UNIVERSITY OF CALIFORNIA
SANTA CRUZ

DESIGNING TO SUPPORT AUTHORSHIP PLAY IN EMERGENT NARRATIVE GAMES

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTATIONAL MEDIA

by

Max Kreminski

September 2020

The Thesis of Max Kreminski
is approved:

Professor Noah Wardrip-Fruin, Chair

Professor Michael Mateas

Quentin Williams
Interim Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Designing to Support Authorship Play in Emergent Narrative Games

by

Max Kreminski

Prior interactive narrative research has extensively investigated the question of how game designers can tell stories through games, but little investigation has been done of how and why players use games to tell stories of their own. This thesis investigates player storytelling practices around emergent narrative games as a form of *authorship play*. First, we examine existing play practices centered on the authorship of written stories based on play experiences. We find that some players deliberately seek out emergent narrative games as storytelling partners, and argue that this is due in part to the creativity support features common to several games within this genre, which can help players overcome certain barriers to creativity—including fear of the blank canvas, fear of judgment, writer's block, and perfectionism. Second, we present *Why Are We Like This?* (*WAWLT*), a mixed-initiative co-creative storytelling game based on design insights gleaned from this analysis. We discuss the AI architecture of *WAWLT*, focusing especially on how it makes use of *story sifting* and *social simulation* technologies to provide players with creativity support, and present the results of preliminary playtesting. Finally, we conclude with some brief discussion of the implications of our work in this design space, including our priorities for future work.

To the wildcats, who run the UC

Acknowledgments

The text of this thesis includes reprints of the following previously published material:

Max Kreminski and Noah Wardrip-Fruin. 2019. Generative Games as Storytelling Partners. In *Proceedings of the Fourteenth International Conference on the Foundations of Digital Games*.

Max Kreminski, Melanie Dickinson, Michael Mateas, and Noah Wardrip-Fruin. 2020. *Why Are We Like This?*: The AI Architecture of a Co-Creative Storytelling Game. In *Proceedings of the Fifteenth International Conference on the Foundations of Digital Games*.

The co-authors Noah Wardrip-Fruin and Michael Mateas directed and supervised the research which forms the basis for this thesis. The co-author Melanie Dickinson was the co-designer and co-developer of *Why Are We Like This?*, and created the *Why Are We Like This?* system diagram featured in chapter 3.

Chapter 1

Introduction

Interactive and generative narrative systems can be placed along a spectrum, with *strong story* systems at one end and *strong autonomy* systems at the other [38,49]. In strong story systems, a drama manager or similar computational agent attempts to direct the course of the story from the top down, while in strong autonomy systems, the story instead arises in a bottom-up fashion from the interactions of autonomous characters. The latter class of systems, sometimes known as *emergent narrative* systems [34], includes some of the great popular and commercial successes of interactive and generative narrative: games like *The Sims 2* [40] and *Dwarf Fortress* [17], both of which are widely known for the compelling emergent stories they produce. None of these systems, however, are reliably successful at producing compelling stories on their own. Instead, for their full narrative potential to be realized, they seem to require a human player to operate them.

In recent years, emergent narrative scholarship has been advanced considerably by James Ryan's work, which introduces what Ryan terms a *curationist* approach to interactive emergent narrative [52]. Under the banner of curationism, Ryan attempts to disentangle simulation technologies (which produce sequences of events, but not *stories* per se) from the *story sifter* systems that curate narratively potent events from the "raw material" of simulation and the *narrativizer* systems that transform these "nuggets" of narrative material into fully realized stories. In so doing, Ryan proposes that the pipeline by which scattered simulation events are first organized, then sifted, and finally assembled into stories contains a number of distinct roles, each of which can be played by either a computational system or a human interactor.

Simultaneously, Mirjam Eladhari has recently called for the interactive digital narrative (IDN) research community to pay closer attention to the *retellings* that players produce of their experiences with IDN systems [14]. Citing *Sims* player stories as an example, Eladhari argues that the existence of retellings for a particular IDN system can serve as evidence of the system's success at producing a compelling narrative experience for the player. Larsen et al. extend this analysis of retellings by drawing a distinction between the *afterstory*, or the "virtual" narrative experience that exists in the human interactor's mind after play, and the retelling proper, which is based on a further refinement of this afterstory [31]. Much like Ryan's work, this nascent body of theory on retellings and afterstories attempts to disentangle several distinct stages that may occur in the process by which events in a computational system are transformed into narrative—stages that past work has often tended to conflate.

These parallel strands of work both call attention, in different ways, to the complexity of narrative production in emergent narrative systems. To unite these strands of work, we believe it is necessary to reframe the role of the player in interactive emergent narrative. Drawing on both the theory of curationist emergent narrative and the theory of retellings, we assert that *emergent narrative* is in some ways a misnomer: the narratively compelling player stories for which emergent narrative systems are known do not merely *emerge*, but are in fact actively constructed by a human interactor in dialogue with a computational system. As such, we believe that it is both valuable and necessary to frame some forms of player interaction with interactive emergent narrative systems as a kind of *shared authorship* [55] or *mixed-initiative co-creativity* [32], in which players actively seek out these systems as storytelling partners with the end goal of crafting a compelling story in mind. We refer to these play practices as a form of *authorship play*.

The recognition of authorship play in interactive emergent narrative games and systems raises two key questions:

1. Why do players seek out these systems in particular as storytelling partners, even when the creators of these systems didn't initially envision them primarily as storytelling partners? What about the design of these systems leads players to use them for authorship play?

2. How could we build on existing scholarship in interactive emergent narrative technology to better support authorship-oriented play experiences?

In this thesis, we seek to provide preliminary answers to these questions. First, we examine existing practices of play in interactive emergent narrative games to show that some players seek these games out primarily as storytelling partners. Drawing on the theory of *creativity support tools* [60], we suggest that players may find that their creativity is supported by the way that these games, through several design elements common to this game genre, mitigate a number of well-documented barriers to creativity that may keep players from engaging in creative activity except when scaffolding is provided. We then present and discuss the design and technical dimensions of *Why Are We Like This?* (*WAWLT*), an emergent narrative game explicitly intended to serve as a storytelling partner for the player. The design of *WAWLT* was informed by insights gleaned from our analysis of existing authorship play practices, and *WAWLT* includes a number of features intended to go beyond existing emergent narrative games in providing first-class support for narrative authorship as a mode of play.

Chapter 2

Generative Games as Storytelling Partners

2.1 Background

There is a close relationship between creativity and play. Nevertheless, expressing one's self creatively is hard, and can remain difficult even in explicitly playful contexts. Many internal barriers can interfere with the creative mindset, and players who are unused to thinking of themselves as creative may balk at the prospect of game mechanics that force them to create as part of play. Specific design strategies may need to be employed to get and keep these players in a mindset where they are willing and able to express themselves creatively.

The idea that games and other playful interactive systems can serve as sites of creative self-expression is not a new one. In particular, several distinct strands of thought have emerged to offer approaches to the design of playful interactive systems intended to enable creativity, all of which offer useful perspectives on the problem of encouraging players to create as part of play.

Yannakakis and Liapis [32] introduce a new subfield, termed *mixed-initiative co-creativity*, at the intersection of digital creativity support tools and computational creativity. Mixed-initiative co-creativity research concerns itself with artificially intelligent systems that are intended to collaborate with humans on creative tasks, exercising somewhat more control over the output of the creative process than mere tools but still allowing human creators to retain a good deal of control over the output as well.

Samuel's dissertation [55] discusses the notion of *shared authorship* in interactive narrative play experiences, especially focusing on games and other interactive experiences that aim to make the player feel as though they are producing a story or other "narrative artifact" in collaboration with the game or system.

Finally, and perhaps most importantly in the context of this chapter, Compton and Mateas [9] introduce the idea of *casual creators*: a particular class of digital creativity support tools that blur the lines between tools and toys and are intended to support casual use. The chapter makes a distinction between *goal-directed* creativity, in which the individual engaging in the creative activity is attempting to accomplish a particular goal or has a particular outcome in mind, and *autotelic* creativity, in which the individual engaging in the creative activity is doing so primarily for enjoyment of the creative process rather than out of the desire to achieve a specific outcome. Casual creators are intended to support the latter kind of creativity.

Of these three approaches, only the shared authorship approach concerns itself directly with games as such. Nevertheless, although games are rarely positioned explicitly as creativity support tools, many players continue to make use of games for the purpose of creating things—sometimes even with creation as a primary goal. Therefore, in our view, it might well be worthwhile to evaluate games that players seek out and use for creation as if they are creativity support tools.

In this chapter, we first catalogue four potential barriers to creativity that may discourage creative expression by players or get in the way of the player's creative process. We then narrow our focus to a particular form of player creativity, which we describe as *dialogic retelling* of play experiences, and present several examples of successful dialogic retellings. We extract common design elements from the games that were used to produce these retellings and show how these design elements may help players circumvent the barriers to creativity that we discuss. Finally, we draw a connection between these extracted design elements and the key identifying elements of "gardening games," with the goal of showing that this particular class of generative games may offer a new perspective on how to facilitate player creativity through design [28].

2.2 Barriers to Creativity

The process of creative self-expression can be both enjoyable and beneficial to those who undertake it. Therefore, we want to make games and playful systems that enable and encourage players to express themselves creatively. However, getting and keeping players in a state of mind where they are willing to be creative can be difficult. The very idea of creativity can be intimidating. Many people are unused to thinking of themselves as creative, and may conceive of an activity that they consider creative (such as writing, drawing, or music-making) to be the exclusive domain of the naturally talented, rather than a set of acquired skills that they themselves can learn. Honest creative self-expression requires a degree of vulnerability, leaving many people hesitant to perform creative activities when they feel that others may be watching and judging them for the creative decisions they make. A blank canvas or an empty page can strike terror into the hearts of even the most experienced artists and writers; similarly, if a game hands the player a blank canvas, a set of tools, and instructions to create something, the player may well freeze up immediately due to their inability to answer the question of what they should create. And finally, even if the creative process goes well at first, "writer's block" can set in at any time, leaving players stuck at a point where they don't know how to proceed. To deal with these problems, games and playful systems that want players to express themselves creatively need to find ways to lower the perceived stakes of participating in creative activities.

2.2.1 Fear of the Blank Canvas

One common phenomenon that may inhibit creativity is the sense of fear or intimidation that many creators report when first faced with a blank canvas or blank page (either literal or metaphorical) at the start of the creative process. It is often said that "constraints breed creativity," and the blank canvas represents a highly unconstrained state; this lack of constraints may leave creators paralyzed by uncertainty regarding where and how to begin.

To alleviate the fear of the blank canvas, Victor [67] has stressed the importance of making it possible for users to "create by reacting," pointing out that many creators do not begin the creative process with a fully formed idea of what they want to create already present in their heads. Instead, they frequently begin by doing their chosen medium's equivalent of "pushing paint around on the canvas" or "noodling around" on a musical instrument, taking any steps necessary to get past the blank canvas and reach a point at which they can begin to create by reacting to and adjusting something external they can perceive. In Victor's words,

> An essential aspect of a painter's canvas and a musical instrument is the immediacy with which the artist gets *something there to react to.* A canvas or sketchbook serves as an "external imagination", where an artist can grow an idea from birth to maturity by continuously *reacting to what's in front of him.*

In the context of casual creators, Compton and Mateas [9] have also proposed "no blank canvas" as a design pattern with the specific intent of avoiding or mitigating the effects of this barrier.

2.2.2 Fear of Judgment

Another creativity-inhibiting phenomenon that many creators report struggling with involves the fear of criticism, judgment, or other negative assessment of the things they create. Fear of being judged for the things they create may leave creators hesitant to be as expressive, vulnerable or original in their work as they would like.

In some cases, tools intended to provide their users with creativity support end up exacerbating the fear of judgment through their attempts to provide users with feedback on their designs. Cross's early experiments in the simulation of computer-aided design [10], in which one human participant played the role of an architect and another played the role of an artificially intelligent architectural design tool, suggest a potential reason for this phenomenon. Cross tested two scenarios, which he described as the "forward" and "reverse" scenario; in the "forward" scenario, the architect was responsible for creating designs, while the computer critiqued and gave feedback on how to improve these designs. In this scenario, the architects often found the collaboration to be both difficult and stressful, even if they appreciated the feedback on their designs. Meanwhile, in the "reverse" scenario (in which the "computer" participant was responsible for generating possibilities and the "human" for modifying, critiquing and refining these possibilities), the architects were much less likely to report the activity as stressful and much more likely to describe it as easy, enjoyable, or even fun.

Compton and Mateas [9] address the fear of judgment and the way that tool-provided feedback may aggravate that fear by suggesting "entertaining evaluations" as a design pattern for casual creators. One example of this can be seen in the "abstract generative art game" *BECOME A GREAT ARTIST IN JUST 10 SECONDS*, which juxtaposes the player's glitch-art creation with a classic masterpiece painting and rates the player's work by percentage similarity to the obviously unattainable goal state. Paradoxically, by presenting players with a goal state that clearly cannot be reached using the tools provided by the game, *GREAT ARTIST* may actually assist players in overcoming the fear of judgment—partly by allowing them to blame their "failure" to reach the goal state on the limited tools with which they have been provided rather than on their own artistic vision, and partly by framing the process of judgment itself as playful and inherently ridiculous rather than a true arbiter of what constitutes great art.

2.2.3 Writer's Block

Even when the creative process is going well, it is nevertheless quite possible to suddenly find one's self at an impasse, unable to think of what step one should take next in order to continue the process. This phenomenon, commonly known by practitioners as "writer's block" or "artist's block," can be substantially disconcerting when it occurs, and strategies for dealing with it are a frequent topic of discussion in practitioner-focused books about, for instance, "how to write." Writer Anne Lamott, for instance, dedicates a whole chapter of her book *Bird by Bird: Some Instructions on Writing and Life* to the subject of writer's block [30]. She characterizes the subjective experience of suddenly becoming creatively blocked as follows:

> A blissfully productive manic stage may come to a screeching halt, and all of a sudden you realize you're Wile E. Coyote and you've run off the cliff and are a second away from having to look down.

Evidently, strategies for avoiding or mitigating the effects of this condition are much to be desired where they can be found.

2.2.4 Perfectionism

One final creativity-inhibiting phenomenon commonly described by creators is perfectionism, in which a creator feels that imperfections or flaws in their work must not be tolerated. This can lead them to avoid ever finishing or publishing projects, instead falling into a state of perpetual revision in which any imperfection "must be" corrected before the work can be presented to an audience or considered complete. It can also lead creators to frequently restart from the beginning of a project, rather than "play out" to completion an iteration of their vision that they know to be flawed. Under some conditions it can even manifest as a form of total creative paralysis that prevents any attempt to begin work.

As with writer's block, strategies for overcoming perfectionism are much in demand among practitioners. Natalie Goldberg, another writer, specifically instructs aspiring writers to set their expectations for their own work as low as possible in order to avoid paralysis due to perfectionism:

> Sit down with the least expectation of yourself; say, "I am free to write the worst junk in the world." [...] I've had students who said they decided they were going to write the great American novel and haven't written a line since. [19]

Likewise, as with writer's block, design strategies that can help would-be creators mitigate the effects of perfectionism are therefore highly desirable.

2.3 Which "Player Creativity"?

Our objective is to study how games can facilitate player creativity. Player creativity, however, can take many forms. What do we mean when we talk about "player creativity"? What specific kinds of creative activity do we aim to concern ourselves with in this chapter?

First and foremost, we wish to investigate cases in which the player's creative process is shaped by engagement with the unique formal properties of digital games. With this restriction in mind, we do not wish to highlight the form of player creativity seen in, for example, machinima, in which the player uses the game as a kind of canvas on which to paint or an engine for generating illustrations to accompany a pre-authored script. Instead, we prefer to focus on a kind of creativity in which the game's systems are allowed to push back against, resist, react to, or redirect in unexpected directions the player's creative intent, producing a kind of adversarial yet generative relationship between the player and the system. In this sense, our focus is on the form of player creativity that has been identified as "co-creativity," rather than on creative practices that use games (or creative tools provided within games) as something like a traditional "inert" or non-reactive creative medium.

Moreover, we also wish to concern ourselves specifically with cases in which the player's creative process results in the construction of some concrete artifact. Narrowing our focus to this form of player creativity makes it much easier to frame our evaluation of a game's "creative potential" in terms of existing creativity support research. Although creativity may indeed be employed in, for instance, the process by which players generate novel puzzle solutions in puzzle games, these forms of player creativity do not lead to the production of concrete artifacts that we can evaluate, and so we choose to exclude them from the scope of this chapter.

Taken together, these two restrictions on the kinds of player creative activity we are interested in studying naturally lead us to examine one particular form of player creative activity as a focus for our investigation: namely, the practice of *retelling*.

2.4 Retellings as Co-Created Artifacts

Retellings, as defined by Eladhari, are narrative artifacts created by players as recountings of their play experiences [14]. Eladhari suggests that the existence of retellings of play experiences within a particular game or interactive narrative system may be taken as evidence that players found the experience compelling, and—by extension—as an indicator that the game or system in question is somehow successful. Moreover, Eladhari also proposes the analysis of corpora of player retellings as a way to understand and critique interactive narrative systems.

We propose to apply the framework of co-creativity to the problem of understanding how games support player creativity. From this perspective, the practice of retelling represents a form of player creative activity that produces concrete artifacts (the retellings themselves) in collaboration with the game. In retelling, the game and the player essentially work together as storytelling partners to produce a narrative that the player would not have been likely to produce alone. By examining the narrative artifacts produced by this co-creative process, we hope to find evidence of how the game's design may have had an impact on the co-creative process by which the artifacts were produced—or, in other words, of how the game may have acted to facilitate or support player creativity.

For our purposes, we will refer to the kinds of retellings in which we are interested as *dialogic retellings*. A retelling is dialogic when the player who creates it is meaningfully in dialogue with the game's systems during the creative process: rather than acting upon the game to bring it into line with a preexisting creative vision devised primarily outside the play experience itself, the player accepts creative input from the game in real time, allowing its systems to change the direction of the story at will and even directly contravene the player's creative intent. We draw a contrast between these and *monologic retellings*, in which the player more often ignores, rejects, discards, or overwrites the game's creative input than accepts it, treating the game less like a creative partner and more like a kind of raw material to shape according to their externally formed creative will.

One good example of monologic retelling can be found in some of the *Cities: Skylines* videos by donoteat01, who uses the game as a backdrop to tell stories about urbanism [11]. In this case, the game is used more as a subordinate generator of illustrative imagery than a source of novel creative input; buildings, roads and the like are placed, manipulated and destroyed as needed to produce appropriate accompanying imagery for the creator's voice-over narration, largely without regard for their role in the game's ongoing process of simulation. Although the creator is indeed relating (or retelling) the events of a play experience, the game's systems are not at any point allowed to meaningfully interfere with the preordained story the creator intended from the beginning to tell.

2.5 Sample Retellings

Here we present three examples of dialogic retellings. We selected these partly due to their prominence (all three are relatively well known) and partly due to the way in which they each demonstrate evidence that the player did not merely use the game as an inert stage upon which to act out a predefined script, but allowed the game's systems to exercise meaningful creative input and influence or even outright determine the overall direction of the story. It is our hope that, by examining these retellings in parallel, other common elements will emerge, potentially showing how the design of the games in question facilitated the creative process of the players who authored these retellings.

2.5.1 *Matul Remrit*

Matul Remrit [64] is a *Dwarf Fortress* retelling that follows the exploits of a small band of dwarves as they attempt to found a stable, self-sustaining settlement (the titular Matul Remrit) in a hostile, uncaring land. Most of the story is told from the perspective of the individual procedurally generated dwarf characters; the events of each in-game month are narrated by a particular character, in the form of a succession of entries from that character's diary. Unusually for a retelling of a play experience in a single-player game, it represents a collaborative effort between four individuals: writer Kevin Snow; illustrator George Kavallines; musician Thomas Ferkol, who created the soundtrack for several video interludes; and editor Andi McClure. Among the community of *Dwarf Fortress* players and fans, who have collectively generated at least hundreds of published retellings, *Matul Remrit* is one of the best-known; the DFStories.com website, a repository of *Dwarf Fortress* retellings, lists it as a "highlight from the Hall of Legends" [12].

Here it is worth pointing out that, from a high-level structural perspective, the vast majority of *Dwarf Fortress* retellings are essentially identical to one another. The game's systems naturally lend themselves to the construction of stories in which a small and ragtag band of founders construct a settlement, are initially successful despite minor setbacks, gradually build up the size and complexity of their settlement, inevitably succumb to some combination of hubris and natural disaster, and suddenly disintegrate, with the last survivors of the wreckage perishing in various gruesome ways. It could fairly be said that, if you've heard one *Dwarf Fortress* story, you've essentially heard them all. Nevertheless, players keep returning to *Dwarf Fortress* with the goal of constructing retellings in mind, suggesting that something other than plot-level originality motivates their desire to work with this game in particular as a creative partner.

The story of *Matul Remrit* proceeds in largely the same way as any other *Dwarf Fortress* story. Attacks by groups of elves soon shape up to be the biggest obstacle to the settlement's continued survival and growth; eventually, their numbers dwindling and facing an attack by an elven force of overwhelming size, the few remaining dwarves sacrifice themselves and the settlement to have their revenge, destroying not only themselves but also the invading force.

Throughout the story, the game's generative processes keep things moving forward by continually supplying the characters with new motivations, goals, and problems. The propulsive effect of the real-time processes that control the game world's calendar, the schedule of elf and monster attacks, and so on may have helped to prevent writer's block from setting in at any point in the (relatively lengthy) story: even when uncertain of how to proceed next, the creative team could always elect to let the simulation keep running without any specific guidance, which would inevitably generate a new scenario that had to be resolved in short order.

Moreover, the bounded nature of the player's access to the game's generative processes ensures that the player cannot just freely "pull on" the generator until they obtain the desired results. The game chooses what to generate and when to generate it without input from the player, and in so doing encourages you to remain where you are and "play out" the situation at hand (even if it seems that the current situation will inevitably result in the failure of your settlement) rather than abandon your settlement to seek a fresh start elsewhere in the world.

2.5.2 *Alice and Kev*

Alice and Kev [5] is a *Sims 3* retelling by Robin Burkshaw that follows two "homeless" Sims, a father (Kev) and a daughter (Alice), as they attempt to survive without access to either the ordinary features of an in-game home or forms of money-making that the author considers to be "unrealistically easy." The author makes a point of remaining faithful to the events of their actual play sessions, stating that their goal is to tell the story "with the minimum of embellishment," and repeatedly reaffirms their commitment to stepping back and allowing the Sims to do whatever it is that they want to do on a fairly regular basis. As a result, the overall direction of the story is evidently guided as much or more by *The Sims 3*'s systems than by the retelling's human author.

In *Alice and Kev*, as in *Matul Remrit*, the game's generative processes continually supply the characters with new impulses to act on. One especially powerful moment in the story takes place when Alice, having finally managed to secure a job, receives her first ever paycheck. Gripped by a momentary impulse, rather than using it to begin improving her own living situation, she instead elects to donate the entirety of her hard-earned money to charity [6]. At this point, the author of the story explicitly expresses reluctance to allow the simulation to dictate what happens next, and even considers intervening to prevent it. In this moment, the game's systems are in direct conflict with the player's creative intent—and yet, reaffirming once again a commitment to allowing Alice and Kev to live their own lives, the author eventually allows this scene to play out as the game has determined it should.

2.5.3 *Pro Vercelli*

Pro Vercelli [44] is a *Football Manager 2009* retelling by sportswriter Brian Phillips in which the author catalogues his efforts to restore the titular Pro Vercelli soccer team to greatness. The team—today relegated, in both the game's simulation and the real world, to relative obscurity in spite of its run of championship successes in the early 1900s—makes for a natural underdog, and the author further embellishes his retelling of the team's story with in-character sections narrated from the point of view of various procedurally generated characters, mostly players and staff for the team. The *Football Manager* games are unusual among sports games in that they do not allow the human player to directly control their team's actions during a simulated match; instead, the player remains solidly in the role of the team's manager, with match outcomes being determined semi-randomly based on the statistics and characteristics of the simulated players on the competing teams.

As with *Dwarf Fortress*, *Football Manager 2009* stories have a tendency to resemble one another quite closely. Successful playthroughs typically follow the player's selected team from whatever initial status they occupy to a position of relative security as one of the top teams in the game's simulated world, capable of reliably winning championships and standing against the other powerhouse teams. As might be expected, *Pro Vercelli* deviates little from this formula.

One remarkable moment in *Pro Vercelli* takes place toward the beginning of the series, as the author attempts to take advantage of his first opportunity to make trades for players with other teams [43]. Due to a misunderstanding of a certain nuance of the game's budgeting rules, he finds himself engineering a deal that he lacks the funds to complete—and discovering his error only once he has already committed to the first half of the trade. This leaves him unable to cleanly undo the trade, and forces him to scramble to make more trades in order to compensate for his mistake.

In the context of creativity support tools, it is often generally accepted that the user should always be freely permitted to cleanly undo their actions wherever possible. Here, however, it seems likely that the lack of any straightforward way to cleanly undo the mistake is the reason that the mistake made it into the story—and the reason that it is narrativized as a moment of characterization for the team's manager, rather than explained away as an extradiegetic lapse (or, more likely, omitted entirely). In this sense, by limiting or taking control away from the player, the game forces the player to incorporate imperfections into the story they are telling, resulting in what is arguably a more compelling narrative overall.

2.6 Design Elements that Facilitate Dialogic Retelling

The retellings highlighted here share several common elements. In all of these cases, the author of the retelling started out with little high-level sense of where the story was "meant to" go; in this sense, they were truly making it up as they went along in accordance with what happened in the simulated game world, rather than imposing their will on the simulation. In all of these cases, the author of the retelling sought out the game primarily in order to co-create a story with the game, rather than producing the retelling incidentally based on a play experience they originally pursued for some other reason entirely. In all of these cases, the story, world, or simulation naturally grew richer and more interesting over time. In all of these cases, the simulation had a level of autonomy; if allowed to keep running, things would keep happening even in the absence of input or intervention from the player. Finally, in each of these cases, the player's control over the simulation was limited, rather than total: the player could not elect to fully impose their will on the simulation even if they wanted to, and the simulation would regularly push back against, resist, subvert, or redirect their creative intent.

It is our belief that these similarities between the retellings themselves are due in large part to similar design elements shared by all of the games that were used to construct them. We present these common design elements here.

2.6.1 Generativity

First, and perhaps most obvious, of the common elements between the selected games is the presence of generativity. *Dwarf Fortress* is known for featuring a fully procedural world, and generative processes that introduce new situations (such as monster attacks, or the arrival of traveling caravans whose members could be persuaded to join the player's settlement) continue operating in the background as the player plays. *The Sims 3* is likewise deeply procedural, with generative processes finding their most prominent expression in the autonomous actions of both nominally player-controlled and non-player characters. And *Football Manager 2009* uses generative processes to periodically introduce new characters into the simulation, as well as to determine the outcome of matches between opposing teams.

Generativity plays several roles in the context of player creativity. First and foremost, generating an initial scenario helps to circumvent fear of the blank canvas by ensuring that players are never faced with a completely empty or unconstrained starting condition. Moreover, when generative processes are allowed to continue operating past the starting point of a playthrough, they can help to ensure that writer's block does not set in by continually providing the player with new problems to address and prompts to react or respond to. Finally, by giving players a way to disclaim design decisions (essentially by saying that "the game did that, not me"), it may help to diffuse the player's sense of responsibility for creative decisions and thereby mitigate their fear of being judged for the things they create.

2.6.2 Incrementality

Another key design element shared by all three games examined here is the presence of a simulation that can continue to run even in the absence of significant new input from the player. In *Dwarf Fortress* and *The Sims 3* alike, simply allowing the game to continue running without entering any input will cause the game world to continue growing and changing in real time; in *Football Manager 2009*, the player's input is required to advance time, but on most timesteps, the player is generally free to allow the simulation to "stay the course," i.e., to advance time without making any significant changes. This ability of the simulation to proceed without substantial player input ensures that it is nearly impossible for the player to remain truly blocked indefinitely. Even in a situation where the player has no idea how to proceed, merely allowing the simulation to continue running will inevitably lead to the introduction of new prompts and possibilities before long.

On the subject of dealing with writer's block, Raymond Chandler—a writer of hard-boiled detective fiction—is alleged to have once given the following piece of oft-repeated advice: "When in doubt, have a man come through a door with a gun in his hand." Each of the games examined here has at least one process that periodically and automatically performs an equivalent function. Matul Remrit was plagued by the periodic arrival of elven raiding parties; Alice and Kev were both moved by sudden impulses to take actions the player would not have been likely to deliberately suggest; and the cyclical nature of the professional soccer calendar in *Football Manager* ensured that Pro Vercelli would always find itself facing an opportunity to substantially change its lineup once per in-game year. In each case, by periodically injecting novelty into the simulation in the form of formulaic but reliably disruptive interventions, incremental processes helped to keep the highlighted retellings moving along, even (and especially) when they might otherwise have slowed down or ground to a halt.

2.6.3 Boundedness

A third shared design element between all of the games examined here is *boundedness*: specifically, boundedness of access to the game's generative processes by the player once a playthrough has begun. Unlike in some games (such as *Minecraft*) where the generator is invoked in response to the player's actions, for instance to generate new terrain for the player to explore whenever they travel to the edge of the currently generated area, all of the games examined here place the ability to invoke the generator deliberately out of the player's hands. In *The Sims 3*, Sims receive new urges largely independently of the player's actions; in *Dwarf Fortress*, the game chooses when and where to spawn enemies, travelers, and the like without any input from the player; and in *Football Manager*, the cyclical yearly calendar is responsible for determining when matches take place and when new procedurally generated characters will be made available.

In each case, preventing the player from deliberately invoking the game's generative processes once the game is in motion helps push back against the perfectionist tendency to restart at the first sign of trouble, encouraging players to "play out" the consequences of even problematic or undesirable events. The opportunities to which the player has access are limited; whether or not they like the results they get from acting on these opportunities, they are not permitted to freely re-roll the dice until they get a more desirable outcome.

2.6.4 Limited Player Control

One effect of the other design elements of the highlighted games is that player control over the outcome of their actions remains ultimately limited. Unlike in games that are traditionally seen as placing player creativity front and center, these games—due to their use of largely player-independent generative processes—tend to subvert or complicate the player's intent, rather than allowing events to play out exactly as the player intended. In other words, rather than using the game as a canvas upon which they are free to draw whatever they would like, players in these games tend to find themselves forced into a form of creative compromise with the systems at work.

Paradoxically, the ways in which these games tend to subvert or even outright frustrate the player's creative intent may in fact play a role in helping players overcome certain barriers to creativity. In *Pro Vercelli*, for instance, the way the author is essentially strong-armed by the game's lack of a clean undo feature into accepting and narrativizing an objectively suboptimal decision suggests that limiting the player's control may in some cases serve as an effective means of combatting perfectionism.

2.7 Gardening as a Mode of Play

The notion of gardening as a mode of creative play can be traced back to Wardrip-Fruin's *Expressive Processing* [68, p. 309–310], which likens the experience of playing *SimCity* to that of gardening. More recently, Kreminski and Wardrip-Fruin [28] have introduced the term *gardening games* to describe a class of PCG-based games [63] (in other words, games that feature player interaction with generative systems as a core element of gameplay) whose play experiences similarly resemble gardening. Kreminski and Wardrip-Fruin describe several key features of gardening games, including generativity with bounded player access to the generator; the presence of a simulation or generative process that can continue running even in the absence of player input; default play patterns that tend to make the game world richer and more interesting over time, rather than to gradually drain or deplete it of its play potential; and limited player control in general. These features match up closely with the design patterns extracted from the games used to construct the three dialogic retellings highlighted in this chapter, suggesting that “gardening games” may be an apt term to adopt for the class of games that aim to facilitate player creativity using the means described here.

There is a tradition in computational design of using anthropomorphic roles such as “clerk,” “partner,” “wizard,” “surrogate,” and “accountant” to classify computational systems by the space they occupy in the design process [66]. In light of the way that players sometimes deliberately make use of “gardening games” like the ones described here for the purpose of constructing stories, it may be appropriate to extend this taxonomy with an additional non-anthropomorphic role for the computer: that of the garden, with the human collaborator in turn taking up the role of the gardener. The use of gardening as a metaphor for co-creative processes involving human collaboration with generative systems is also endorsed by generative music pioneer Brian Eno, who draws a distinction between two forms of composing: a more traditional style in which the composer’s role is seen as analogous to that of an architect, dictating every aspect of a piece of music from the top down, and a more modern co-creative style in which the composer takes on the role of a gardener whose purpose is to set up, tend, and curate the work produced by generative systems [15].

Although most games categorized as gardening games were not explicitly created with facilitating player creativity as a primary design goal, many of these games nevertheless seem to function well in this context. Players go about crafting retellings of their experiences in these games because they think there is something of value in doing so. In Eladhari’s terms, players recognize in these games a certain kind of “narrative potential”—or what could potentially be construed even more broadly as “co-creative potential"—and are drawn to it [14]. Thus, based on how players appropriate their play experiences in gardening games as raw material for dialogic retellings to a seemingly disproportionate extent, we think there is something important to be learned from these games about how to facilitate player creativity.

2.8 Discussion

The games examined here are far from the only games for which notable individual dialogic retellings, or sizable corpuses of dialogic retellings, exist. Many strategy game player communities, for instance, have long-standing traditions of compiling "after-action reports"—many of which focus on blow-by-blow accounts of battles rather than narrative—for other players to read. It seems to lend credence to our suggestions here, however, that many of the other games for which large numbers of *narrative-focused* dialogic retellings exist feature similar design elements to the games we highlight. Narrative-focused after-action reports, for instance, enjoy particular popularity among players of the Paradox-published grand strategy games *Crusader Kings II* and *Stellaris*, both of which make unusually heavy use of generativity in comparison to other strategy games.

At the same time, many games that are often held out as particularly conducive to player creativity (such as *Minecraft*, to give one example) do not seem to have inspired similar traditions of dialogic retelling of play experiences. Although it is difficult to determine with any confidence that dialogic *Minecraft* retellings are objectively rare relative to the game's popularity, we can nevertheless hypothesize as to why *Minecraft* retellings seem to be monologic in nature (using *Minecraft* more as a medium to be bent to the storyteller's will than a creative partner) more often than not. We suggest that, although *Minecraft* makes extensive use of generativity to craft its procedural world, it employs generativity very differently than the games we highlight in this chapter. In particular, *Minecraft* employs generativity almost exclusively *reactively* (e.g., by generating more terrain in response to a player who walks to the edge of the already-generated space.) This is unlike the games we highlight in this chapter, which tend to employ generativity *proactively* to keep the narrative moving forward and head off writer's block before it has a chance to set in.

2.9 Conclusions

By examining three cases in which players made use of games as storytelling partners to craft dialogic retellings, we found evidence of common design elements between the games used to craft these retellings. We also found that these design elements, taken together, seem to provide assistance to players engaged in creative activity, potentially enabling them to mitigate the effects of or outright circumvent several identified barriers to creativity. These design elements correspond closely to the key features of "gardening games"—a class of generative games recently identified by Kreminski and Wardrip-Fruin—and their presence in a game may help attract players who seek to tell stories based on their play experiences to certain games over others.

It remains to be seen whether the design elements that this chapter proposes as conducive to player creativity in the construction of retellings can also be generalized to facilitate other kinds of player creativity. We would like to claim that the barriers to creativity addressed by these design elements are agnostic to the particular creative medium in which the player is working, and therefore that the design elements we highlight should apply equally well to facilitating forms of player creativity that are non-narrative in nature, but in practice, more evidence is needed to support this claim.

2.9.1 Future Work

Our findings here suggest several potential paths forward. First and foremost, in order to test the effectiveness of the design patterns highlighted in this chapter at facilitating player creativity, we intend to construct a game that makes use of these design patterns and evaluate its effectiveness as a storytelling partner.

This chapter largely attempted to analyze the design patterns that games use to facilitate creativity from a distance, without talking to the players who created the dialogic retellings we chose to highlight. Going forward, it may be worthwhile to interview players who use games as creative partners about their experiences in order to better understand what appeals to them about the experience of working with the particular games they select.

Finally, by more closely examining individual games that make use of the design patterns described here, we may be able to identify additional patterns that these games use to faciliate player creativity—some of which may not be shared between all of the games examined, but which can nevertheless be incorporated into a wider catalogue of the tools games use to facilitate creative activity by players. One potentially promising pattern found in certain strategy games, including *Crusader Kings II* (which makes use of many of the other patterns discussed here), involves the automatic maintenance of a player-accessible log or timeline of notable game events, which the player can then use as a reference or piece of raw material in their efforts to construct dialogic retellings of their play experiences.

Chapter 3

Why Are We Like This?

3.1 Background

Why Are We Like This? (*WAWLT*) is an AI-supported digital storytelling game, previously reported on in [23] and [24]. In *WAWLT*, one or more players (ideally two) work to construct a story in a pastiche of the cozy mystery genre, supported by an AI system that serves to provide players with inspiration and keep the story moving forward, even when players are unsure what should happen next.

WAWLT's setting is grounded in the familiar context of academic research, dealing with a small community of researchers who are temporarily stranded during a symposium at a remote venue. Over the course of a play session, because different characters have access to different *story sifting patterns* that they use to make narrative sense of the world, characters end up telling themselves different stories about the events that have transpired in the world so far – and, therefore, end up acting in conflicting ways

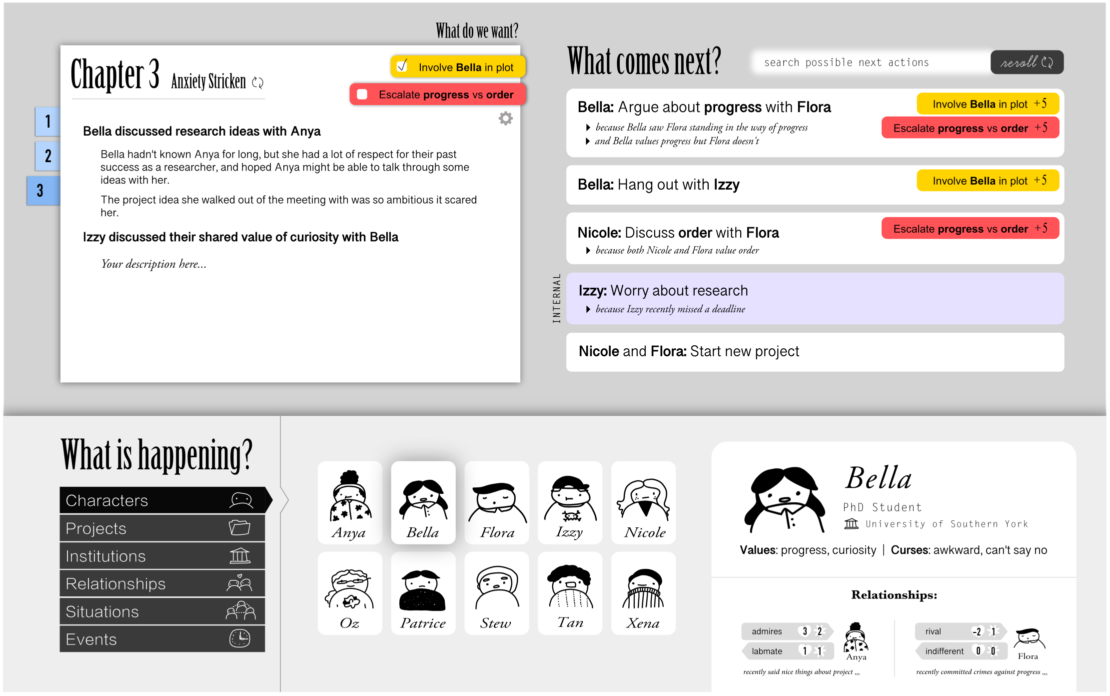

Figure 3.1: The main *WAWLT* interface, with the running transcript of the story so far in the upper left, action suggestions in the upper right, and the storyworld investigator on the bottom, focusing on a specific simulated character.

based on their conflicting evaluations of the same evidence. The closed environment of the symposium venue acts as a pressure cooker, exacerbating initially minor tensions between characters until a variety of plausible motivations exist for characters to commit severe crimes.

WAWLT represents an example of *AI-based game design* [13] inspired by the study of existing player storytelling practices [14, 27] in simulation-driven games like *Dwarf Fortress* [17], *The Sims 2* [40], and *Crusader Kings II* [21]. We particularly set out to provide creativity support features that would help players overcome four major barriers to creativity documented in [29]: fear of the blank canvas; fear of judgment; writer's block; and perfectionism. Further design inspiration was drawn from tabletop storytelling games like *Microscope* [50] and *The Quiet Year* [3], and from the AI-augmented improvisational theater experience *Bad News* [58], of which we wanted to create a "home version" that a small number of amateur players could set up and play through on their own without any special training.

The main contribution of this chapter is a *computational caricature* [62] of an AI architecture that enables a mixed-initiative co-creative storytelling play experience. Like other caricatures, our architecture is intended first and foremost to clearly present a central claim about the design space of co-creative storytelling games: that machines should support player storytelling practices by providing players with intelligent plot direction suggestions, drawn from an ongoing social simulation and guided by player utterances in a machine-understandable *intent language*. Secondarily, we also aim to provide readers with a small number of reusable abstractions that might be generally applicable in other, similar systems.

In the remainder of this chapter, we first describe our AI architecture at a high level, through the lens of an idealized vision of a typical *WAWLT* play experience. We then describe the key subsystems in our overall AI architecture and the roles of these subsystems in producing the desired player experience of co-creative storytelling. Throughout these sections, for illustrative purposes, we draw comparisons between our AI architecture and several other simulation-based storytelling architectures, including Ensemble [57], Versu [16], Ceptre [36], and Tale-Spin [41]. Finally, we discuss the results of early playtesting and possible directions for future development.

3.2 Related Work

The current iteration of *WAWLT* was conceived first and foremost as an analogue to tabletop storytelling games, which attempt to provide scaffolding for a collaborative storytelling process between a group of human players. *Microscope* [50] in particular offers a wide variety of creativity support features, particularly the *palette* – a way for players to collectively negotiate what they do and do not want to see in the story – and a recursive story structure that enables players to "dive deeper into" any part of the story that they would like to further flesh out. *The Quiet Year* [3] involves the collective production through play of a physical artifact, namely a map of the world that the players have created, which players can take with them after play as a reminder of the play session. And *Archives of the Sky* [47] provides mechanisms for structuring character conflict around *values* held both by individual player characters and the larger society in which they exist. All of these features have directly inspired design elements in the current version of *WAWLT*.

WAWLT is a *mixed-initiative co-creative* system [32], and can be viewed as a *casual creator* [9] for cozy mystery stories set in a particular context. Other casual creators for storytelling, such as *Writing Buddy* [56], and other mixed-initiative co-creative systems intended to be used in a storytelling context, such as *Mimisbrunnur* [65], have provided valuable design inspiration for *WAWLT*, but have not fully embraced the use of a fine-grained simulated storyworld in the way that we aim to here.

The same is true of co-creative writing processes driven by language models, such as the Botnik *Predictive Writer* app [4], the *Creative Help* system [51], and the writing practices described by Manjavacas et al. [35] and Sloan [61]. Moreover, language model-based systems are particularly flawed from a creativity support perspective due to their lack of an explicit world model. Because of this lack, they frequently go off track or make suggestions that clearly contradict previous statements, forcing users to spend time and energy repeatedly reminding them of established facts via prompting techniques. This distracts from the useful creativity support features they provide (suggestions about "what happens next") and exacerbates the problem of maintaining consistency, which even experienced authors may already find difficult on their own.

WAWLT is built around *story sifting* in both its implementation and its design, making central use of the story sifting and simulation engine Felt [25]. Story sifting approaches to emergent narrative attempt to address the challenges of narrative generation through curation: by allowing a simulated storyworld to run, generating a massive chronicle of mostly-uninteresting simulated events, and then searching within this chronicle for patterns of narratively compelling events, it is possible to provide players with compelling stories or microstories without baking knowledge of how to tell a compelling story directly into the simulation engine itself. Story sifting, originally known as "story recognition", was first proposed as an open design challenge for interactive emergent narrative by Ryan et al. [53], and further refined by Ryan in his dissertation, Curating Simulated Storyworlds [52].

Several existing play experiences make use of story sifting technology in some way, but none of them attempt to center story sifting as a player-facing game mechanic as we aspire to in *WAWLT*. *Dwarf Grandpa* [18] runs sifting retroactively on the history of a Dwarf Fortress world to pull out and highlight the stories of certain kinds of vampires. *Bad News* [58], an interactive theater experience with both human and computational components, involves a process of live sifting in which a human "wizard" (one of the performers, rather than a member of the audience) interacts with a Python console to pull out interesting information about the simulated storyworld in which the story is set and feeds this information in real time to the human actor portraying the simulated characters. One of our own past prototypes [22] also makes use of story sifting, albeit without centering it to the extent that we attempt to here.

3.3 Architecture

To give a high-level overview of the *WAWLT* AI architecture and how the various subsystems fit together, we will first present a brief walkthrough of an idealized *WAWLT* play session. We will then elaborate on each of the indivdual subsystems in sections 3.3.1-3.3.7.

At the start of the play session, two players sit down and generate a new *WAWLT* scenario. The game generates a storyworld state database containing an initial cast of characters and institutions and some basic relationships between them. It then performs *backstory simulation* to quickly generate a history for the community as a whole. This setup process is described in section 3.3.1.

Control is then handed off to the players, who are prompted to choose some

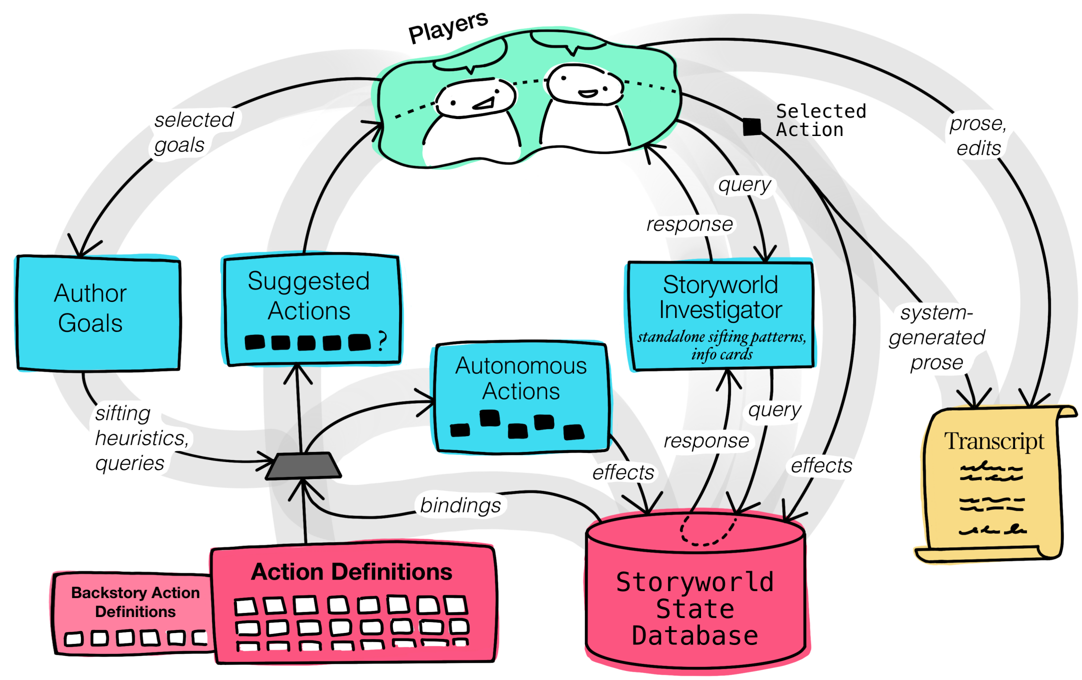

Figure 3.2: An overall system diagram of *WAWLT*, showing the important modules and data flows. Pink subsystems (action definitions and the storyworld state database) consist of inert data; blue subsystems (author goals, suggested actions, autonomous actions, and the storyworld investigator) act on this data; and the transcript emerges from player actions over the course of play. Subsystems depicted in this diagram are discussed in greater detail in sections 3.3.1-3.3.7.

author goals (discussed in section 3.3.3) and a subset of all the characters at the symposium to participate in the first *scene*. The players do not yet know anything about the history of the storyworld and the preexisting relationships between the characters, so they pick a couple of characters with interesting-sounding names and traits to participate in the first scene. They also select a single author goal: "cast suspicion on Vincent", one of the two characters they selected.

The system prompts the players with five different *action suggestions* (discussed in section 3.3.4), many of which seem likely to motivate Vincent to pursue revenge against another character. This is because the suggestions are guided by the author goals that the players have selected. The players deliberate for a bit about which of these actions to perform, and eventually select an action in which Mikayla, another character in the scene, makes a disparaging comment about Vincent's research. A terse system-generated description of this action is added to the *transcript* (discussed in section 3.3.7), with an editable text box below it in which the players are free to write their own more detailed description of this action. The action's *effects* (discussed in section 3.3.2) are also run, updating the storyworld state database.

The players decide they want to find out more about Mikayla and Vincent, so they use the *characters* tab of the *storyworld investigator* (discussed in section 3.3.6) to look up Mikayla's and Vincent's *information cards.* They discover that Mikayla and Vincent are both doctoral students in the same lab, advised by a third character, Lea. They also discover that Mikayla and Vincent had previously worked on a major project together, but eventually both left the project over personal differences. This information helps the players develop a clearer picture of the relationship between Mikayla and Vincent up until this point, and allows them to write dialogue in their description of the insult action that makes reference to these past events.

After selecting several more system-suggested actions for Mikayla and Vincent to perform within this scene, the players decide that they have accomplished their current author goal of casting suspicion on Vincent. They end the scene, and are prompted to pick characters and author goals for a new scene.

In the meantime, other characters not participating in the first scene have also performed a number of *autonomous actions* (discussed in section 3.3.5), guided partly by the players' author goals, but with a greater degree of randomness involved in action selection. The players use the storyworld investigator to see what other events have happened recently, and use this to guide their selection of participants and author goals for the second scene.

The players also disagree briefly about whether the next scene should focus on establishing a conflict between two *characters* (Alex and Rashida) or between two *values* (comfort and safety). Eventually, they choose to compromise, selecting three author goals: one goal that explicitly focuses action suggestions on the establishment of a values conflict, and two goals that instruct the system to involve both Alex and Rashida in the plot as much as possible.

Generally, as the players proceed from the start to the end of the play session, they gradually shift from spending most of their time investigating the history of the storyworld and opening up new loose ends to spending most of their time trying to pull threads together and bring the story to a satisfying conclusion. The choice of author goals from scene to scene closely follows this arc: early in play, many of the players' goals focus on escalating tension and introducing new points of contention between characters, but in the last few scenes, the players select goals that steer action suggestions toward reconciliation between characters instead.

The following subsections describe individual subsystems of the overall architecture in greater detail.

3.3.1 Storyworld State Database

The state of the *WAWLT* storyworld is stored in a DataScript [46] database managed by the Felt [25] story sifting and simulation engine. Storyworld entities represented in the database include:

- **Characters.** A character has a name; a *role*, representing their job (e.g., "PhD student" or "janitor"); two *values* drawn at random from a pool of ten; and several other personality traits, which restrict the actions this character is willing and able to perform.

- **Relationships.** A relationship entity stores one character's view of another, including both *impressions* formed of that character's actions and *role relationship* information (for instance, in the case of an advisor/student relationship or a marriage). It also contains a numeric *charge* value representing the source character's overall attitude toward the target, with positive values reflecting a positive attitude and vice versa. There are two relationship entities for every pair of characters, one pointing in each direction.

- **Impressions.** An impression represents a source character's evaluation of a target character based on a particular introspection event. It has a *score*, which is summed together with the scores of other impressions between the same two characters to produce an overall running evaluation of the target character by the source. It also has a pointer back to its *cause*: the event that led to its creation. A single relationship may only be defined by up to three positive and three negative impressions simultaneously. Stronger impressions tend to displace weaker ones over time.

- **Projects.** A project is an ongoing effort by one or more characters. It has a set of *contributors*, a name, a numerical *drama level* representing its troubledness, and some other detail information.
- **Institutions.** An institution is an organization with which characters can be affiliated. These are mostly used in generation of character backstories during backstory simulation.
- **Events.** An event is a record of an action performed by a character. It has at minimum an *event type* drawn from the action definition; an *actor*, the character who performed it; a short textual *description*; and a *timestamp*. It may also have a number of *tags*.

At the beginning of each play session, a cast of 10 characters is initialized with random names, roles, values, and personality traits. Each pair of characters is assigned a random relationship charge value in both directions, for a fully connected character affect graph. The established character roles are then used to probabilistically set role relationships between particular characters where appropriate ("works for", "advises", etc).

Backstory simulation using higher-level, lower-resolution action definitions is then run to flesh out character life history and the shared intellectual and social history of the community for several years prior to meeting at the symposium. This backstory simulation draws from a pool of special actions, not available during gameplay, that operate at a larger time scale. Examples of these actions include “graduate”, “join institution”, “write paper together”, “publish a controversial book”, “take industry position”, and “take leave to parent”. Such coarse-grained actions don’t make as much sense in the context of a scenario playing out over the course of a few days, but they help bootstrap rich social context quickly, and make the characters feel more real.

Backstory simulation is intended to mitigate the *fear of the blank canvas* [29] in players by providing an interesting situation from the start, with many sites of potential narrative interest for players to explore together and begin building their story from. The generated backstory can also be referenced and queried at any time via the storyworld investigator. Browsing backstory can be useful as inspiration for creating a shared understanding of a persistent character, for sources of possible motivation for surprising character actions, or for finding and fleshing out a character who is new to the story so far.

After initial setup, all subsequent simulation actions (both chosen by players and autonomously running in the background) draw from a finer-grained pool of action definitions that make sense to happen in the course of a few days, on location at the symposium. These actions operate on the same state database initialized by the backstory simulation, gradually evolving it over the course of play.

```
Felt.registerAction('worryAboutOthersProjectDrama', {
  tagline: '?n1: Notice that ?n2 is struggling with project "?projname"',
  where: [
    // there's an active project to which c2 is a contributor
    '?proj "state" "active"',
    '?proj "projectContributor" ?c2',
    // c1 is a character who likes c2
    '(likes ?c1 ?c2)',
    // three increaseProjectDrama events involving c2 and proj
    '?e1 "tag" "increaseProjectDrama"', '?e1 "actor" ?c2', '?e1 "project" ?proj',
    '?e2 "tag" "increaseProjectDrama"', '?e2 "actor" ?c2', '?e2 "project" ?proj',
    '?e3 "tag" "increaseProjectDrama"', '?e3 "actor" ?c2', '?e3 "project" ?proj',
    '(< ?e1 ?e2 ?e3)',
    // extra info for display purposes
    '?proj "projectName" ?projname',
    '?c1 "name" ?n1',
    '?c2 "name" ?n2'
  ],
  event: (vars) => ({
    actor: vars.c1,
    target: vars.c2,
    project: vars.proj,
    effects: [
      {type: 'addImpression', source: vars.c1, target: vars.c2, value: 2}
    ],
    text: `${vars.n1} became concerned about ${vars.n2}'s struggles\
with project "${vars.projname}"`,
    tags: ['projects']
  })
});
```

Figure 3.3: An example *WAWLT* action definition.

3.3.2 Action Definitions

Character actions in *WAWLT* are defined as Felt actions. They have preconditions, which take the form of Felt sifting patterns, and effects, which update the database when the action is performed. When an action definition is registered, Felt precompiles its sifting pattern to a Datalog query, which can later be run against the storyworld state database to return all possible sets of parameter bindings for this action.

We draw a distinction between *external actions*, which involve characters acting on the world outside of themselves, and *introspection actions*, which involve characters reflecting on past events through the lens of a particular narrative frame. External actions, for instance, might include actions like "discuss a shared value with Rashida", "insult Vincent", or "sabotage Alex's experiment"; introspection actions might include actions like "speculate that Lea dislikes me" or "conclude that Mikayla is a mean person." Introspection actions often produce *impressions* that influence the introspecting character's attitude toward another character. By separating out character reasoning into a category of first-class actions that players can observe and perform directly, and that are added to the transcript for players to elaborate on within their stories, we hope to expose character reasoning to the players explicitly – and, thereby, to mitigate the *Tale-Spin effect* [68], in which a system is mistaken by players as less intelligent than it actually is due to insufficient exposure of the internal processes.

Introspection actions bring *story sifting* – the process of matching and narrativizing sequences of past simulation events – into the storyworld as a diegetic component of how simulated characters make sense of the world around them. In this way, story sifting plays a central role in *WAWLT*'s implementation of character subjectivity.

Most introspection actions require that the first matched event in a sequence of events took place within a certain window of recency. This helps ensure that characters don't suddenly decide to spend their time thinking about actions that took place a long time ago, unless something (such as a special-case introspection action, without recency-gating on matched events) specifically prompts a reevaluation of those past events in particular.

Impressions define characters' perceptions of other characters. One character's perception of another can be influenced by up to three positive and three negative impressions simultaneously. Impressions are formed through introspection actions, and have numerical strength values indicating how strongly (and in what direction) they influence the holder's perception of the target. The holder's overall numerical "charge", or liking, toward the target is the sum of the scores of all the holder's currently held impressions of the target. Stronger impressions displace weaker ones upon formation, so a character's perception of another is generally defined by their strongest positive and negative impressions of that character. Impressions can be communicated from one character to another, albeit generally in weakened form, through gossip actions.

Impressions are *WAWLT*'s answer to character knowledge modeling, which is a common element of similarly simulation-driven narrative systems. Tale-Spin [41], Talk of the Town [54], and Versu [16], for instance, all model character knowledge phenomena at a fine-grained level, tracking per-character awareness of individual facts about the world and allowing character knowledge to change substantially over time. In *WAWLT*, however, we avoid modeling character knowledge phenomena directly due to complexity. Instead, we assume that all characters know about every event as soon as it transpires, but that most characters don't care about most events until these events somehow become directly relevant to them. Character *knowledge* is thus replaced in our system by subjective impressions, which are substantially fewer in number and more narratively consequential than granular facts about the world.

From a social simulation perspective, *WAWLT* also differs substantially from previous similar systems in its handling of character motivation. Ensemble [57], for instance, treats character motivations – *volitions* in Ensemble terminology – as transient and implicitly derived, due to the system's automatic re-computation of volitions after every action. Our motivation model, in contrast, is more like Ceptre [36]'s "tokens": character motivations modeled as consumable "resources" in a linear logic framework, produced by actions and persistent in the world state until explicitly consumed by another action. Thus, motivations in *WAWLT* are both less transient and less implicit (due to their production by explicit character actions, generally introspection actions) than volitions in Ensemble and other, similar social simulation engines.

3.3.3 Author Goals

Author goals are intended to provide players with a compositional *player intent language* [37] that they can use to explicitly communicate their current creative intent to the system. This intent language is one of the main novel features in the *WAWLT* architecture, and – we argue – an important feature for mixed-initiative co-creative architectures in general to support, due to its central role in enabling the system to respond intelligently to changes in player intent on a moment-to-moment basis.

Author goals currently include the following:

- Involve *character* in plot as *role*
- Cast suspicion on *character*
- Dispel suspicion on *character*

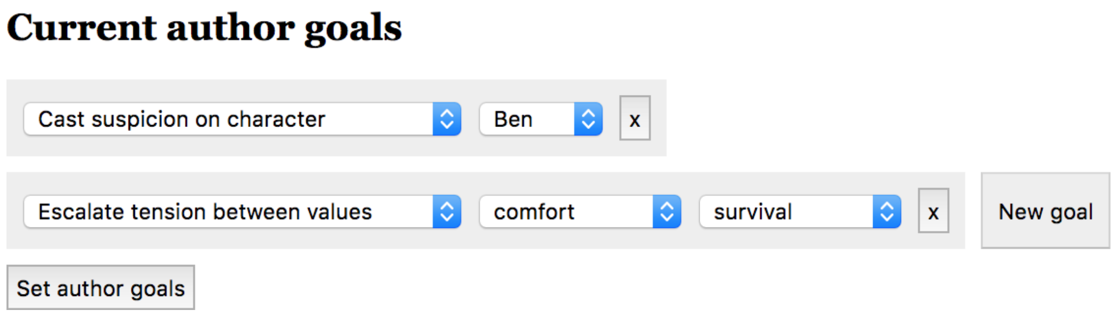

Figure 3.4: The *WAWLT* author goal selection interface.

- Escalate tension between *character* and *character*
- Defuse tension between *character* and *character*
- Escalate tension between *value* and *value*
- Defuse tension between *value* and *value*
- Introduce false lead
- Dismiss false lead
- Custom author goal: *textual search query* (*weight*)

Italics in goal names indicate parameter slots. For instance, if players want to focus on actions involving a particular character, they can add this character to a *character* slot in any goal that provides one. Alternatively, players can also leave this slot empty, in which case the system will treat it as a wildcard that stands for "any character."

The *custom author goal* option in particular allows players to specify finer-grained constraints on action suggestions than would otherwise be possible. When specifying a custom author goal, players can use a lightweight textual query language to restrict action suggestions to actions that contain certain substrings in their taglines; actions belonging to certain categories, such as "introspection actions" or "actions that involve projects"; and to filter action suggestions in a variety of other ways. Players can also specify the numerical weight that this custom author goal will contribute to any matched actions. This textual search mechanism greatly expands the expressiveness of the author goal language, but also requires the players to know what they're looking for and how it might be described in this system. Therefore, we expect that custom author goals will be used more frequently later in gameplay sessions, and by players who have a better sense of the space of all possible actions within the simulation.

Author goals are used to rank all currently possible actions in order to provide players with action suggestions. Each author goal is associated with a heuristic function that takes a possible action as an argument and returns a numerical score representing the relevance of this possible action to this author goal. Whenever the system needs to provide players with action suggestions, it first generates a list of all currently possible actions, then evaluates these possible actions against the set of currently active author goals. Every active author goal contributes a score to each possible action, and the list of possible actions is sorted by the total combined score from all active author goals, so that the most goal-relevant actions appear closest to the top of the suggestions list. This provides players with support in navigating the space of currently possible actions, which may contain hundreds of possible actions at any given point.

In addition to ranking action suggestions for characters participating in the current scene, the system also uses author goals to guide the autonomous actions of characters who are currently “in the background.” The process by which autonomous actions are selected and performed is described in greater detail in section 3.3.5.

Author goals are also intended to help players negotiate their intent with one another by making this intent explicit. Because players have to agree on the author goals that they are entering into the system, and because they are prompted to adjust their author goals at the start of every new scene, the system encourages players to regularly discuss their intent with one another, and players may have to explicitly argue for the things they want to have happen in the story. In this way, author goals can function similarly to the “palette” mechanism in tabletop storytelling games like *Microscope* [50], which requires players to explicitly discuss what they do and do not want to include in the story they are creating together.

3.3.4 Action Suggestions

The action suggestion interface displays the five most highly ranked possible actions for characters who are involved in the current scene. Whenever the storyworld state database is updated, *WAWLT* automatically recalculates which actions are currently possible. For each registered action definition, Felt executes the action’s compiled Datalog query against the database, returning all possible sets of bindings for this action, and instantiates a “possible action” for each set of bindings. This complete list of possible actions is then used to drive both action suggestions for the players and autonomous actions performed by characters in the background.

Action rankings are based primarily on the current author goals, each of which

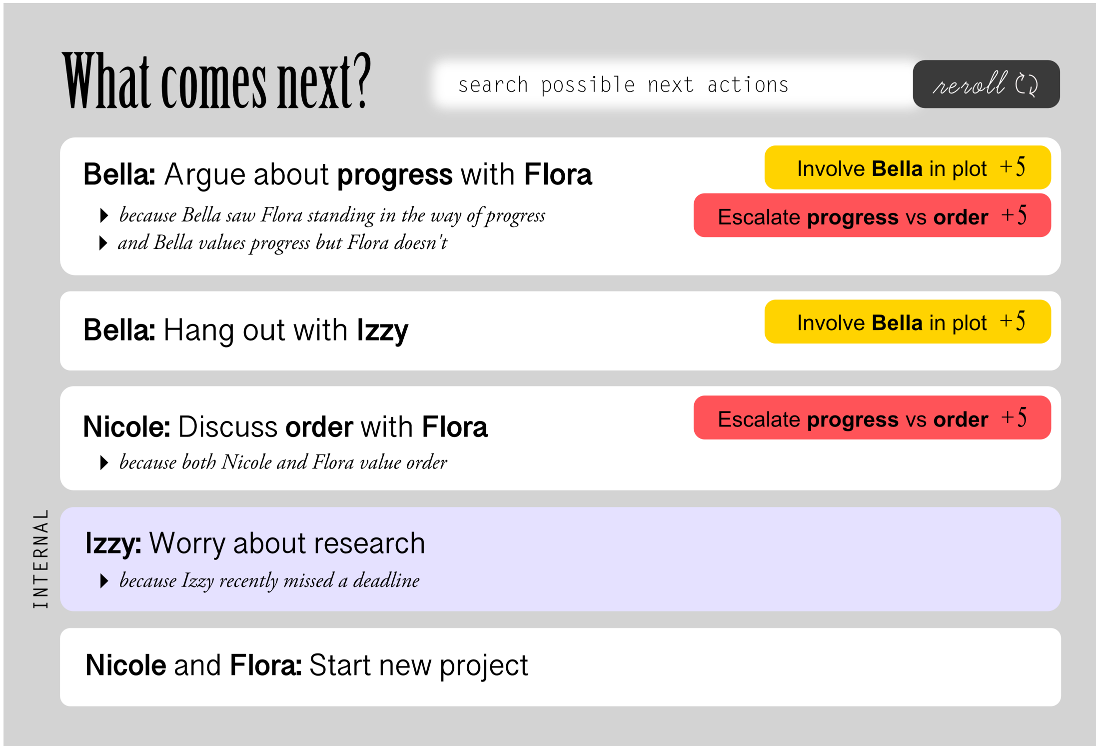

Figure 3.5: The *WAWLT* action suggestion interface, showing the current top five next possible actions for players to choose among to enact and add to their transcript. Possible actions are scored according to the current, player-set author goals (e.g., "involve character (Bella) in plot," "escalate tension between value (progress) and value (order).")

contributes a positive score to possible actions that can be read as advancing this author goal in some way – and, possibly, a negative score to possible actions that may be seen as detracting from the realization of this author goal. Action rankings also take into account the base weight specified in each action definition as an indicator of the action's base narrative significance, since players may find it confusing if relatively mundane actions (e.g., "gossip about weather") are ranked above more inherently dramatic actions (e.g., "accuse of murder"). This base weight is multiplied together with the total score from author goals to produce an overall score for each action, which is then used to sort

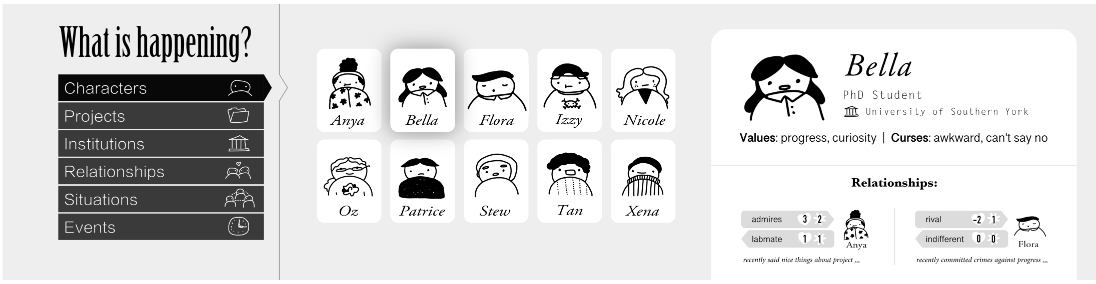

Figure 3.6: The *WAWLT* storyworld investigator interface, showing a portion of a character information card.

the list of possible actions and display the top five.

In the future, we may also expand the calculation of action scores to consider other factors. For instance, in order to prevent the same few actions from appearing as highly ranked suggestions over and over again, we may apply a ranking penalty to instances of the last few actions that were performed. In the context of introspection actions that involve a character ruminating about or reflecting on a particular set of past events, we may also take into account the subjective significance of the matched events to the character who is doing the rumination. Event significance calculation may consider such factors as the recency of the matched event, the strength of the relationships between the ruminating character and the event's participants, and perhaps some of the other factors suggested by the Indexter [7] model of event salience. These event salience heuristics would likely begin to resemble generic *sifting heuristics* [52] as proposed by Ryan, due to their basis in general-purpose models of why a sequence of events might be perceived as an interesting story.

Possible actions in the action suggestion interface are displayed alongside some details about why this action is currently possible. This includes a display of what active author goals contribute to the surfacing of this action suggestion, and with what strength; a short description of all the previous actions that partially caused this action, with the ability to expand the causality trace backward (to view the causes of the causes, and so on); and a short description of any other preconditions for this action, such as the presence of certain character traits on the action's protagonist, if any such preconditions exist. Introspection actions are also clearly marked as such, to distinguish them from external actions.

A search bar at the top of the action suggestion interface allows players to use the same lightweight textual query language used in custom author goals to rapidly filter action suggestions without having to add or update any author goals. If multiple top action suggestions have equal overall scores, actions with the same score are shuffled when suggestions are retrieved; we therefore also provide players with a "reroll" button, which allows them to re-randomize the sort order of actions with equal weights, possibly changing which actions are displayed in the suggestion interface.

3.3.5 Autonomous Actions

Action suggestions concern possible actions for characters who are active in the current scene. However, characters who aren't involved in the current scene can also perform actions autonomously in the background. Like player-visible action suggestions, autonomous actions are influenced by author goals. Autonomous actions, however, are chosen via a weighted random selection process over some of the higher-scoring possible actions for offscreen characters, rather than selected by the player directly. For efficiency, autonomous actions reuse the set of calculated possible actions that are used to provide players with action suggestions.

Autonomous actions are intended to help mitigate writer's block by ensuring that the storyworld will always continue to develop in significant ways, even if the players can't think of anything interesting to do in a particular scene. We are inspired here by the design property of *incrementality* [29], which was found to be helpful in supporting player storytelling in some existing simulation-driven games. After a scene, players can use the storyworld investigator to check in on what the characters not involved in the scene have been up to in the background, which might inspire new directions for the story.

3.3.6 Storyworld Investigator

Playtesting of previous versions of *WAWLT* revealed that players needed a way to proactively browse the full history of the simulated storyworld while writing their stories. In response, we created the *storyworld investigator*, which provides players with fine-grained tools for investigating the history and current state of the storyworld. The investigator is divided into several tabs, each of which displays a complete list of all instances of a certain type of storyworld entity – such as characters, relationships, projects, institutions, and events – and lets the player view *information cards* containing more detailed information about these entities.

Information cards are linked together with hyperlinks to enable rapid exploration of the web of storyworld entities. For instance, while viewing the information card for a particular character, links under the "Relationships" section allow rapid navigation to information cards containing more detailed information about this character's relationships with other characters, including the events and impressions that played a role in shaping each character's perception of the other.

The storyworld investigator also provides a *situations* tab, which allows players to make proactive use of story sifting to discover potential sites of narrative interest in the storyworld. This tab is equipped with a number of premade parametrized sifting patterns designed to help players locate emergent situations that might complicate the story in interesting ways: for instance, relationships in which the two involved characters have strongly incompatible assessments of one another, escalating cycles of revenge between characters, or long-standing instances of jealousy. As with author goals, players can leave these sifting patterns unconstrained or constrain them by specifying parameter values, for instance to view only instances of jealousy that involve a particular character.

In general, we expect that players will make especially extensive use of the storyworld investigator toward the start of a play session to familiarize themselves with the world's backstory, including the existing relationships between characters. We also expect that players will frequently use the investigator when deliberating between scenes about what situations they intend to explore next – primarily in order to discover untapped or neglected sites of narrative potential (especially through the *situations* tab), to explore autonomous actions performed by "out-of-focus" characters during the most recently completed scene (especially through the *events* tab), and to learn more about characters and other entities they might want to spotlight in the future.

3.3.7 Transcript

The transcript holds a running record of "the story so far": short system-generated summaries of all player-accepted action suggestions since the start of the play session, interleaved with more detailed player-generated prose descriptions of these actions. By the end of the play session, the transcript will constitute an "artifact of play" summarizing the events of play, much like the map in *The Quiet Year* [3] or the board in *Threadsteading* [2].

By giving players a way to annotate events with their own descriptions, we aim to provide support for *extrapolative narrativization* [27]: a player storytelling behavior in which players seize on and elaborate minor details in the stories they tell about their play experiences, regardless of whether these details are explicitly modeled in the simulation. Free-text descriptions of events give players a place to decide for themselves what aspects of an event are most important, and to establish and reference recurring story elements that are not modeled in the computational system.

Player-generated prose descriptions of events are not read, interpreted or reasoned over by the system in any way. This allows players to flexibly "retcon", or revise their descriptions of past events to match an updated understanding of where the story is going. Event description text remains editable indefinitely and doesn't impact forward simulation, so future actions cannot be invalidated by edits to the descriptions of previous events. Allowing the system to read and respond to player-generated text would compromise this flexibility: if player-generated text was incorporated into the simulation directly, subsequent edits to a block of player-generated text might compromise the system's interpretation of an event that was used as the foundation of a running chain of causality between several additional events in the meantime.

By treating player-generated prose as opaque to the system, we also aim to distinguish *WAWLT* from co-creative writing systems based on language models, such as *Creative Help* [51]. These systems treat prose-level suggestions as first-class while making no attempt to model causality at the level of discrete events or overall plot structure. We believe that these systems, although useful for injecting moment-to-moment "what happens next" suggestions that keep players from becoming completely stuck, fail to provide other, more important forms of creative support. In particular, because they lack an explicit world model, these systems frequently issue suggestions that contradict previously stated facts about the world, exacerbating the existing problem of maintaining consistency within a fictional world and distracting from the creativity support features they do provide. We prefer to offload the difficult problem of maintaining consistency to the computer, which can excel at this task given the right kind of architecture, and free up the humans to focus on authoring prose. For the sake of clarity, we thus focus the *WAWLT* AI architecture primarily on the provision of plot-structural or event-level suggestions, grounded in an explicit world model whose consistency is maintained by the system.

3.4 Playtesting

We conducted playtests with three solo players and two groups of paired players. Initially, each playtester was given a brief introduction to the project and the different parts of the user interface. Playtesters were then instructed to think aloud during their interaction with the game for 5-15 minutes at the player's discretion. Paired players engaged with a single instance of *WAWLT* simultaneously on a single computer, under the same conditions as the individual playtesters.

Broadly speaking, we found that the current version of *WAWLT* already supports player creativity in some of the intended ways, and is capable of producing an enjoyable play experience. Playtesters had little difficulty making use of the game's primary mechanics once they were introduced. All playtesters, even those who initially struggled to make the pieces of their story fit together into a larger storyline, eventually found themselves excited or curious to discover what would happen next in the story. All playtesters also expressed overall enjoyment of the play process. Six of seven playtesters (including all four paired playtesters and two of three solo playtesters) reported some sense of ownership over the story they produced through play. Moreover, the paired playtesters in particular expressed a great degree of enjoyment of the play experience; desire to continue working on the story (to such an extent that they were vocally disappointed that they could not continue at the conclusion of the playtest session); and feeling that what happened in the storyworld was somehow "real".

Nevertheless, there were also some significant points of confusion among players. Five players (including three of the four paired players) reported a sense of directionlessness at least once during the play process, suggesting that the system's action recommendations were not always sufficient to provide players with a sense of narrative structure. In one paired-playtesters group, both players initially assumed that author goals were intended primarily to be used by the system to filter and prioritize action suggestions, without realizing that they were also intended to be used as a way to encourage multiple simultaneous human players to negotiate intended story directions. Debriefing after the playtest also indicated that three players (including both of the paired players in one group) at some point forgot that the author goals existed, although the paired players "rediscovered" the author goals when a minor creative conflict briefly emerged between them.

The success of the paired playtesters in particular suggests to us that the creativity support features we provide in *WAWLT* are, like their counterparts in tabletop storytelling games, perhaps useful for individual players, but especially transformative when helping to scaffold and structure co-creativity in a multiplayer context, where negotiation between players regarding the content and direction of the story they intend to tell becomes a central part of play. As a result, we have since begun to consider *WAWLT* a multiplayer game primarily, while still aiming to support solo play as a viable mode of use.

3.5 Discussion

3.5.1 Story Sifting

At an implementation level, story sifting in *WAWLT* is functionally identical to precondition matching where some of the preconditions happen to involve sets of related storyworld events. However, from a design perspective, story sifting is not just precondition matching, but matching *and applying a particular subjective narrative frame* to the matched events. The execution of sifting patterns produces a variety of valid possible "readings" or interpretations of the same set of events, and even though these interpretations may be mutually incompatible, it is up to another subsystem – or to the player – to determine which readings to build a story around going forward. This distinction between sifting and precondition matching is made clear in *WAWLT*'s use of sifting patterns to implement subjective character reasoning, but is also present in more conventional sifting-based systems, such as Ryan's Sheldon system [52]: many valid sifting pattern matches are never promoted to a "real" part of the story, and a number of aesthetic judgment calls are made by the system as to which ones will be included in the story that gets told. Sifting patterns therefore bear some similarity to "rhetorical devices" in *Terminal Time* [39], producing a particular reading of a set of events that may be used or discarded depending on its suitability to the author's current rhetorical goals.

In this sense, we have found that story sifting can be viewed as a *thematics for narrating the operation* of precondition matching, in the same sense that Agre argues

AI techniques in general tend to simultaneously provide “both a method for designing artifacts and a thematics for narrating [their] operation.” [1] As in *WAWLT*, using the language of story sifting to discuss what is functionally precondition matching can help to inspire design approaches that might not have been considered otherwise, despite their longstanding technical possibility.

3.5.2 Simulation Design

Ryan, in his dissertation [52], proposes two key design patterns for simulations intended to support story sifting: *causal bookkeeping*, or explicit tracking of the causal relationships between storyworld events; and *contingent unlocking*, or the use of action preconditions that check for the existence of certain past events to support gradually escalating event “storylines”, such as a “flirt” event between two characters unlocking a “secret tryst” event between these same characters as a later possibility.

WAWLT makes extensive use of causal bookkeeping, albeit with a small twist. In Ryan’s Hennepin simulation engine, causal relationships are tracked directly between external actions taken by characters on the world, and actions may explicitly cause other subsequent actions to be queued for later performance at the moment the first action in a causal chain is performed. In *WAWLT*, however, we impose an authoring convention that external actions never cause other external actions directly: instead, external actions may cause introspection actions, which may in turn produce interpretations that cause other external actions. In this sense, causality in *WAWLT* is more frequently inferred after the fact (by characters performing introspection actions that project a particular causal interpretation onto a set of matched past events) rather than explicitly determined at the moment an action is performed (as with queuing of future actions in Hennepin.) Nevertheless, we – like Ryan – find it useful for the system to keep track of the causal relationships between actions, especially for the purposes of debugging and exposing character motivations to players.

Contingent unlocking, meanwhile, is used extensively within *WAWLT*'s simulation design, and differs from Ryan's interpretation of contingent unlocking in few ways if any. In particular, contingent unlocking enables plausibly gradual escalation of conflict in the *WAWLT* storyworld from less to more dramatic interactions between conflicting characters. Certain interpretation actions unlocked by repeated antagonism, for instance, effectively enable one *WAWLT* character to adopt the attitude that "this is the last straw", unlocking a variety of retaliation actions that allow for the further escalation of a simmering conflict between two characters who have vaguely disliked one another for a while.

3.5.3 Author Goals

Author goals were initially implemented via speculative execution of possible actions. Under this formulation, every author goal included a Datalog query fragment which could be executed against the simulation state database to determine the extent to which the goal in question had been realized. Each possible action was executed speculatively, producing one updated version of the simulation state database for each action, and each author goal's query was then run against the current and all speculatively updated versions of the database. By taking the difference between the number of matches found for each author goal query before and after each possible action was performed, a score was derived for each possible action indicating its overall suitability to the current author goals. These scores were then used to rank the actions.

This approach provided substantial authorial leverage, as author goals implemented in this way do not need to include any knowledge about what specific events are possible in the storyworld: events can be judged as advancing or detracting from author goals based on their actual effects alone. However, although elegant, this approach proved too inefficient to be practical with nontrivial numbers of actions and author goals, so the current implementation of author goals instead uses heuristic functions to evaluate possible actions. Each author goal's heuristic function takes a possible action as an argument and returns a numerical score indicating this possible action's suitability for this author goal. This makes goal evaluation more computationally tractable, but entangles action authoring with author goal implementation in a way that substantially decreases authorial leverage, frequently forcing action authors to modify the heuristic functions of relevant author goals when they introduce a new action or make a substantial change to an existing action definition. The use of goal heuristic functions that rely on action *tags* rather than specific action names to determine goal relevance has somewhat mitigated this added authorial burden, because it is relatively easy to tag new actions in such a way that they become "visible" to author goals that are already looking for existing actions with these same tags, but this merely reduces the frequency of the problem rather than eliminating it entirely.

In the future, we may return to using speculative execution of possible actions as a subsequent action evaluation step, after first using author goal heuristic functions to filter down the set of all possible actions to a more reasonable subset. Under this process, goal heuristic functions would first nominate some number of possible actions for consideration, and only these candidate actions would be fully evaluated via speculative execution.

3.5.4 Effect Handlers

A wide variety of existing generative narrative systems, including planning-based systems [45, 70] and linear logic-based systems like Ceptre [36], define character actions in terms of preconditions and effects. Action effects, in turn, are commonly defined in terms of add and delete lists, which describe facts to add and delete from a blackboard or database of logic sentences when an action is performed. Effects in Felt actions are defined similarly, but Felt provides an additional layer of abstraction – called *effect handlers* – over add and delete lists. Effect handlers are Felt's implementation of *procedural effects*, a feature of some planning-based systems (such as the system driving NPC behavior in *F.E.A.R.* [42]) that allows for the implementation of dynamic changelists. We found this feature to be particularly beneficial in our implementation of *WAWLT*.

A Felt effect handler is a function, defined by the authors of a particular Felt simulation domain, that takes in both the storyworld state database and some other parameters, then returns a list of facts to add and delete from the database. Essentially, effect handlers represent a way to make changes to the database in terms of the overall effect you want to achieve (e.g., "create an advisor/student relationship between these two characters") rather than the specific individual facts you want to add and delete from the database.

From a simulation implementation perspective, this separation between intent and implementation allows for the construction of safer and cleaner interfaces for high-level changes that frequently involve the simultaneous editing of many individual facts. Robust effect handlers can be written to change *all* relevant individual facts consistently whenever you make a specific kind of high-level change, so you don't have to manually enforce this through action authoring discipline or worry about forgetting some individual changes within a larger changelist intended to achieve a particular overall effect. This also greatly eases refactoring of how simulation updates are performed, since all changes to individual facts take place through a smaller number of cleanly defined effect handlers: actions are not permitted to change individual facts directly, they may only invoke effect handlers.

Simultaneously, from an action authoring perspective, this separation between interface and implementation allows action authors to use and reuse effect handlers without knowing how they work exactly, shielding them somewhat from the internal complexity of the database. We also found that, for action authors, a well-defined set of effect handlers can effectively serve as a rapidly skimmable catalog of the kinds of state changes that actions can make in the storyworld. This can serve to inspire the design of new character actions by hinting at the possibility of introducing more ways to flavorfully achieve each available effect.

Finally, because effect handlers are first-class functions that receive the entire database as an argument, they are substantially more flexible than hand-authored literal changelists, and are able to consider broader context when calculating the sets of specific changes that ought to be made to individual facts when trying to achieve a specific outcome in the database. For instance, consider the `addImpression` effect handler, which is used to conditionally update the set of impressions influencing one character's evaluation of another. This complicated effect handler makes several queries against the state of the database to determine whether or not it should even add the new impression at all; which (if any) existing impressions it should displace to "make room" for the new impression; and how the overall relationship between the characters should be updated as a result. This effect handler, whose behavior is core to the *WAWLT* social simulation, could not be replaced by a static changelist; the changelist of an action that adds an impression to the database while obeying the elaborate rules governing impression formation must necessarily be dynamic.

3.6 Conclusions and Future Work

In this chapter, we presented the AI architecture of *Why Are We Like This?* (*WAWLT*), an AI-supported storytelling game intended to provide explicit support for the kinds of player storytelling practices seen in many simulation-driven games. *WAWLT* makes extensive use of story sifting, both to implement character subjectivity and to provide players with tools for investigating the history and current state of the storyworld. *WAWLT*'s architecture is also intended as an argument for a central claim: that machines should support player storytelling practices by providing players with intelligent plot direction suggestions, drawn from an ongoing social simulation and guided by player utterances in a machine-understandable intent language, realized here in the form of *author goals*.

Neither *WAWLT*'s current setting (a remote research symposium venue occupied by a community of temporarily stranded researchers) nor its current genre (a pastiche of the cozy mystery genre) are inevitable, and both were chosen largely because they appealed to the game's authors. In the future, we hope to treat this setting as one of many swappable "playsets", allowing players to explore other settings and genres using the same co-creative infrastructure and tooling. This is made possible in large part by the fact that the *WAWLT* storyworld is defined mainly in terms of Felt actions, which were designed to be easy to author and agnostic to the fine-grained details of any particular simulation domain. The creation of an alternative *WAWLT* playset would thus consist largely of defining a new initial world state generator and a new set of actions for characters to perform. However, it is also worth noting that some of the genre framing in *WAWLT* is provided by author goals, which are more difficult to implement and would likely require significant revision by playset developers.

In the future, we intend to conduct a larger and more thorough evaluation of a more complete version of *WAWLT*. Such an evaluation will likely involve several parts. Quantitative survey questions taken from the Creativity Support Index [8] may be used to assess how well the game performs as a creativity support tool, perhaps enabling direct comparison with other co-creative storytelling tools, such as *Writing Buddy*, *Creative Help*, and *Mimisbrunnur*. In addition, following the recommendations of Kreminski et al. [27], open-ended interviews with players may be analyzed in conjunction with the stories they create to discover how the different features of the game tend to influence the authorship process.

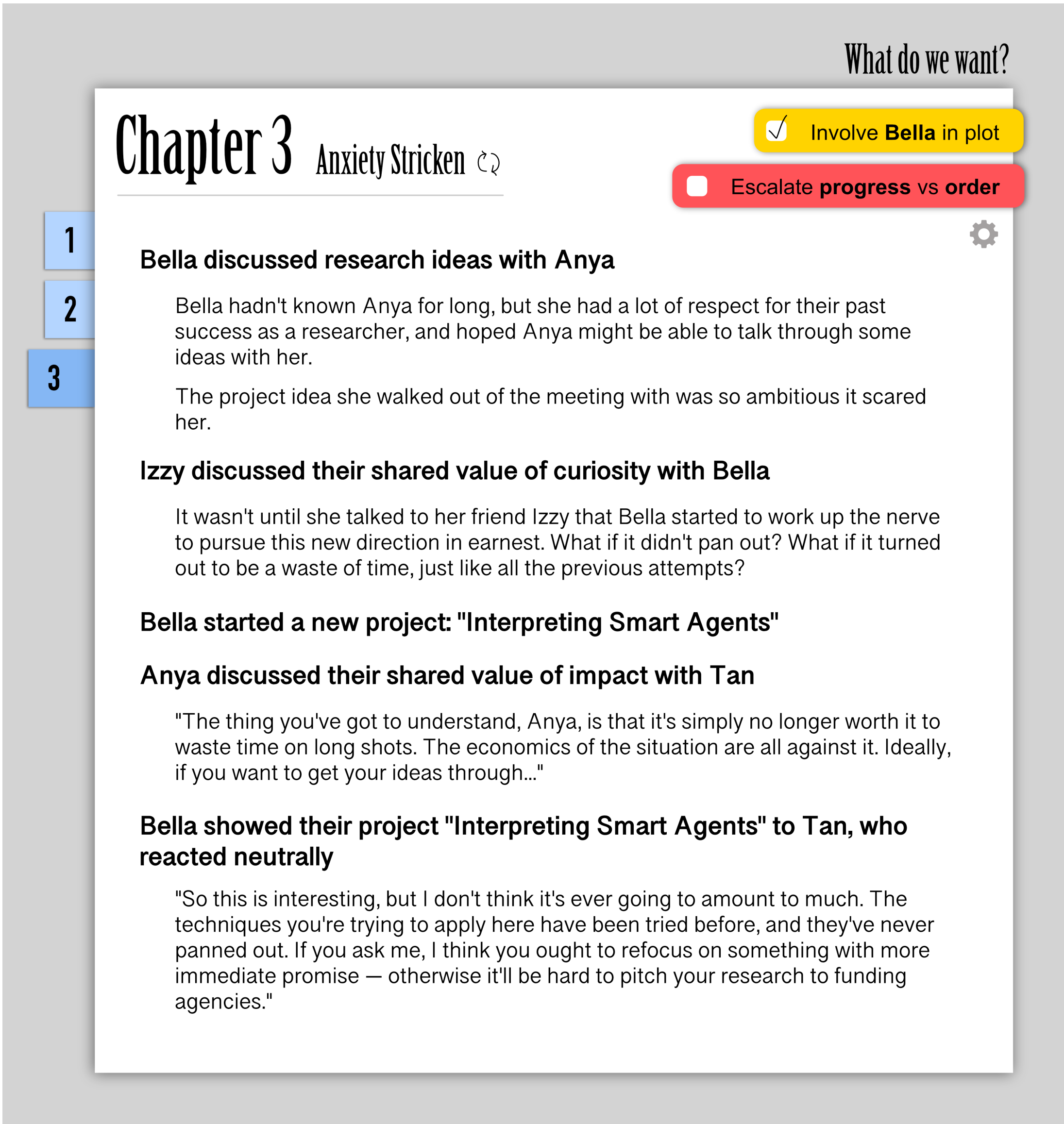

Figure 3.7: The *WAWLT* transcript editing interface. Bold text in the transcript is system-generated, non-bold text is authored by the players.

Chapter 4

Conclusion

We conducted a preliminary analysis of emergent narrative *authorship play*: a set of play practices in which players use simulation-driven emergent narrative games as storytelling partners due to their creativity support capabilities. We examined three culturally significant dialogic retellings of play experiences in three different emergent narrative games, and found evidence that four key features of these games—generativity, incrementality, boundedness, and limited player control—helped to support the development of these stories during the authoring process. Specifically, our analysis—by focusing on points at which each of the stories in question could have "gone wrong", but didn't—revealed features that likely helped the authors of these stories overcome certain common barriers to creativity, which might have caused the storytelling process to fail if left unaddressed.

Guided by this analysis, we then developed *Why Are We Like This?* (*WAWLT*), a mixed-initiative co-creative storytelling game that provides explicit support for emergent narrative authorship play. *WAWLT* incorporates all of the creativity-supporting design elements identified by our analysis to some extent, while also introducing several new creativity support features to the standard emergent narrative game design formula—including story sifting, author goals, and the transcript. Initial playtesting showed that players consistently found the process of storytelling with *WAWLT* to be enjoyable; felt a sense of ownership over the stories they created; and were sometimes excited or curious to discover what would happen next in the stories they created. However, playtesting also showed that players in the current version of *WAWLT* suffer from an occasional sense of aimlessness. Going forward, we hope to address this issue by providing the system with a more sophisticated model of high-level narrative structure, which can then be used to guide action suggestions.

Altogether, we interpret our playtest results as evidence that *WAWLT* successfully provides support for player storytelling practices. At present, it is difficult to determine exactly how much each of the several creativity support features in *WAWLT* contributed to the overall success of the design. Teasing apart the roles that these various features play in the storytelling process may be the target of further evaluation in the future. Regardless, however, we believe that *WAWLT* represents a strong prototype for a new class of emergent narrative play experiences explicitly designed to support authorship play.

Bibliography

[1] Philip Agre. Toward a critical technical practice: lessons learned in trying to reform AI. In Geoffrey Bowker, Susan Leigh Star, William Turner, and Les Gasser, editors, *Bridging the Great Divide: Social Science, Technical Systems, and Cooperative Work*, pages 131–157. Mahwah, NJ: Erlbaum, 1997.

[2] Lea Albaugh, April Grow, Chenxi Liu, James McCann, Gillian Smith, and Jennifer Mankoff. Threadsteading: playful interaction for textile fabrication devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 285–288, 2016.

[3] Avery Alder. The Quiet Year. `buriedwithoutceremony.com/the-quiet-year`, 2013.

[4] Botnik. Predictive Writer. `botnik.org/apps/writer`, 2017.

[5] Robin Burkinshaw. Alice and Kev. `https://aliceandkev.wordpress.com`, 2009.

[6] Robin Burkinshaw. Selflessness. `https://aliceandkev.wordpress.com/2009/06/16/selflessness`, 2009.

[7] Rogelio E Cardona-Rivera, Kara B Cassell, Stephen G Ware, and R Michael Young. Indexter: a computational model of the event-indexing situation model for characterizing narratives. In *Proceedings of the 3rd Workshop on Computational Models of Narrative*, pages 34–43, 2012.

[8] Erin Cherry and Celine Latulipe. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(4), 2014.

[9] Kate Compton and Michael Mateas. Casual creators. In *International Conference on Computational Creativity*, pages 228–235, 2015.

[10] Nigel Cross. *Simulation of Computer Aided Design.* PhD thesis, University of Manchester Institute of Science and Technology (UMIST), 1967.

[11] donoteat01. Power, politics, & planning: Episode 2: Urban freeways. `https://www.youtube.com/watch?v=rseaKBPkRPU`, 2018.

[12] DFStories.com editors. Dwarf Fortress Stories. `http://dfstories.com/all-stories`, 2019.

[13] Mirjam P Eladhari, Anne Sullivan, Gillian Smith, and Josh McCoy. AI-based game design: enabling new playable experiences. Technical report, UC Santa Cruz Baskin School of Engineering, Santa Cruz, CA, 2011.

[14] Mirjam Palosaari Eladhari. Re-tellings: the fourth layer of narrative as an instru-

ment for critique. In *International Conference on Interactive Digital Storytelling*, pages 65–78. Springer, 2018.

[15] Brian Eno. Composers as gardeners. https://www.edge.org/conversation/brian_eno-composers-as-gardeners, 2011.

[16] Richard Evans and Emily Short. Versu—a simulationist storytelling system. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2):113–130, 2013.

[17] Bay 12 Games. Dwarf Fortress. bay12games.com/dwarves, 2006.

[18] Jacob Garbe. Simulation of history and recursive narrative scaffolding. project.jacobgarbe.com/simulation-of-history-and-recursive-narrative-scaffolding, Feb 2018.

[19] Natalie Goldberg. *Writing Down the Bones: Freeing the Writer Within*, pages 11–12. Shambhala, 2005.

[20] Jason Grinblat and C Brian Bucklew. Subverting historical cause & effect: generation of mythic biographies in Caves of Qud. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, page 76. ACM, 2017.

[21] Paradox Interactive. Crusader Kings II. paradoxplaza.com/crusader-kings-ii/CKCK02GSK-MASTER.html, 2012.

[22] Max Kreminski, Devi Acharya, Nick Junius, Elisabeth Oliver, Kate Compton, Melanie Dickinson, Cyril Focht, Stacey Mason, Stella Mazeika, and Noah Wardrip-

Fruin. Cozy Mystery Construction Kit: prototyping toward an AI-assisted collaborative storytelling mystery game. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 2019.

[23] Max Kreminski, Melanie Dickinson, Michael Mateas, and Noah Wardrip-Fruin. Why Are We Like This?: Exploring writing mechanics for an AI-augmented storytelling game. In *Proceedings of the Electronic Literature Organization Conference*, 2020.

[24] Max Kreminski, Melanie Dickinson, Michael Mateas, and Noah Wardrip-Fruin. Why Are We Like This?: The AI architecture of a co-creative storytelling game. In *Proceedings of the Fifteenth International Conference on the Foundations of Digital Games*, 2020.

[25] Max Kreminski, Melanie Dickinson, and Noah Wardrip-Fruin. Felt: a simple story sifter. In *International Conference on Interactive Digital Storytelling*, pages 267–281. Springer, 2019.

[26] Max Kreminski, Isaac Karth, and Noah Wardrip-Fruin. Generators that read. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 2019.

[27] Max Kreminski, Ben Samuel, Edward Melcer, and Noah Wardrip-Fruin. Evaluating AI-based games through retellings. In *Proceedings of the AAAI Conference on*

Artificial Intelligence and Interactive Digital Entertainment, volume 15, pages 45–51, 2019.

[28] Max Kreminski and Noah Wardrip-Fruin. Gardening games: an alternative philosophy of PCG in games. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 2018.

[29] Max Kreminski and Noah Wardrip-Fruin. Generative games as storytelling partners. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 2019.

[30] Anne Lamott. *Bird by Bird: Some Instructions on Writing and Life*, page 177. Knopf Doubleday, 2007.

[31] Bjarke Alexander Larsen, Luis Emilio Bruni, and Henrik Schoenau-Fog. The story we cannot see: On how a retelling relates to its afterstory. In *International Conference on Interactive Digital Storytelling*, pages 190–203. Springer, 2019.

[32] Antonios Liapis, Georgios N Yannakakis, Constantine Alexopoulos, and Phil Lopes. Can computers foster human users' creativity? Theory and praxis of mixed-initiative co-creativity. *Digital Culture & Education*, 8(2):136–153, 2016.

[33] Sandy Louchart, Ivo Swartjes, Michael Kriegel, and Ruth Aylett. Purposeful authoring for emergent narrative. In *Joint International Conference on Interactive Digital Storytelling*, pages 273–284. Springer, 2008.

[34] Sandy Louchart, John Truesdale, Neil Suttie, and Ruth Aylett. Emergent narrative,

past, present and future of an interactive storytelling approach. In *Interactive Digital Narrative: History, Theory and Practice*, pages 185–199. Routledge, 2015.

[35] Enrique Manjavacas, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont. Synthetic literature: writing science fiction in a co-creative process. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, pages 29–37, 2017.

[36] Chris Martens. Ceptre: A language for modeling generative interactive systems. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[37] Chris Martens and Matthew A Hammer. Languages of play: towards semantic foundations for game interfaces. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 32–41. ACM, 2017.

[38] Michael Mateas and Andrew Stern. Towards integrating plot and character for interactive drama. In *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium*, pages 113–118. Menlo Park: AAAI Fall Symposium Series, 2000.

[39] Michael Mateas, Paul Vanouse, and Steffi Domike. Generation of ideologically-biased historical documentaries. In *AAAI/IAAI*, pages 236–242, 2000.

[40] Maxis. The Sims 2. `ea.com/games/the-sims/the-sims-2`, 2004.

[41] James R Meehan. TALE-SPIN, an interactive program that writes stories. In

Proceedings of the 5th International Joint Conference on Artificial Intelligence, pages 91–98, 1977.

[42] Jeff Orkin. Three states and a plan: the A.I. of F.E.A.R. In *Game Developers Conference*, 2006.

[43] Brian Phillips. Pro Vercelli: A Conspiracy So Vast. `http://www.runofplay.com/2008/12/08/pro-vercelli-a-conspiracy-so-vast`, 2008.

[44] Brian Phillips. Pro Vercelli. `http://www.runofplay.com/category/vercelli`, 2009.

[45] Julie Porteous. Planning technologies for interactive storytelling. In *Handbook of Digital Games and Entertainment Technologies*. Springer, 2016.

[46] Nikita Prokopov. DataScript. `github.com/tonsky/datascript`, 2014.

[47] Aaron Reed. Archives of the Sky. `archivesofthesky.textories.com`, 2018.

[48] Martin Rhodes, Simon Coupland, and Tracy Cruickshank. Enhancing real-time sports commentary generation with dramatic narrative devices. In *Joint International Conference on Interactive Digital Storytelling*, pages 111–116. Springer, 2010.

[49] Mark Owen Riedl and Vadim Bulitko. Interactive narrative: An intelligent systems approach. *AI Magazine*, 34(1):67–77, 2013.

[50] Ben Robbins. Microscope: A fractal role-playing game of epic histories. `lamemage.com/microscope`, 2011.

[51] Melissa Roemmele and Andrew S Gordon. Creative Help: a story writing assistant. In *International Conference on Interactive Digital Storytelling*, pages 81–92. Springer, 2015.

[52] James Ryan. *Curating Simulated Storyworlds*. PhD thesis, UC Santa Cruz, 2018.

[53] James Owen Ryan, Michael Mateas, and Noah Wardrip-Fruin. Open design challenges for interactive emergent narrative. In *International Conference on Interactive Digital Storytelling*, pages 14–26. Springer, 2015.

[54] James Owen Ryan, Adam Summerville, Michael Mateas, and Noah Wardrip-Fruin. Toward characters who observe, tell, misremember, and lie. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[55] Ben Samuel. *Crafting Stories Through Play*. PhD thesis, UC Santa Cruz, 2016.

[56] Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. The design of Writing Buddy: a mixed-initiative approach towards computational story collaboration. In *International Conference on Interactive Digital Storytelling*, pages 388–396. Springer, 2016.

[57] Ben Samuel, Aaron A Reed, Paul Maddaloni, Michael Mateas, and Noah Wardrip-Fruin. The Ensemble engine: next-generation social physics. In *Proceedings of the*

Tenth International Conference on the Foundations of Digital Games (FDG 2015), pages 22–25, 2015.

[58] Ben Samuel, James Ryan, Adam J Summerville, Michael Mateas, and Noah Wardrip-Fruin. Bad News: an experiment in computationally assisted performance. In *International Conference on Interactive Digital Storytelling*, pages 108–120. Springer, 2016.

[59] Doug Sharp. Story vs. game: the battle of interactive fiction. `web.archive.org/web/20040404061317/www.channelzilch.com/doug/battle.htm`, 1989.

[60] Ben Shneiderman. Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, 50(12):20–32, 2007.

[61] Robin Sloan. Writing with the machine. `robinsloan.com/notes/writing-with-the-machine`, May 2016.

[62] Adam M Smith and Michael Mateas. Computational caricatures: probing the game design process with AI. In *Workshops at the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.

[63] Gillian Smith, Elaine Gan, Alexei Othenin-Girard, and Jim Whitehead. PCG-based game design: enabling new play experiences through procedural content generation. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, page 7. ACM, 2011.

[64] Kevin Snow, George Kavallines, Thomas Ferkol, and Andi McClure. Matul Remrit. `http://www.bravemule.com/matulremrit`, 2013.

[65] Ingibergur Sindri Stefnisson and David Thue. Mimisbrunnur: AI-assisted authoring for interactive storytelling. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, pages 236–242, 2018.

[66] Theodora Vardouli. Computer of a thousand faces: anthropomorphizations of the computer in design (1965-1975). *Dosya*, page 24, 2012.

[67] Bret Victor. Learnable programming. `http://worrydream.com/LearnableProgramming`, 2012.

[68] Noah Wardrip-Fruin. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. MIT Press, 2009.

[69] Georgios N Yannakakis, Antonios Liapis, and Constantine Alexopoulos. Mixed-initiative co-creativity. In *Foundations of Digital Games*, 2014.

[70] R Michael Young, Stephen G Ware, Kara B Cassell, and Justus Robertson. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, 37(1-2):41–64, 2013.