealized versions, developing the system would provide valuable learning opportunities for all collaborators.

As the project involved an industry partner, there were concerns that a generative system which did not make use of AI would quickly become obsolete, particularly with OpenAI's recent improvements to GPT3 and the creation of an API [6]. However, it quickly became apparent that while GPT3 is exceptionally good at some kinds of text generation, narrative remains difficult. Even one of the more advanced and extensively developed applications of GPT3, *AI Dungeon*, generates stories that are 'garbled and incoherent, more often than not' [10]. Therefore, while it may be that AI could be useful for an element of narrative generation, it was considered unlikely that an AI system would provide a sufficient solution for the industry partner's aims and objectives at present.

3 APPROACH

While most of the systems, approaches and exemplar games above are described in terms of being top-down (author or plot driven) or bottom-up (character or world-driven), the approach taken in this project, while broadly bottom-up from a generative perspective, might also be thought of as ‘back-to-front’ in terms of its design methodology. This is because the starting point was to manually author idealized character backstories and analyze them to determine the necessary elements of the generator.

During the initial co-design workshop, it was decided that a fantasy nautical theme would be used, incorporating piracy, travel and settlements alongside relationships between characters. However, it was also agreed that the system should be generalizable if possible, so that changing key texts could change the genre to, for example, science fiction or high fantasy. Therefore a few potential generator elements and their relationships were sketched into an outline plan which would be genre agnostic (for example, character relationships, world events, travel, combat etc.).

Following this initial co-design workshop, three short sample outputs were written. Beginning with outputs builds on Aylett and Louchart’s bottom-up authorial process [4], but also incorporates Bailey’s notion of developing ‘the story [...] as an artefact’ [5]. As the idealized outputs were human-authored, it was acknowledged that they would contain some information that would not be possible to generate, but, as the name implies, they were intended to provide inspiration as to what an ideal system would generate, rather than an exact template for the prototype generator’s outputs. They were kept short – around 500 words, as it was unlikely generated backstories would need to be any longer than this. Each short text was also intended to demonstrate a different aspect of a potential generator to aid discussion in the second co-design workshop. In each instance, the texts were annotated to explain the authorial reasoning behind certain choices (see Appendix A1). Although authorial process would not be simulated, this was a specific request from EvoRevo’s developers, who felt it would help them better understand which elements were essential to generation and which were lower priority.

The first was a retelling of William Goldman’s *The Princess Bride*. This novel and film were chosen as inspiration because of the fantasy setting, piratical subplot and large cast of characters. The output was essentially a short story listing all the characters and their exploits, with the view that in the generator each character name would be clickable giving their backstory and connections to the other characters. For some this would be extensive, for others, very brief.

The second focused in on one mundane character backstory and described that character’s life as a fish gutter, their hope to leave and do something more, and their final acceptance that they will remain gutting fish forever. This output was intended to show both some of the relationships (e.g. familial connections, friendships) and other elements (job roles, aspirations, failures etc) that might be needed within the generator, and give a sense of how a fairly detailed complete output might look.

Example three was a single character’s backstory this time told via the framing device of a narrator. While this shared many similarities with example two in terms of its demonstration of character

relationships, key events and necessary abstract concepts, it was primarily an exploration of having a distinct pre-authored narrative ‘voice’ to frame the generated contents. There is an element of this framing concept present in *Wildermith*, but the narrator has no clear identity. The idea here was that if an issue similar to *Wildermith*’s lack of ‘iconic’ characters arose, the effect could be masked somewhat with a more iconic narrator.

EvoRevo and InGAME agreed that example two offered the most desirable and achievable output. During the second co-design workshop, some of the elements necessary to generate such an output were discussed and used to inform the development of the character backstory generator. The full annotated version of this output is available in the Appendix (A1). While some annotations explained authorial reasoning behind certain choices (for example, the fact that ‘She has less than a handful of pennies when she hears the news’ is open to interpretation by the reader to draw them in), others were written with generation in mind (e.g. age and financial circumstance are what drive the character to act – elements which could be generated), while still others highlighted text that would probably need to be pre-authored (e.g. the descriptive sentences that give colour, sound and scent to the narrative world). Crawford reached a similar conclusion when trying to generate soliloquies with his Deikto engine, noting that its outputs ‘would lack such elegant phrases’ as those found in *Hamlet* because ‘the action would be presented in its simplest, most basic form’ [7]. While this use of pre-authored text mitigates against this potential simplicity in some respects, the basic nature of the outputs is not always an issue in structural terms. Since the generated content is intended to provide backstories for non-player characters rather than an overarching narrative, there is less of a need for a traditional ‘plot’ than in some of the exemplars.

3.1 Simulation Design

The team started by identifying the following key parts of the story that should be simulated for a minimum viable product technical demonstration: locations or towns, familial relations, and character details (names, occupations, skills, traits, and goals). Keeping this in mind, the simulation was designed with three types of entities at its core – Characters, Ships, and Towns.

A core simulation loop was designed consisting of five repeating phases: Town events, ship events, character events, character actions, and year-end setup. Character entities exist, act, and interact with each other in the world over several hundred years, sped-up for convenience. Characters have names, some defining characteristics, familial ties, occupations, and a set of likes and dislikes. Characters take up different goals based on their situation, their characteristics, and their location. Some goals may involve travel, which occurs via ship. When a character is on a ship, their lives are simulated with ship-based rather than town-based events. Towns can have events which are either beneficial or catastrophic to their residents. Each town has set features which affect the likes and dislikes of its residents. A map is procedurally generated at the start of the simulation which keeps track of island-like landmasses in the world, their locations, and which ones are inhabited. A character’s actions and interactions in the simulated world can lead to conflicts. Somewhat similar to Aylett and Louchart’s model, rather

than using a global plot, the conflicts interrupt character actions and affect their choice of goals. However, in our system, rather than attempting to model ‘cognitive-affective architecture’ [4], we are simply trying to simulate character arcs by placing dramatic obstacles and opportunities in their way.

3.2 Tool Design

The technical demonstration was designed as a tool for designers and writers to use to support the practice of developing narrative-based role-playing games. Three key principles were kept in mind for this purpose to ensure writers could build the intended experience without friction or the need to learn new programming languages [14]:

- 1) All data, in this case in the form of authored text input, must be separated from the simulation code and be easy to edit;
- 2) The simulation should be designed to be independent of the content of this input to as large an extent as possible to allow for narrative flexibility;
- 3) Outputs from the system must be easy to filter, view and edit.

4 OUTCOME

A prototype character backstory generator was created which met the project’s key objectives of creating varied character stories while also offering an ‘author-friendly’ method of inputting content. Authored text is read via CSV files, meaning they are easily edited with Excel once the author understands the basic structure (which is fairly self-explanatory). In order to meet Lessard et al’s other criteria of identifying and fixing errors and predicting runtime variations [13], visualized simulation was used in combination with the ability to export character stories as text files.

The procedurally generated map is visualized with a simple 2D texture reflecting the positions of landmasses and dynamically showing changes like the formation of settlements and movements of characters on ships. This supports the user’s understanding of the simulation and helps with spotting and debugging simulation errors. Textual errors can be found and edited through comparing the text outputs with the original CSV entries.

4.1 Simulation Entities

An initial world state is simulated with characters born and later travelling to begin new settlements. Characters have starting characteristics (levels of comfort, wanderlust, desperation, good and evil) which can be impacted by world events (famine, enemy attacks, gold rushes). Changes to characteristics prompt setting and/or changing of character goals. The four main goals used in the prototype are: 1) Family (set when the character has a high comfort level), 2) Travel (set when wanderlust is high), 3) Piracy (prompted by high levels desperation) and 4) Healing (prompted when a family member or loved one is unwell and causing the abandonment of other goals). Characters may achieve their goals, or they may need to adjust them, or obtain other goals first – for example, a family goal will fail if there are no eligible suitors nearby, which may prompt travel, which may first require an accumulation of wealth. If their settlement is attacked by pirates or monsters, they may die before achieving a goal.

4.2 CSV Entries

All authoring of input text is completed via CSVs editable in Excel. In Table 1 below is an example of a CSV for character actions. While both the default output text and alternate phrasing are shown together in Table 1 below, these are actually separate CSVs in the generator. This shows the ease with which the content can be expanded – additional CSVs can be added, and new entities can also be added relatively easily within each CSV (in collaboration with the project’s programmers). Authored content is separated out from code and generated content in the system and can be used as needed. This allows for flexibility and creativity, as long as the writer is mindful of which type of data each element represents within the sentence structure. For example, in ‘DiscoveredTown’ X represents the character’s name, Y represents the name of the ship they travelled on, and Z represents the name of the town. Each of these elements are procedurally generated. However, the writer may refer to them in a completely different order, as in the ‘DiscoveredTown’ ‘Alternative Phrasing’ example, as long as this still results in a coherent sentence.

4.3 Text Outputs

Outputs in the form of exportable text documents (See Appendix A2) allow writers to review full transcripts of backstory simulation. This allows both corrections of errors and typos and offers a means of understanding how the disparate CSVs are combined at runtime. Being able to review texts in this way also provides a clear indication of where further content (and/or code) is needed to avoid repetition and create a more pleasing flow to the backstory. Generated stories can also be filtered based on log size, goals achieved, type of occupation, and levels of stats such as good and evil. Again, this makes it easier to locate potential errors, and to identify the most and least interesting character backstories. While the generator is currently at an early stage, initial outputs were created sharing many of the key features identified in the textual analysis. For example, the character Dereni Delan falls in love (prompted by their goal of having a family): ‘Dereni Delan is completely taken by Ethar Delan’, but their feelings are not returned. Therefore, after several attempts at proposing ‘Dereni Delan realises that their love for Ethar Delan will always be unrequited’ and their goal changes from family to travel. Fully authored sentences (e.g. ‘Dereni Delan rolls out some bolts of fabric and cuts them to length.’) are deployed at appropriate moments to break up the text and add more description (Appendix A2). In a more complete iteration of the generator, more of these would be available to ensure less repetition.

Obviously, the generator is not without flaws at both a simulation and textual output level. As is evident from the examples given above, the system does not prevent those with the same family name from initiating romance with one another. A full family system was theorized which would track family trees long term, but in the current version, character knowledge is limited to immediate family. The repetition of full names throughout and the fact that some phrases are repeated multiple times in quick succession makes for clunky reading. Friendship is not particularly accounted for in the CSV files, and so friends only really appear when they are saving their friends’ lives and then rapidly disappear again. While it is perhaps not clear from the example given, a simulation error

Table 1: Character Actions CSV

Action	Output Text	Alternative Phrasing
FirstMeet	X met Y for the first time today.	X's eyes met Y's across a crowded room.
GiftPositive	X gave Y Z, and Y really liked it!	X presented Y with Z, and Y was delighted.
GiftNegative	X gave Y Z, but Y was completely offended!	X was horrified to receive Z from Y.
Death	X passed away due to old age.	X lived until the ripe old age of Z.
Marry	X and Y got married.	X and Y were joined in matrimony.
Reproduce	X and Y brought Z into the world.	X and Y became proud parents to little Z.
Travel	X started a voyage from Y on Z.	X boarded Z, leaving Y behind.
ReachedTown	X reached Y on Z.	After a long voyage on Z, X reached Y.
DiscoveredTown	X reached a new land via Y and established a town named Z.	Z was established by X, who had arrived there aboard Y.
DeathConflict	X was killed by Y.	X perished at the hands of Y!
NewOccupation	X started work as Y.	X accepted a job as Y.

became obvious through multiple outputs in that most characters end up as monster hunters because it is the most lucrative and in-demand job, and many of the other goals require at least some degree of wealth to achieve. However, the key project goals of author-friendliness, textual and structural variation, and a lack of narrative contradiction were met, and many of the issues identified above could be resolved through tweaks to the generator's code and/or the text CSVs.

5 CONCLUSION

Kybartas and Bidarra assert that generative systems are 'usually most successful when the author maintains complete control over content, and the computer's duty is to provide important services for the author', [11] while Adams suggests a slightly freer approach, where 'the individual elements [...] combine, producing results which can be varied, surprising, and yet still responsive to your adjustments' [2]. Ultimately, the generator created allowed complete authoring control via the CSVs, but still resulted in varied, surprising outputs through the combination of multiple system elements.

The use of three idealized creative writing outputs to initiate the design conversations meant that the industry partners could select the most suitable output for their needs, however this process could have been streamlined. Either a single idealized output could be written, or three outputs all exploring different aspects of the same narrative concept. However, a clear brief would be required in advance in order to achieve this.

Using the selected idealized creative writing output, the team worked backwards to create a generator containing entities which interact to produce varied character backstories. This approach incorporates authorial practice but places its primary focus on the outputs themselves rather than attempting to replicate creative processes. While initially written with a fantasy nautical theme in mind, the generator has potential for creating backstories in other genres by simply changing CSV content to match the desired genre. The system also has sufficient flexibility at a sentence level to allow writers to get creative with the structure of phrases around entity data. The generator is early stage, but demonstrates the potential

of both the system itself and the use of idealized creative writing outputs in the design process.

ACKNOWLEDGMENTS

This project is funded by InGAME: Innovation for Games and Media Enterprise, part of the AHRC Creative Industries Clusters Programme. Grant no: AH/S002871/1

REFERENCES

- [1] Tarn Adams and Zach Adams. 2006. Dwarf Fortress. Kitfox Games, Montreal.
- [2] Tarn Adams. 2015. Simulation Principles from Dwarf Fortress. In S. Rabin (Ed.). 2015. Game AI Pro 2. pp. 519–521. CRC Press, Boca Raton. http://www.gameaipro.com/GameAIPro2/GameAIPro2_Chapter41_Simulation_Principles_from_Dwarf_Fortress.pdf
- [3] Nate Austin. 2021. Progen in Wildermyth: Storytelling. Video. Retrieved from <https://www.youtube.com/watch?v=A5BGDbLFRrE>
- [4] Ruth Aylett and Sandy Louchart. 2013. I contain multitudes: creativity and emergent narrative. In Proceedings of the 9th ACM Conference on Creativity and Cognition. C&C '13, Sydney, Australia, 337–340. 3 pages. <https://doi.org/10.1145/2466627.2466659>
- [5] Paul Bailey. 1999. Searching for Storiness: Story-Generation from a Reader's Perspective. In Working Notes of the Narrative Intelligence Symposium, Manlo Park, California, 157–164. 7 pages. <https://aaai.org/Library/Symposia/Fall/fs99-01.php>
- [6] Tom B. Brown *et al.* 2020. Language Models are Few-Shot Learners. arXiv: 2005.14165. Retrieved from <https://arxiv.org/abs/2005.14165>
- [7] Chris Crawford. 2021. Is Language-Based Interactive Storytelling Possible? Erasmatazz. May 31 2021. <http://www.erasmatazz.com/library/fourth-generation-interacti/reconsideration-of-fundamen/is-language-based-interacti.html>
- [8] Chris Crawford. 2005. Chris Crawford on Interactive Storytelling. New Riders, Berkley, CA.
- [9] Firaxis Games. 2004. Sid Meier's Pirates! Atari, New York.
- [10] Derek Garcia. 2021. Why AI Dungeon is a Work in Progress, and Still Not Safe. Screenrant. May 27 2021. <https://screenrant.com/ai-dungeon-not-safe-children-kids-safeguards-problems/>
- [11] Quinn Kybartas and Rafael Bidarra. 2016. A Survey on Story Generation Techniques for Authoring Computational Narratives. IEEE Transactions on Computational Intelligence and AI in Games, 9, 3, (March 2016), 239–253. <https://doi.org/10.1109/TCIAIG.2016.2546063>
- [12] Latitude. 2019. AI Dungeon. Latitude, Utah.
- [13] Jonathan Lessard, Etienne Brunelle-Leclerc, Timothy Gottschalk, Marc-Antoine Jetté-Léger, Odile Prouveur and Christopher Tan. 2017. Striving for author-friendly procedural dialogue generation. In Proceedings of the International Conference on the Foundations of Digital Games (FDG '17), ACM, Hyannis, Massachusetts, 6 pages. <https://doi.org/10.1145/3102071.3116219>
- [14] David Lightbown. 2015. Designing the user experience of game development tools. CRC Press, Boca Raton.
- [15] Sid Meier. 2020. Sid Meier's Memoir!: A Life in Computer Games. W.W. Norton and Company, Manhattan.

- [16] James Ryan. 2018. Curating Simulated Storyworlds. https://www.researchgate.net/publication/330855103_Curating_Simulated_Storyworlds
- [17] James Ryan, Michael Mateas and Noah Wardrip-Fruin. 2015. Open Design Challenges for Interactive Emergent Narrative. In *Proceedings of the International Conference on Interactive Digital Storytelling (ICIDS '15)*, Springer Lecture Notes in Computer Science, Aalborg, Denmark, Volume 8, 14-26, 12 pages. https://doi.org/10.1007/978-3-319-27036-4_2
- [18] James Ryan, Adam Summerville, Michael Mateas, Noah Wardrip-Fruin, 2015. Towards Characters Who Observe, Tell, Misremember and Lie. In *2nd Workshop on Experimental AI* (November 2015) Santa Cruz.
- [19] Pei Wang. 2019. On Defining Artificial Intelligence. *Journal of Artificial General Intelligence*, 10, 2, (January 2019), 1-37. <http://dx.doi.org/10.2478/jagi-2019-0002>.
- [20] Worldwalker Games. 2021. Wildermyth. Worldwalker Games, Texas.

A APPENDICES

Below is the annotated idealized output chosen to form the inspiration for the character backstory generator. It was originally annotated using Word's commenting feature, but has been footnoted to meet layout requirements.

A A.1 ANNOTATED IDEALIZED OUTPUT: MUNDANE CHARACTER BACKSTORY

Clara grows up on the docks of a tiny fishing village. Her father works on the fishing boats. Her mother is taken ill after the birth of Clara's younger sister, Lucy, and so Clara takes care of them both. Money is tight, and when Clara turns sixteen¹ Convergence of motivations (age and finances) drives character movement. she's sent to the large port up the coast to work as a fish gutter, a task she excels at.² Potential futures for this character are somewhat limited due to financial and social circumstances. She impresses the older women³ Less important characters go unnamed in this example, but this could just as easily be a named individual for ease of generation. who organize the fish preparation, and they take her under their wing, treating her as their own daughter.

While her new home is by no means cosmopolitan, Clara is very taken by the hustle and bustle of the port and dreams of joining the crew of a merchant ship and travelling to tropical climes.⁴ Change in location triggers an aspiration. She begins saving a penny a week from her wages to buy her passage.

She has less than a handful of pennies when she hears the news⁵ This could be interpreted by the reader in a couple of ways – she earns very little, or receives the news very soon after her decision to save. These small ambiguities invite the reader to participate in creating the meaning of the text. – her father was lost at sea, swept overboard in a terrible storm. She sends the pennies home to Lucy to make arrangements for the funeral. There aren't enough for her to attend herself. Her sister takes that as a slight and stops responding to her letters.

¹Convergence of motivations (age and finances) drives character movement.

²Potential futures for this character are somewhat limited due to financial and social circumstances.

³Less important characters go unnamed in this example, but this could just as easily be a named individual for ease of generation.

⁴Change in location triggers an aspiration.

⁵This could be interpreted by the reader in a couple of ways – she earns very little, or receives the news very soon after her decision to save. These small ambiguities invite the reader to participate in creating the meaning of the text.

A merchant ship docks at the port and Clara stares up at its figurehead⁶ As Clara's dream has already been stated,

she receives a reminder of it here, although this isn't explicitly described as such to the reader. as she works her way through baskets of mackerel and buckets of cod, coiling their discarded guts into pink-grey piles⁷ This kind of descriptive text is very important to help build the narrative, but difficult to generate coherently, therefore pre-authored descriptions interspersed may be a solution. She memorises every detail of the figurehead, a leaping dolphin flecked with gold leaf. She begins saving pennies again.⁸ Because Clara's dream remains unchanged and has not been realised, she reverts to a previous action (saving money). Again, this requires no further explanation, as the reader will determine her reasoning themselves from the context.

Winter rolls around and she could go back to the village for the off season, but she doesn't know how she'll be received by her sister, so she stays at the port with one of the older gutting women and her family. She spends her savings in the local tavern, buying her hosts tankards of mead and fine cuts of lamb in thanks for their hospitality.⁹ This latest piece of information may offer an alternative reading of the earlier 'handful of pennies' – perhaps Clara has a trait which makes her frivolous with money.

At the height of the season, when the fish guts are really stinking in the sunshine¹⁰ Sights and smells are really important for evoking a sense of time and place. Clara's sister writes again. Their mother's condition has worsened – she must come home at once if she wants to see her at all. Clara digs into her savings once more¹¹ Although this same action has been repeated several times, each time the outcome has been different. arriving in time to spend a final few hours with her ailing mother. Her sister is still somewhat bitter over being left to hold things together alone for so long, but she's keen to introduce Clara to her handsome fiancé so she can gloat a little. Seeing her sister settled, Clara returns to the port to live more permanently with her older friends.

The Dolphin is in the dock again, offloading spices and silks. Clara still hasn't the money for her passage and fears she never will. She returns to her gutting stool, her quick knife flashing in the sun.¹² This doesn't really end but instead implies the same loop could continue until her death.

B A.2 ACTUAL OUTPUT (UNEDITED)

Meror Delan and Yurae Bath welcome their new baby, Dereni Delan

Dereni Delan's coffers are filling up with gold.

⁶As Clara's dream has already been stated, she receives a reminder of it here, although this isn't explicitly described as such to the reader.

⁷This kind of descriptive text is very important to help build the narrative, but difficult to generate coherently, therefore pre-authored descriptions interspersed may be a solution.

⁸Because Clara's dream remains unchanged and has not been realised, she reverts to a previous action (saving money). Again, this requires no further explanation, as the reader will determine her reasoning themselves from the context.

⁹This latest piece of information may offer an alternative reading of the earlier 'handful of pennies' – perhaps Clara has a trait which makes her frivolous with money.

¹⁰Sights and smells are really important for evoking a sense of time and place.

¹¹Although this same action has been repeated several times, each time the outcome has been different.

¹²This doesn't really end but instead implies the same loop could continue until her death.

A furious Sea serpent causes wanton destruction throughout Swancester .

2 soldiers repel the creature.

11 monster hunters battle the monster.

The repairs required after the attack leave Dereni Delan bankrupt.

Following the rampant deaths and injuries in Swancester , 1 townspeople take up the mantle of healer.

Dereni Delan wants children.

As they turn 9 , Dereni Delan begins a new career as a Tailor inspired by their interest in Plants .

Dereni Delan is falling for Banar Volerept .

Dereni Delan is enjoying life.

Dereni Delan longs to see more of the world.

Dereni Delan rolls out some bolts of fabric and cuts them to length.

Dereni Delan mourns the loss of their close friend, Banar Volerept .

Etteo Oglam is completely taken by Dereni Delan .

Edran Scegoron is falling for Dereni Delan .

Dereni Delan has lost interest in the idea of having children.

Dereni Delan wants children.

Dereni Delan is completely taken by Ethar Delan .

Dereni Delan offers Ethar Delan an engagement ring!

Ethar Delan has eyes only for Velimiwe Agnopus..

Dereni Delan realises that their love for Ethar Delan will always be unrequited.

Dereni Delan proposes to Ethar Delan..

Ethar Delan has eyes only for Mirorwyn Nemthach..

Dereni Delan realises that their love for Ethar Delan will always be unrequited.

Dereni Delan has lost interest in the idea of having children.

Dereni Delan joins a voyage on The Katana from Swancester with Ethar Delan Velimiwe Agnopus Emeloda Nemthach Mirorwyn Nemthach .

Dereni Delan's passage onboard Katana comes to an end at Saltland with fellow passengers Ethar Delan Velimiwe Agnopus Emeloda Nemthach Mirorwyn Nemthach .

Saltland is famed for being full of Cats .

Dereni Delan has heard rumours of Sea serpent and Kraken in the area.

Dereni Delan worries over what the future may hold.

Dereni Delan wants to travel.

Edran Scegoron is falling for Dereni Delan .

A furious Sea serpent causes wanton destruction throughout Saltland .

1 soldiers repel the creature.

15 monster hunters battle the monster.

Edran Scegoron proposes to Dereni Delan..

At the age of 26 , Dereni Delan and Edran Beochain get married.

As they turn 26 , Dereni Delan tires of their job as a Tailor and finds a new vocation as a MonsterHunter due to their passion for Sea serpent .

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan is seriously injured during their work as Monster-Hunter .

Dereni Delan is seriously injured during their work as Monster-Hunter .

Dereni Delan 's health is waning.

Ethar Delan wants to save their friend, Dereni Delan .

Ethar Delan 's efforts are rewarded when Dereni Delan makes a full recovery.

Dereni Delan beats the odds and survives!

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan is content with their existence.

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan soaks in the bath, trying to remove the stench of monster blood.

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan 's life is fading.

Ethar Delan vows to do everything in their power for their suffering friend, Dereni Delan .

Ethar Delan 's efforts are rewarded when Dereni Delan makes a full recovery.

Dereni Delan beats the odds and survives!

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan is seriously injured during their work as Monster-Hunter .

Dereni Delan 's life is fading.

Ethar Delan wants to save their friend, Dereni Delan .

Ethar Delan saves Dereni Delan's life.

Dereni Delan makes a dramatic recovery!

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan suffers a terrible injury while working as Monster-Hunter .

Dereni Delan mourns the loss of their close friend, Ethar Delan .

Dereni Delan is seriously injured during their work as Monster-Hunter .

The town of Saltland has 2 new healers to keep the population healthy and thriving.

Dereni Delan 's health is waning.

Finaren Ioigeth vows to do everything in their power for their suffering friend, Dereni Delan .

Finaren Ioigeth 's efforts are rewarded when Dereni Delan makes a full recovery.

Dereni Delan beats the odds and survives!

Dereni Delan is seriously injured during their work as Monster-Hunter .

Dereni Delan is seriously injured during their work as Monster-Hunter .

Dereni Delan is seriously injured during their work as Monster-Hunter .

Dereni Delan sharpens their weapons, trying to stay ready for the next creature attack.

Dereni Delan 's health is waning.

Finaren Ioigeth wants to save their friend, Dereni Delan .
Finaren Ioigeth saves Dereni Delan's life.
Dereni Delan makes a dramatic recovery!

Dereni Delan is seriously injured during their work as Monster-Hunter .
Dereni Delan suffers a terrible injury while working as Monster-Hunter .