Cozy Mystery Construction Kit

Prototyping Toward an AI-Assisted Collaborative Storytelling Mystery Game

Max Kreminski, Devi Acharya, Nick Junius, Elisabeth Oliver, Kate Compton, Melanie Dickinson,
Cyril Focht, Stacey Mason, Stella Mazeika, Noah Wardrip-Fruin
UC Santa Cruz
Santa Cruz, California
{mkremins,dacharya,njunius,elaolive,kcompton,mldickin,cfocht,stcmason,jmazeika,nwardrip}@ucsc.edu

ABSTRACT

This paper presents a case study in the experience-first prototyping of a generative game. Our goal in this process was to create a PCG-based mystery story construction game centered on a social simulation of characters and their motivations, and driven by a set of core themes and experiences we wanted players to encounter. In pursuit of this goal, we created a series of prototypes to test how a variety of generative and AI-based techniques—including character generation, character action suggestion based on game state, story sifting, and social simulation—may be used in support of collaborative storytelling. In this paper we catalogue these prototypes and what we have learned by creating them, detailing design elements we found to be successful in supporting player creativity and that may be useful to the developers of similar games and experiences going forward.

CCS CONCEPTS

• **Applied computing → Computer games**; • **Human-centered computing** → Interaction design theory, concepts and paradigms;

KEYWORDS

co-creativity, player creativity, prototyping, design patterns, collaborative storytelling, tabletop games, AI-based game design, PCG-based game design

ACM Reference Format:
Max Kreminski, Devi Acharya, Nick Junius, Elisabeth Oliver, Kate Compton, Melanie Dickinson, Cyril Focht, Stacey Mason, Stella Mazeika, Noah Wardrip-Fruin. 2019. Cozy Mystery Construction Kit: Prototyping Toward an AI-Assisted Collaborative Storytelling Mystery Game. In *Foundations of Digital Games 2019 (FDG19).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Motive, means, and opportunity: any good crime drama will teach you that all three must be considered in the solving of a crime. Of these, means and opportunity—the *how* of mystery scenario construction—have been well considered in the domain of procedural mysteries. It is not uncommon for a mystery to be set up as a pile of clues to be collected. These are generally facts about the world: a broken bat, an eyewitness placing a character in the foyer. Then, like an elaborate zebra puzzle, the facts are lined up to reveal a logical conclusion, the perpetrator revealed through the elimination of all other suspects.

In many mystery stories, whether in traditional or procedural media, motive is also simply part of such a puzzle. A suspect can only be accused if their motive has been established—and it may be a motive that feels as distant to us as the "hand wrought dueling pistols, curare and tropical fish" that Raymond Chandler noted many puzzle-oriented mysteries employ as their crimes' means. [4]

This "puzzle mystery" tradition, perhaps best exemplified by the works of Ellery Queen, is a rich one. And the logical structure of its plots makes this tradition an attractive target for procedural generation. And yet we believe that procedural mystery generation will be impoverished if this tradition is the only one represented. So our work has focused on generative support for mystery game experiences in a different tradition. This tradition is perhaps best exemplified by Chandler—but not because of the "hard boiled" settings or main characters often associated with him. Rather, because of his focus on character relationship and emotion as the center of his stories—with motive only understandable in this context—sometimes sacrificing the logical structure of the mystery plot in the process.

This is different from the puzzle focus in the Queen tradition, and also different from the action-oriented tradition found in the "pulp" publications in which Chandler began his career. Though Chandler argued it was central even there: "My theory was that readers just thought that they cared about nothing but the action; that really, although they didn't know it, they cared very little about the action. The thing they really cared about, and that I care about, was the creation of emotion through dialogue and description." [5]

In this paper, rather than presenting technical progress toward a recognized goal in procedural content generation, we attempt to present a design exploration of new possibilities for the use of generative methods in games, guided by an understanding of genre and a set of design goals for the experience we want to create. We first introduce the design pillars and intended play experience that informed our design decisions and guided our prototyping. We briefly survey related work in mystery scenario generation, collaborative story construction play experiences, and the use of social simulation in narrative games. We then describe a series of prototypes we developed over the course of approximately six months of work, gradually iterating toward an enjoyable play experience that

FDG'19, August 26-30, 2019, San Luis Obispo, California, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

adheres to our design pillars while using AI and generative methods to provide players with creativity support. Next, we discuss several features we found to be especially supportive of player creativity during the prototyping process. Finally, we briefly discuss broader learnings from the prototyping process as a whole and possible next steps for this work.

2 DESIGN GOALS

2.1 Story and Aesthetic Design

Our project seeks to support a mystery genre experience about character relation, emotion, and motive. For us, the heart of motive is character interactions, the social buildup preceding a crime and the social fallout after it has occurred. We focus our investigation not on the where or the how (as one playing *Clue* might), but rather on the why. What drives a character, particularly in a world in which there are no monsters, to commit crime, potentially as severe as murder?

We investigate how to best expose this dimension of mystery to players, making use of AI and PCG in conjunction with human reasoning and feeling. Our current design direction is toward experiences in which two human players with differing responsibilities work cooperatively with an AI system to construct mysteries in which the social and emotional motivations are the driving force of the story's construction.

We want the emotional tone of these stories to work within the "cozy" mystery tradition. In our research on exemplars, we have found that this tradition actually has four key features, which may be more or less present in any given work. In our design discussions we refer to these as the four axes of coziness: Sociological (everyone is good, there are no monsters), Structural (plot wraps up nicely), Nonviolent (no or little violence), and Thematic (the story revolves around things like knitting, dog shows, or rose breeding). Our prototypes thus far have held close to the cozy end of the sociological axis (it is the key driver of our emotional tone), with the eventual goal of supporting players who wish to have an experience close to the cozy end of the structural axis, varying wildly on the physical/violence axis, and with a thematic setting (a snowed-in observatory, full of researchers) that is probably cozy for us (and many readers of this paper) but not for the average mystery fan.

2.2 Play Experience Design

The eventual play experience we envision, shaped by the process of prototyping we document in this paper, involves two players both taking on the role of a storyteller. One of the players, the *Agatha player* (named in honor of mystery author Agatha Christie), is primarily in charge of making decisions about higher level plot concerns, similar to the beat construction in *Façade*. The other player, the *detective player*, is focused more on the individual characters' goals and actions, similar to the actions in *Prom Week* or the social moves in Versu. The computational system supporting the players is primarily responsible for keeping track of what has happened and what has been deemed "true" by the players up to that point. This system also needs to help mediate possible character actions and plot developments to provide a degree of constraints and suggestions to help the players continuously work towards the creation of a complete story.

Several aspects of this design—most notably the distinct and asymmetrical roles of the two human players and the placement of the computer in a supporting role to this collaboration—were inspired heavily by *Bad News* [27]. However, we realized early on that to extend a *Bad News*-like experience to a broader audience would require extensive creativity support. *Bad News*'s success hinged not only on the underlying town simulation but also on the capabilities of two human participants other than the "player." First of these was the *actor*, who would have to repeatedly take up the role of a different simulated character and improvise convincingly as that character at a moment's notice. And second was the *wizard*, who would sit behind the scenes during a playthrough of *Bad News* and pore over the simulation state in real time, seeking out narratively interesting information to feed the actor—sometimes even directly in response to a player's question, with the actor unable to answer the question until the wizard fed them the necessary information. Both the actor and the wizard thus needed deeply specialized skills to make the experience operate successfully, and the majority of ordinary players could not be expected to function in similar roles without a great deal more support from the system.

Moreover, even if the system was successful at scaffolding player creativity, we suspected that there was no way to make the experience work with general audiences other than to relax the expectations the experience as a whole would place on its participants. As a result, we found ourselves drawing further design inspiration from "GM-less" tabletop story construction games like *Microscope* [23] and *The Quiet Year* [1]. Ultimately, we decided, what we really wanted was something less like a conventional mystery story generator and more like a *casual creator* [7] for a certain specific kind of mystery story.

3 RELATED WORK

3.1 Mystery Generation

A number of other projects have undertaken the task of procedurally generating mystery stories or scenarios, some specifically in service of player experience in the context of games. The generation of static mystery stories dates back at least as far as 1971, with Klein's work in narrative generation [15, 16] serving as an early example.

More recently, a wide variety of approaches to the procedural generation of mystery stories or scenarios for players to experience or explore interactively have been proposed. Stockdale [31] has generated playable murder mystery scenarios by generating networks of characters, selecting a single motive/perpetrator/victim combination at random, and then generating character interaction histories "in reverse" to ensure that the murder seems plausible. Barros et al. [2] have made use of open data to generate murder mystery scenarios involving networks of associated historical figures. Mohr et al. [21] have used Dynamic Epistemic Logic [8], "the logic of changing knowledge and beliefs", to generate mystery scenarios for players to solve, with a focus on ensuring that all character actions are plausibly motivated. Their work also enables a style of player interaction that includes the interrogation of characters who are capable of lying in motivated ways (for instance, to cover up their involvement in a crime.)

Among commercial experiences, *The Shrouded Isle* [10] stands out for its extensive reliance on the player's investigation and gradual revelation of the attributes of procedurally generated characters as part of gameplay. Although the game does not advertise itself as incorporating a mystery scenario generator, the natural course of play hinges on the player's understanding of emergent procedural mysteries whose solutions will frequently have a direct impact on the player's choices about which characters to trust.

Adjacent to the challenge of mystery generation is Horswill's and Robison's recent work on operationalizing the questionnaire-based character creation process in *Dread* [12]. By tagging answers to a variety of character creation questions with their logical implications about a character who would give those answers, it is possible to use a SAT solver to generate random characters whose backstories are consistent with the answers the player provided. This same architecture could be employed to generate progressively smaller sets of valid solutions to a mystery scenario as the player uncovers clues—or, in the context of mixed-initiative mystery story construction, as the user adds constraints to the scenario. Some of the prototypes we report in this paper have made tentative steps toward the use of answer set programming [29] to similar ends.

Broadly speaking, most of these projects have focused on the generation of *logically consistent* mystery stories or scenarios. Some have added to this focus a further goal of ensuring that players or readers are presented with a sufficient set of clues to enable them to solve the mystery with only the information made available to them through the course of reading the story or playing the game. The interactive experiences discussed here universally place the player in the role of the detective, with player interaction essentially boiling down to the solving of an elaborate logic puzzle.

We are interested in the mystery genre primarily for its thematic focus on the search for truth. Moreover, we are more interested in empowering players to construct their own mystery stories than in presenting them with generated mystery scenarios they must solve. As a result, we draw on existing mystery generation work only lightly, and only insofar as it can be repurposed in support of player creativity.

3.2 Social Simulation in Narrative Games

In pursuit of interactivity and reactivity, narrative games have, to varying degrees, incorporated social simulation to guide character behavior. Some of this variance is based on whether the games are focused on the experiencing of the story, as is the case with *Façade* and *Blood and Laurels*, or the discovery of the story, as is the case with *Prom Week* and *Bad News.*

Façade, being an interactive drama, is primarily concerned with maintaining a coherent plot and experience through its beat structuring. It uses light elements of social simulation, the abstract social games, as part of organizing and choosing beats and tracking state rather than centering the entire experience on the playing of these games [18]. This lack of player focus on the social games is partially due to *Façade* maintaining a theatrical aesthetic and relying primarily on the characters' expressiveness, rather than traditional interface elements, to convey the state of the social simulation.

While *Blood and Laurels*, by virtue of being built with Versu, is an interactive drama like *Façade*, it has a more decentralized method of structuring its plot. The individual agents in *Blood and Laurels* are the ones primarily responsible for knowing their motivations and goals and taking action rather than being constrained by beats in the way *Façade*'s characters are [9]. Additionally, in Versu stories, characters and roles are defined separately, allowing for a combinatorial approach to casting characters in roles—similar to the exploration of generative characters we describe in Section 3.1.

In contrast to our positioning of players as different aspects of a storyteller, both *Façade* and Versu cast players in a specific role for each playthrough. This means that whatever creative power a player has in these stories is limited mostly to the creativity felt by an actor [17] rather than a storyteller and the acting out of the particular version of the story is the goal of these experiences. As a result of this focus, the social simulations found in these games are concerned with maintaining the integrity of the world and plot more than allowing a collaborative construction of a story.

Unlike our work discussed in this paper, the initial work toward Comme il Faut (CiF), a social simulation system built around modeling character interaction through social games [19], began before development on *Prom Week*, the key experience created with it [20]. In this way *Prom Week*'s development was, at least partially, guided by the constraints of CiF's initial design. We have approached the development of a social simulation-focused game from the opposite direction, first identifying an experience we want to give players and then developing a social model to help implement those design goals.

Bad News [27] is a computational theater experience where a player must notify the next of kin of a deceased person. The bulk of the computational side of *Bad News*'s social simulation is run before the game begins and is updated on the fly through direct human input. Additionally, all character interactions are mediated through a live human actor. This live performance allows for a significantly higher amount of flexibility in characterization and interaction than any of the purely computational systems we have described and is part of the reason we are interested in building a live, local multiplayer experience.

While social simulation has been used successfully as part of narrative experiences, it has not been brought to bear in the world of story construction to the same degree. One of our goals with this project is to marry the liveness found in the play of *Bad News* with the liveness of the social simulation found in the likes of *Prom Week* to allow for a reactive and constrained environment to build stories in. As Laurel describes in *Computers as Theater* [17], constraints on players (and actors) can encourage rather than discourage creativity and we view our usage of social simulation as providing these constraints to help encourage player creativity.

3.3 Story Construction Play Experiences

In his dissertation, Reed notes that "sculptural fiction," a type of interactive narrative centered around the creation of a story graph rather than the traversal of one, is a reaction to the limitations of graph based interactive story games [22]. This explicit move away from pre-built graphs was one of the major inspirations of the direction we decided to explore using the prototypes discussed in Section 3. Reed points to *The Ice-Bound Concordance* as a culmination of his work with sculptural fiction as it embodies the core elements of sculptural fiction [22]:

- Low-cost and reversible decisions
- The expressive exploration of possible stories
- A system that highlights potentially interesting additions
- A high-level view of the entire story

Our goal of creating a mixed-initiative table-top-esque experience means that, unlike with *The Ice-Bound Concordance*, the computational system does not have to be the primary facilitator of all of these elements. It is, however, still responsible for highlighting what it thinks are relevant additions and helping make decisions malleable. By offloading part of the expressive exploration of stories to the two players, they in turn have a greater stake in the *shared authorship* [1] [26] with the computational system. This goal of shared authorship is also our reason for giving the two players separate parts of the whole story domain to take ownership over and having the computer mediate between those levels.

In providing players with additional scaffolding to support the creative process at the cost of full control over the kinds of stories they can tell, these story construction experiences can arguably be usefully viewed as casual creators [7] for the storytelling domain. This is the perspective that we take in our own work: we are attempting to build a casual creator that supports the collaborative construction of a certain specific kind of mystery story, and the features of our system are therefore explicitly intended to provide support for player creativity.

4 PROTOTYPES

Over the past six months, we have constructed a series of prototypes to develop our design. In keeping with Gingold and Hecker's guidelines for prototyping [11], we crafted each prototype to answer a specific question or questions about the design space in which we are working. Prototyping work alternated between our lab's weekly group meeting and a series of smaller, more task-focused meetings that were set up on a week-by-week basis. Meetings ran from 1-3 hours each, with the size of the group varying from 3-6 for smaller meetings up to 10 for whole-group meetings.

4.1 Character Generation

For character generation, we focused on creating prototypes modeling characters and their relationships to one another. One of the elements we wanted to explore in this space was what elements we wanted to be fundamental to each character, and how that could drive the kinds of interactions between characters and thus stories we told with those characters. We also wanted to investigate how creating characters would work in a generative system. Would our method for generating characters and their relationships to one another consistently provide enough variety for each playthrough to seem novel, while still providing consistently compelling stories for players? And if not, how could we tweak the current way we generate characters or make them in new ways to mitigate these problems?

4.1.1 Paper Prototypes. For our character generation paper prototypes, we began with modeling out what elements we wanted characters to have. These were elements that should make the character unique, and help to create interesting stories between characters in a generative storytelling process. We modeled several distinct traits for each character, including:

- Two *values*, representing what the character holds important and defining what actions they would consider to be a crime. Values also served as a point of affinity or contention with other characters. We initially began prototyping with a set of eight values, chosen based on the themes we desired to emphasize within the game and their imagined potential for interesting storytelling: *science*, *survival*, *communalism*, *funding*, *comfort*, *order*, *faith*, and *progress*. Our hope is that, if we reify the themes we hope to address as mechanics, players will naturally find themselves constructing stories that turn on these themes. We are inspired here by a similar approach taken in the design of *The Shrouded Isle* [10], which the designers refer to as an instance of *subjective simulation design* [13].
- One primary and one secondary *character trope*, pulled from a manually filtered list of character tropes originally drawn from TVTropes.org. We include these primarily to support the players in imagining a personality for each character.
- *Occupation*, defining the character's role in the observatory. This could help to inform the character's motivations, and could also be used to build out relationships between characters, as mentioned below.

We also used paper prototyping to model characters' relationships to one another. Because this generative storytelling game is built on a social graph of relationships between characters, it was important to provide these relationships, both to help players know more about each character, and to provide more context for that character's motivations (for instance, why one of them might or might not be the perpetrator of a crime). In our prototypes we modeled several different elements of relationships between characters:

- Formally recognized relationships, such as the professional relationship between a professor and a student or the romantic relationship between a married couple. These were treated as semi-permanent and unlikely to change in the course of play.
- Valence of characters in their feelings of one another—whether a character liked, disliked, or felt neutral to another character. These could change through play, for instance if one character performed an action that another disliked.
- Other character relationship modifiers—we looked at other interesting ways to connect characters to one another, such as defining characters as rivals to one another, or denoting that one character has a secret crush on another.

Paper prototype evaluation happened over several different play sessions. They mainly involved writing down character traits and relationships on cards, passing them out randomly, and then describing various reasons why each character might commit a crime and how elements such as relationships and traits might influence characters' affinities to one another. We used different combinations

[1] Samuel defines shared authorship in his dissertation as "the act of creating something with someone else that could not have existed without the both of you."

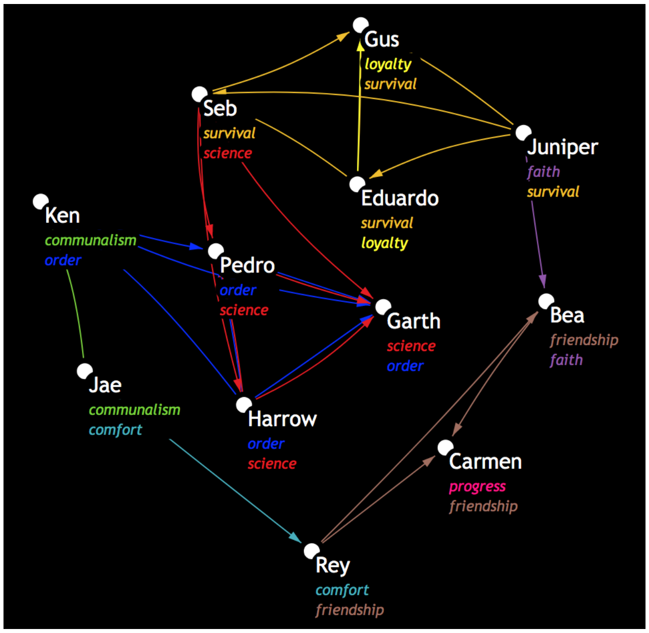

Figure 1: A visualization of a generated cast of characters. Colored arrows between characters indicate which values they share; shared values draw characters closer together.

of character traits and relationships, and ended up incorporating those that worked well together in our digital character generation prototype.

4.1.2 Digital Prototypes. From our paper prototypes for character generation, we were able to construct a digital prototype of a social graph generator. This generator worked as a quicker stand-in for methods we had previously used to generate characters and their relationships to one another, and that could then be used for other prototypes, such as generating transcripts or crime graphs from a list of characters. Because the digital prototype used constraint solving to build out the list of characters and relationships, it could also more easily ensure that the list of characters and their relationships made sense with one another, for instance ensuring that every student had a professor they worked for. These variables and constraints could be easily tweaked and their outputs evaluated using the digital prototype, so that we could figure out combinations that make sense and lend themselves to diverse and interesting stories.

Running the generator, we get a list of characters and relationships such as the following:

```
has_modifier("Bob","celebrity_status").
has_modifier("Georgia","has_funding").
has_modifier("Henry","secret_expert").

has_profession("Alice","professor").
has_profession("Bob","student").
has_profession("Georgia","student").
has_profession("Henry","professor").
has_relationship("Alice","Henry","mentor_of").
has_relationship("Bob","Henry","working_for").
has_relationship("Georgia","Henry","working_for").
has_relationship("Henry","Bob","rivals").
has_relationship("Henry","Bob","professor_of").
has_relationship("Henry","Georgia","professor_of").

has_value("Alice",1,"Order").
has_value("Alice",2,"Funding").
has_value("Bob",1,"Science").
has_value("Bob",2,"Funding").
has_value("Georgia",1,"Faith").
has_value("Georgia",2,"Comfort").
has_value("Henry",1,"Order").
has_value("Henry",2,"Faith").

personality("Alice","primary","Parent Figure").
personality("Alice","secondary","Optimist").
personality("Bob","primary","Innocent").
personality("Bob","secondary","Socially Awkward").
personality("Georgia","primary","Bad-to-the-bone").
personality("Georgia","secondary","Outspoken").
personality("Henry","primary","Eccentric").
personality("Henry","secondary","Optimist").
```

This list of characters and relationships could then be evaluated by creating transcripts of stories based on the list of characters.

4.2 Transcripts

Our intention in creating transcripts was to model the play experience, particularly the interactions between players and the computational system. By playing out different interaction patterns between the detective player, the Agatha player, and the system, we could test out the interactions we believed would support the kinds of experiences we wanted players to have, and discover what kinds of actions we wanted players to be able to take that were not already afforded to players. We could also use transcripts to examine the kinds of stories that we could generate through this interaction, evaluating story length, plot arcs, and the dynamics between characters in each of the stories. In the construction of these transcripts, we had a human step in to play the role of the AI system; this enabled us to assess the kinds of prompts we wanted the system to generate, clarify what the system would need to know in order to generate those prompts, and evaluate different styles of interaction between between the three different agents.

The majority of possible player/system interactions we modeled involved both the detective player and the Agatha player working in tandem with the knowledge base to construct a story. Under this interaction model, the story begins with the system providing a story hook to the players (for example, "Alice's [research] has been sabotaged."). The detective player can then interact by investigating various system-recognized words (represented here in square brackets), choosing a character from the list of all characters to perform this action (for example, "Bob investigates research."). Three possible results of this action are generated by the system and given to the Agatha player, for instance:

- Data about the [astronomical event] that Alice was collecting was altered and is now useless.
- Data about the [astronomical event] that Alice was collected was deleted.
- An [important tool] for Alice's research was broken.

The Agatha player chooses one of these options as the continuation of the story. This updates the knowledge base and adds this event to the story, which can be expanded upon by the Agatha and detective players. With this new part of the story now in place, the detective player can pick another action to take, either with their current or a different character, and the loop begins again.

We created text transcripts by taking turns performing each step (one person performing the system step, the next performing the Agatha step, the next performing the detective step) and transcribing the results in a online collaborative text editor. Each person was able to see and react to what the others typed, and the group was able to collectively discuss higher-level elements of the story—such as possible directions in which the story could proceed, the pacing of development, and specific moments of character interaction we wanted to see realized—as the transcript was written.

4.3 Crimegraph + Buckets

Our next prototype focused on the structure of character actions and the possible relationships between them, with an eye to answering one question in particular: given an aesthetic prohibition on characters who are "evil", "monstrous", or otherwise driven by nature to perform severe destructive acts, could we devise a plausible form of "motivation arbitrage" by which a basically morally good character among a cast of other basically morally good characters could nevertheless end up committing a severe crime?

With this question in mind, we began to use our existing cast generator to support the manual collaborative construction of "crimegraphs": graph structures that describe the relationship between character actions, with an eye to gradual and plausible escalation in crime severity. For instance, a perceived slight by one character against another might motivate the second character to take direct or indirect revenge on the first, and the action taken in revenge might be slightly more severe than the original slight itself. As in earlier transcript construction sessions, we repeatedly met as a group, generated a cast of characters, and then used this as the seed from which to manually and collaboratively construct a crimegraph involving the generated cast. These crimegraphs were drawn out on a whiteboard as we constructed them, so each session resulted in the collaborative construction of a concrete artifact similar to the written transcripts from earlier sessions. Sometimes we started with a high-severity seed crime, such as murder, and worked backwards to retroactively construct a plausible chain of motivations. On other occasions, we began with a low-level "crime" (some as apparently innocuous as accidentally stealing a colleague's lunch from the communal fridge, failing to clean up a common area, or perceived rudeness, "motivated" only by innate character traits such as carelessness) before gradually working our way upward to higher-severity actions.

In the process of manually constructing these graphs, we began to routinely find ourselves blocked by uncertainty regarding how to proceed, particularly at the points of trying to decide what actions characters might reasonably perform; what motivations they might have for performing these actions; and what kinds of evidence might be left behind by a crime. As such, we collaboratively developed several lists of crimes, motives, and evidence types from which we could randomly draw items for inspiration. These began as physical decks of cards but were soon moved into a Tracery [6] grammar from which we could rapidly and repeatedly draw arbitrary combinations of items, such as a single crime, a single motivation, and three evidence types. Storing these "buckets" of static options digitally also enabled us to rapidly revise and extend them with additional options.

Insofar as we were successfully able to construct plausible crimegraphs while using the buckets for inspiration, this prototype confirmed that our conception of motivation arbitrage could function as intended. Manual construction of crimegraphs, however, proved to be a slow and laborious process, and the consensus of the team was that the computer was not being sufficiently leveraged to support this process. Our frustration with the manual process of crimegraph construction led directly to the implementation of our final prototype.

4.4 Closing the Loop

Our final prototype to date was intended to answer three questions. First, could filtering the pool of possible character actions according to the current state of the social simulation result in more plausible-seeming automated suggestions for character actions? Second, what could be gained by "closing the loop" and allowing players to inform the system of which character action they chose to perform? And third, to what extent would giving players access to a set of *story sifting* [24] functions that they could use to search the simulation for narratively interesting situations help them develop the story more rapidly and in more compelling directions?

This prototype consisted of several parts. First, we recreated an existing cast generator from an earlier prototype and encoded its output as facts in a logic database, specifically a Clojure implementation of a Datalog [32] database. Then, we devised a pool of possible character actions, each one associated with a query that can be run against the database to bind a character who might perform this action and other variables as needed (such as a "target" for the action, e.g., another character) on a per-action basis. When players query for possible character actions (either globally or in a search for actions that a specific selected character might perform), the prototype iterates over the possible actions and attempts to establish a set of valid bindings for each one, filtering out those actions for which no valid set of bindings can currently be established. Then it randomly selects a number of successfully instantiated actions from the list and presents these to the player as options. The player may then pick one of these actions to perform, resulting in the game state being updated as described in the action definition. At any point, players may also run any of a set of query functions against the simulation to seek out narratively interesting situations such as grudges, jealousies, and other possible sites of conflict or development within the storyworld.

Though we implemented only a limited pool of possible character actions for this prototype, we found that filtering actions based on the current social state did result in a subjective improvement to the relevance of the suggested actions. Initially, some generic fallback actions had a tendency to come up too often, even when more flavorful actions with more specific preconditions were possible, and to crowd out these more flavorful actions by their presence. This was primarily a consequence of our simple unweighted random approach to the selection of possible actions, and was straightforwardly mitigated by weighting actions with more specific preconditions as more likely to be suggested when their preconditions are met—or, in other words, by adopting a naïve *salience-based* [28] approach to action selection.

The subjective effects of "closing the loop" by allowing player-selected actions to update the social simulation state, on the other hand, seemed to be negligible. Even when one character took an action that several other characters all interpreted as the commission of a severe crime, the next pool of suggested actions would often freely mix direct reactions to the high-severity action with other, much more innocuous or unrelated actions. This seems to suggest that there is a real need for an action suggestion system that is more deeply aware of the salience of possible actions from a player-facing perspective: a system, perhaps, that is capable of prioritizing the suggestion of actions that directly respond to recent actions, or that otherwise prioritizes actions based on some notion of which characters, relationships, situations, or other storyworld entities are currently "in focus". Earlier work on perceived event salience in narrative generation [3, 14] has suggested that events that share a common *protagonist*, *time*, *space*, *casuality*, or *intentionality* are likely to be perceived by readers as more salient to one another; in future prototypes, we may explore the possibility of using these dimensions to further prioritize possible action suggestions.

This prototype provided players with only a few story sifting functions (for instance, one for identifying potential jealousies between characters, and another for identifying pairs of characters in which one character likes the other and the other character dislikes the first.) Nevertheless, players consistently stated that they found these functions helpful, particularly when they felt they had reached an impasse in developing the current line of the story. Providing them with a set of tools for locating other potentially narratively interestingly situations besides the one they were currently focusing on supported the development a "braided" plot structure, with several intertwined threads of narrative intrigue between which the narrative focus would occasionally move. Subjectively, this seemed to result in the construction of more compelling stories overall.

In addition to the questions we explicitly set out to answer, we also found that this prototype was successful at encouraging conversations about character motivation among the players. However, the system's understanding of character motivation remained largely internal rather than being exposed to players, resulting in frustration when players wanted the system to help them develop their own understanding of a particular character's or action's possible motivations. Going forward, we intend to be more explicit in surfacing possible motivations for each suggested action, so that players can see what motives the system believes to be in play.

We have also discussed the possibility of separating impulses from actions, so that players can view the full list of a character's current impulses and then filter possible actions for that character according to a specific impulse, or view which of several possible impulses might reasonably motivate a given suggested action. Under such a two-part structure, impulses would take the form of abstract motivated intentions, such as a character's desire to somehow take revenge on another character who had harmed them, while actions would function as concrete realizations of one or more impulses. A character seeking revenge on a scientist might sabotage the scientist's experiment, while a character seeking revenge on a skiier might hide or damage the skiier's ski equipment; in either case, the abstract impulse to get revenge would serve as motivation for the vengeful character to perform concrete actions that are tagged as harmful to the target of their vengefulness.

5 LEARNINGS FROM PROTOTYPES

Based on our prototyping process so far, we found that certain design elements present in one or more prototypes stood out as strongly supportive of player creativity, while others proved less useful, impractical to implement, or fell by the wayside for other reasons. Below, we briefly catalogue some of the design elements we intend to retain going forward, including some reasoning as to why each element might prove effective from a creativity support perspective.

5.1 Editable Text Transcript

One element of gameplay that felt like a fun and rewarding way to support creativity was allowing players an editable transcript of the story that they could use to expand out the story during play. This interface is a collaborative text editor that populates with each system prompt, but can be added to by players, creating a transcript of the entire story created in the play session. Although the system provides text prompts for what happens next, we found that much of the fun and flavor of the story came from both the Agatha and detective players expanding on the story prompts provided by the system. One example here is taken from a transcript session where a character, Cindy, is discovered with a dead body and asked if she was the murderer. The system (played here by a person) provided a prompt:

> Cindy quickly tries to defend herself

which was expanded out by the Agatha and detective players to read:

> Cindy looks up from Dan's body, immediately stammering, "W-what? No! No, of course not!"

Another instance of this is in characters explaining elements of the world to one another:

> System prompt: Alice suggests Fred might have been involved, because he was excluded by Dan
>
> Expanded text: "That doesn't make any sense! If anyone was going to kill Dan it would've been Fred. Dan's always leaving him out."

Finally, this can just serve as a colorful retelling of the system prompt:

> System prompt: Kate accuses Cindy of Dan's [murder]
>
> Expanded text: "YOU MURDERED DAN!" Kate shouts.

While this editable text expansion does have some limitations—for instance, as the system does not reason over the player-provided text in any way, it may not recognize significant changes made to the story by the players—we found that in our own transcripts these constraints actually helped support player creativity. By providing a prompt and a limited amount of story to narrate each turn, we mitigate some of the blank-canvas paralysis that comes with creative work, and may be especially daunting for long-term storytelling. By making this editable text transcript available inline as part of the gameplay experience, we want to encourage player creativity and storytelling directly in the game as part of the play process, and ensure they do not have to go elsewhere for telling the stories of the characters, as is often the case with chronicling the stories of other simulation games.

5.2 Action Generation

Another element of the prototypes we found successful, particularly in our transcript writing process, was having the system generate suggestions for what could happen next in the story, then providing prompts to players to continue the story based on these possibilities. By providing the Agatha player with a limited set of options for continuing the story, we reduce the difficulty of making decisions about the story, making each turn for that player relatively simple. On a higher level, we reduce the creative difficulty of storytelling in general from an open-ended problem (what should happen next, what do these characters do) to an easier choice of picking an option from the list of system prompts. This also allows players the ability to choose options that are narratively interesting or follow along with what they want to see happen in the world.

Because the story is in part built through this action generation and system prompting, we can utilize the system to effectively achieve certain ends. One advantage of this is that we can use the system to survey the possibility space and thus surface to the players narratively interesting paths for the story to continue down. This can also be used to steer the player down different kinds of story paths, from practical needs for the story (such as providing different ways to end the story for players) to meeting more of our thematic goals for the project. For instance, if action generation focuses on the connections between members of the community and their values and interactions, we can use this action generation system to drive stories that have these elements at their core.

5.3 Two Levels of Story / Simulation Structure

One of the aspects of storytelling we noticed through our prototyping was the elements of story structure as the story developed. Many of our transcripts focused on individual actions taken by each character and the immediate response, while others focused on broad strokes of the story structure from beginning to end, without worrying about each character's individual actions. In general we found some tensions between low-level actions (what characters would like to do in each moment) moving the high-level story structures such as overarching narrative arcs and actions in the world that run counter to what characters would want for themselves. We wanted players to consider both elements—as both characters and story authors—to create an overall cohesive story still highly motivated by characters' desires. But without a way of formalizing this in play, we found, especially in transcripts, that one or the other was often lost, with stories either too caught up on turn-to-turn actions without any kind of resolution or higher-order scenes, or overly-rigid plot arcs that didn't build off of moment-to-moment character interactions. In order to facilitate both of these elements of storytelling, we hope to create a two-level action structure. This would allow players to have control of both high-level structures to the story as well as control at the character level, determining what happens moment-to-moment. The hope is that by separating these out into two levels, players will be able to work on both an authorial level and control low-level moves made by characters, and that both of these levels will better be able to compliment one another.

5.4 Story Sifting

Story sifting, also known in some sources as *story recognition* [25], refers to the process of "automatically recognizing interesting narrative material embedded in the morass of accumulated simulated material" that is generated in the course of simulating a storyworld [24]. Our final prototype provided players with a set of story sifting functions, expressed in terms of Datalog queries, that they could run against the logic database at any time to discover narratively interesting possibilities that they might be overlooking.

From a creativity support perspective, we found two key benefits to this approach. First, providing players with tools to proactively seek out new narratively interesting situations gives them a way to shift focus when they feel they have reached an impasse, often to an as-yet overlooked or underdeveloped part of the storyworld. Second, whenever players make use of these tools, they are reminded of underdeveloped situations that might be worked into the story or serve as candidates for future development later—especially helpful when players are attempting to manage several narrative threads at once and may need to be reminded of threads that they have temporarily elected to leave on the back burner.

In the future, we may further extend our use of story sifting by automatically running some story sifting queries periodically in the background and proactively surfacing their results to players, rather than waiting for players to seek out these tools themselves.

6 CONCLUSIONS

Although our eventual intended play experience remains a work in progress, our prototyping process was decidedly successful in guiding our exploration of an unusual design space for PCG-based [30] games. Designing based on a set of specific player experience goals, crafting prototypes to seek answers to specific design questions, and allowing humans and analog systems to stand in for AI or generative processes proved effective in supporting our development of creativity support techniques for collaborative story construction games. Moreover, it is our hope that—by documenting not only the results of our prototyping process, but also the process itself—our experiences may serve as a guide to the developers of other, similar PCG-based games going forward.

REFERENCES

[1] Avery Alder. 2013. The Quiet Year. https://buriedwithoutceremony.com/the-quiet-year.

[2] Gabriella Alves Bulhoes Barros, Michael Green, Antonios Liapis, and Julian Togelius. 2019. Who killed Albert Einstein? From open data to murder mystery games. *IEEE Transactions on Games* (2019).

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

[4] Raymond Chandler. 2002. *The Simple Art of Murder*. Knopf Doubleday Publishing Group. https://books.google.com/books?id=z40A6qE9tucC

[5] Raymond Chandler, Dorothy Gardiner, Kathrine Sorley Walker, and Paul Skenazy. 1997. *Raymond Chandler Speaking*. University of California Press. https://books.google.com/books?id=aI1NaxoYsJYC

[6] Kate Compton, Ben Kybartas, and Michael Mateas. 2015. Tracery: An author-focused generative text tool. In *International Conference on Interactive Digital Storytelling*. Springer, 154–161.

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

[8] Markus Eger and Chris Martens. 2017. Character beliefs in story generation. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[9] Richard Evans and Emily Short. 2014. Versu - a Simulationist Storytelling System. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 2 (2014), 113–130.

[10] Kitfox Games. 2017. The Shrouded Isle. http://www.kitfoxgames.com/shrouded-isle.

[11] Chaim Gingold and Chris Hecker. 2006. Advanced Prototyping. https://www.gdcvault.com/play/1013252/Advanced.

[12] Ian Horswill and Ethan Robison. 2018. What's the Worst Thing You've Ever Done at a Conference? Operationalizing Dread's Questionnaire Mechanic. *Proc. Experimental AI in Games* (2018).

[13] Jongwoo Kim. 2018. Subjective Simulation Design: Ludonarrative Congruence in The Shrouded Isle. https://www.youtube.com/watch?v=oV5PjvcVky8.

[14] Christopher Kives, Stephen G Ware, and Lewis J Baker. 2015. Evaluating the pairwise event salience hypothesis in Indexter. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

[15] Sheldon Klein, John F Aeschlimann, David F Balsiger, Steven L Converse, Claudine Court, Mark Foster, Robin Lao, John D Oakley, and Joel Smith. 1973. *Automatic novel writing: A status report*. Technical Report TR186. University of Wisconsin.

[16] Sheldon Klein, John D Oakley, David J Suurballe, and Robert A Ziesemer. 1971. *A program for generating reports on the status and history of stochastically modifiable semantic models of arbitrary universes*. Technical Report TR142. University of Wisconsin.

[17] Brenda Laurel. 2013. *Computers as Theater* (2nd. ed.). Addison-Wesley Professional.

[18] Michael Mateas and Andrew Stern. 2005. Structuring Content in the Façade Interactive Drama Architecture.. In *AIIDE*. 93–98.

[19] Joshua McCoy, Michael Mateas, and Noah Wardrip-Fruin. 2009. Comme il Faut: A system for simulating social games between autonomous characters. (2009).

[20] Josh McCoy, Mike Treanor, Ben Samuel, Brandon Tearse, Michael Mateas, and Noah Wardrip-Fruin. 2010. Authoring game-based interactive narrative using social games and Comme il Faut. In *Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate*. Citeseer.

[21] Henry Mohr, Markus Eger, and Chris Martens. 2018. Eliminating the impossible: A procedurally generated murder mystery. *Proc. Experimental AI in Games* (2018).

[22] Aaron Reed. 2017. *Changeful Tales: Design-Driven Approaches Toward More Expressive Storygames*. Ph.D. Dissertation. UC Santa Cruz.

[23] Ben Robbins. 2011. Microscope. http://www.lamemage.com/microscope.

[24] James Ryan. 2018. *Curating Simulated Storyworlds*. Ph.D. Dissertation. UC Santa Cruz.

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

[26] Ben Samuel. 2016. *Crafting Stories Through Play*. Ph.D. Dissertation. UC Santa Cruz.

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

[28] Emily Short. 2016. Beyond Branching: Quality-Based, Salience-Based, and Waypoint Narrative Structures. https://emshort.blog/2016/04/12/beyond-branching-quality-based-and-salience-based-narrative-structures.

[29] Adam M Smith and Michael Mateas. 2011. Answer set programming for procedural content generation: A design space approach. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (2011), 187–200.

[30] Gillian Smith, Elaine Gan, Alexei Othenin-Girard, and Jim Whitehead. 2011. 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*. ACM, 7.

[31] Andrew Stockdale. 2016. ClueGen: An exploration of procedural storytelling in the format of murder mystery games. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[32] Jeffrey D Unman. 1989. *Principles of database and knowledge-base systems*. Computer Science Press, New York.