GENERATION OF POTENTIAL NARRATIVES IN INTERACTIVE FICTION

GENERATION OF POTENTIAL NARRATIVES IN INTERACTIVE FICTION

BY

ADRIAN SCIGAJLO, B.Eng.

A THESIS

SUBMITTED TO THE COMPUTING & SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF McMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2023)	McMaster University
(Computing & Software)	Hamilton, Ontario, Canada

TITLE:	Generation of Potential Narratives in Interactive Fiction
AUTHOR:	Adrian Scigajlo B.Eng. (Software Engineering & Game Design), McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Jacques Carette
NUMBER OF PAGES:	xiv, 110

Lay Abstract

Writing stories for video games and interactive fiction is a time consuming and complex affair. The resulting web of possible events and pathways through the non-linear story is known as a potential narrative. Computer assisted tools for creating potential narratives could cut down on production costs and open the door for stories that are able to adapt to the player in real-time. In order for these generated stories to be viable they must be coherent, credible and allow for player agency. These criteria describe whether stories make logical sense and are satisfying for the user to interact with. Taking that into account we present a model for coherent interactive stories which keeps track of the story world and the events that could happen within it. As a proof of concept we implement this model with a prototype to generate traversals through a manually authored potential narrative.

Abstract

Digital media has created more opportunities for stories to be interactive, allowing the user to participate in plot events and even change the direction of a story. Interactivity can make stories more engaging however if the plot can change as a result of user input then the story becomes non-linear. The resulting system is known as a potential narrative because a narrative is only presented through user interaction. Non-linear stories require extra considerations and content compared to a linear story of similar length. Computer generated content for video games has grown as a field and this raises the possibility of using computer assistance for the creation and management of interactive stories. This thesis explores the paradigm required for both computers and developers to understand potential narratives. The success of a potential narrative requires the same coherency and credibility that a regular story does, in addition to a new layer known as player agency. Coherency refers to the logical causal progression of a plot, credibility is the verisimilitude of the presented story world and player agency is how satisfying the interactive elements are. These three criteria are the guiding principles for the creation of an intuitive potential narrative model. We present a state transition model for potential narratives as well as a prototype using relational programming to generate coherent traversals through a manually authored potential narrative.

This one is for my family, my friends, and the G-ScalE Lab,

For making me look forward to the future.

Acknowledgements

First and foremost I'd like to thank Dr. Carette who continued to work with me despite a rocky start and work that dragged on far too long. With his encouragements I built the right mindset to tackle larger self-driven projects.

I'd also like to thank my Mom who was there for me during this entire process making sure that I learned the lesson of finishing what I started.

And finally the strong support group of friends who were always around to keep up the positive vibes. This includes my brother Alex, all of the G-ScaLe lab crew who I am proud to call my friends, Jonas who kept me inspired with all the game design chats I could ever want, my boys James and Rob, and all the folks on CFEC who I hung out with almost daily.

Contents

List of Figures

Definitions, and Abbreviations

Definitions

Emergent Narrative

A narrative that is not human authored but emerges from the interactions of players and the underlying systems.

Game Mechanics

A set of rules that govern and guide the player's actions.

Hypertext Fiction

An interactive fiction where the player uses hyperlinks to navigate text passages.

Interactive Fiction

Software presenting an interactive story using a simulated world and a potential narrative.

Multi-User Dungeon

Similar to a text based adventure game but runs in real time and meant to be played with multiple players.

Narrative Refers to the presentation of a story to an audience, for example through the use of character perspectives and plot events.

Player Agency

The feeling of control a player has over an interactive story.

Plot A series of connected events.

Potential Narrative

A system which relies on human collaboration to create a narrative.

Potential Narrative Traversal

The narrative presented to the user as a result of their interaction with a potential narrative.

Simulated World

The story space of a potential narrative including the environments, objects, characters, and events.

Story Sequence of events which make up a plot and are presented to the player through a narrative.

Story Beat Refers to an important plot event.

Suspension of Disbelief

Allowing oneself to believe in something that isn't true, for example to become emotionally attached to fictional characters and feel as if the events of a story are happening to them.

Text Based Adventure Game

A work of interactive fiction that uses a text-accepting, text-generating system and a structure of rules where an outcome is sought.

World State A snapshot of a simulated world at one moment in the story.

Abbreviations

AI Artificial intelligence

IF Interactive Fiction

MUD Multi-User Dungeon

NPC Non-Player Character

Declaration of Academic Achievement

I, Adrian Scigajlo, declare this thesis to be my own work.

To the best of my knowledge, the content of this document does not infringe on any copyright. Copyrighted material in this thesis was used with permission.

While I was the primary contributor, Dr. Jacques Carette provided support and guidance during the research process.

Chapter 1

Introduction

Narratives and games have been a part of human history for as long as history has been recorded. Narratives are an important tool for understanding the world around us (Bruner and Bruner, 1990) and games can entertain us, teach us, and help us connect with one another (Huizinga, 2016). Digital narratives and games have existed for a relatively short time but interactive computer programs have changed the entertainment landscape (Alpert, 2007). New areas of research and engineering have been dedicated to designing interactive systems and the user experience. Rule sets that once had to be verified and calculated by a human can be processed by a computer allowing for the creation of vastly more complex and interconnected rules all for the purpose of creating worlds for us to play in.

Text based adventure games are one of the first examples of these interactive worlds. Authors would use the text input and output functions available to early computers to write interactive stories while accepting commands from the user to navigate a simulated world. These scripted stories allow for no variation besides what the creator had prepared for. When building for a large audience the author must consider user preferences, different user skill levels and the user's understanding of the story. For entertainment it is also desirable that an interactive story have player agency. This is when the user feels that their inputs have an impact on the narrative and different inputs could create a substantially different experience. Interactive worlds which account for different user preferences and allow for many possible narrative sequences are time-consuming to create. Research into this field seeks to give computers a level of narrative intelligence so that they can assist with creating expansive narrative worlds and adapt the narrative to a user's particular needs or preferences.

1.1 Potential Narratives

An interactive story is best described by the term potential narrative (Motte, 1986). An interactive story depends on human inputs to determine what is presented next and so it uses human collaboration to create a narrative. The system can contain many potential variants of the story presentation and so it is referred to as a potential narrative. A clear demonstration of potential narrative can be found in hypertext fiction, an evolution of text based adventure games that uses text passages connected by hyperlinks. To interact with the system a user selects a link to navigate to the next passage and in doing so they traverse through a potential narrative. Different paths through the passages represent different narratives that a user can experience. In this way the potential narrative can be thought of as an event space of all the possible events. When the simulated world grows in complexity and the user is given more ways to interact with the story the potential narrative also grows more complex. Each new interaction requires human generated content to represent each new possible state.

Computer-generated content is already used to assist in some applications, for example in the simulation of non-player characters in The Sims 4 (Maxis, 2014). Some systems such as Dwarf Fortress (Adams and Adams, 2006) or Spore (Maxis, 2008) go even further by simulating the interactions of an entire world. These systems don't have explicitly authored story content and instead rely on emergent narrative, where stories arise from the interplay of many smaller system components. Drama Managers are another area of research that focus on creating stories through the interactions of believable agents with simulated goals and emotions (Roberts and Isbell, 2008). With this context we can now describe the goals, approach, and contributions of this thesis.

The overarching goal of this thesis is building models and tools for computer assisted generation of potential narratives. Although this goal extends beyond the scope of this thesis it is the motivation which directs the research done here.

1.2 Approach and Scope

The original research question that motivated our work was the following:

Research Question: How can you encode the characteristics of interactive stories for use in computer assisted narrative generation?

Interactive storytelling is a broad area and so the decision was made to focus solely on Interactive Fiction, which is a digital storytelling medium encompassing text-based adventure games, hypertext fiction and more. It is an ideal medium for testing experimental ideas because of the low barrier to entry (Valls-Vargas, 2013). Unlike many digital games they can include but do not require sound design or graphics to create a compelling experience. For this reason we primarily analyze the potential narratives of interactive fiction for our work.

To examine story characteristics we aim to create a set of evaluation criteria which consider the logical and aesthetic properties of human authored interactive narratives. Afterwards with the analysis and evaluation criteria we aim to model potential narratives in a computer friendly format. One of the considerations for modeling a potential narrative is to identify the content that makes up the story and determine which elements can be computer-generated and what should be manually authored. The need to model potential narratives led to new research questions:

Research Question: What paradigm can be used to intuitively encode potential narratives for the purpose of generation?

Research Question: How can formal story structures be applied to the computer generation of potential narratives?

With a logical model of potential narrative we can then use relational programming to generate potential narrative traversals. The ability for a computer system to sample different traversals through a potential narrative is a key step on the path to generating them.

1.2.1 Limitations

As a result of only analyzing interactive fiction the user interactivity examined in this thesis is very limited. Digital games have an immense variety of interactive mechanics that our work only examines through their effects on a narrative traversal. The prototype reflects this by focusing primarily on the evaluation criteria of coherency and not on credibility or player agency which are left to the author.

In addition this thesis only includes very simple non player characters. Although these are a crucial part of storytelling the current focus is on ensuring that the resulting traversals are coherent for the player controlled character before including additional agents into the system. More work on generating believable characters in stories can be found in Riedl's *Balancing Plot and Character* (Riedl and Young, 2010).

1.3 Contributions

The scope of this thesis can be summarized into a thesis statement.

Thesis Statement: Using a state transition model of the potential narratives of interactive fiction enables computer generation of coherent traversals through relational programming.

Following the approach above yielded these contributions:

- **Interactive Fiction Analysis:** An analysis of interactive fiction with a focus on the potential narrative.
- **Evaluation Criteria:** A set of qualitative evaluation criteria for evaluating interactive stories.
- **Potential Narrative Model:** A state transition model of potential narratives.

- **Traversal Generation Prototype:** Prototyping work done to generate traversals through a human authored potential narrative using relational programming.

1.4 Plan of Thesis

Through the analysis of interactive fiction and the states, events, mechanics, and outcomes that form the potential narrative we will present three evaluation criteria for interactive stories: credibility, coherency, and player agency. These describe how interactive stories should have a logical progression of events, how they should maintain the verisimilitude of the simulated world, and how a player should feel that they have some control over the narrative. Our analysis also demonstrates how potential narratives can be described by events which modify a world state. This is codified into a state transition model of potential narratives. Finally we show a prototype using relational programming that is capable of finding coherent traversals through a human authored potential narrative of over a hundred events written using the state transition model.

Chapter 2

Storytelling

Storytelling is the act of conveying a series of events to an audience. It is extensively used in both entertainment and education as a means of communicating ideas in a more readily understandable way. The word story is often used interchangeably with plot or narrative. In this thesis a plot describes a sequence of events that are connected by cause and effect while narrative will refer to the presentation of connected events to an audience. For example the audience may experience a plot through a narrator who gives their perspective as a character in the world. We may also refer to a story space which encompasses all events which take place in a story as well as the locations, objects, history, characters, and all other story existents. A story should begin in one place and finish in another without interruption. (Jarrell, 2002) This can be a new physical location or state but can also be a social or emotional transformation. This thesis will explore some of the ways computer assisted authoring can be used to create stories.

2.1 Interactive Storytelling

"Interactive narrative is a form of digital interactive experience in which users create or influence a dramatic storyline through their actions. The goal of an interactive narrative system is to immerse users in a virtual world such that they believe that they are an integral part of an unfolding story and that their actions can significantly alter the direction or outcome of the story." (Riedl and Bulitko, 2013)

In a scripted narrative only the feeling of agency can be achieved but non-linear narratives can allow significant alterations. This comes at a cost since every potential narrative encounter must be implemented even if a user doesn't experience it during a session. In addition the author must strive to ensure a coherent narrative experience independent of what choices the user makes.

The space of an interactive story is all of the relevant content within the game or implied by the game, also known as the simulated world. The story events encoded into the game make up the plot. Narrative is the entire process of playing through the game itself which presents the story events using audiovisual media or through the game mechanics themselves. The player may take control of a character who narrates a story, or in some games the player themselves may end up being the narrator while the plot itself is created through their decisions.

In interactive stories the user is usually given control over a character who is part of the simulated world. Both the possible actions of the user-controlled character and the impact those actions have on the story vary from one work to another. In some games the player is given a scripted story event as a reward for completing a puzzle and in other works the puzzle itself may be a story event with multiple solutions where each solution takes the story in a different direction. This will be explored more in depth as we analyze interactive fiction.

2.2 Story Structures

Analysis of stories has revealed patterns within them that authors sometimes use for guidance in creating their own. One of the early examples of a formal structure for stories is Propp's Morphology of a Folk Tale (Propp, 2010) which identified common themes of Russian folk tales to create sequential functions that described specific plot events. Another can be found in The Hero with a Thousand Faces (Campbell, 2008) which explores stories told about the archetypal hero in myths. Some structures such Freytag's Pyramid (Freytag, 1894) are more generalized and focus on creating dramatic impact by building up to a climactic moment before resolving the story.

Story structures are sometimes called Plot Grammars, and they are numerous and varied in scope and intended usage. In this thesis, The Writer's Journey (Vogler, 2007) will be used to provide direction for the type of stories to be generated. The Writer's Journey is a variant of the Hero's Journey which is tailored to provide structure for stories instead of being an analysis of mythological heroes. The Writer's Journey was selected because it was found to be a good fit for interactive stories compared to other structures. (Dunfield, 2018)

Chapter 3

Interactive Fiction

Interactive Fiction refers to software presenting an interactive narrative using a text based interface. The term originally referred to text based adventure games but has grown to encompass multi-user dungeons and hypertext fiction. Authoring tools for creating hypertext fiction such as Twine (Klimas, 2009) minimize the technological barrier for authors looking to create an interactive story. This has made interactive fiction a popular choice for prototypes and experimental ideas. These same features make the genre a good starting point for research on applying computer assisted authoring and design techniques to interactive narratives. By building a model of interactive fiction games this thesis shows where these techniques could be applied.

3.1 Text Based Adventure Games

In a text based adventure game the user interacts with a simulated world by inputting text commands to control a character and reading the corresponding output from the game. This process builds a narrative of the user's adventure through the environment since the final transcript of these interactions will detail the player's exploration of the world, the mysteries they unraveled, and the puzzles they solved. Since these games are single player they often focus more on puzzles and plot, unlike multi user dungeons, which are their multi player counterpart and focus on interacting with other players and fighting.

3.1.1 Elements of Text Based Adventure Games

To understand where story generation techniques could be applied we examine the four elements that make up text-based adventure games (Montfort, 2011):

- a text-accepting, text-generating computer program;
- a potential narrative, that is, a system that produces narrative during interaction
- a simulation of an environment or world.
- and a structure of rules within which an outcome is sought, also known as a game.

Text-Accepting and Text-Generating Program

When a user begins to play they are presented with a description of their character's immediate surroundings and sometimes the character themselves. In this description they are made aware of nearby objects of interest and are expected to write text commands to explore and interact with their surroundings. The following verbs are standard commands in many text based adventure games: The player can use "examine" for more information about something in the room and can move their character from one room to another using "go." The character usually has an inventory and can carry and use objects with "take" and "use." Different commands may have the same outcome, for example "go door" may be accepted instead of "go east." Uncommon verbs or nouns are often mentioned before they are needed or provided through hints since guessing an exact word can be frustrating for the player. An example of a text-accepting and text-generating system can be found in one of the first well-known adventure games: Zork (Lebling et al., 1979).

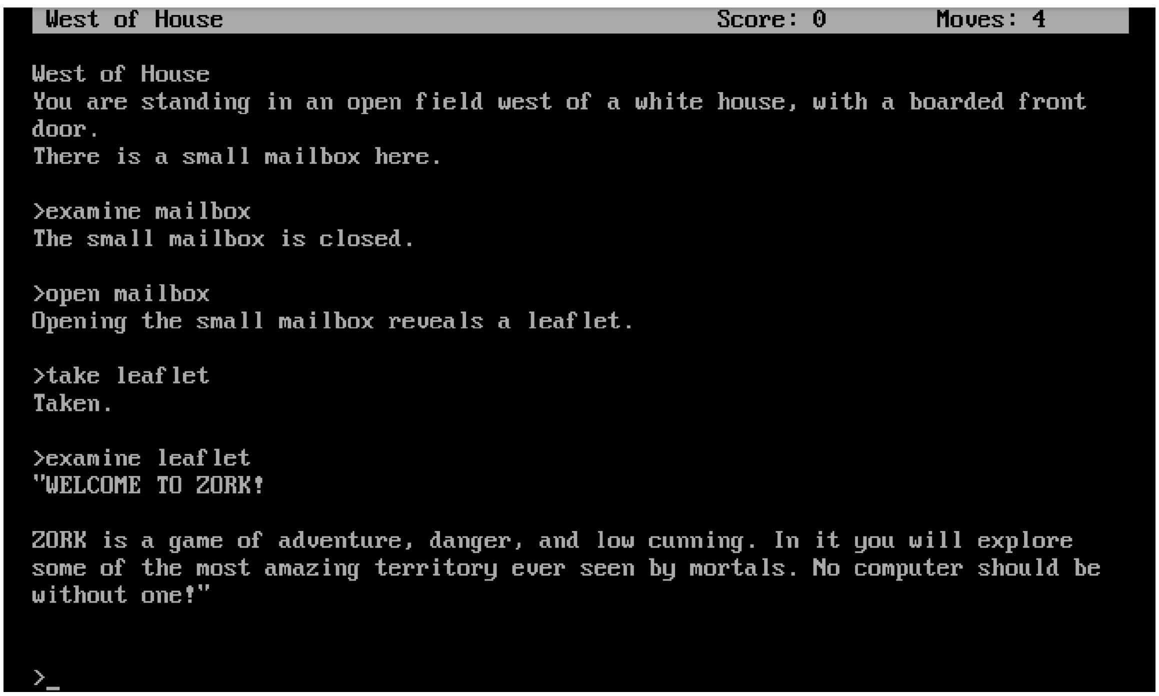


Figure 3.1: Zork, a Text Based Adventure Game (Lebling et al., 1979)

Potential Narratives

Playing through a text based adventure game results in a transcript which details all the character's interactions with the simulated environment. This transcript shows how the player experienced the story and so it can also be called a narrative. For this reason text based adventure games can be called potential narratives; a narrative is created as a consequence of playing the game. This is distinct from literature or film where the creator chooses how the user will experience a story. In a text based adventure game the creator influences and directs the user's experience but they cannot guarantee an exact traversal through the game. This element is not unique to text based adventure games; it exists in interactive narratives as a whole.

Simulated World

The potential narrative exists within a larger story space which can include the different environments, objects, characters, and the history of these things. A text-based adventure game simulates the world that contains all these elements. The text-generating system provides information about the world to the player who then uses the text-accepting portion to interact with the world. For example the system may tell the player that a forest is located to the east and the player may then go east to move their character into the forest to explore it. In this way a text-based adventure game simulates the different environments of the world and the player character's ability to traverse from one location to another. Many text-based adventure games allow the player to pick up objects and move them to other locations. For example picking up a key to open a door in another area and gain access to a new location. The level of simulation varies greatly but all text-based adventure games must simulate a space for the story to reside in.

Structure of Rules

The interactive text-generating system, potential narrative, and simulated world provide us with the user interface, the narratives that the user may experience, and the space that the user will interact with. The final element is a structure of rules within which an outcome is sought, also known as a game. For example, Hollywood Hijinks (Anderson and Cyr-Jones, 1986) is about a treasure hunt in a mansion where the player will win if they can collect 10 treasures hidden throughout the game world. Collecting 10 treasures to win is an explicit goal and the player will learn that they can find these treasures by exploring and solving puzzles. Not all Interactive Fiction has a goal like this, Exhibition (Finley, 1999) has the player explore a simulated gallery with no puzzles or score. The player can look at paintings and interact with several characters who have unique perspectives on each painting in the gallery. There is an argument to be made that the goal is to explore the entire gallery and speak to every character but there is no way to win the game as it has no rules. During its release in 1999 a work like this would not have been considered a game, however game design has evolved since then to accept that interactive media with no rules or goals can be games too.

3.2 Multi User Dungeons

Multi user dungeons allow for multiple players to create their own character and play together within the simulated environment. One of the main differences is that the game runs in real time instead of waiting for the player's input. They are often set in fantasy worlds and have a focus on simulated combat where the player fights monsters and completes quests in order to progress. Interaction between players involves role-playing as their chosen character which can lead to formation of complex social structures such as hierarchical groups of players that have rivalries or alliances with other groups. *MUDs are out of scope as their narrative is typically emergent* because it results from interactions between players in a multiplayer simulated world. The focus of this thesis will be on designer-guided narratives experienced by a single player.

3.3 Hypertext Fiction

Hypertext fiction refers to works of electronic literature where the reader uses links to traverse a series of written sections. This term is predated by choose your own adventure novels which are physical books that use internal references to create a nonlinear narrative. The reader must traverse down potentially branching paths using page references to jump from one section to the next. In hypertext fiction instead of a page reference the reader is presented with a hyperlink that leads to another section. When faced with a branching path the reader maybe be required to make choices that change the outcome of the story. In this way text based adventure games and hypertext fiction both have potential narratives.

The hyperlinks in any given passage are the only interaction the user can have with the system at that given passage. Text Based Adventure games maintain the same level of interaction throughout the game however in Hypertext Fiction this is not the case. It is up to the author how much the player involvement there is between sections. For example some hypertext fiction may have long linear segments with no interaction besides pressing a link to proceed the next passage while others create an

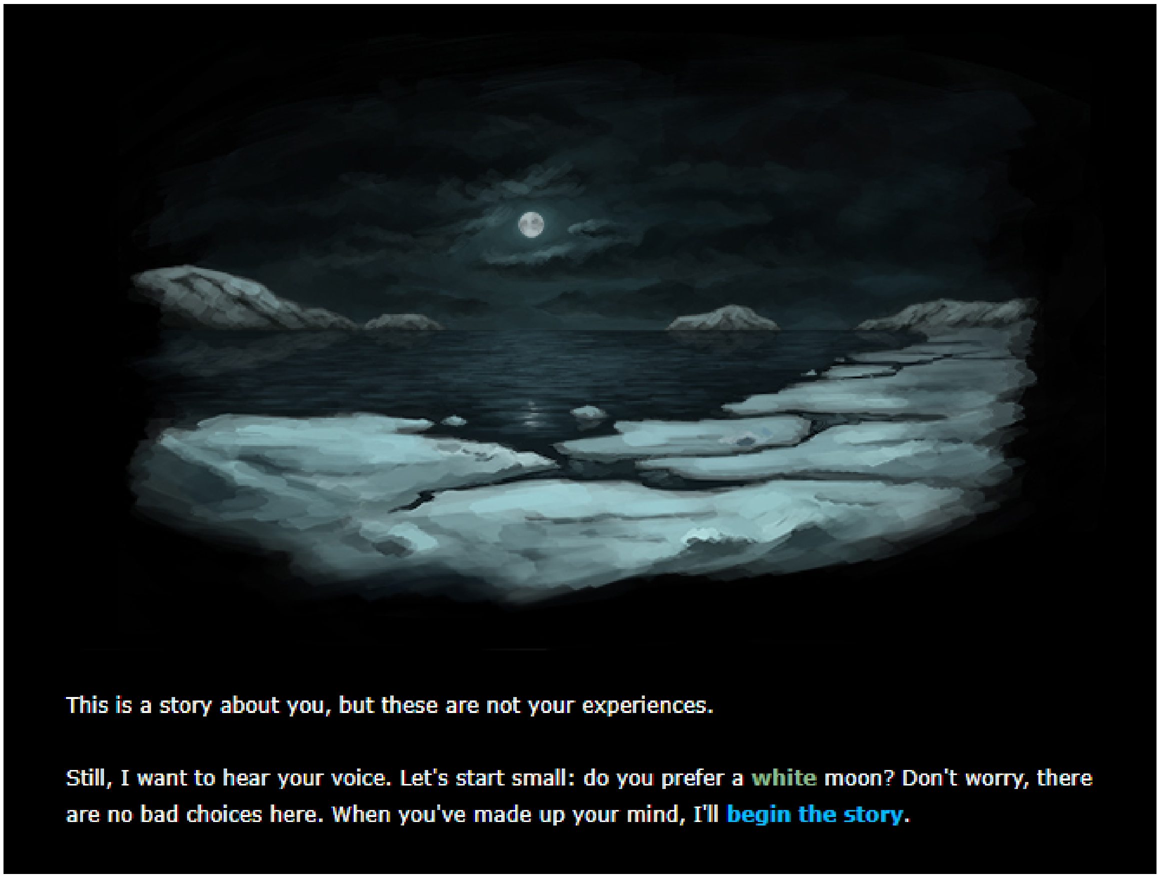


Figure 3.2: Beneath Floes, a Hypertext Fiction (Snow, 2015)

adventure experience of exploring a simulated environment and solving puzzles. A simulated world in hypertext fiction will be built out of explicit options to answer questions such as "What room do I go to next?" and "What object do I examine?"

3.4 Synthesis

Potential Narratives are the core focus of this thesis. This feature describes how interactive fiction presents a story as a result of user interaction. The input system, simulated world, and rules provide structure for the user interaction while the potential narrative turns this interaction into a story. In order for computer assisted story generation to create stories for interactive fiction it must create potential narratives. The following chapter will examine the structure of interactive fiction and analyze some examples to understand what potential narrative is.

Chapter 4

Content of Interactive Fiction

To understand how interaction becomes narrative involves we will examine what a user can interact with during play and how those inputs build a narrative. In broad terms the user will interact with the game mechanics, the environment, and characters within the game to produce story beats. Each of these story beats builds the player's understanding of the story. Not every interaction has to push the plot forward by changing the story world, they can also provide context or texture. This distinction between meaningful and non-essential events has been examined before and they are referred to as Kernels and Satellites respectively (Cohan and Shires, 1988). Kernels are required for a logical understanding of the story while satellites may improve the story but are optional for understanding. Kernel events and the specific player interactions that create them are the primary focus of our analysis.

4.1 Interactive Elements

The content of interactive fiction roughly falls into five categories. A work of interactive fiction can contain all or only some of these elements (Montfort, 2011).

- **Game Mechanics:** The game elements such as inventory or health points.
- **Rooms:** The physical layout of the game
- **Puzzles:** Challenges which will be presented to the player solved using the mechanics.
- **Objects of Interest:** Any elements which are unrelated to puzzles but create a more coherent experience
- **Characters:** Both the player character and non-player characters that can be interacted with.

Any model of Interactive Fiction should be able to represent these elements.

4.2 Game Mechanics

The structure of rules found in text based adventure games are also known as game mechanics. These are rules which provide feedback and guide the player's actions in game (Sicart, 2008). For example some Interactive Fiction gives the player a concrete goal and a way to measure their progress towards that goal. A game could tell the player to find all the hidden treasures in a mansion and include a score counter on the interface to show how many treasures they have collected so far. An initial goal gives the player a reason to engage with the simulated world and feedback to show the progress they have made through the game. Other mechanics can be included to introduce different challenges. For example a game might include an inventory system which requires the player to pick which items they want to carry with them and which to leave behind or a health point system which tracks how injured the player character has gotten.

Interactive fiction often connects the user's progress through the narrative with their progress towards goals that the game mechanics create. In the above example of a goal that requires a player to find all the hidden treasures in a mansion the game designer can reveal story beats while the player explores for the treasures or as a result of finding these goal items. In this way the designer creates a Potential Narrative; by engaging with the goal of searching for treasure the player will experience a narrative as a result of their actions. If the treasures do not have to be collected in a set order or not all treasures have to be collected then different playthroughs can result in different narratives. This uncertainty is a key feature of potential narratives, through clever use of rooms and other elements a designer can ensure that every player will experience a coherent narrative.

4.3 Rooms

The simulated world of a typical interactive fiction is not an open world represented by one contiguous field, instead it is split into discrete connected locales. These locations are called Rooms by convention but they can also represent different sections of a cave or forest. Each room is populated with interactive elements representing all the player relevant things in that location. We can represent the player's ability to traverse rooms and explore the game world as a node graph where each node is a room and each connection represents the ability to move between those rooms. Generally each room will be relevant to the player by providing progress towards the game's goal or more information about the world. Access to rooms can be prevented until some condition has been met, for example a room that is locked until the player can find the key.

Besides providing story information in their descriptions, rooms can also section the narrative into discrete parts by grouping related story information. For example if the player must investigate a murder then the crime scene could be a room or set of rooms where the player is expecting to find relevant evidence. By controlling the access to different sections a designer can present the story beats in a coherent order by ensuring that information required to understand later beats is presented first. The traversal between rooms itself can also be a player decision that leads to significant story beats, such as in the case of one-way passages or forks in the road where the player has a choice over traversal.

4.4 Puzzles

Interactive fiction will present the player with problems such as a locked door or a greedy dragon which must be solved in order to reach the goal. Problem solving in the simulated world is an imitation of real-world problem solving but with often fantastical problems that fit the fictional world. These puzzles are the primary challenge of most interactive fiction games. The player must manipulate objects, overcome physical obstacles, and persuade, evade, or fight non-player characters within the game world in order to complete the game. Many works also include classical puzzles such as anagrams, cryptograms, riddles and mazes (Sorolla, 2011). Cataloging the variety of puzzles is out of scope for this thesis, however the lock and key pattern is ubiquitous and one of the common tools used by the designer to create a coherent narrative.

4.4.1 Lock and Key

A Lock and Key puzzle is the most common challenge found in Interactive Fiction. An object (key) must be found and used in order to overcome a challenge (lock). This can take the form of an actual lock with a hidden key but it can take many other forms, such as a greedy dragon that will only let the player pass if given gold or a password that must be deciphered before a character will open a door. The lock can keep the player out of a room or prevent interaction with an element which can be used to separate the potential narrative into sections. Filling a world with only literal locks and keys will not create something that is engaging to play. Mapping problem and solution to objects that make contextual sense contributes to the verisimilitude, however this is superficial as pointed out in *The Craft of Adventure* (Nelson, 1995) since the player will quickly figure out the one-to-one relationship of problem to solution in what is described as Get-X-Use-X syndrome. This is a problem where the player finds an object and uses it to solve exactly one puzzle before discarding it.

The IF Theory Reader (Sorolla, 2011) suggests some structural remedies to the Get-X-Use-X problem:

- Solutions requiring more than one object
- Objects relevant to more than one solution
- Problems having more than one solution

- Objects irrelevant to problem and problems without solutions (Objects of Interest)

Locks which block access to interactive elements also prevent the player from experiencing the story beats associated with those elements. This separates the potential narrative into events which can happen before and after the lock. If certain story events are necessary for understanding later ones then a designer can attach those story events to a lock and key puzzle which must be solved before progressing. For example if the player is tasked with investigating a murder then the game could require the player to find sufficient evidence in order to progress the story. Then the next section could have the player character interviewing suspects about their findings because player knowledge about the evidence can be assumed. By using lock and key puzzles interactive fiction writers can create larger structures with branching pathways that stay coherent because the player must encounter certain story beats.

4.5 Objects of Interest

Interactive elements which are unrelated to puzzles are crucial to creating a believable simulated world. A simulated environment would be populated with objects that make sense to be there but aren't necessarily associated with the other challenges of the game. These elements can also be problems with no solution, for example a barricaded door that prevents entry. A barred door makes the simulated world seem larger than it actually is and makes the player ask questions about why the door was barricaded, which can be answered by story beats found later. In this way objects of interest can provide the player with information about the world or additional context for the story but a designer cannot assume a player will get this information. Objects of interest should generally not be used to present kernel story beats that are critical to understanding the plot because player's may ignore or miss these elements.

4.6 Characters

Interactive Fiction works tend to have a player character that will be controlled but they can also include other characters that the player will interact with. These are other beings in the simulated world that the player can learn information from or trade with, or provide challenges the player must overcome. In this way they are related to the other interactive elements however there are other considerations when adding a non player character to the world. This is because a fully implemented character should:

- React to events;
- make conversation of some kind or another;
- understand and sometimes obey instructions;
- wander around the map in a way consistent with the way the player does;
- have some attitude to the player, and some personality (Nelson, 1995).

Requirements like these make characters more believable for the player, however they increase the complexity required in a potential narrative substantially. Character believability is about ensuring that characters do not ruin the player's suspension of disbelief and will be discussed more in the evaluation chapter. Making interactive characters is an entire research topic on its own, for example characters are thought to be more readily comprehended when the audience is able to infer the goals and intentions of the characters (Bates et al., 1994). *Narrative Planning: Balancing Plot and Character* uses planning algorithms to create intentional agents that work towards their personal goals in order to generate stories (Riedl and Young, 2010).

Many works of interactive fiction have no characters or have very simple characters in order to keep the potential narrative well scoped. These simple characters can be interpreted as another way to present mechanics, puzzles, or objects of interest and will often not be very interactive beyond that. While fully realized and intentional characters are a rich area for further work in generating potential narratives they are out of scope of this thesis, which will focus mostly on simple characters.

4.7 The Dreamhold Analysis

The Dreamhold by Andrew Plotkin was selected for this analysis because it was designed for players who have never played interactive fiction before. It is a fantastic introduction to the common elements of interactive fiction for players who are new to the text-based adventure genre. In The Dreamhold the player character wakes up with amnesia in a wizard's house and must explore the world in order to regain their memories and figure out what happened. Dreamhold's primary objective is a key and lock puzzle where the player must find seven colored masks, an iron key and a copper key to complete the game. To do this the player must explore the simulated world and solve other puzzles to collect these items. As the player interacts with the simulated world they will regain more of their memories and be able to piece together the story. During play the player is scored based on how many masks they collect in addition to secret items which are optional objectives.

The player experiences the narrative of The Dreamhold by solving puzzles and examining rooms and objects. The player is taught early on that finding a mask is rewarded with score and a flashback presenting important plot information. This encourages the player to seek out more masks with the end goal of discovering every mask. Although the player has freedom to explore The Dreamhold however they'd like, the game uses lock and key puzzles as well as distance from the starting point to ensure that story beats are encountered in a coherent order. The connections between rooms and important interactive elements can be drawn as a directed graph like the one included below.

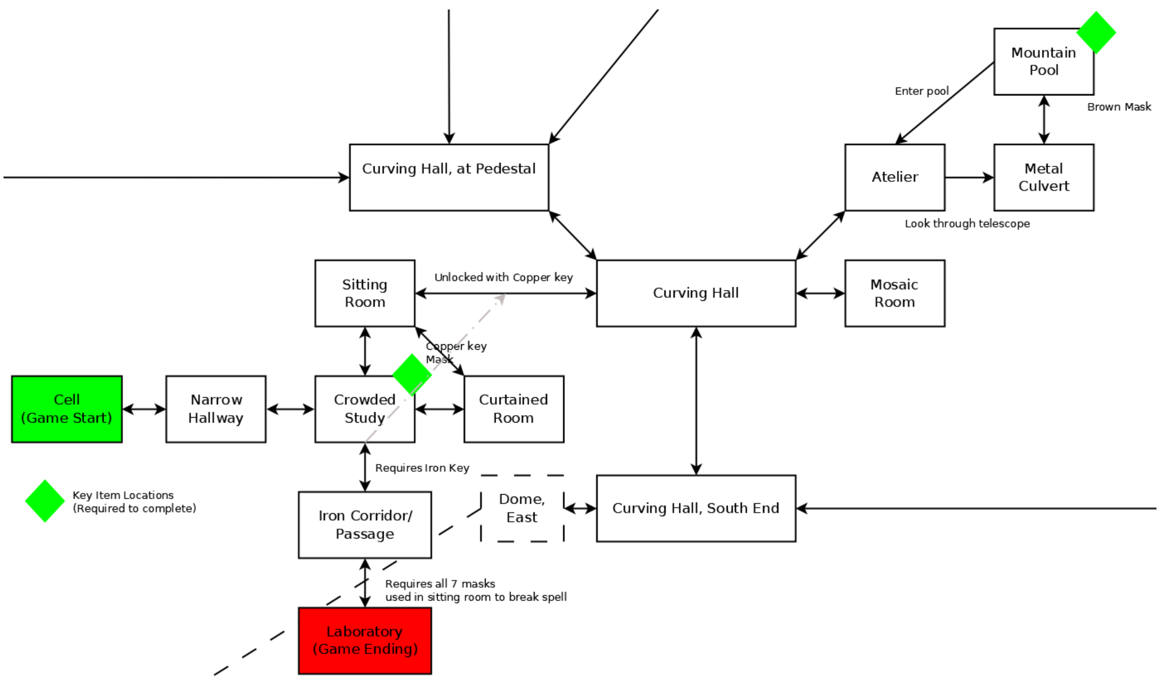


Figure 4.1: Partial The Dreamhold room map

This figure represents a section of the text based adventure game The Dreamhold. Each rectangle is a room in the simulated world and the connecting arrows show how the player can traverse between them. The arrows show direction of travel and whether the passage only goes one-way and text beside the arrows describes traversal conditions such as a lock that must be opened. The player begins in the highlighted green room and ends the game in the highlighted red room. Although the end of the game is just a few rooms away from the start, the player cannot enter that room without first using the iron key and all seven masks. Rooms that contain items required for progression are marked with a green diamond. The full room breakdown for The Dreamhold can be found in Appendix A: Dreamhold Room Analysis.

In order to find these key objects the player must search for them by looking at rooms and examining objects. In The Dreamhold this is done using the look command and examine command respectively. When entering a room for the first time the player will get a description of the room which sets the scene. Afterwards the player can use the look command while their character is in the room to get an updated description at any time. The player can then use the examine command to get more detailed information about anything mentioned in the room description, however the player must specify what they are examining. An example is pictured in Figure 4.2 where the player uses the look command inside of the Crowded Study to get a description of the room and then uses the examine desk command to get a detailed description of the desk.

Almost everything described in the room can be examined further and the resulting descriptions can reveal more elements. In Figure 4.2 the desk description reveals a mask as another element of room which can be examined or interacted with. The mask can be taken to add it to the player's inventory of items which also rewards them with score and this is demonstrated in Figure 4.3. Score is a game mechanic which gives the player a goal to collect all seven masks in The Dreamhold and keeps track

Crowded Study

>look

Crowded Study
You stand amid clutter, oddments, shelves and stacks. The chamber is comfortably broad, centered around a massive and well-worn wooden desk. But so much is jammed in, around the desk and up the panelled walls, that you barely find room to stand. Books; papers; dried plants; animals stuffed and preserved, antique instruments, candles of any hue. You don't recognize half of it, and you can't name half the rest.

There are wide doorways on the north and east sides of the room. A door of heavy wood stands closed to the south, and stairs descend into dimness to the west.

You see an immense book on the desk. Behind the desk is a tall glass-fronted cabinet. Beside that is a small brass trunk, which is closed.

>examine desk
The desk is heavy and plain, a massive wooden construction. Surprisingly, the surface is bare, except for an immense book.

Something in the chaos catches your eye. It looks like a white papier-mache mask, lying beside the desk.

Figure 4.2: Look and Examine commands in The Dreamhold

of their progress. The player can also put the masks on to discover more about the story by restoring some memories which links story progression to game progression. The transcript of putting on the first mask is shown in Figure 4.4. The Dreamhold also has optional goals which are not required to complete the game but give bonus score at the end.

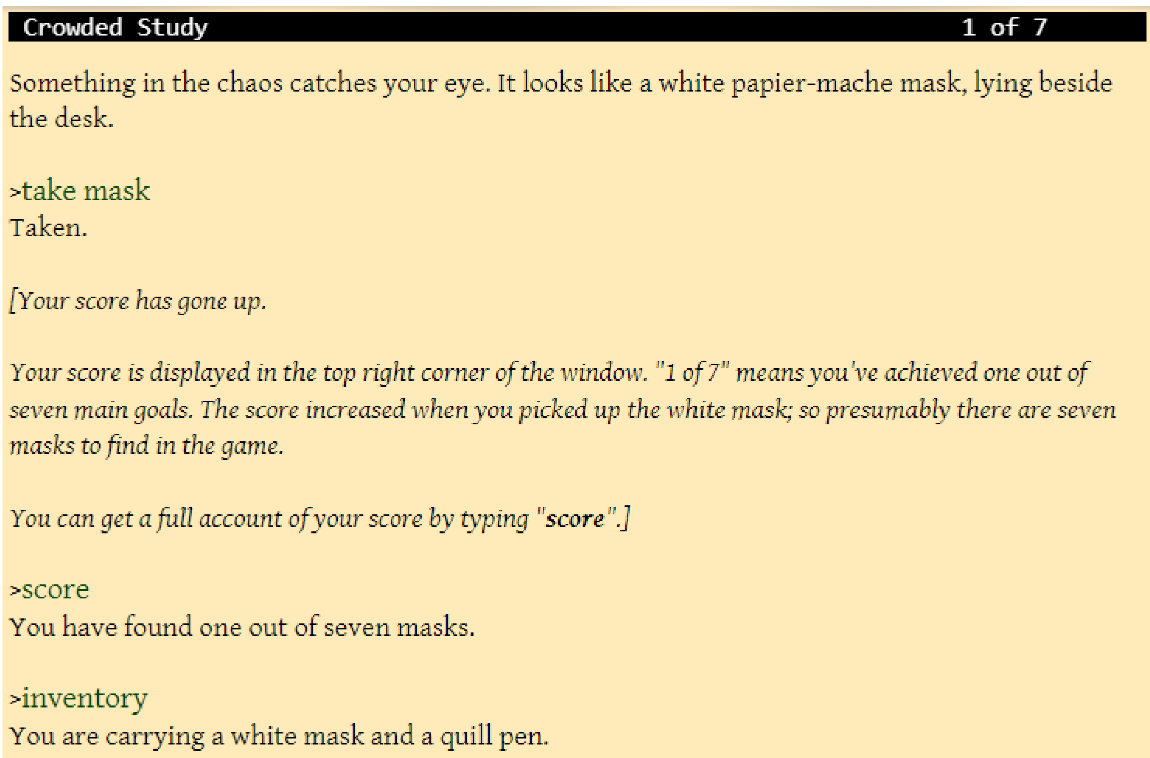
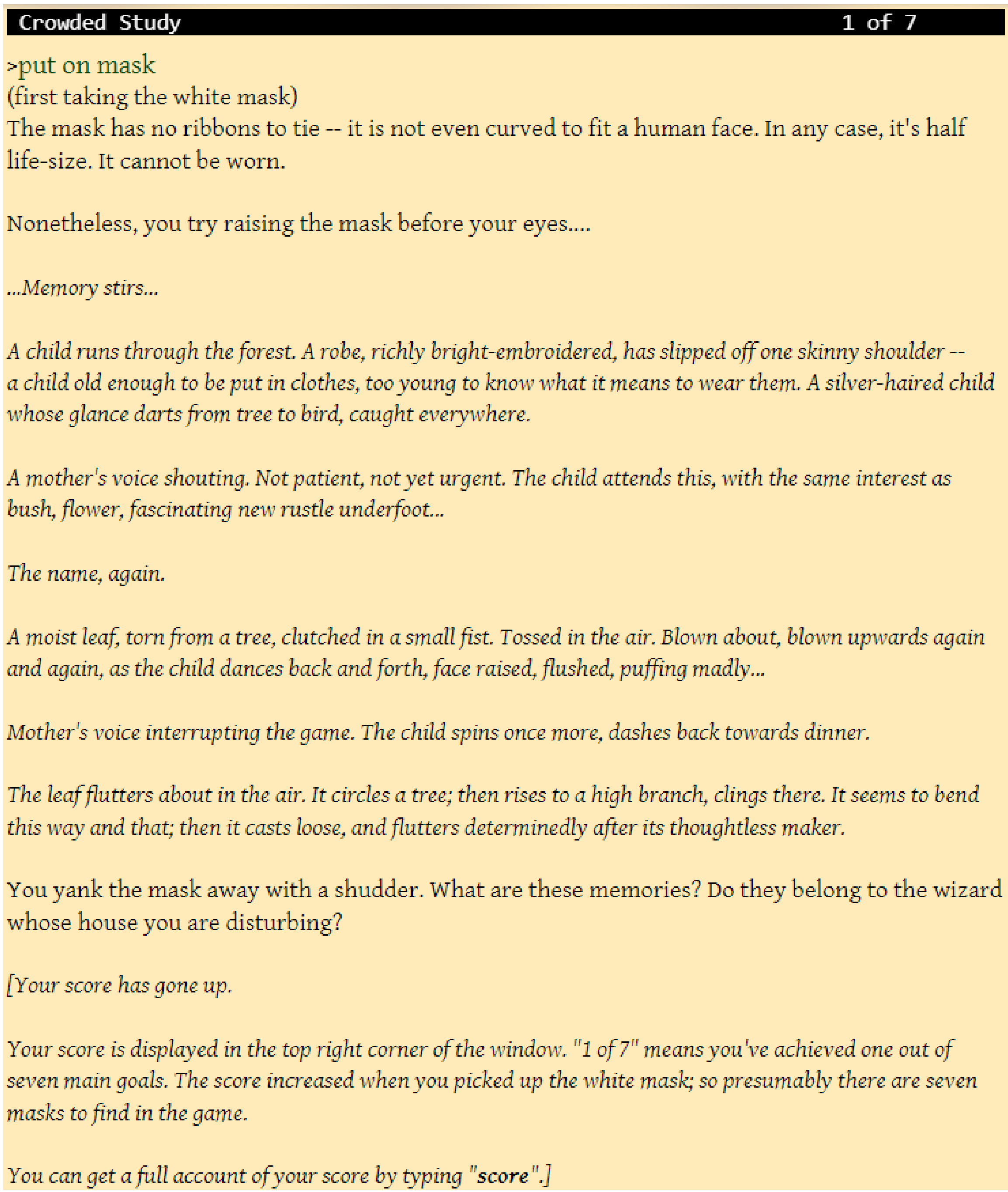


Figure 4.3: Examining a scoring item and collecting it.



Crowded Study **1 of 7**

>put on mask
(first taking the white mask)
The mask has no ribbons to tie -- it is not even curved to fit a human face. In any case, it's half life-size. It cannot be worn.

Nonetheless, you try raising the mask before your eyes....

...Memory stirs...

A child runs through the forest. A robe, richly bright-embroidered, has slipped off one skinny shoulder -- a child old enough to be put in clothes, too young to know what it means to wear them. A silver-haired child whose glance darts from tree to bird, caught everywhere.

A mother's voice shouting. Not patient, not yet urgent. The child attends this, with the same interest as bush, flower, fascinating new rustle underfoot...

The name, again.

A moist leaf, torn from a tree, clutched in a small fist. Tossed in the air. Blown about, blown upwards again and again, as the child dances back and forth, face raised, flushed, puffing madly...

Mother's voice interrupting the game. The child spins once more, dashes back towards dinner.

The leaf flutters about in the air. It circles a tree; then rises to a high branch, clings there. It seems to bend this way and that; then it casts loose, and flutters determinedly after its thoughtless maker.

You yank the mask away with a shudder. What are these memories? Do they belong to the wizard whose house you are disturbing?

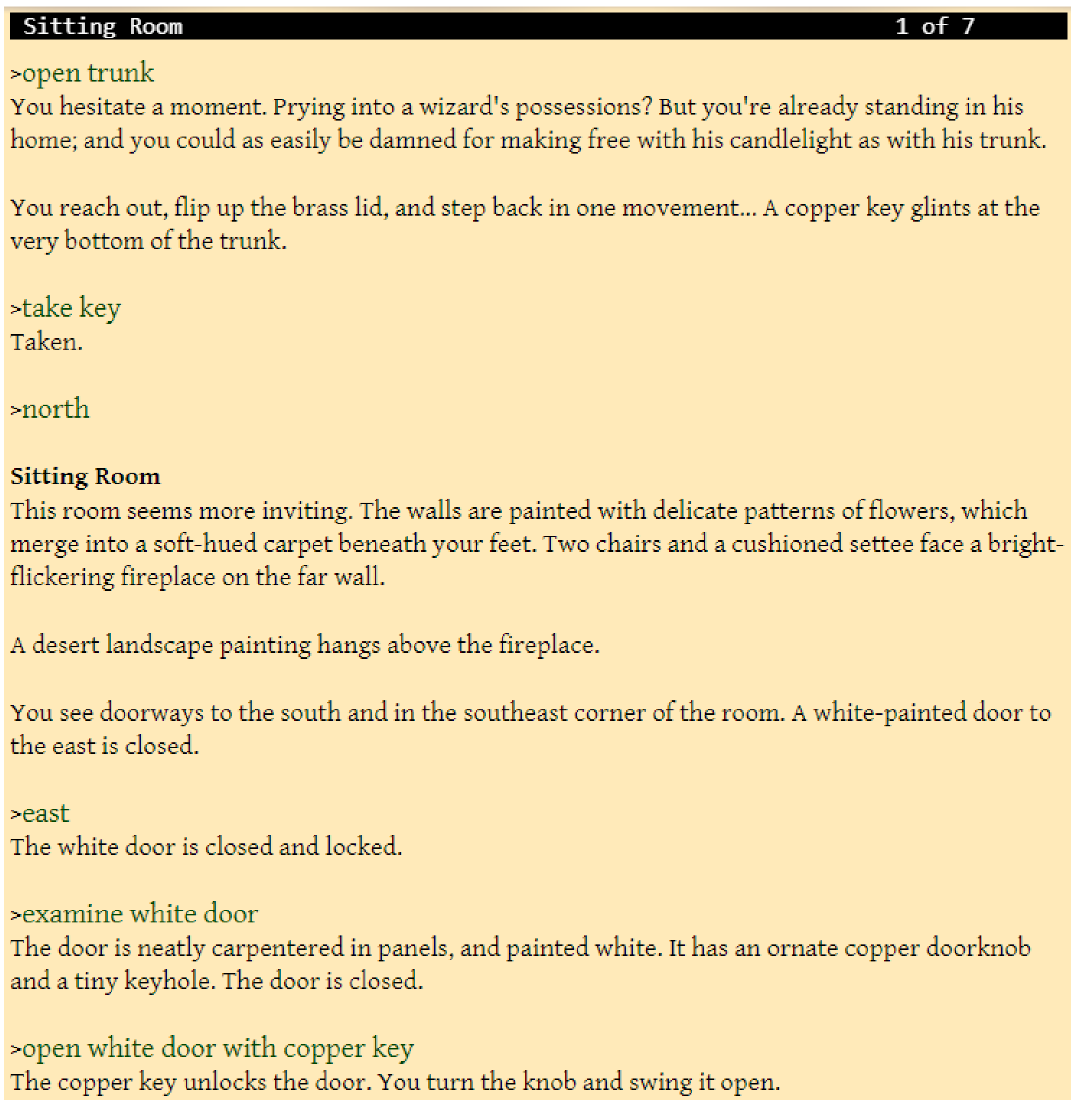
[Your score has gone up.

Your score is displayed in the top right corner of the window. "1 of 7" means you've achieved one out of seven main goals. The score increased when you picked up the white mask; so presumably there are seven masks to find in the game.

*You can get a full account of your score by typing "**score**".]*

Figure 4.4: Putting on a mask rewards the player with a flashback

Solving puzzles in The Dreamhold also requires interacting with objects found using the look and examine command. A straightforward lock and key puzzle in the first few rooms serves as a gentle introduction to this concept. This puzzle is a literal lock and key, the player must open a trunk in the Crowded Study to take a copper key that will open a door with a copper lock in the Sitting Room to the north. This process is demonstrated in Figure 4.5.



Sitting Room 1 of 7

>open trunk
You hesitate a moment. Prying into a wizard's possessions? But you're already standing in his home; and you could as easily be damned for making free with his candlelight as with his trunk.

You reach out, flip up the brass lid, and step back in one movement... A copper key glints at the very bottom of the trunk.

>take key
Taken.

>north

Sitting Room
This room seems more inviting. The walls are painted with delicate patterns of flowers, which merge into a soft-hued carpet beneath your feet. Two chairs and a cushioned settee face a bright-flickering fireplace on the far wall.

A desert landscape painting hangs above the fireplace.

You see doorways to the south and in the southeast corner of the room. A white-painted door to the east is closed.

>east
The white door is closed and locked.

>examine white door
The door is neatly carpentered in panels, and painted white. It has an ornate copper doorknob and a tiny keyhole. The door is closed.

>open white door with copper key
The copper key unlocks the door. You turn the knob and swing it open.

Figure 4.5: Finding and using a key to open a lock in The Dreamhold

It should be noted that there is another locked door within the Crowded Study however the player will find that the copper key does not fit the lock. This means that fully exploring this one room gives the player a lock without a key, a key without a lock, and a scoring mechanic encouraging the player to find all seven masks. The smaller subtasks provide early direction for the player's exploration and reward the player with more areas to explore when they are completed. Other puzzles found during exploration can be much more involved, requiring the player to manipulate complex mechanisms in an orrery or manage water levels in a cistern to access upper and lower levels.

The order of rooms and locking of areas directs the player's traversal through the game but the player still has freedom over their actions. A player could explore all of the Sitting Room, and Curtained Room and before returning to the Crowded Study to search for the key. Or they might find the key immediately and skip the white mask and Curtained Room until they return later. This is what is meant by potential narrative because the presentation of the story depends on player interaction. The Dreamhold does not decide what actions a player will take but it does influences the player's actions using the descriptions and layout of the rooms. For example because examining the desk is required to find the white mask, the desk is the first prominent object described in the look command and the rest of the description is centered around the desk. Both the white mask and copper key are found within a room the player must pass through and it is the first room that contains interactive objects. While the game cannot force the player to do things, it should draw the player's attention to important elements and structure itself so that the player builds an understanding of the story and mechanics as they progress.

Potential narrative can be seen as a set of discrete player interactions that change the story world and increase the player's understanding of the story in the process. The Dreamhold was specifically selected to illustrate the idea of potential narrative. Each interaction in The Dreamhold isn't necessarily meaningful but there is a set of events that form the core experience. Each of these events changes the world state and can have a required world state before they occur. For example finding the copper key is a requirement to open the copper lock and opening the copper lock is a requirement to any event that occurs in the newly unlocked rooms. The potential narrative is the collection of all these potential events, their requirements, and their effect on the story.

Chapter 5

Related Work

This thesis focuses on *deliberative* narrative generation systems where a computer selects an appropriate series of events based on some parameters. This is different from *simulation-based* narrative generation where a narrative emerges from the interactions of many subsystems (emergent narrative) (Riedl and Young, 2010). Narrative generation systems can be further categorized depending on what parts of a story they generate and the technique they use to accomplish it. These concepts are explored more in depth in a survey of story generation techniques done by Kybartas and Bidarra (2017). This chapter gives an overview of related story generation techniques and where to find more information on the systems that implement them.

Story generation is computer assisted story development where the system can automate parts of the story authoring process. The goal of story generators is to create a plot which satisfies some goal while remaining causally and temporally coherent. Generation systems are distinct depending on what technique they use and which story elements they focus on. For example drama managers simulate believable agents and generate stories based on their interactions while narrative planners use a computer planning based approach to create a series of coherent story beats. Most use a mixed-initiative approach where a human author provides some information to the system (Kybartas and Bidarra, 2017). The benefit of computer assisted story development in interactive storytelling is that the system can use information of user actions and preferences to tailor the experience. This would make it possible for the user to influence a storyline without manually authoring the consequence of each action. This section will cover grammar based generation, drama managers, case based reasoning, genetic algorithms, linear logic, AI planning, petri nets, and neural network generation.

5.1 Grammar Based Generation

Stories can be generated by encoding a plot grammar such as the Hero's Journey into a set of relationships between story world elements and plot events. Commonly these will describe actions that characters can take part in, the requirements for both the characters and story world before the event can occur and the effect that event has on the story world. By keeping track of the story state and applying the authored relationships, the generator can produce plots matching the plot grammar. Structuralist analysis of stories have been used in this approach, including the use of Propp's structure to create a grammar for computer generation of new folk tales (Gervás, 2013). Grammar based generation has been successfully used to generate other types of stories such as the betrayal narratives generated by BRUTUS. (Bringsjord and Ferrucci, 1999)

Interactive applications such as SQUEGE (Onuczko et al., 2008) and ReGEN (Kybartas and Verbrugge, 2014) both generate quests for Role Playing Games using grammar rewriting and locations, characters, and items from the game to create new stories for the player to experience. ReGEN uses a formal representation of the game as a series of nodes and edges which can scale to a desired level of complexity. Formal grammars for stories have been criticized for being unable to take meaning and symbolism of the individual events into account and therefore being unable to represent every story (Garnham, 1983). Grammar based generation is well suited to applications which have a specific story structure such as a quest in a role-playing game.

An example of simple graph rewriting used for grammar based generation is in the figure below. The story is represented as a series of story events connected by small arrows. The large arrows represent rewrite steps and the states outlined in blue are the ones that are selected to be rewritten according to a rewrite rule. In this case a quest to slay a dragon is then represented as finding a magic sword and then dealing with the dragon. Then there is another rewrite rule on dealing with the dragon that adds a branching path to the story where the dragon can be slain with the magic sword or befriended.

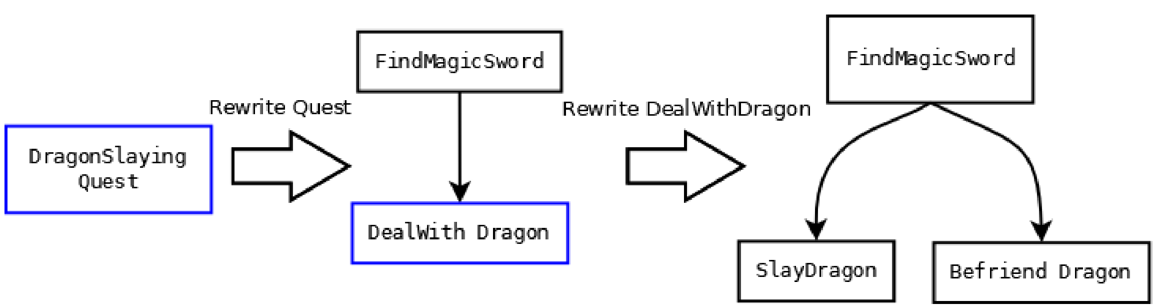


Figure 5.1: A Graph Rewriting Example

5.2 Drama Managers

Drama Managers are systems which create a story experience in real time by reacting to player interactions with agents within the system. It is created with some knowledge about the desired story structure and uses state of the simulated world and the inputs of the player in order to make decisions on what should occur next. An example of an implemented drama manager is Facade (Mateas and Stern, 2003), where the player plays the role of a long-time friend visiting a couple having marriage problems and the conversations and arguments that occur that night. The drama manager keeps track of the objects that the player interacts with and will naturally try to include narrative beats involving those objects into the story. For example in Facade there is a painting that will trigger a discussion about a vacation the couple went on. There are also other events on timers, such a mixing a drink or showing off a souvenir that can naturally interject conversations. By doing this drama managers create player agency since with enough cues and reactions its possible that every natural player action will have a relevant response. One downside to this approach is that it requires much more manually authored content and careful consideration of potential player actions.

5.3 Case Based Reasoning

Case based reasoning generates stories by using a database of manually authored plots to decide how stories should be constructed. The database has stories written as their constituent plot events according to some model so the generator can suggest solutions which match how similar stories have approached each case. One such generator uses Propp's structure for folk tales and a database of folk tales written according to the Propp functions (Díaz-Agudo et al., 2004). The paper suggests this approach is better for creating stories that are dramatically impactful since it works off of proven cases rather than just focusing on the coherency of stories which may produce uninteresting results. MINSTREL is another case based generator which tries to generate short stories about King Arthur and His Knights of the Round Table. This model focuses on the authorial intent to provide a moral message through the plot and a wealth of manually authored stories in order to try and generate stories with a moral message (Turner, 1993). Tearse et al. (2013) have modernized the MINSTREL system as well.

5.4 Genetic Algorithms

Genetic algorithms determine the fitness of a given plot or plot segment and then algorithmically mutate the story towards a desired fitness threshold while still meeting story requirements. The plot generation involves an initial pool of stories which undergo recombination and mutation to generate a new pool of stories which are ranked using a fitness function. The fitness is then used as a weighting to select a new group to continue the mutation process. A feature of genetic algorithms is that the thresholds can be tightened to shrink the generation space for reliable plots or they can be loosened to allow for a greater variety of results. The fitness function varies greatly depending on implementation, for example HEFTI (Ong and Leggett, 2004) allows authors to determine the plot fitness of different events to try and achieve the result the author desires while PlotGA (McIntyre and Lapata, 2010) uses a fitness function trained on a database of previous stories which determines fitness primarily based on the coherency of the story.

5.5 Planning Based Generation

Planning Based Generation creates stories by starting at an initial story world state and using AI planning algorithms to search for a plot which arrives at a desired end state, also known as the goal. All story existents are encoded as propositions and the world state keeps track of the current existents that define the world as a set of propositions. The world state can be modified according to a list of actions which may consume propositions and introduce new ones to reflect the changed world state. These actions represent plot events and the planning algorithm uses the list of potential actions and the initial world state to search for a series of plot events that reach the intended goal state. Depending on the system there can be multiple goal states, optional goal states, and goal states which change while the planning algorithm is running. For example STORY CANVAS (Skorupski and Mateas, 2010) focuses on author defined end goals for the story while FABULIST (Riedl and Young, 2006) uses a combination of author and individual character goals.

Research into planning algorithms that use character goals involve modeling the emotions and behavior of the agents within stories to make them more believable and dramatic storylines (Riedl and Bulitko, 2013). Defining attributes of characters gives more context so that the planning algorithm can make better decisions about what the character would do. In particular the Intent-Driven Partial Order Causal Link (IPOCL) (Riedl and Young, 2010) planner creates more believable stories by ensuring that character actions are consistent with their perceived goals and desires. This is known as intentionality and the IPOCL planner attempts to balance it with the author's goals which none of the characters may intend. .

The approach used later in this thesis has many similarities to planning. Below is a visualization of a simple planning algorithm as a state transition diagram. The blue initial state represents where the planner would start and the connecting arrows are actions which change the story into new states. A planning algorithm tries to find a trace through this state transition diagram which leads to the goal state, represented in green. Some of these events have requirements which are included in square brackets, for example in order to slay the dragon the knight must have a magic sword and not be friends with the dragon. One way to find this trace is to search through all possible actions at the initial state and then search the resulting states and continue this process until a goal state is found.

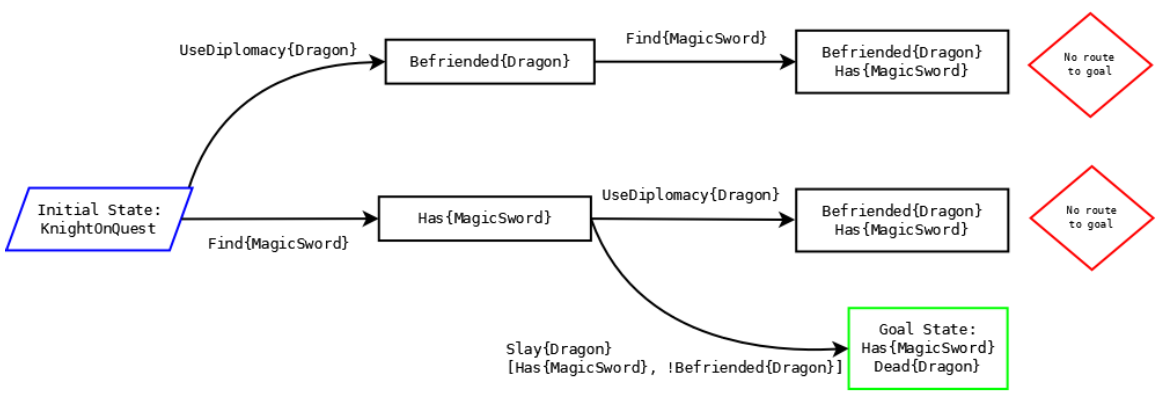


Figure 5.2: Planning Algorithm Example

5.6 Linear Logic

Linear logic was created by Girard (1987) and would later be proposed as a representation of non-linear narratives by Bosser et al. (2010). A linear logic deduction manipulates resources that cannot be duplicated or dropped and are consumable by rules. These resources and rules can model different states and the way those states can be modified.

Linear logic has been proposed as a way to assist authors in the creation of interactive stories (Dang et al., 2011). Story generation using linear logic involves mapping the narrative existents as resources such as "The knight has a magical sword" and so the world state ends up as a list of these propositions. Then by using linear logic connectives it is possible to model events which change the state of the world, such as "If the knight is present, and the knight has a magical sword, and the dragon is present, then transition to a state where only the knight is present and the dragon has been slain." By consuming and creating new propositions to represent narrative events that change the world it becomes possible to use linear logic principles to derive the events required to reach a goal state thus generating a series of events representing a story. More information can be found in *Programming Interactive Worlds with Linear Logic* (Martens, 2015). Our thesis takes a very similar approach to this technique, the details of that can be found in the model chapter.

5.7 Petri Nets

Petri nets (Murata, 1989) are a graphical and mathematical modeling tool created to study and describe concurrent computer processing. It is represented as a graph of *places* (circles) and *transitions* (rectangles). Arrows connect *places to transitions* or *transitions to places*. Each place can also have markings on it represented as black dots. When every place connected to a transition has a marking it can *fire*, removing a marking from each input place and putting a marking on each output place. An example petri net modeling a simple story can be found in Figure 5.3:

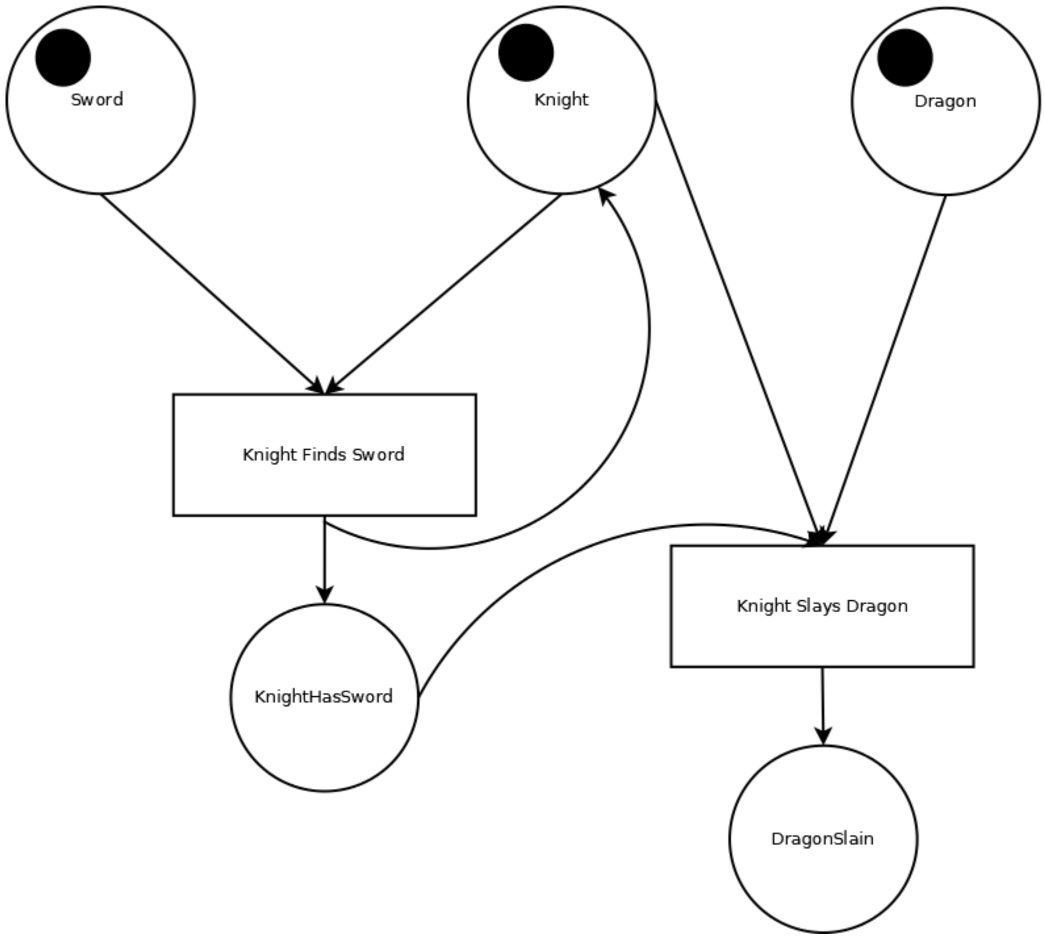


Figure 5.3: Petri Net Example

In this example, the `Knight Finds Sword` transition can fire because of the markings on *Sword* and *Knight*. This would consume those two markings and create a new marking on *KnightHasSword* and replace the one on *Knight*.

Although not widely used in the industry (Djaouti, 2013) petri nets have applications in game design, having been used to model game mechanics (Manuel and Roque, 2009) and inspiring a visual game design tool called Machinations (Dormans, 2011). Petri nets have also been used in interactive story generation as a way to manage non-linear stories with virtual actors (Brom and Abonyi, 2006), as a role-playing game quest plot generator (Lee and Cho, 2012), and a method to validate interactive stories (Dang et al., 2013).

5.8 Neural Network Generation

Neural Networks can generate stories by being trained on a large set of human authored content so that the system can predict what word, phrase, or event should come next in a story. This approach has the ability to tackle the problem of open story generation where the domain of characters, places and objects does not need to be defined. Specific training sets are often selected from the same class of story to improve the coherency of resulting stories. For example AI Dungeon (Walton, 2019) is trained on a set of choose your own adventure stories in order to create an interactive story for the user. To accomplish this it uses the Generative Pre-trained Transformer (GPT) (Radford et al., 2018) which itself is trained on an enormous amount of human written text in order to accomplish natural language processing.

Chapter 6

Evaluation Criteria

This chapter proposes evaluation criteria for potential narratives. Cognitive psychology has determined that comprehension of a narrative is correlated with the causal structure and the attribution of intentions to the characters that participate in events. (Magliano and Graesser, 1991) The criteria will be split into three categories: Coherency, Credibility, and Player Agency. Coherency ensures the resulting story has a causally sound progression and credibility refers to the believability and integrity of the story. Player agency is unique to interactive stories and it deals with how much a player feels they are able to intentionally modify the story. Many of the concepts come from the storytelling section in the book *Fundamentals of Game Design* (Adams, 2014).

6.1 Coherency

A story is coherent when it maintains temporal and spatial consistency while still progressing forward. The sequence of plot events (but not necessarily the narrative) travels forward in time from one event to the next. The causality of events is strongly correlated with audience comprehension. There must be a relationship between the temporally ordered plot events such that a preceding event changes the story world so that the next event can occur. Spatial consistency requires that all existents must be in appropriate locations for the events that occur. For example two actors must be in the same location to speak face to face, so that if they are in different areas they must move before being able to speak. If a story is coherent then the audience should also be able to predict possible outcomes by following the logical chain of events.

6.1.1 Temporal and Spatial Consistency

Temporal consistency is the system's ability to ensure that the order of plot events maintains causality. Preconditions must be met before an event can occur, and the postcondition of an event will remain true in the story world unless changed by another event. It should be noted that the narrative can provide the audience with information about a different time in the plot through a technique called flashbacks or flashforwards. The time period which the audience is being shown may change but the consistency of each plot event should still hold true. Spatial consistency refers to the system ensuring that the actors of events need to be in the appropriate locations before and after the event occurs. These can be represented as precondition and postcondition that deal with character location, for example a conversation event requiring the precondition that two characters share the same location or a travel event having the postcondition of changing a character's location. One technique to ensure temporal and spatial consistency is to create a simulated story world that models the necessary states. Preconditions of a plot event must be satisfied in the world state before the action can occur and afterwards the world state changes to satisfy the postconditions of the action.

We can demonstrate inconsistency with the story of a knight being sent out on a quest to slay a dragon. The knight must first collect a magic sword and then travel to the dragon's lair to do the deed. If the dragon is slain by a magic sword wielding knight who has not yet found the sword then the story is temporally inconsistent. If that same knight was to slay the dragon from within the comfort of their castle while the dragon resided in its lair then we could also call the story spatially inconsistent.

6.1.2 Forward-Progressing

In order for a story to be coherent the narrative must progress towards a conclusion which is different from where the story began. There should be visible changes to the world state that indicate a plot progression and the events should not seem arbitrary. The progression does not need to be dramatically meaningful in order to be forward-progressing, however a reader typically expects to experience all major plot events before the story concludes. This forward progression of many generated stories follows an authorial intent, since the end goal or individual character goals are provided to the system before generation. This is especially true in planning based story generation since it always fulfills some authored goals as part of the generation process. In interactive stories the potential narrative should encourage the player to move towards key plot points or be built such that these world changing events are part of the core game progression and cannot be missed.

Returning to our knight and our dragon, let us imagine the hero preparing for their quest by taking care of their regular knightly duties. The story then describes the knight performing mundane tasks while the dragon situation is ignored and no progress is made towards their quest. While this continues then the story is not forward progressing. (Even if the story was examining the cowardice of our knight then there should be consequences as a result of their inaction!)

6.2 Credibility

Credibility refers to a feeling that although the story takes place within a fictional and even impossible world, it is believable in context and does not feel arbitrary or repetitive. This is sometimes referred to as the audience's suspension of disbelief, where the credibility of the narrative helps the audience be emotionally attached to the characters because they feel as if the events are really happening to them.

Character Believability

Character Believability means that actions performed by characters do not have a negative impact on the audience's suspension of disbelief. This is often dependent on the depiction of the character's behavior and appearance in addition to the internal attributes of a character such as personality and intent. Stories are also more readily comprehended by the audience if they are able to infer the intentions of characters that participate in story events (Bates et al., 1994). Having goal-oriented characters is crucial to believability (Charles et al., 2003) since in order for a character to be believable the audience should be able to predict their motivations and intentions through observation. This is explored in *Balancing Plot and Character* (Riedl and Young, 2010) which tries to generate narrative sequences in which the characters will be perceived to be intentional agents.

The goal of our knight is to complete the quest they were given by slaying the dragon. If the knight is shown to be cowardly then the character should act accordingly. The audience won't find a supposedly cowardly character believable if they bravely march into the dragon's lair without external pressure or other justifications. They may seek other solutions or find a way to conquer their fear since their goal is still to complete the quest.

6.2.1 Not Repetitive

The clear repetition of plot events quickly erodes the credibility of a story. This is most noticeable if a plot begins to loop through the same actions and states even if it eventually continues towards the intended goal. Story Generation systems often use a library of plot events to generate plots, which can lead to a risk of repeating similar events if the algorithm decides to choose the same event. Generation of interactive stories which combine gameplay mechanics with story are often at risk of feeling repetitive since there are only so many types of player interaction in a game.

If a knight slays one dragon and is immediately sent out to slay another in exactly the same manner then the story will feel repetitive. To justify using a similar event later in the story it must be meaningfully different. Since the story must be forward-progressing both the characters and events should be changed in some way. Perhaps the knight has overcome their cowardice and approaches the task differently, or another dragon is seeking revenge for earlier actions of the knight!

6.2.2 Not Arbitrary

The audience should feel that events occurring in the story aren't senseless and random. The audience will expect the story to progress a certain way based on which characters have been focused on and what has already been presented. In other words the internal model of the story world should be consistent to the narrative being presented. Although the reader's expectations are often subverted to increase the impact of a particular event, it must come with some justification. At this point it's enough that a generated story just be consistent and not attempt these dramatic twists. One way to make sure events are not arbitrary is ensure future events are strongly tied to earlier events. By having earlier actions create the conditions required for later events to happen the story has a sense of causality and the plot feels as if it fits together.

A knight that is given a dragon slaying quest should undertake suitably heroic actions relevant to their goal. The audience may feel the story is arbitrary if the knight decides to open a bakery instead of searching for a magic sword and no justification is provided. The act of searching for the magic sword becomes justified later since it is the tool used to complete the quest. In the same way the opening of a bakery could become justified if a later event involves the knight befriending the dragon over a shared love of baking.

6.3 Player Agency

Player agency refers to a feeling of control a player has over the direction of an interactive story. As the player traverses a potential narrative they will make choices and expect the story to react to their decisions. When a player experiences the narrative that results from their actions there is an implication that different decisions would have led to different outcomes. Interactive stories will often try to create a feeling of player agency by allowing the player to have more freedom over the actions they can take and making those actions meaningful, however creating player agency using an enormous branching potential narrative that covers every possible decision is time and resource intensive. Often just the feeling of player agency is enough to create a satisfying interactive story and so potential narratives utilize foldback points and good feedback to the player's actions which gives a sensation of control without needing an unreasonable story scope. This is also sometimes related to work done on player modeling, which learns a model of a player's preferred style of play and can select content in the story to reflect that. (Thue et al., 2007)

6.3.1 Freedom of Player Action

Player intent will not always exactly match the possible actions available in the potential narrative. Freedom of player action is how well a player feels that their intent is represented in the decisions they are capable of making. Intent is influenced by their understanding of the player character's place in the story and their ability to interact with that world. The game mechanics constrain what a player can input into the story but also create intent by suggesting possible actions. The player feels disconnected from the story if there is a strong intent that a player cannot fulfill, for example if a game mechanic allows the player to befriend or attack previous characters and then without justification forces the player to befriend a character they intended to attack.

In an interactive story the player character could be the dragon-slaying knight.

If they were to experience the story as it was presented earlier a player would find very little freedom in their actions since the only option is to find the magic sword and then slay the dragon. Learning that the dragon has a secret love of baking could create the intent to befriend them and that intent could be fulfilled by the option to open a bakery. Giving a choice between befriending or slaying the dragon increases freedom of player action.

6.3.2 Impact of Player Actions

The impact of a player action is how much a player feels their actions direct the plot. While freedom of player action refers to a potential narrative's capacity to accommodate the player's intent, impact of player action is how much their actions change their traversal through a potential narrative. This is the feeling that if the player made different decisions, their experience would also have been different. If a story has high freedom but little impact it will feel as if the player's intent does not matter and they are only observers to a story instead of an active participant within the world. Interactive stories must often balance freedom and impact so that although a player may be presented with fewer options, each option leads to a distinct pathway. Creating a compelling interactive story by allowing the player's actions to have impact while still maintaining a coherent and credible narrative is a typical problem for story generators, with some approaches also using player modeling to take player preferences into account (Sharma et al., 2007).

A player choosing to slay or befriend a dragon should leave lasting changes on the plot. There is an expectation that other characters in the world should react to their choice and the dragon may become a recurring character later. Having the ability to call upon their new friend to assist by flying them to other locations or intimidate their enemies would make their actions feel impactful. If the dragon is killed shortly after they have been befriended then it can feel as if the player's choice did not matter because the story state ends up in the same place either way.

6.3.3 Player Story Control

The thesis *Development of an Emergent Narrative Generation Architecture* (Schudlo, 2014) describes granularity of story control as the level of control a system has over story events and how reliant the system is on player actions to progress the story. In their definition an interactive story with a large grain only concerns itself with the major plot points to control the overall direction of the story while the user has some control within these larger beats. This means the player can manipulate how these plot points are achieved or what other tangential events they experience before the next key story event. Having a small grain or low granularity means the story is micromanaged and the player is always constrained by the authors intentions.

Here we refer to this concept as story control and relate it to player agency since it deals with the player's influence over the ordering of plot events. When the player has high story control the potential narrative only constrains the major plot events and the player is free to experience other story beats or tangentially related events in the order that interests them most. While freedom of player actions refers to the player's ability to react to events according to their intent, story control is the player's ability to freely pursue different areas of the potential narrative. High player story control describes open world games such as *Skyrim* (Bethesda, 2011) where the player forges their own path and can experience different sub-stories without many restrictions in addition to the main story line. Low player story control means the potential narrative is constrained to only a few author-intended traversals. This typically applies to games with a linear or even branching story, although these games can still have high freedom and impact!

Interactive stories can have more than just a single quest for the player to undertake. The quest to slay a dragon may be just one event on a larger adventure through a fantasy world. The quest to slay the dragon may be their main goal but they can prepare for it by undertaking other tasks and experiencing different parts of the potential narrative that can impact their story. If the player is able to pursue other narrative arcs such as exploring a haunted catacombs or defending a farmer's crops then they have some level of story control.

Chapter 7

Modeling Potential Narratives

Interactive stories allow the audience to take part in and influence plot events and the resulting web of possible experiences is known as the potential narrative. In this chapter we suggest a model for potential narratives with computer assisted generation in mind. The model will cover three main components: discrete events occurring within a world state, the directed state transition graph containing all possible traversals, and how user inputs direct those traversals. This model is focused on single player experiences with a curated story (as opposed to an emergent narrative that arises out of game mechanics.)

7.1 Events and World States

The core of this model is built on the definition of story as a series of causally linked discrete plot events that change the story world. When an author writes a story they will seek to ensure their audience feels it is coherent, however a computer system must be given the structure needed to verify this. In this model the world state contains propositions to describe the current status of the story and simulated world. Events are written with preconditions and postconditions which describe the required world states before and after the event occurs. These conditions make it is possible to encode temporal and spatial coherency into the events of a story. An example of this in a simple linear story is pictured below as a state transition diagram with predicates to show coherency.

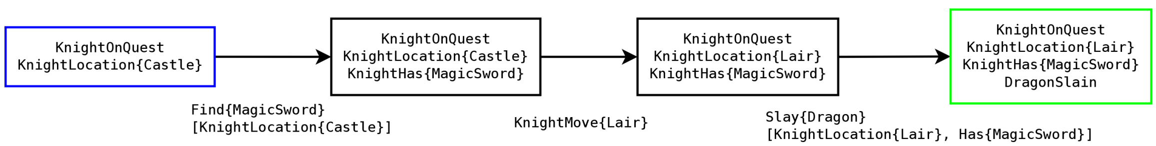


Figure 7.1: A linear story showing event preconditions

Propositions can also contain story metadata to help keep track of the story as a whole instead of only simulating the world. This could include data about formal story structures such as Propp's morphology or The Hero's Journey to make sure the resulting story fits into these patterns. Encoding a story structure into the propositions can allow the stories to benefit from aspects of the structures. For example since The Hero's Journey describes how a character goes on an adventure, encounters a crisis and returns transformed, it can be used to help the generated stories be forward-progressing and not repetitive.

7.2 Structure of Potential Narratives

A state transition graph made with all possible applications of events can represent all potential stories in that space. The structure of this graph will depend on the events that create it and the level of interactivity. Human authored potential narratives tend to show recurring patterns. The simplest possible story structure is linear, however stories can also have branches, parallel paths, or a dynamic hierarchy. More information on story structures can be found in *Fundamentals of Game Design* (Adams, 2014). Non-linear story structure is required for player agency and some common non-linear patterns are listed below.

Branching

A branching story structure has new pathways at player decision points. This structure is present with high levels of player freedom, impact, and story control as each decision leads the story into unique states. As a downside it also requires the most development resources. The high cost of creating unique content that players can miss entirely means fully branching potential narratives are uncommon. Unless these branches can be computer generated in their entirety this approach is impractical for larger projects since any substantial amount of manual authoring will lead to an unreasonable production cost.

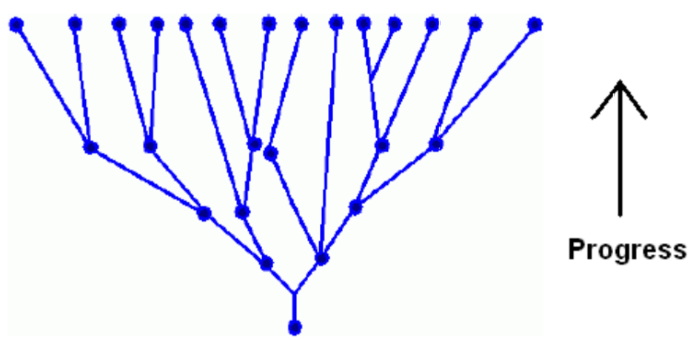


Figure 7.2: Branching Story Structure (Beyak and Carette, 2011)

Parallel Paths (Foldback)

Parallel paths or foldback patterns provide choices which change the direction of the story however every decision eventually leads back into the same event. This structure provides the illusion that player choices substantially change the story direction without needing to create too many unique states. This can give some sense of freedom and story control however extra effort must be put in to make sure the player feels the impact of their actions. This is a popular choice for potential narratives in the game industry since the player must experience all the major content created in just one play through while still having some sense of player agency.

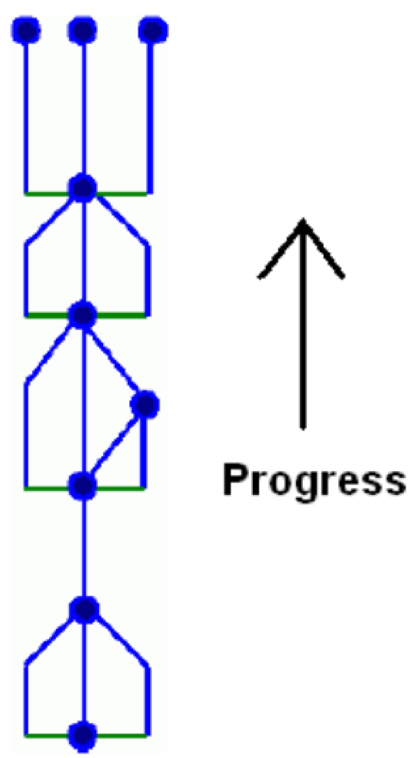


Figure 7.3: Foldback Story Structure (Beyak and Carette, 2011)

Dynamic Hierarchical

Dynamic Hierarchical potential narratives group events into sections to keep story complexity down. This is the structure that is seen most often in Interactive Fiction that wants to create a high level of player agency with a reasonable scope. Related plot events are grouped together into small and manageable sections with only a few possible exit states. Selecting which group of events to pursue gives the player story control and within the individual groups there can be a high level of freedom and impact. Connecting many of these sections together can tell a complex story without requiring the same level of content creation as a branching structure because there are only a few possible propositions that result from each group.

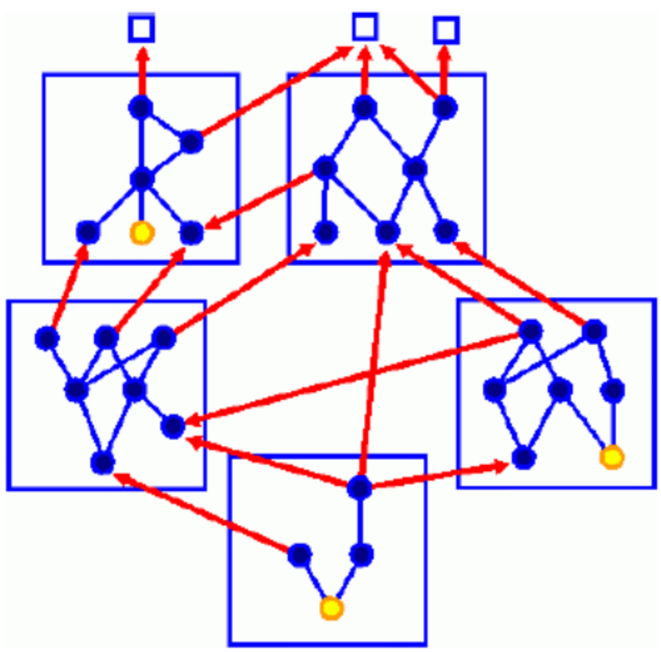


Figure 7.4: Dynamic Hierarchical Structure, Beyak and Carette (2011)

7.3 Modeling Interactivity

A potential narrative is meant to be traversed by a player who is an active participant in the story so that narrative branch points are a result of player decisions. It is important to note that each branch point is not necessarily the result of an immediate player decision such as choosing which fork in a road to go down. The result of a player decision may be delayed and change the story at a later point, or even multiple later points. The story can also branch as a result of indirect player decisions such as the outcome of a battle, or the status of their inventory.

The story metadata can also include information about the player's experience. If the player encounters certain themes or information this can be noted down to influence their path later or to avoid presenting existents that would feel arbitrary if player didn't have prior knowledge. There is a lot of freedom for authors to create different interactions depending on the story domain and what is in encoded into events.

7.4 Synthesis

Potential narratives can be modeled as many potentially interactive events forming a world state transition diagram with features as follows:

- The building block of this model is the event, which can represent an interactive decision point, a game mechanic, or the actions of a non player character. Each event is described by the requirements for an event to happen and how the story changes after it does.

- The representation of all these events in a state transition diagram creates the potential narrative's structure. The type of structure depends greatly on the design of the events, for example if the designer wants to ensure that the player experiences a certain event then this will lead to a foldback pattern.

- Besides simulating the world, events can also include story metadata such as player models or formal story structures. This metadata can be used to help create desirable traversals in the potential narrative, such as a traversal that fits the pattern of The Hero's Journey.

Further work on this model would involve recreating many popular interactive stories in this model and closely examining the resulting structures and design of events in order to refine this representation of potential narrative.

Chapter 8

Prototype Implementation

This chapter will cover our current design and implementation of a prototype for a potential narrative generator based on the state transition model. The goal of this prototype is to find possible player traversals through a story domain encoded according to our model. Generation of potential narrative for an arbitrary story domain is out of scope for this thesis. Output from the generator is not meant to be interactive as it is meant to examine possible player traversals and not for creating a playable product.

8.1 Implementation

This prototype was built using the *core.logic* (Nolen and Hickey, 2011) programming library for *Clojure*. Clojure is a dialect of Lisp which is a functional programming language. More information on Clojure is available on the official website https://clojure.org/ which contains the rationale behind the language and how to get started with writing Clojure programs.

core.logic is a Clojure implementation of relational programming based on miniKanren from William Byrd's dissertation *Relational Programming in miniKanren: Techniques, Applications, and Implementations* (Byrd, 2009). The core.logic code repository can be found here: https://github.com/clojure/core.logic and an interactive primer on core.logic coding can be found here: https://nextjournal.com/try/helios/a-core-logic-primer. The code used in this prototype extends an implementation in the *miniKanren* chapter of *Seven More Languages in Seven Weeks* (Tate et al., 2014) with modifications for extra functionality and optimization. The original code can be downloaded from the publisher's website The Pragmatic Bookshelf: https://pragprog.com/titles/7lang/ The full code with our modifications and example can be found on our **github:** https://github.com/asciga/IFgen-prototype.

The example story domain used for generation is written especially for this prototype, making use of *The Writer's Journey* (Vogler, 2007) to generate stories that follow this story structure. The example is encoded in the potential narrative model of worldstates and events so that the output of the prototype is a series of events representing a traversal.

There are similarities between planning and narrative generation. They both refer to a sequence of events each of which describes a transformation from one state to another state. As long as the operators in a plan are possible in a story world then the plan is able to describe a story (Riedl and Young, 2010). Planning algorithms solve the planning problem which takes a domain theory, an initial state, and a goal state described by a set of propositions and finds a sound sequence of actions to transition the initial state to the goal state. The domain theory describes how the state is able to change which can include what operators the planning problem will use to transition to the goal state.

8.1.1 Narrative Planning

There are many parallels between AI planning and the potential narrative traversal generation of our model. They both take a domain theory and initial world state and generate a sequence of state transformations to a given goal state. The domain theory in our case is the story space of all possible events and propositions. Narrative planning is explored in depth in *Narrative Planning: Balancing Plot and Character* Riedl and Young (2010). Linear logic has been shown to be a candidate to create plans (Martens et al., 2013) and this prototype makes use of the logic programming provided by *core.logic* in order to generate traversals through a potential narrative.

8.1.2 Relational Programming

The prototype uses a representation of potential narrative written as a database of story events fed into a *core.logic* method which defines the goal predicates of a valid traversal so that the program can find a valid traversal which satisfies a given goal state.

Potential Narrative Representation

We used a manually authored database of possible story events to represent the potential narrative. These follow the definition of event in the model chapter, being composed of two arrays of propositions representing the preconditions and postconditions. Each event also contains a string that briefly describes what the event represents in the story. An example event can be found below:

```
[[:Test :Deadline] [:UnchartedSpace :Deadline]
"With the deadline coming up so quickly the hero must take
drastic action. They pilot their ship into uncharted space"]
```

In this event the propositions *:Test* and *:Deadline* must be present in the world state for the event to occur. The propositions *:UnchartedSpace* and *:Deadline* will be present in the world state after the event occurs. Note that *:Deadline* appears twice which means that this proposition must be in the world state before and after the event. This is because propositions in the preconditions are consumed, so *:Test* won't be present in the world state afterwards. More on the way these events were created and the meaning of the propositions can be found in the event authoring section.

Valid Traversals

The database of events representing the potential narrative is then imported into the *core.logic* program. As mentioned earlier, this prototype extends coding done in *Seven More Languages in Seven Weeks* (Tate et al., 2014). The authored events are turned into a database of relations from precondition to postcondition. This database is used in the `storyo` method to find a series of these relations that results in the desired goal state from a starting state. We refer to a series of relations that reach a goal state with all event conditions satisfied as a valid traversal.

`storyo` is a relation with three parameters, a starting state, the desired goal state, and a list of actions describing the traversal from start to goal. Relations in relational programming are a list of goals that must be true in order for the relation to hold. A valid traversal can be described by the following goals:

- A story event is valid if the preconditions and postconditions are satisfied.

- A valid event is added to a list of events representing the traversal.
- When the world state contains all goal propositions, the traversal is valid and the relation is fulfilled.
- If the goals are not met, recursively call `storyo` to check the new state for another valid event.

For the first goal of valid preconditions and postconditions for a given event the code uses a second relation known a `actiono` which takes the parameters of a starting state, the modified state, and the event which will modify it. This is called in `storyo` as follows:

```
(actiono start-state new-state action)
```

The `actiono` relation has the goal of finding a valid event relation known as `ploto` and then removing the preconditions and adding the postconditions to the given state to produce the new-state. This code was optimized by swapping the order of `ploto` and `foroall` to reduce function calls and compute traversals an order of magnitude faster. This optimization works because the ordering of goals matters in *core.logic* since the program tries to satisfy the goals from the top. If `foroall` is run first the program will search for all possible combinations of propositions to check which of those combinations are valid relations in the database. By running `ploto` first the search space is constrained to only relations in the database. The extra performance allowed the extension of the original code to handle events with multiple inputs and multiple outputs through the relations `foroall` and `remberoall` which handle the preconditions and postconditions respectively. The code for `actiono` is below:

```
(defn actiono [state new-state action]
        (fresh [in out temp]
                (ploto in out)
                (for-all #(membero % state) in)
                (rembero-all in state temp)
                (appendo out temp new-state)
                        (== action [in out])))
```

The next goal is fairly straightforward, once a valid event has been found the program uses `conso` to construct a list called *actions* out of *action* which is the valid event and *new-actions* which is the rest of the traversal up until this point.

```
        (conso action new-actions actions)
```

The final two goals are part of a single conditional, either the goals are met and the traversal can be returned or the program must recurse to search for more events. The code uses `conda` to represent this which is satisfied once any of the clauses succeed and ignores the rest.

```
        (conda
                [(everyg #(membero % new-state) end-elems)
                (== new-actions [])]
                [(storyo* new-state end-elems new-actions)])))
```

`conda` has two goals under it, the first of which uses `everyg` to check if the new story state contains all of the goal propositions. If every goal proposition is a member of the story state then the traversal is valid, `storyo` is satisfied and the prototype finishes. If this goal is not met then `storyo` is called recursively with the new state, the same goal, and the new traversal. This recursion continues looking for possible events in the potential narrative to build the traversal until the goal state is met. The `storyo` function is composed using these four goals as follows:

```
(defn storyo [start-state end-elems actions]
        (storyo* (shuffle start-state) end-elems actions))


(defn storyo* [start-state end-elems actions]
        (fresh [action new-state new-actions]
        (actiono start-state new-state action)
        (conso action new-actions actions)
        (conda
                [(everyg #(membero % new-state) end-elems)
                (== new-actions [])]
                [(storyo* new-state end-elems new-actions)]))))
```

Note that `storyo` is the relation used to start the process which shuffles the starting state before running `storyo*` which is where the goals of a valid traversal are checked. `actiono` and `storyo` implement the potential narrative model in a very concise way compared to other implementations. This is one of the reasons why relational programming was selected for this prototype, more on the other attempted prototypes can be found in the Failed Implementations section. In order to run the program a user must use the *core.logic* `run` function and call the storyo relation.

```
        (with-db story/story-db
```

```
(run 1 [q]
(story/storyo [:start] [:end] q)))
```

The `run` function executes goals until a certain number of results are found. The parameters are first the amount of results we want, a name for the binding *q*, and the goal relation to be run, `storyo`. We provide `storyo` with the desired start state, ending states, and then instead of a traversal we give the name of the binding we want as a result, *q*. The above input tells `run` that we will provide the starting state and ending state and we want it to search for one valid traversal according to the goal `storyo`. It should be noted that the relation could also search for possible starting states or ending states if given a valid traversal. Although this versatility is not used in this thesis, it is another reason why relational programming was selected for the prototype as it may have interesting applications for further research.

Event Authoring

The next step of the prototype was to build a larger potential narrative to test the traversal generation. The primary consideration for writing this potential narrative was to ensure that the resulting traversals were coherent and credible. In the end the example consisted of 114 potential events which could generate traversals telling a three act story structured around *The Writer's Journey* (Vogler, 2007). The theming is a science fiction adventure telling the story of a spaceship captain who must go on a heroic adventure with their crew. Writing the potential narrative involved fulfilling the three act structure with a variety of events and then ensuring the propositions created by those events are consumed by later ones so it does not feel arbitrary.

The Writer's Journey is a story structure which describes a heroic adventure as twelve stages in three distinct acts. The stages describe in broad strokes what information should be presented to the reader and how the story changes. It should be noted this is not a strict set of rules or a claim that all adventure stories must follow this structure, it is a set of guidelines and a starting point for writers looking to create their own. The first act is described in five stages:

Act One

- **The Ordinary World:** The backstory and theme are established
- **Call to Adventure:** An inciting incident forces the hero into action
- **Refusal of the call:** A hero may avoid the call
- **Meeting with the mentor:** A character provides some help or wisdom
- **Crossing the first threshold:** The turning point where the adventure truly begins

These stages are turned into propositions to denote which part of the story a particular event is meant to fulfill. This is demonstrated at the very beginning of the event database which can be found in **Appendix B: Prototype Story Elements**:

```
[[:start] [:OrdinaryWorld]
"Start generating Act 1."]

[[:OrdinaryWorld] [:SpaceshipMission :CalltoAdventure]
"In a sci-fi world a spaceship with a capable crew is
restocking supplies."]
```

```
[[:CalltoAdventure] [:StrangeReadings :Refusal]
"Picked up strange readings in a solar system "]


[[:CalltoAdventure] [:DistressCall :Refusal]
"Picked up a Distress Call in a system "]
```

The traversal begins at *:OrdinaryWorld* which sets the theme and transitions into the *:CalltoAdventure* afterwards. When *:CalltoAdventure* is part of the story state the generator is able to pick from a list of events which fulfill that story structure. In the example above it could select either *:StrangeReadings* or a *:DistressCall*. That proposition is added to the story state in addition to *:Refusal* which denotes an event from the next stage should be selected.

The other important consideration when writing events is that the propositions added in earlier stages should potentially be utilized by later ones. This makes the story seem less arbitrary and in the case of an interactive story would make the player feel as if their actions had impact. In the previous example *:DistressCall* is added to the story state and so it should be used later in the story and preferably in several places and different traversals. Here are some events which use *:DistressCall* in the prototype:

```
[[:Test :DistressCall] [:UnchartedSpace :DistressCall ]
"With the urgency of the DistressCall. They pilot their
 ship into uncharted space."]


[[:Approach :DistressCall] [:AsteroidBelt :Ordeal]
```

```
"The ship approaches the source of the distress call,
 somewhere in an asteroid belt."]

[[:Reward :DefeatedPirates :DistressCall]
[:FakeDistressCall :PirateRevenge :RoadBack]
"The pirates had set the distress call as a trap! "]



[[:Reward :SwarmDeactivated :DistressCall ]
[:DroneDisabledShip :AIRevengeSwarm :RoadBack]
"With the swarm deactivated the hero follows the distress call
 to a disabled ship that was attacked by the drones."]
```

Output

The output of this prototype is a traversal through the given potential narrative starting at the given start state and ending once the world state contains the goal propositions. A traversal is formatted as a list of events which show each state transition towards the goal. Here is an example output for act one of a traversal:

```
(([[:start] [:OrdinaryWorld]] [[:OrdinaryWorld]
        [:SpaceshipMission :CalltoAdventure]]
[[:CalltoAdventure] [:DistressCall :Refusal]]
[[:Refusal] [:NoRefusal :MeetingMentor]]
[[:MeetingMentor] [:OldFriend :CrossingThreshold]]
[[:CrossingThreshold] [:Deadline :Test :Ally :Enemy]] ...
```

This traversal shows us that the call to adventure is a *:DistressCall*, the hero does not refuse the call to adventure and they are mentored by an *OldFriend* before they are forced into action because of time pressure from a *:Deadline*. These propositions may then be used by later events to create a variety of credible traversals. This is demonstrated as the output continues into act two:

```
... [[:Test] [:SpacePirateAttack :SpacePirates]]
[[:Enemy :SpacePirates] [:EnemyPirate :Approach]]
[[:Ally] [:AllyCrew]]
[[:Approach :DistressCall] [:AsteroidBelt :Ordeal]]
[[:Ordeal :EnemyPirate :AsteroidBelt] [:PirateBeltAmbush]]
[[:PirateBeltAmbush] [:HeroCapturedByPirates]]
[[:HeroCapturedByPirates :AllyCrew] [:HeroEscapesPirates :Reward]] ...
```

In act two the story establishes a *:Test*, *:Ally*, and *:Enemy* for the hero. The *:Test* is a *:SpacePirateAttack* which also introduces the *:SpacePirates* who end up being selected as an enemy for the story through *:EnemyPirate*. However the hero is not alone as the crew of their spaceship prove themselves to be a valuable ally in *:AllyCrew*. Throughout all of this the hero begins the *:Approach*, seeking out the *:DistressCall* which began this quest. Unfortunately they must first go through an *:Ordeal* as it turns out the pirates had set the distress call as an ambush in *:PirateBeltAmbush* and so the hero is captured in *:HeroCapturedByPirates*. Thankfully the loyal crew hatches an escape plan and because of *:AllyCrew* the *:HeroEscapesPirates* and the story continues.

8.2 Failed Implementations

There were a few other attempted implementations using different methods before we arrived at relational programming using *core.logic*. Prototyping began with a Haskell program called IFGen which used the *Fast Downward Planning* system (Helmert, 2006). Although it worked for small examples it was difficult to extend due to inexperience with functional programming and primarily relying on an external package. Afterwards we started a prototype called TweeGen inspired by *Twine* (Klimas, 2009) which uses a graph representation for stories. The idea was to use the GraphRewriting package (Rochel, 2010) to directly generate a graph representation of a story that could be input into Twine. Again although it worked for toy examples it was difficult to extend as writing rewrite rules to represent a story felt unintuitive and the code wasn't straightforward.

Eventually the prototype evolved into using relational programming after discovering a miniKanren distribution for Haskell called *DSKanren* (Gratzer, 2014). Soon afterwards while researching other miniKanren implementations the core.logic library was found for Clojure which led to the current prototype design. With more Haskell experience DSKanren could potentially provide the same functionality but the existence of related code which could be extended for our purposes was more appealing for a first prototype.

Chapter 9

Conclusion

Digital interactive stories present their own challenges compared to other media. Interactivity means the stories will often be non-linear which creates additional considerations to make sure the story is coherent and that the interactive components are satisfying for the player. This satisfaction comes from a sense of agency the player has over a story, however if those interactions change the story then content must be made for all the possible story variations. This content may not even be experienced in a single play through, making interactive stories expensive to create relative to linear non-interactive stories of similar length. Despite these costs potential narratives provide a uniquely engaging experience. The research done in this paper is meant to provide a paradigm to evaluate and encode the properties of potential narratives towards the creation of story generation tools.

The use of specialized story tools is not widespread in the games industry (Djaouti, 2013) since development studios usually rely on game specific manual authoring (Despain, 2020). The systems to manage large non-linear narratives can be very complex, involving thousands of flags across many non-player characters to represent different possible story states. Research into using logic programming to verify the coherency of these systems is promising and this thesis expands that by examining interactive fiction and creating an intuitive model for encoding potential narratives. This began with an analysis of interactive fiction and the creation of a set of qualitative evaluation criteria to direct the model creation.

After some research in the area the model we selected was similar to a state-transition diagram with important story events codified as transitions of a world state. This computer friendly representation is also easy to grasp for manual authoring. The model contains only the basic functionalities for representing interactive narratives, more on how this could be expanded will be in the future work section. To prototype the model we selected relational logic which has been used before in other research. For our purposes this was a promising proof of concept with many possibilities for extension.

9.1 Future Work

Further research on this thesis will focus on three distinct areas, the model, the evaluation criteria, and the relational programming implementation. In this section we will examine possible research directions for each area and provide an outline to pursue them.

Continued refinement of the model is the first concern. Since the model should be able to represent the potential narratives found in the industry it would be prudent analyze many different interactive fiction stories using our model. If there are elements that cannot be represented within the model then these would be good candidates for further development. One example is the representation of numeric variables such as hit points or currency which is common in interactive fiction but not mentioned by our current model.

With several modeled potential narratives the next step is to analyze them for structural patterns or other similarities. Some of these patterns are already mentioned in the model chapter such as dynamic hierarchical potential narratives but there may be others. Having a database of potential narrative is also an opportunity to examine which of our qualitative evaluation criteria could be quantitative. A thorough analysis of more potential narratives through the lens of our evaluation criteria should make it more clear which criteria could be represented. Coherency and player agency both rely on the structure of the potential narrative which make them good candidates for this.

The relational programming prototype could use further development as well. Additional functionalities could be implemented such as the ability to group propositions using tags, representing numeric variables and how they change, and the ability to select if a proposition isn't consumed by an event. The prototype could also use more optimization. One potential method for this is to investigate the use of pruning during the traversal search to trim impossible searches to improve computation time.

Finally after all these tasks we would revisit the generation of potential narratives instead of only generating traversals through a potential narrative.

Appendix A

Dreamhold Room Analysis

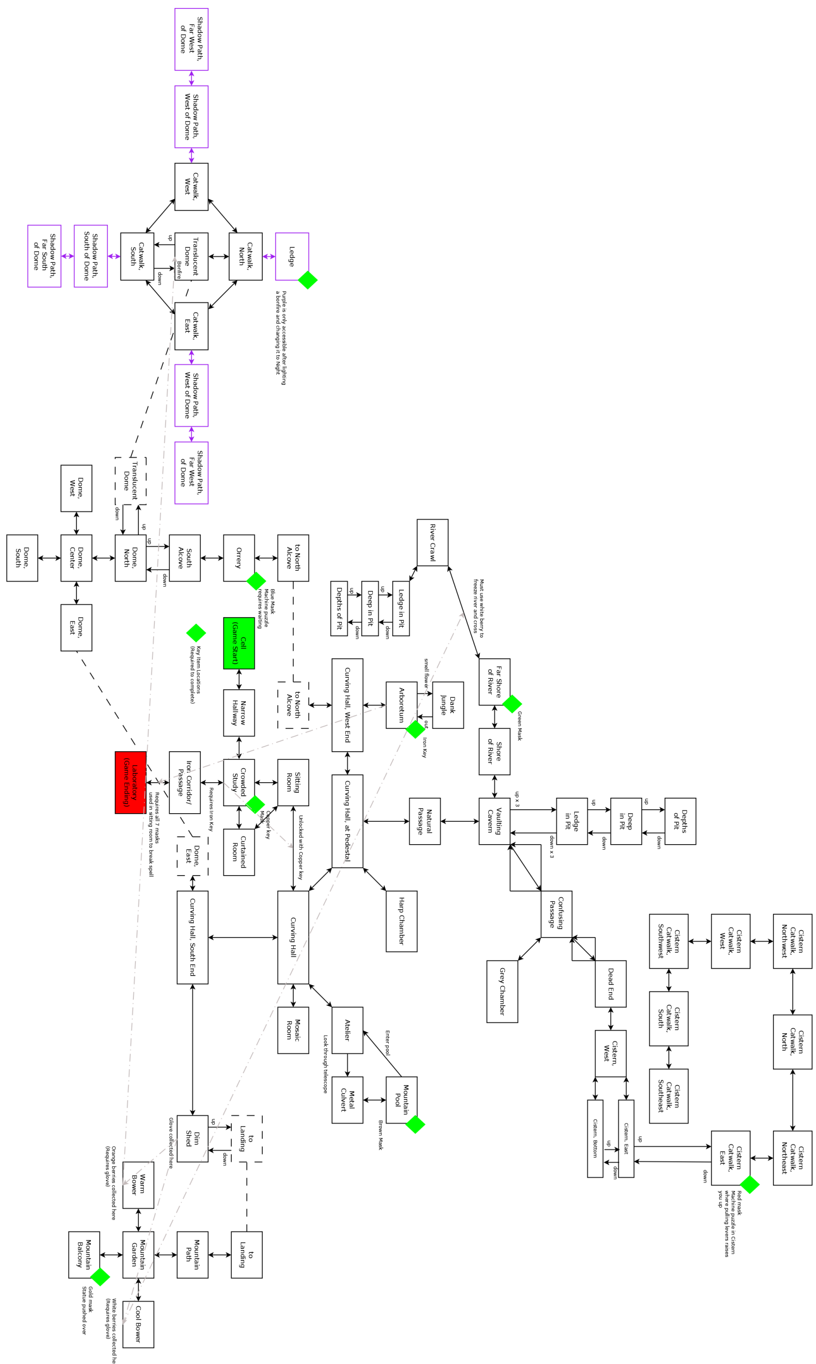


Figure A.1: A room map for The Dreamhold

Appendix B

Prototype Story Elements

```
;;Here is where the relations are coded, they take the following format:
;;[[:input] [:output]
;;"Text description"]

(in-ns 'ifgen.story)

(def story-elements

[[[:start] [:OrdinaryWorld]
"Start generating Act 1."]

;;Act 1 Ordinary World
[[:OrdinaryWorld] [:SpaceshipMission :CalltoAdventure]
"In a sci-fi world a spaceship with a capable crew is restocking supplies."]
```

```
;;Act 1 Calls to Adventure
[[:CalltoAdventure] [:StrangeReadings :Refusal]
"Picked up strange readings in a solar system "]

[[:CalltoAdventure] [:DistressCall :Refusal]
"Picked up a Distress Call in a system "]

[[:CalltoAdventure] [:TransportingCargo :Refusal]
"Tasked with transporting Cargo"]

[[:CalltoAdventure] [:FerryingPassengers :Refusal]
"Tasked with ferrying passengers."]

;;Act 1 Refusing the call, motivations
[[:Refusal] [:MotivatedbyMoney :MeetingMentor]
"Doesn't want to go but needs the money."]

[[:Refusal] [:AfraidofAdventure :MeetingMentor]
"Afraid of what might happen on the journey."]

[[:Refusal] [:NotAbandonFamily :MeetingMentor]
"Afraid of leaving family and friends."]

[[:Refusal] [:NoRefusal :MeetingMentor]
```

```
"The character does not refuse the call."]

;;Act 1 Mentors
[[:MeetingMentor] [:OldFriend :CrossingThreshold]
"And old friend sends a message."]

[[:MeetingMentor] [:Expert :CrossingThreshold]
"An expert is assigned to travel with the hero."]

[[:MeetingMentor] [:MysteriousBenefactor :CrossingThreshold]
"A mysterious benefactor contacts the hero."]

[[:MeetingMentor] [:Starmaps :CrossingThreshold]
"The hero consults old navigation charts to find their way."]

[[:MeetingMentor] [:NoMentor :CrossingThreshold]
"The hero consults old navigation charts to find their way."]

;;Act 1 Crossing The Threshold
[[:CrossingThreshold] [:JourneyBegins :Test :Ally :Enemy]
"The spaceship sets off on its journey."]

[[:CrossingThreshold] [:ShipMalfunction :Test :Ally :Enemy]
"The spaceship sets off on its journey and immediately has some
```

```
malfunctions to be fixed"]

[[:CrossingThreshold] [:Deadline :Test :Ally :Enemy]
"The deadline has been moved up and the spaceship must depart now."]

[[:CrossingThreshold :TransportingCargo]
[:AttemptToSteal :Test :Ally :Enemy]
"Someone has tried to steal the cargo, the hero has to go now!"]

;;Act 2 Tests, Allies, Enemies

;;Pirate theme
[[:Test] [:SpacePirateAttack :SpacePirates ]
"The ship is attacked by SpacePirates"]

[[:Ally :MotivatedbyMoney :SpacePirates] [:AllyPirate :MotivatedbyMoney]
"Promising the pirates a cut of the profits, the hero convinces them to
join their side."]

[[:Enemy :SpacePirates] [:EnemyPirate :Approach]
"A large crew of space pirates is gunning for the hero."]

;;Rogue AI theme
[[:Test :Deadline] [:UnchartedSpace :Deadline ]
```

```
"With the deadline coming up so quickly the hero must take drastic action.
They pilot their ship into uncharted space"]

[[:Test :DistressCall] [:UnchartedSpace :DistressCall ]
"With the urgency of the DistressCall. They pilot their ship into
uncharted space."]

[[:Enemy :StrangeReadings] [:EnemyRogueAI :StrangeReadings :Approach]
"Picking up a burst of the readings the hero was searching for, their
ship is attacked by unmanned drones."]

[[:Enemy :UnchartedSpace] [:EnemyRogueAI :Approach]
"While flying through uncharted space, the ship is attacked by
unmanned drones."]

[[:Ally :UnchartedSpace :MysteriousBenefactor] [:AllyRogueAI]
"An artificial intelligence reveals itself to be the mysterious benefactor,
offering further assistance."]

[[:Ally :UnchartedSpace :Starmaps] [:AllyRogueAI]
"The hero encounters a rogue articial intelligence in uncharted space.
 After offering the starmaps to it, it is willing to help"]

;;BountyHunter Theme
```

```
[[:Test] [:BountyHunterAttack ]
"The hero must fight off a bounty hunter's ship."]

[[:Enemy :FerryingPassengers :BountyHunterAttack]
[:EnemyBountyHunter :FerryingPassengers]
"One of the passengers is being hunted by a bounty hunter."]

[[:Enemy :BountyHunterAttack] [:EnemyBountyHunter :Approach]
"Someone has put a bounty on the hero's head and he is now being hunted."]

;;Other Allies
[[:Ally :ShipMalfunction :Expert] [:AllyExpert]
"The Expert assigned to the ship is able to repair the malfunctions,
 they are reliable!"]

[[:Ally] [:AllyCrew]
"At a spacestation the hero spends some time reinforcing
bonds with the crew. They can count on them."]

;;Coming out of this section it's possible to have:
;; Tests: SpacePirateAttack, UnchartedSpace, BountyHunterAttack
;; Allies: AllyPirate, AllyRogueAI, AllyExpert, AllyCrew
;; Enemies: EnemyPirate, EnemyRogueAI, EnemyBountyHunter
```

```
;;Act 2 Approach to Inmost Cave

;;Basically all preparations that are done before the ordeal.
;; Kinda based on ally/enemy
;;can also have a guardian.


;;Upgraded Ship

[[:Approach :AllyRogueAI] [:Approach :UpgradedShip :Ordeal]
"The friendly artificial intelligence upgrades the hero's ship,
improving its capabilities."]

;;Asteroid Belt
[[:Approach :StrangeReadings] [:AsteroidBelt :Ordeal]
"The ship approaches the source of the strange readings somewhere in an
asteroid belt."]

[[:Approach :DistressCall] [:AsteroidBelt :Ordeal]
"The ship approaches the source of the distress call, somewhere in an
asteroid belt."]

;;Waiting to Dock
```

```
[[:Approach] [:WaitingToDock :Ordeal]
"Arriving at their destination, the ship waits to dock at a space station"]

;;Passenger goal

[[:WaitingToDock :FerryingPassengers] [:SpaceStation :FerryingPassengers]
"After some waiting the ship is accepted into the station and docks,
unloading the passengers."]

[[:WaitingToDock :TransportingCargo] [:SpaceStation :TransportingCargo]
"After some waiting the ship is accepted into the station and docks
unloading the cargo."]

;;Cargo goal
[[:WaitingToDock :TransportingCargo] [:IllegalCargo]
"A cargo scan flags the hero's ship as carrying illegal goods! They have
to talk their way out of this one"]

[[:IllegalCargo :AllyPirate]
[:AllyPirate :SpaceStation :TransportingCargo]
"Although the hero is refused entry to the station, their pirate friends
find them a way to smuggle it on board."]

[[:IllegalCargo :NotAbandonFamily]
```

```
 [:NotAbandonFamily :SpaceStation :TransportingCargo]
"The hero gives a heartfelt speech to the docking authorities about
family back home, he needs this job done he doesn't
know anything about the cargo."]

[[:IllegalCargo :AllyCrew] [:AllyCrew :SpaceStation :TransportingCargo]
"The crew of the ship show the hero a hidden smuggling compartment
for the cargo that'll let them hide it and dock."]

;;approaches places: SpaceStation, AsteroidBelt

;;Act 2 Ordeal

;;Pirates attack station
[[:Ordeal :EnemyPirate :SpaceStation] [:PiratesAttackStation]
"While docked at the station it is attacked by a pirate armada!"]

[[:PiratesAttackStation :UpgradedShip]
[:UpgradedShip :FoughtOffPirates :Reward]
"The hero uses their upgraded to ship to fight off the entire armada."]

[[:PiratesAttackStation :AllyCrew]
[:AllyCrew :FoughtOffPirates :ShipDamaged :Reward]
"The hero and their loyal crew all agree the station must be protected.
```

```
They fight valiantly but the ship is damaged and must
dock into the station."]

;;AI takes over Station
[[:Ordeal :EnemyRogueAI :SpaceStation] [:AIStation]
"When the hero is being pulled into the station he realizes the station
has been taken over by an Artifical Intelligence!"]

[[:AIStation :AllyExpert] [:ExpertSacrifice :StationDestroyed :Reward]
"The expert sacrifices themselves to rescue the hero and their ship,
staying behind to make the station self destruct."]

[[:AIStation :AllyPirate] [:PiratesSaveStation :Reward]
"The hero's newfound pirate friends bring their whole fleet to fight
the AI station."]

[[:AIStation :Starmaps][:AITakesShip :Reward]
"The hero gives up their ship and starmaps in return for the safety
of the people on board the station."]

;;Hero caught by bounty hunter
[[:Ordeal :EnemyBountyHunter :SpaceStation] [:CaughtByBountyHunter]
"The bounty hunter has told the station authorities about the bounty,
the ship is captured!"]
```

```
[[:CaughtByBountyHunter :OldFriend] [:OldFriendRescue :Reward]
"The hero is put in prison when it turns out the bounty hunter
knows their old friend! They help the hero break out of prison
and back to their ship"]

[[:CaughtByBountyHunter :AllyPirate] [:PirateRescue :Reward]
"The pirates the hero befriended set a plan to break the hero out
of prison and have him join them."]

;;Ambushed by pirates in an asteroid belt
[[:Ordeal :EnemyPirate :AsteroidBelt] [:PirateBeltAmbush]
"Pirates laid a trap for the hero in the asteroid belt!"]

[[:PirateBeltAmbush :UpgradedShip] [:DefeatedPirates :Reward]
"With the upgraded ship the hero is able to defeat the pirates."]

[[:PirateBeltAmbush] [:HeroCapturedByPirates]
"The hero's ship is disabled and they are captured by the pirates!"]

[[:HeroCapturedByPirates :AllyCrew] [:HeroEscapesPirates :Reward]
"The hero and their loyal crew execute a daring escape on one of
their ships."]
```

```
[[:HeroCapturedByPirates :AllyExpert] [:HeroEscapesPirates :Reward]
"With the help of the Expert, The hero and their crew execute a daring
escape on one of their ships."]

;;A swarm of rogue AI drones
[[:Ordeal :EnemyRogueAI :AsteroidBelt] [:RogueAISwarm]
"A swarm of Rogue AI drones are mining this asteroid belt."]

[[:RogueAISwarm :AllyExpert] [:SwarmDeactivated :Reward]
"The expert engineer creates a devices that deactivates the swarm
at least temporarily."]

[[:RogueAISwarm] [:EscapeSwarm :Reward]
"The hero makes a daring escape after scouting the rogue swarm of
machines, deftly flying between asteroids."]


;;Showdown with the bounty hunter
[[:Ordeal :EnemyBountyHunter :AsteroidBelt] [:BountyHunterShowdown]
"The Bounty Hunter tracked the hero out here where they have a showdown."]

[[:BountyHunterShowdown :UpgradedShip] [:DefeatedBountyHunter :Reward]
"With the upgraded ship the hero is able to fight off the bounty hunter."]
```

```
[[:BountyHunterShowdown] [:HeroCapturedByBountyHunter :Reward]
"The Bounty Hunter disables the hero's ship and captures them."]

[[:HeroCapturedByBountyHunter :AllyCrew] [:BountyHunterShip :Reward]
"With the help of their loyal crew, the hero takes over the bounty
hunter's ship and escapes!"]



;;recombine into a few paths.

;; a few main routes for this
;;
;; Ship is Destroyed, new ship from Pirates or Bountyhunter
;; Ship is destroyed, stuck on space station.
;; Ship isn't destroyed, celebration on board ship.

;;Act 2 Reward
;;let's focus on the completion of the goal, cargo / signal / etc.

;;Celebration / Campfire / Woo we did it!
;; :StrangeReadings
;; :DistressCall
;; :TransportingCargo
;; :FerryingPassengers
```

```
;;Pirates attack station :StationRescued, :StationEvacuated
[[:Reward :FoughtOffPirates :FerryingPassengers]
[:DeliverySuccess :PirateRevenge :RoadBack]
"The hero pulls through the passengers are safely delivered."]

[[:Reward :FoughtOffPirates :TransportingCargo]
[:DeliverySuccess :PirateRevenge :RoadBack]
"The hero docks with the station and transfers the cargo they were
meant to deliver."]


;;AI takes over Station :ExpertSacrifice :StationDestroyed,
;; :LootedStation, :AITakesShip
[[:Reward :StationDestroyed :FerryingPassengers]
[:StationEvacuated :PassengersOnShip :AIRevengeSwarm :RoadBack]
"With the expert's sacrifice, the hero rescues not only the passengers
 now without a destination, but also the station's crew."]

[[:Reward :StationDestroyed :TransportingCargo]
[:StationEvacuated  :CargoOnShip :AIRevengeSwarm :RoadBack]
"With the expert's sacrifice and destruction of the station, the cargo
is left on board the ship."]
```

```
[[:Reward :PiratesSaveStation :FerryingPassengers]
[:DeliverySuccess :AIRevengePirate :RoadBack]
"Together with the pirates the hero manages to fight off the AI and
deliver the passengers safely."]

[[:Reward :PiratesSaveStation :TransportingCargo]
[:DeliverySuccess :AIRevengePirate :RoadBack]
"Together with the pirates the hero manages to fight off the AI and
deliver their cargo safely,"]

[[:Reward :AITakesShip :FerryingPassengers] [:DeliverySuccessNoShip]
"The ship is unloaded of passengers before the AI takes it over, leaving
the hero with no ship but a successful mission. "]

[[:Reward :AITakesShip :TransportingCargo] [:DeliverySuccessNoShip]
"The ship is unloaded of cargo before the AI takes it over, leaving the
hero with no ship but a successful mission. "]

[[:DeliverySuccessNoShip] [:StayOnStation :RoadBack]
"The hero decides to stay on the space station for some time before
 looking for an opportunity to return home."]

;;Hero caught by bounty hunter :OldFriendRescue :PirateRescue
```

```
[[:Reward :OldFriendRescue :FerryingPassengers]
[:DeliverySuccess :NewBountyHunter :RoadBack]
"During detainment the passengers are let onto the station by the
authorities."]

[[:Reward :OldFriendRescue :TransportingCargo]
[:DeliverySuccess :NewBountyHunter :RoadBack]
"Once released the hero is able to transfer the cargo to the space
station."]

[[:Reward :PirateRescue :FerryingPassengers]
[:DeliverySuccess :PirateBetrayal :RoadBack]
"During detainment the passengers are let onto the station by the
authorities."]

[[:Reward :PirateRescue :TransportingCargo]
[:DeliverySuccess :PirateBetrayal :RoadBack]
"The pirates take a few pieces of cargo as payment for their rescue but
allow the hero to deliver the rest."]

;;Ambushed by pirates in an asteroid belt
;; :DefeatedPirates :HeroEscapesPirates
[[:Reward :DefeatedPirates :StrangeReadings]
[:ValuableAsteroid :PirateRevenge :RoadBack]
```

```
"With the pirates defeated the hero investigates the strange readings
which turned out to be an asteroid with exotic minerals."]

[[:Reward :DefeatedPirates :DistressCall]
[:FakeDistressCall :PirateRevenge :RoadBack]
"The pirates had set the distress call as a trap! "]

[[:Reward :HeroEscapesPirates :StrangeReadings]
[:ValuableAsteroid :PirateRevenge :RoadBack]
"The hero loses the pirates in the asteroid belt, eventually circling
back around to the strange readings and find an asteroid with exotic
minerals"]

[[:Reward :HeroEscapesPirates :DistressCall]
[:FakeDistressCall :PirateRevenge :RoadBack]
"The pirates had set the distress call as a trap! "]

;;A swarm of rogue AI drones :SwarmDeactivated :EscapeSwarm
[[:Reward :SwarmDeactivated :StrangeReadings]
[:DroneReadings :AIRevengeSwarm :RoadBack]
"With the swarm deactivated the hero finds that the readings are coming
from the drones themselves."]

[[:Reward :SwarmDeactivated :DistressCall ]
```

```
[:DroneDisabledShip :AIRevengeSwarm :RoadBack]
"With the swarm deactivated the hero follows the distress call and finds
a disabled ship floating through space that was attacked by the drones."]

[[:Reward :EscapeSwarm :StrangeReadings]
[:DroneReadings :AIRevengeSwarm :RoadBack]
"Once the hero escapes the drones, they find that the readings are coming
from the drones themselves."]

[[:Reward :EscapeSwarm :DistressCall]
[:DroneDisabledShip :AIRevengeSwarm :RoadBack]
"The hero manages to evade the drones and find their way to the source of
the distress call, a disabled ship floating through space."]

;;Showdown with the bounty hunter :DefeatedBountyHunter :BountyHunterShip]
[[:Reward :DefeatedBountyHunter :StrangeReadings]
[:ValuableAsteroid :NewBountyHunter :RoadBack]
"After defeating the bounty hunter the hero investigates the strange
readings which turned out to be an asteroid with exotic minerals."]

[[:Reward :DefeatedBountyHunter :DistressCall]
[:FuelDistressCall :NewBountyHunter :RoadBack]
"After defeating the bounty hunter the hero rushes to the distress call,
finding a ship that has run out of fuel!"]
```

```
[[:Reward :BountyHunterShip :StrangeReadings]
[:ValuableAsteroid :NewBountyHunter :RoadBack]
"Taking over the bounty hunter's ship the hero investigates the strange
readings which turned out to be an asteroid with exotic minerals. "]

[[:Reward :BountyHunterShip :DistressCall]
[:FuelDistressCall :NewBountyHunter :RoadBack]
"Taking over the bounty hunter's ship the hero rushes to the distress call,
finding a ship that has run out of fuel!"]

;;Act 2 can end with the following:



;; Pirate/Bounty Hunter Station resolutions :DeliverySuccess, :StayOnStation
;; AI station resolutions :StationEvacuated (PassengersOnShip / CargoOnShip)
;; Pirate / Bounty hunter belt attack :ValuableAsteroid, :FakeDistressCall,
;; :FuelDistressCall
;; Drone belt attack resolutions: :DroneReadings, :DroneDisabledShip

;; Act 3 The Road Back
;; Handling the mission success stuff. How do we bring an enemy back?
;; it should be the same enemy to be coherent.
```

```
;;We should add the pursuing enemy here too
;; :PirateRevenge , :NewBountyHunter , :AIRevengePirate ,
;; :AIRevengeSwarm , :PirateBetrayal

;;DeliverySuccess paths
[[:RoadBack :DeliverySuccess] [:DeliverySuccess :HeroFlyingHome]
"After succeeding in their delivery mission,
the hero begins their journey home."]

;;Evacuation comes with PassengersOnShip and CargoOnShip too,
;; to be resolved in resurrection
[[:RoadBack :StationEvacuated] [:StationEvacuated :HeroFlyingHome]
"After evacuating the station personnel onto the ship,
the hero begins their journey home."]

[[:RoadBack :FakeDistressCall ] [:HeroFlyingHome]
"Noting down the distress call as a fake the hero begins their journey home."]

[[:RoadBack :ValuableAsteroid ] [:CrystalCargo :HeroFlyingHome]
"The hero spends some time mining the asteroid for valuable crystals
before starting their journey home."]

[[:RoadBack :FuelDistressCall ] [:EscortingShip :HeroFlyingHome]
"The hero docks with the stranded ship to transfer fuel to them, before
```

```
starting to escort them back home."]

[[:RoadBack :DroneDisabledShip  ] [:EscortingShip :HeroFlyingHome]
"The hero docks with the disabled ship, finding the crew still alive. The
crew completes their repairs before starting to escort them back home."]

[[:RoadBack :DroneReadings] [:InfoOnAI :HeroFlyingHome ]
"Analyzing the readings from the drones provides some insights into their
workings, the hero decides it best to take the information home."]

;;Handle the potential final conflicts for act 3
[[:HeroFlyingHome :PirateRevenge] [:PirateAmbush :Resurrection]
"Unaware of it, the pirates from earlier have regrouped and set an ambush
for the hero!"]

[[:HeroFlyingHome :AIRevengePirate] [:AIPirateAttacks :Resurrection]
"Unaware of it, some of the allied pirates ships has been
taken over by the AI!"]

[[:HeroFlyingHome :AIRevengeSwarm] [:SwarmPursuit :Resurrection]
"Fragments of the station come to life after the explosion, chasing the
hero on their journey home!"]

[[:HeroFlyingHome :NewBountyHunter] [:BountyHunterPursuit :Resurrection]
```

```
"The bounty hunter managed to contact their friends, soon the hero finds
more bounty hunters on their tail!"]

[[:HeroFlyingHome :PirateBetrayal] [:PirateBetrayal :Resurrection]
"The Pirates decide to take their chances to get the hero's bounty
for themselves."]

;;Staying on station
[[:RoadBack :StayOnStation] [:StayOnStation :Resurrection]
"Staying on the station the hero spends some time helping the people
on the station rebuild."]



;; Act 3 Resurrection
;;One last big conflict! Big climax of the story!
;; This act starts with :PirateAmbush, :AIPirateAttacks, :SwarmPursuit,
;; :BountyHunterPursuit, :PirateBetrayal, :StayOnStation

;;Pirate ambush finish, what variants?
[[:Resurrection :PirateAmbush] [:ReturnElixir]
"The pirate fleet regroups and ambushes the hero right before they get back
home, who fights them off"]

[[:Resurrection :AIPirateAttacks] [:ReturnElixir]
```

```
"Some of the pirate's ships have been taken over by the AI and they attack
the hero who manages to disable the AI control."]

[[:Resurrection :SwarmPursuit] [:ReturnElixir]
"The hero lures the pursuing swarm into an ambush and comes out
victorious!"]

[[:Resurrection :BountyHunterPursuit] [:ReturnElixir]
"Eventually the bounty hunters catch up and force the hero into a pitched
battle! The hero manages to win."]

[[:Resurrection :PirateBetrayal ] [:ReturnElixir]
"The pirates decide to betray the hero and cash in on the hero's bounty,
forcing a battle which the hero manages to escape from."]

[[:Resurrection :StayOnStation] [:ReturnElixir]
"The hero spends their time repairing the station and organizing the
survivors of the AI attack."]

;; Act 3 Return with the Elixir
[[:ReturnElixir] [:end]
"The hero returns home after their long adventure and a completed mission."]

[[:ReturnElixir :NotAbandonFamily] [:end]
```

```
"The hero returns home after their long adventure and a completed mission,
happy to see their family again"]

[[:ReturnElixir :CrystalCargo] [:end]
"The hero returns home after their long adventure and sells the valuable
crystal cargo to repair the ship."]

[[:ReturnElixir :EscortingShip] [:end]
"The hero returns home with the ship they have been escorting, who are
grateful for the rescue."]

[[:ReturnElixir :InfoOnAI] [:end]
"The hero returns home with info on the artificial life form that
must be studied. "]

])
```

Bibliography

Ernest Adams. 2014. *Fundamentals of game design*. Pearson Education.

Tarn Adams and Zach Adams. 2006. Slaves to Armok: God of Blood Chapter II: Dwarf Fortress. (2006).

Frank Alpert. 2007. Entertainment software: Suddenly huge, little understood. *Asia Pacific Journal of Marketing and Logistics* 19, 1 (2007), 87–100.

Dave Anderson and Liz Cyr-Jones. 1986. Hollywood Hijinks.

Joseph Bates et al. 1994. The role of emotion in believable agents. *Commun. ACM* 37, 7 (1994), 122–125.

Bethesda. 2011. The elder scrolls V: Skyrim / Bethesda Game Studios.

Lucas Beyak and Jacques Carette. 2011. SAGA: A DSL for story management. *arXiv preprint arXiv:1109.0776* (2011).

Anne-Gwenn Bosser, Marc O Cavazza, and Ronan Champagnat. 2010. Linear logic for non-linear storytelling. In *19th European Conference on Artificial Intelligence*. IOS Press, 713–718.

Selmer Bringsjord and David Ferrucci. 1999. *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine.* Psychology Press.

Cyril Brom and Adam Abonyi. 2006. Petri-nets for game plot. In *Proceedings of AISB artificial intelligence and simulation behaviour convention, Bristol*, Vol. 3. 6–13.

Jerome Bruner and Jerome Seymour Bruner. 1990. *Acts of meaning: Four lectures on mind and culture*. Vol. 3. Harvard university press.

William E Byrd. 2009. *Relational programming in miniKanren: techniques, applications, and implementations.* Ph. D. Dissertation. Indiana University.

Joseph Campbell. 2008. *The hero with a thousand faces*. Vol. 17. New World Library.

Fred Charles, Miguel Lozano, Steven J Mead, Alicia Fornes Bisquerra, and MO Cavazza. 2003. Planning formalisms and authoring in interactive storytelling. Fraunhofer IRB Verlag.

Steven Cohan and Linda M Shires. 1988. *Telling stories: A theoretical analysis of narrative fiction.* Psychology Press.

Kim Dung Dang, Ronan Champagnat, and Michel Augeraud. 2013. A methodology to validate interactive storytelling scenarios in linear logic. In *Transactions on Edutainment X*. Springer, 53–82.

Kim Dung Dang, Steve Hoffmann, Ronan Champagnat, and Ulrike Spierling. 2011. How authors benefit from linear logic in the authoring process of interactive storyworlds. In *Interactive Storytelling: Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28–1 December, 2011. Proceedings 4*. Springer, 249–260.

Wendy Despain. 2020. *Professional techniques for video game writing*. CRC Press.

Belén Díaz-Agudo, Pablo Gervás, and Federico Peinado. 2004. A case based reasoning approach to story plot generation. In *European Conference on Case-Based Reasoning*. Springer, 142–156.

Damien Djaouti. 2013. Game design tools for collaboration. `https://www.gamedeveloper.com/business/game-design-tools-for-collaboration`

Joris Dormans. 2011. Simulating mechanics to study emergence in games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 7. 2–7.

Bethany Rachel Dunfield. 2018. *Exploration of Narrative Structure in Games for Story Creation*. Ph. D. Dissertation. Carleton University.

Ian Finley. 1999. Exhibition.

Gustav Freytag. 1894. *Freytag's Technique of the Drama*. Scott, Foresman.

Alan Garnham. 1983. What's wrong with story grammars. *Cognition* 15, 1-3 (1983), 145–154.

Pablo Gervás. 2013. Propp's Morphology of the Folk Tale as a Grammar for Generation. In *OASIcs-OpenAccess Series in Informatics*, Vol. 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Jean-Yves Girard. 1987. Linear logic. *Theoretical computer science* 50, 1 (1987), 1–101.

Daniel Gratzer. 2014. DSKanren Documentation. `https://hackage.haskell.org/package/ds-kanren-0.2.0.1/docs/Language-DSKanren.html`

Malte Helmert. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26 (2006), 191–246.

Johan Huizinga. 2016. *Homo ludens: A study of the play-element in culture*. Angelico Press, Ranchos de Taos, NM.

Randall Jarrell. 2002. *Randall Jarrell's Book of Stories*. New York Review of Books.

Chris Klimas. 2009. Twine is an open-source tool for telling interactive, nonlinear stories. `https://twinery.org/`

Ben Kybartas and Rafael Bidarra. 2017. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 3 (2017), 239–253.

Ben Kybartas and Clark Verbrugge. 2014. Analysis of ReGEN as a graph-rewriting system for quest generation. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 2 (2014), 228–242.

P David Lebling, Marc S Blank, and Timothy A Anderson. 1979. Zork: A computerized fantasy simulation game. *IEEE Computer* 12, 4 (1979), 51–59.

Young-Seol Lee and Sung-Bae Cho. 2012. Dynamic quest plot generation using Petri net planning. In *Proceedings of the Workshop at SIGGRAPH Asia*. 47–52.

Joseph P Magliano and Arthur C Graesser. 1991. A three-pronged method for studying inference generation in literary text. *Poetics* 20, 3 (1991), 193–232.

Araújo Manuel and Licinio Roque. 2009. Modeling Games with Petri Nets. In *DiGRA - Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*. Brunel University. http://www.digra.org/wp-content/uploads/digital-library/09287.37256.pdf

Chris Martens. 2015. *Programming interactive worlds with linear logic*. Ph. D. Dissertation. Carnegie Mellon University Pittsburgh, PA.

Chris Martens, Anne-Gwenn Bosser, Joao F Ferreira, and Marc Cavazza. 2013. Linear logic programming for narrative generation. In *Logic Programming and Nonmonotonic Reasoning: 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings 12*. Springer, 427–432.

Michael Mateas and Andrew Stern. 2003. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, Vol. 2. 4–8.

Maxis. 2008. Spore. *PC Game* (2008).

Maxis. 2014. The Sims 4. *PC Game* (2014).

Neil McIntyre and Mirella Lapata. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 1562–1572.

Nick Montfort. 2011. Toward a theory of interactive fiction. *IF theory reader* 25 (2011).

Warren F Motte. 1986. A Primer of Potential Literature. *University of Nebraska Press* (1986).

Tadao Murata. 1989. Petri nets: Properties, analysis and applications. *Proc. IEEE* 77, 4 (1989), 541–580.

Graham Nelson. 1995. *The craft of adventure.* Interactive Fiction Archive.

David Nolen and Rich Hickey. 2011. Clojure/core.logic. `https://github.com/clojure/core.logic`

TeongJoo Ong and John J Leggett. 2004. A genetic algorithm approach to interactive narrative generation. In *Proceedings of the fifteenth ACM conference on Hypertext and hypermedia*. 181–182.

Curtis Onuczko, Duane Szafron, and Jonathan Schaeffer. 2008. Stop getting side-tracked by side-quests. *AI Game Programming Wisdom* 4 (2008), 513–528.

Vladimir Propp. 2010. *Morphology of the Folktale.* Vol. 9. University of Texas Press.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

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

Mark O Riedl and R Michael Young. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing* 24, 3 (2006), 303–323.

Mark O Riedl and Robert Michael Young. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39 (2010), 217–268.

David L Roberts and Charles L Isbell. 2008. A survey and qualitative analysis of recent advances in drama management. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 4, 2 (2008), 61–75.

Jan Rochel. 2010. GraphRewriting Documentation. `https://hackage.haskell.org/package/graph-rewriting-0.7.10/docs/GraphRewriting.html`

Nicholas A Schudlo. 2014. *Development of an Emergent Narrative Generation Architecture for Videogames*. The University of Western Ontario (Canada).

Manu Sharma, Santiago Ontanón, Christina R Strong, Manish Mehta, and Ashwin Ram. 2007. Towards Player Preference Modeling for Drama Management in Interactive Stories.. In *FLAIRS Conference*. 571–576.

Miguel Sicart. 2008. Defining game mechanics. *Game studies* 8, 2 (2008), 1–14.

James Skorupski and Michael Mateas. 2010. Novice-friendly authoring of plan-based interactive storyboards. In *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Kevin Snow. 2015. Beneath Floes. Itch.IO.

Roger Sorolla. 2011. Crimes Against Mimesis. *IF Theory Reader* (2011).

Bruce Tate, Ian Dees, Frederic Daoud, and Jack Moffitt. 2014. *Seven more languages in seven weeks: Languages that are shaping the future*. Pragmatic Bookshelf.

Brandon Tearse, Peter Mawhorter, Michael Mateas, and Noah Wardrip-Fruin. 2013.

Skald: minstrel reconstructed. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 2 (2013), 156–165.

David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen. 2007. Interactive Storytelling: A Player Modelling Approach.. In *AIIDE*. 43–48.

Scott R Turner. 1993. *Minstrel: a computer model of creativity and storytelling.* University of California, Los Angeles.

Josep Valls-Vargas. 2013. Narrative extraction, processing and generation for interactive fiction and computer games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 9. 37–40.

Christopher Vogler. 2007. *The writer's journey.* Michael Wiese Productions Studio City, CA.

Nick Walton. 2019. AI Dungeon. `https://play.aidungeon.io/main/home`