(Re)telling Chess Stories as Game Content

Eric Buckthal
California Polytechnic State University
1 Grand Ave.
San Luis Obispo, California
ebucktha@calpoly.edu

Foaad Khosmood
California Polytechnic State University
1 Grand Ave.
San Luis Obispo, California
foaad@calpoly.edu

ABSTRACT

Procedural story generation is an active and increasingly important area of interactive entertainment. Some of the most difficult challenges with the concept are generation and managing of dramatic elements with high degree of variation that are both internally consistent and entertaining. If an authoring system could access a large repository of proven sets of domain specific dramatic action sequences with guaranteed internal structure, consistency and known entertainment value, could it generate unique stories with the same drama in a different domain? We consider chess games as activities capable of being dramatic and entertaining in their own domain. Chess can inspire the same kind of audience reaction as novels, plays, and films. It is subject to the same kind of cultural intrigue. In this paper, we use chess games as drama-producing events to generate traditional stories in multiple genres that contain no chess-specific content. Our system accepts chess games in Portable Game Notation (PGN) and outputs text stories in any of the supported genres: Romeo and Juliet, Star Wars (rebels versus empire), zombie story and combat story. We find that to a significant degree audiences familiar with chess can correctly match a story to the chess game it was generated from, suggesting some level of transfer with respect to dramatic elements.

Keywords

story generation, drama management

1. INTRODUCTION

We are interested in generating stories to serve as content in interactive entertainment products. In-game stories-as-content exist within the larger game narratives already: The main plot backstory, character biographies, object histories, in-game novels, newspaper content, movies or plays. Almost all AAA games, especially those in the action-adventure genre, contain backstory [10] or many other stories used for background, motivation, plot exposition, or just as decoration to create a more immersive universe. As we argue, hand-crafting all these stories can quickly become intractable.

Generating stories artificially using traditional story generation methods has too many shortcomings [16] for most situations. Purely automatic generation tends to be either just as resource-intensive, or lead to simplistic, lower-quality plots without much variation.

We propose a new approach to this problem: by using chess games. We believe records of chess games represent real, cohesive, self-contained and conflict-driven plots involving major and minor characters. These are often the very elements that are difficult for automatic story generators to create. These records exist in abundance. Further, some chess games are easily describable as "dramatic" and "exciting" in terms of involving "plot" twists and countering expectations. There are established metrics to measure progress or setback at any point in a chess game. There are well-known move combinations with measurable consequences that can be detected and described. Chess itself is a human created game based on a narrative, so the events it describes could be transferable to another story domain with at least some retention of the drama and the emotional evocations. Or so our hypothesis goes.

The structure of the rest of the paper is as follows: Section 2 is about motivation. Section 3 describes related work. Section 4 details the design and implementation of the story generation system. Section 5 analyzes the results of our user study. Section 6 is the conclusion and Section 7 is future work. In Section 8, we present a detailed example story and the PGN input it was generated from.

2. MOTIVATION

Imagine an action-adventure game with a similar setting as *Assassin's Creed IV: Black Flag* (Ubisoft) that features an NPC sailor who has recently been through a large naval engagement involving dozens of ships. Let us say the player is able to engage in conversation about the battle, and has the formal affordances to inquire about the details. The NPC then proceeds to tell a story describing the battle. How would the narrative content required for this conversation be created?

At the moment, for almost all games these narratives are created manually by authors and historians (in case of historical fiction) involved in the project. A real story about the naval engagement is probably written first, then particular details worked out as dialogue responses for the NPC. It is the same way as the character's other responses (back story, main plot fragments, personality-revealing speech, etc.) are woven into the dialogue tree for the character.

Now, let's imagine a dozen stories the NPC could tell about different events. Let's further imagine we want these stories to have certain characteristics depending on the game's main plot requirements: For example we may want to have the battle to have been won or lost, short or long, a clear route or a dramatic come-from-behind victory – all depending on what we want the player to take away from that particular interaction. Oh, and also imagine we want a different and unique story for every play-through!

We hope it is clear that such requirements cannot be fulfilled by hand-crafted story writing in large scale. Stories in the aforementioned example are *content* and much like some other content they could be procedurally generated. But in generating in a purely procedural way, we incur some of the weaknesses of the current methods in procedural story generation. Even games like *Dwarf Fortress*, that generate extensive backstories and world histories have shortcomings like repetitiveness and lack of dramatic tension, as described by Togelius et al. in [16].

3. RELATED WORK

We review various related works in three areas: Chess as narrative, story generation systems and drama management systems. For each section we offer a brief discussion relating and contrasting the works with our own.

3.1 Chess as narrative

Chess is a well-studied classic interactive strategy game. Most commenters are in agreement that chess has some narrative dimensions. Whether the implied fictional setting or historical backstory of chess is essential to the game experience has been subject of much debate – often when chess is used as an example in context of what has been called the "narrativists versus ludologists debate" [6].

Pearce compares the chess "story line" to that of MacBeth [12]. Bowler argues, "...chess has a fiction. Players pretend (even passively) to be two armies with politically divergent interests battling it out to see who wins in a contest of might makes right" [3]. Forrestall goes perhaps further, saying chess phases (opening, middle game and end game) have an "unmistakable resemblance to the 3-act narrative structure of screenwriting and theatre" [5].

Laskin equates chess actions with human emotions when he writes:

> The vocabulary of the chess pieces is richer than commonly perceived: ambition —in performing the work; fury —if obstructed from doing the work; desperation —for undeservingly suffering a misfortune; rejoice —for getting a lucky break; mockery —of the opponent thwarted by the piece; hatred —for anyone threatening the king, notably "anybody" pinning the piece; cussing —opponent's piece which in any way constraints the piece; laughter —in sidestepping a trap, and even the simple witty expression. -*Laskin, 1937* [1]

Vladimir Volkenstein considers peripety as the essence of beauty in a chess game.

> Beauty in chess springs from the expediency of moves involving peripeteia, from the unexpected and seemingly paradoxical resolution of a difficult position. These are the instances when sacrifices take place. Victory achieved through gradual accumulation of tiny advantages without enchanting combinations could be called solid, well-composed, perhaps even instructive and fine, but there is no way it could be called beautiful. -*Volkenstein, 1931* [2]

There can be no doubt that chess games contain a series of events that can be described as structured, internally consistent narratives. This is a record of occurrences and meaningful decisions, subject to strict rules, that are made by two intelligent agents with opposing objectives. We might call the PGN-coded chess game, a kind of shorthand for these narratives. For the aficionados, such narratives capture and communicate the dramatic events of the game that was played. We believe the same narratives can be retold as traditional stories with similar dramatic qualities.

3.2 Story generation systems

Many good systems have been made over the past decades. Here we concentrate on the most relevant of those systems.

An important approach to creative story generation is BRUTUS [4]. The authors of BRUTUS put much emphasis on variability and creativity, particularly "creative distance" (the mark of how different and unexpected the generated story is, compared to the input data). However, maximizing creative distance necessarily comes at the expense of story cohesion and desired discourse goals.

The works of Pérez y Pérez address two goals through an explicit evaluation phase of generated stories. MEXICA [13] is a story generating system that includes an "engagement" and a "reflection" phase. The system attempts to solve a complex constraint-satisfaction problem (engagement) while breaking impasses using situations from previously encoded stories (reflection).

Pablo Gervás in [7] takes an interesting approach to story generation. He uses chess games for similar reasons, but chooses compelling stories by selecting and scoring the best "focalized" experiences within the chess game. Events are described relative to each chess piece and it's limited field of vision. The best such experiences are chosen as the representative fragments of the overall story.

[1] as appears in [8](pp. 78-79).
[2] ibid.

In our view some level of predictability and control of discourse based on input is just as important as creativity. Such control is necessary to produce the desired user experience. The authors in [13] and [14] have presented two different approaches to the discourse control problem. Gervás work does not attempt to transfer or modify the chess domain and contains a necessary one-to-one model of characters to chess pieces.

3.3 Drama management

Roberts, et al. use chess in a case study to demonstrate how an AI drama management system could use analysis of player moves to inform narrative generation systems in order to present better user experiences.

> While the features of chess bear little resemblance to narrative features, they can form the basis upon which narrative reasoning for chess can occur by correlating them to known narrative heuristics such as location flow and plot homing/mixing or determining how they relate to evaluations of suspense and dramatic arc. [15]

While the authors did not build a system processing chess moves, we believe the approach is sound. The challenge is to match crucial dramatic story points to equally strategic points in the chess game. For this to happen, we identify a set of features that help us determine dramatic points in the chess game.

The Interactive Drama Architecture (IDA) [9] proposes three key components to interactive drama: the Writer, the Director, and the User. IDA gives the Writer a reasonable amount of control in plot specification, while considering the desire of the User to control how the drama unfolds. IDA is a first-order logic representation of several components for each scene which consider the scene as the smallest dramatic moment; the Writer's goal is to move the story forward as a whole.

In Search-Based Drama Management (SBDM) [11], a player's concrete experience in the world is captured by a sequence of "Player Moves", abstract plot points that a player activity can cause. A single Player Move may encapsulate 5 or 10 minutes of concrete player activity in the world - moving around, picking up objects, interacting with characters and so forth. When the concrete activity adds up to a story significant event, then a Player Move is recognized. An SBDM has a set of System Moves available that can materially alter the world (e.g. move objects around, change goals in characters heads, etc.) in such a way as to encourage or obviate a Player Move. System Moves give the SBDM a way to warp the world around the player so as to make certain Player Moves more or less likely. Besides the System Moves, the author also provides the SBDM with a story specific evaluation function that, given a complete sequence of Player and System Moves, returns a number indicating the goodness of the story. Whenever the drama manager recognizes a Player Move (plot point) occurring in the world, it projects all possible future histories of Player and System moves, evaluates the resulting total histories with the evaluation function, and propagates these evaluations up the search tree (in a manner similar to game-tree search) to decide which system move to make next which will be most likely lead to a better generated story.

The IDA lessons for our system are mainly that we could benefit from additional annotation leading to richer features which can in turn translate into more complex plot variations. Our game state reconstructor was built primarily with this goal in mind. From SBDM we adopt the idea that a grouping of moves can indicate a discourse act. Though cognitive or semantic groupings would be difficult to determine from a chess record, groupings can nevertheless be made and interpreted.

4. SYSTEM DESIGN AND IMPLEMENTATION

Our system is comprised of five main subsystems: PGN parser, game state constructor, dramatic feature extractor, plot iterator and story skin manager.

The PGN chess game parsing system is used to parse metadata and moves from PGN-coded chess games. The moves within PGN are encoded using the Standard Algebraic Notation (SAN) which does not actually provide the state of the chessboard. A separate state constructor is necessary if we want to recreate the board at every step from a sequence of moves. The state of the board allows us to extract richer features from the game and correlate them to the moves that have been made.

We create a list of features that we extract from the game trace, using the dramatic feature extractor subsystem. Features are generally augmented by labeled chess maneuvers associated with some dramatic descriptions in the chess literature. Some of the features are contextual, that is, they rely on knowing the state of the game after several moves, and some require the eventual outcome of the game.

Our story plot iterator assigns chess features to story plot nodes. Multiple story arcs are possible depending on features. Lastly, the story skin manager uses a multi-level template driven surface text realization approach to instantiate and generate texts associated with plot fragments. For a quick overview of the overall system design see Figure 1.

4.1 Chess move extraction

To generate stories from chess games, we gather 2,367 sample games from `www.supreme-chess.com` [1] in the form of PGN files. Since each game is in a known format, we simply extract the information from each game using regular expressions. The PGN format also includes data about the number of half-moves, players' chess ranking, and the players' home country.

4.2 Chess game state reconstruction

In order to analyze the chess game, we recreate a list of game states representing the location of each piece after each move. The information given in the PGN is sufficient to accurately recreate the detailed chess game. With understanding of the rules of chess, each game state is analysed individually for features derived from the relationships between the player's pieces and the opponent's pieces as well

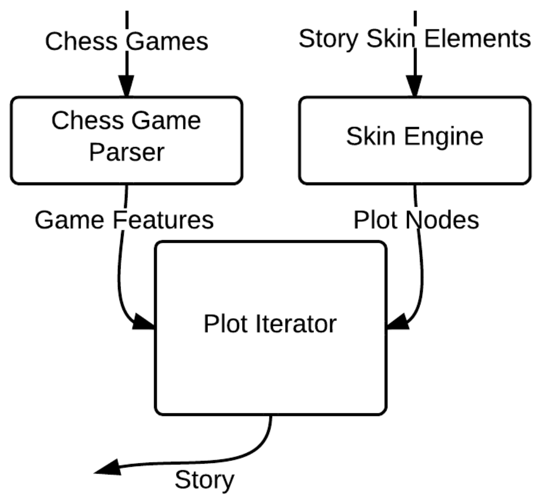

Figure 1: A visual summary of the story generation system

as values of pieces.

4.3 Chess feature extraction

The game of chess is governed by a set of rigid rules, but an impossibly large set of board configurations. In order to narrow down the possible features the system can detect, we consider a set of tactical motif features modelling interesting chess elements and statistical features based on the propagation of a player's techniques. To extract features, we observe the board state after each move. Tactical features only exist when a player attempts to perform an interesting play, but statistical features are derived from each state.

4.3.1 Tactical motif features

We gather the following tactical motif features from the online resource *Chess Tempo* [2].

Pawn Advancement: If a player advances a pawn in the opposing side and there is less than 75% of the original value of all opposing pieces.

Sacrifice: If a player loses a piece it is considered a sacrifice. We've simplified possible sacrifices to be either real or sham. A sham sacrifice occurs when the value lost is taken from the opponent in the next one or two moves. A real sacrifice occurs when the value difference is not restored quickly.

Attack: An attack occurs if a player threatens an opposing piece using one of three tactics. When a player moves one piece, enabling another friendly piece to threaten a previously unthreatened piece, it is considered a *discovered attack*. A player executes a *pin* when it threatens an opposing piece, but if that piece was moved, a more valuable piece would be threatened. A *fork* occurs when a player's piece threatens at least two opposing pieces simultaneously and is unthreatened itself.

Desperado: The desperado is triggered when a player has a piece threatened and undefended, but uses the piece to take an opposing piece in an apparent trade, instead of retreating or defending the piece under attack.

Castle Attack: The castle attack signifies that the opposing player previously castled and now the player has captured any pawn in the file in front of the castled king. Castling by itself is also a feature.

4.3.2 Statistical features

Total Threatening Ratio: The ratio of number of pieces threatening opposing pieces over opposing pieces threatening player pieces.

Total Top Three Moving Ratio: The top three moving count is total of the highest three pieces' move counts. The ratio compares player's to opponent's move counts in order to determine if many or few pieces are used.

Total Top Three Capturing Ratio: The number of captures is the top three capturing count. This ratio compares player to opponent.

Total Value Ratio: The ratio of total value of pieces remaining in each side. In our case, a queen is 9, a rook is 5, a bishop is 3, a knight is 3, and a pawn is 1.

4.3.3 Holistic features

Holistic features are additional story elements associated with the skin that are generated independent of plot nodes but evaluated when plot nodes are chosen by the plot iterator. There are not necessarily holistic features associated with each plot node, but their intent is to incorporate the plot generator's knowledge of the future and past of the chess game as well as the past of our generated story. There are two situations where normal plot generation is augmented by story elements from holistic features.

The first situation occurs when chess features are present that were also present when a resource was spent (e.g. a character was killed) suggesting that this situation is a time to reflect on that resource. The second situation occurs when there is a switch in the current piece value winner. At that time we also are given knowledge about the final outcome of the game. In these situations, elements from a list of corresponding English sentences will be appended to the current plot node's generated text.

4.3.4 Feature translation

While there is little resemblance between features and narrative development, the goal is to find features from chess games that will ultimately translate into interesting plot branches. With each chess game, the system attempts to create a narrative with a dramatic arc similar to the chess game. Sentences of the plot are written and labeled with the corresponding features described in the previous section. There are potentially any number of branches from a specific plot node and the system uses each choice's feature set as the deterministic quality. Plot creation occurs by the plot iterator which is described in a later section.

Table 1 describes example translations in two different skins.

Table 1: Example Corresponding Plot Points

Feature	Skin Plot Point	
	Zombie themed	Romeo and Juliet themed
Sham sacrifice	A character is hurt, but not killed.	Romeo is depressed about the women in his life.
Real sacrifice	A character is killed.	Romeo tries to break up a fight, but Mercutio is killed.
Discovered attack	The characters find a useful tool for their survival.	Romeo falls in love with Juliet.
Pin	The characters successfully set off a trap.	Juliet buys poison from the apothecary and sends Romeo word of her plan.
Fork	A character has communication with another survivor group.	Romeo hears word of Juliet's marriage to Count Paris.
Desperado	A character sacrifices himself to ensure the survival of the group.	Romeo slays Tybalt.
Castle	The characters create a barricade.	Romeo hears of Juliet's death.
Castle attack	Zombies break in through the barricade.	Romeo is confronted by Paris in the tomb.
Total threatening ratio > 1	The zombies seem to be gathering in number, but the survivors kill many.	Romeo kills Paris and finds Juliet.
Top three moving ratio > 1	The zombies are a horde and the survivors must run.	Juliet declares her love for Romeo.
Top three capturing ratio > 1	One survivor is a marksman and proud of his number of zombie kills.	Montague defends Romeo's execution of Tybalt.
Total value ratio > 1	The survivors have plenty of supplies.	Juliet and Romeo find each other alive in the tomb.

4.4 Skins

4.4.1 Plot graph

A plot graph is defined as a set of plot nodes, or events, in a given story universe. It's a super-set of plot nodes for all possible stories within a skin, but it does not specify any story by itself. The architecture allows for plot nodes to link back to previous nodes. An example of what a plot graph may look like for a given skin is shown in Figure 2.

4.4.2 Plot node

A *plot node* is the fundamental unit of the plot graph. It includes templates, all features associated with the templates, and a list of potential next plot nodes. Each plot node also contains a generateText() method that evaluates the current plot and creates an English sentence or group of sentences.

Text generation is done by selecting a random template from the list of templates and doing plot variable instantiation. Variable binding is done recursively, instantiating variables within with other variables from the skin's wordset. While binding, the plot node also keeps track of resources used, deleting a resource word or variable if it is used in a sentence to minimize repetition in text generation. After a template has been converted into a sentence, or sequence of sentences, the template used is removed from the list, so that plot fragment could not be generated again. Once a plot node runs out of templates, the plot iterator can no longer branch to that specific node. This allows for two levels of variability, both on the sentence level and on the word level.

In addition to methods that populate the story, the plot node class also contains methods to determine properties of the plot graph such as shortest distance from the current node to a node that concludes the story and the maximum depth traversed. These properties are ultimately used during the story generation phase of the system to ensure the constructed story matches the chess game.

4.4.3 Templating

The story skin contains a list of nodes each with at least one associated template. Templates are patterns or structures that form a sentence or group of sentences. They are constructed similarly to a standard sentence, but have some sections replaced by variables.

The wordset for a plot is implemented as a Python dictionary constructed from four parts in the skin: words, constants, choices, and resources. Words, the most general type in the wordset, are implemented as a dictionary of lists for each variable. Each list contains one or more strings, from which one will be stochastically chosen for every instance of the variable in the given template. Constants, a type very similar to words, are also implemented as a dictionary of lists. The main difference is that one string is chosen from the list at the beginning and remains constant throughout a given story. Next, the choices type is a list of tuples that contain variables and options. Lastly, the resources type is a dictionary of variables where strings from each list are deleted once they are used. If all values have been deleted the tag defaults to the Words set for replacement.

4.4.4 Story skins

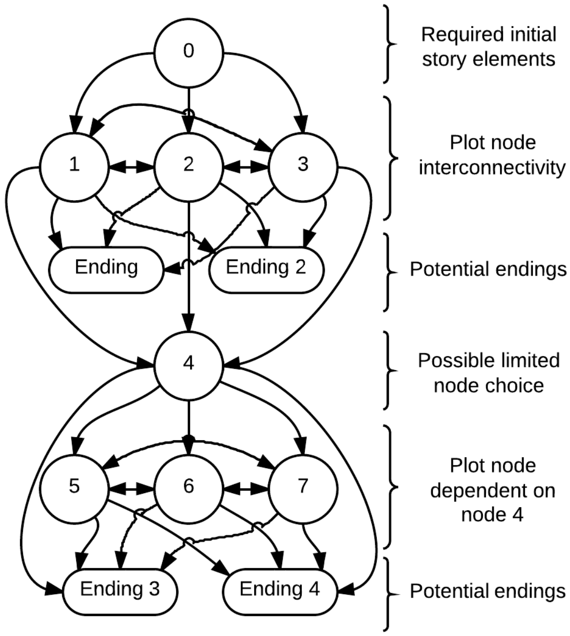

Figure 2: An abridged plot graph for our Zombie story skin. The graph indicates the next possible plot nodes and our plot iterator chooses the best match. A node can be visited multiple times, but the same sentence cannot be regenerated. Some generated stories may never make it to node 4 if the chess game never exhibits certain specific features but the story can still end from almost any node.

Story skins define the interaction of the English sentences, chess features, and plot development attributes. In each skin, a branching plot is developed by hand. Each node represents a plot event and nodes are partially ordered to satisfy a small set of dependencies. Every event is also given a list of applicable features from our chess feature set. We use feature analysis at each plot branch to determine the next story element. Variable binding places new words in each sentence depending on the mood and level of excitement we want to convey in the story. We develop four different skins: The zombie story, Romeo and Juliet, Star Wars (rebels versus Empire) and the war story.

4.5 Plot iterator

The plot iterator is the component of the system that generates story content from chess move features and a plot graph. Each portion of the iterator is discussed in detail in the following subsections.

4.5.1 Chess move grouping

Since most of the games we analyze have large and varying numbers of moves, we cannot map plot nodes (of which there are less than 50) to each individual move in a game. We originally attempted to use aggregate statistics about the game, in conjunction with examining the features of a particular move to determine importance, to influence a weighted drop rate when pruning moves. However, this lead to less than satisfactory results and we ultimately decided to utilize a uniform sized group of moves. The features for each individual move in the grouping are summed to produce the features of the move grouping.

4.5.2 Plot graph traversal

The traversal of the plot graph starts with the first plot node, which is fixed for a given skin. The plot iterator then traverses each node one-by-one, calling each nodes generateText() method to build parts of the story, until it gets to a potential branch (either an omission of an event or a different plot path) in the plot graph. To determine which path to take in the plot graph, the iterator calculates the total feature correlation of the move grouping by summing the weights of each feature for each move. Whichever plot node in the graph has the most similar value to the calculated feature weights of the move grouping is chosen. The plot iterator then repeats this process until it gets to the end of the plot graph.

4.5.3 Finding the best matched features

We tried a few different matching algorithms when attempting to match the chess game features with the story skin features. In the end, we achieve the best results with a simple unweighted k-Nearest Neighbor (kNN) algorithm with discrete valued features. This method favors plot nodes that have the most features in common and have a similar number of features to the chess move grouping.

5. USER STUDY AND RESULTS

Quantifying validity of the experiment is not a trivial task. Our goal is not to make an interesting story, but to reflect the excitement of the chess game, so that we can control which stories to generate for which occasions. To measure how well our generated story actually reflects the input game, we survey 15 Cal Poly students of various chess skill levels. To be eligible, the players had to only know how to play chess.

For each experiment, we select a PGN at random, and generate a story with it using our system. We then generate another story of the same genre except with random chess moves as input. We ask the subject to first watch the chess game being played out using an online PGN viewer. Then he or she is to examine the two stories and select one as being associated with the chess game. 50 such experiments are conducted and in 31 cases the users choose the correct story. This yields a significance of $p < 0.05946$.

Table 2 contains the results broken down by self-identified chess skill level. Not surprisingly, when we asked participants to rate their own chess skills, many put themselves between 1-3 out of 5 and no one rated themselves a full 5 (chess aficionados are hard to come by).

A majority of experiments (62%) resulted in correct matching of the story to the chess game it came from. The results also confirm that more familiarity with chess in general means more likelihood to correctly identify the story. This shows better understanding the chess game makes a positive difference in identifying the correct story. With chess levels of 1 and 2 the results are remarkably close to chance. With levels 3 and 4, they are distinctly above chance.

Table 2: Experiment Results

Chess Proficiency	# Correct	# Total	# of People
1	8	15	4
2	7	14	5
3	10	12	4
4	6	9	2
5	0	0	0
Total	31	50	15

In a number of experiments the alternative story had some overlapping elements with the real generated story. This is because by random chance, many of the same traits could have been chosen to generate the random story. This undoubtedly made the choice more difficult for some of the subjects.

We observe that elements of the chess narrative carry over to the story narrative to *some degree*, which can serve as a proof of concept for our hypothesis.

6. CONCLUSION

Our implementation and user study findings show proof of concept, but there is still much room for improvement. Our user study results are marginally outside the common significance threshold and could be strengthened by a larger user study.

We achieved a high level of non-repetitiveness and variability, generally at least in the hundreds of billions of potential stories per skin. For the Zombie story skin we developed 35 plot nodes each having between 1 and 9 alternative templates which yields 495.3 billion distinct potential unrealized stories. In addition, we have 8 major variables (for example town names or car descriptions) each having between 6 and 126 possible values which results in 91.4 billion combinations. Of course a great many of these stories could read very similar to each other due to only small variations between them. Increasing variability and further plot developments comes at the cost of potentially vague or ambiguous theatrical elements in the final realization.

The chess game feature extraction succeeds in extracting the pivotal themes of the game but could be expanded to get more detailed features. There are virtually unlimited features that can be extracted from a chess game, however. More expertise in chess theory could help to isolate certain unseen elements in the games, as could machine learning or AI feature selection techniques.

While each chess game can be realized through any of the supported skins, the internal structure of some skins are more compatible with some chess games, creating a natural affinity which we currently do not consider.

Our user study has a shortcoming in that we do not measure distinctiveness or "goodness" of stories. By having only alternate stories generated from random chess moves, we lose the chance to control for the level of distinctiveness between the two stories presented to the subject. In the future we can solve this by generating not only random, but maximally divergent stories, or to present the user with N chess games and N stories and ask the user to match them up.

7. FUTURE WORK

This project has many components that could be expanded for future work. The first would be to automate the story skin generation. The limiting factor for the entire project is that the story skins and story graphs are author composed templates. Planning-based author assistance tools such as are designed to alleviate this problem and similar tools could be used for skin generation.

Feature extraction and translation is one of the most significant aspects of generating accurate and interesting stories. The system uses a relatively small feature set. More nuanced features could be developed which could add more detail into the stories. We intend to experiment with more holistic features, as they allow procedural modeling of literary devices such as foreshadowing, flashbacks and *fait accompli*.

Character tracking could also be implemented. Character tracking would mean that each chess piece is assigned to a character in the story and the fate of that chess piece would dictate that characters actions, similar to what was explored in [7]. We need not have parity between characters and chess pieces to accomplish this. We may choose one or a few particularly interesting chess pieces with compelling "lives" to match to story characters.

Ultimately better generated stories from chess games, combined with richer plot material and skins could substantially simplify a sizable problem in automated story generation: control over the dramatic. Many systems attempt to approximate a human judge in order to eliminate incomplete, undesirable or less compelling stories and plot lines. Chess-driven stories are already enriched by an apriori record of rational decisions made within rigid constraints, and chess traces are all but abundant resources. Furthermore, chess moves are not the only things capable of encoding rational human decisions. We imagine many records of decision making activities from Baseball statistics to macroeconomic indicators being used as rich resources for extracting dramatic elements.

8. EXAMPLE

Here we present a complete PGN input and story output.

8.1 PGN Input

0. Nf3 c5 1. c4 Nf6 2. Nc3 d5 3. cxd5 Nxd5 4. d4 cxd4 5. Qxd4 Nxc3 6. Qxc3 Nc6 7. e4 e6 8. Bb5 Bd7 9. O-O Qb6 10. Ba4 Qb4 11. Qc2 Qc5 12. Qe2 Qb4 13. Bb3 Bc5 14. Rd1 Rd8 15. Bf4 Nd4 16. Nxd4 Bxd4 17. Bg5 Rc8 18. Qd3 e5 19. Qg3 f6 20. Be3 Bxe3 21. Qxg7 1-0

8.2 Generated Story Output

```
Ted, Nick, Andrew, and Connor, had banded together as
what seemed like the last survivors. The zombie apoc-
alypse started 2 months ago, but the survivors knew
more humans were still alive.

Zombies broke past the barricades. They all had to
move quickly outside and luckily into their black bank
```

building. The wind kicked up; Connor shivered. There was a faint howl in the distance.

Nick set up a gasoline filled trap while the others stood back in case the zombies surprised him. Nick caught the attention of a horde of zombies and led them straight into the trap which turned into an inferno with a quick shot from Andrew's gun. The zombies burned slowly; the smell was awful.

A huge Scout Zombie threw Ted to the ground and tore off his head. Blood everywhere.

Connor mentioned the gathering of zombies in the distance. The cast of refugees heard a loud growling noise nearby. The streets were filled with zombies crawling on top of a black bank building. It starts to rain. The last thing Andrew needs is wet gear and he makes it known. They look inside a nearby bombed-out office building.

The cast of refugees tries the elevator, but of course the door is pried open. Inside the elevator shaft, zombies start crawling up the side. A Tank Zombie grabs Andrew, in a cunning move, Andrew shoots the elevator cable like he's seen in movies. The elevator comes crashing down and kills the zombies. "This was for Ted".

Through the ceiling, a huge Scout Zombie fell on top of Nick. He became so enraged that he threw the zombie and a dozen other zombies out a window. The next hallway was filled with zombies and the survivors are low on ammunition. Andrew starts the action by blasting a Jockey Zombie from across the hall with his rifle.

Hallway after hallway. The survivors search for anyway to contact help. The top of the bombed-out office building is their goal. Flight after flight they've tried and now at what seems to be the 100th flight stands a Tank Zombie. Andrew uses a rocket launcher to vaporize it.

Andrew began praying. "This is the lowest of the low but you have never let me down before. I have faith". Andrew shoots a large fixture above a Hunter Zombie. The ceiling detaches and crushes the zombies, unfortunately destroying their path.

Andrew kicks open the door to the roof. Outside there are a couple zombies that Connor tosses off the edge. The helicopter circles around and spots the survivors. They're carried away.

9. ACKNOWLEDGEMENTS

The authors thank Cal Poly Computer Science graduate students Connor Lange, Mike Buerli, and Nick Feeney who were part of the team that built the first version of this system. In addition, we thank professor Robert Levinson of UC Santa Cruz for his insightful commentary and advice on this project.

10. REFERENCES

[1] Chess game collection. http://www.supreme-chess.com/chess-games/chess-games-a34.zip, 2012.

[2] Tactical motifs a-z. http://chesstempo.com/tactical-motifs.html, 2012.

[3] S. Bowler. The ludonarrative process. http://www.game-ism.com/2008/05/08/the-ludonarrative-process, 2008.

[4] S. Bringsjord, D. A. Ferrucci, and R. D. Sousa. Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine, 1999.

[5] F. Forrestall. The invisible narrative of games - or - the story of chess, 2009.

[6] G. Frasca. Ludologists love stories too: notes from a debate that never took place. *Level Up, Digital Games Reserch Association (DiGRA)*, 2003.

[7] P. Gervás. Stories from games: Content and focalization selection in narrative composition. In *Proceedings of the First Spanish Symposium on Entertainment Computing*, Madrid, Spain, September 18-19 2013.

[8] A. Katsenelinboigen. *The Concept of Indeterminism and Its Applications; Economics,Social Systems, Ethics, Artificial Intelligence and Aesthetics*. Praeger Publisher, Westport, CT, 1997.

[9] M. Mateas. A neo-aristotelian theory of interactive drama. In *Working notes of the AI and Interactive Entertainment Symposium*, 2000.

[10] D. Myers. The attack of the backstories (and why they won't win). *Proceedings of Digital Games Reserch Association (DiGRA)*, 2003.

[11] M. J. Nelson and M. Mateas. Search-based drama management in the interactive fiction anchorhead. In *Proceedings of the First Annual Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2005.

[12] C. Pearce. *First Person. New Media as Story, Performance and Game*. MIT Press, Cambridge, 2003.

[13] R. Pérez y Pérez and M. Sharples. Mexica: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, pages 119–139, 2001.

[14] J. Porteous and M. Cavazza. Controlling narrative generation with planning trajectories: the role of constraints. In *Interactive Storytelling*, pages 234–245. Springer, 2009.

[15] D. L. Roberts, M. O. Riedl, and C. L. Isbell. Beyond adversarial: The case fo r game ai as storytelling. *Proceedings of Digital Games Reserch Association (DiGRA)*, 2009.

[16] J. Togelius, A. Champandard, P. Lanzi, M. Mateas, A. Paiva, M. Preuss, and K. Stanley. Procedural content generation: Goals, challenges and actionable steps. *Artificial and Computational Intelligence in Games*, pages 61–75, 2013.