MARLINSPIKE: AN INTERACTIVE DRAMA SYSTEM

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE UNIVERSITY OF HAWAI'I AT MĀNOA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMMUNICATIONS AND INFORMATION SCIENCE

AUGUST 2011

By

Zach Tomaszewski

Dissertation Committee:

Kim Binsted, Chairperson Scott Robertson Diane Nahl Andrew Arno John Zuern

Keywords: interactive narrative, interactive drama, scenes, reincorporation, drama manager, evaluation

ACKNOWLEDGMENTS

First of all, thank you to Stacy R. Judy for her love and support. She was the bystander most impacted by my work on this project, yet she remained steadfast in her role as my work-ethic "Jiminy Cricket". She was right: All those extra half-hours really do add up! However, I think I still owe her a few weekends and movie nights.

Thank you to all of my committee members for their enthusiasm and feedback. In particular, I'd like to thank my chair, Kim Binsted, for introducing me to interactive narrative in 2003 and for then sticking with me through the long slog from 2005 to 2011. Her suggestions and detailed comments throughout the process—particularly on last-minute chapter drafts—were also much appreciated. Thank you too to Scott Robertson, who provided both structure and motivation during the long final push on the *Demeter* implementation. His pointers on ANOVA statistics helped too.

Thank you to my alpha and beta testers. Besides those already mentioned above, this includes Brett Becker, Kathy Becker, Mom, and Dad.

Also, thank you to all of the participants who completed my study. All participants volunteered a substantial portion of their time, for which I am very grateful. I would not have been able to complete this dissertation without the data they contributed.

Finally, thank you to the Information and Computer Science department at the University of Hawai'i—Mānoa for providing me with rewarding employment as a teaching assistant for the duration of this project.

ABSTRACT

In an interactive drama, a human player assumes the role of a character in a story. Through this character, the player interacts with objects and other characters in a virtual story world. The interactive drama system then responds to the player by making changes in that world in order to produce a well-formed story shaped by the actions of the player. Thus, an interactive drama experience is much like that of a roleplaying computer game. The difference is that, rather than providing only an open world for the player to explore or else a fairly rigid preset storyline, the story is generated at runtime in response to the player.

Marlinspike is such an interactive drama system. Its design is based on a neo-Aristotelian poetics of interactive narrative developed from the work of Aristotle, Sam Smiley (1971), Brenda Laurel (1991), and Michael Mateas (2004). Marlinspike generates a story by responding to player actions using small preauthored story components called scenes. It selects scenes so as to narratively build upon or *reincorporate* earlier player actions into later story events. This reincorporation serves to make player actions narratively necessary to the finished story structure.

A prototype implementation of Marlinspike was used to produce the text-based game *Demeter: Blood in the Sky*. In an online evaluation with human players, the use of Marlinspike's reincorporation feature produced better-formed internal story structures that also included a greater percentage of players' actions as events within the finished story. However, the evaluation did not reveal a corresponding significant difference in the reported experiences of end users regarding story coherence, story-level agency, or general satisfaction.

Overall, the Marlinspike architecture was still successful in producing an interactive drama experience: a story-like experience generated at runtime in response to a human player acting within a virtual world. Coupled with the lessons learned from the implementation process, Marlinspike provides a solid foundation for future interactive drama development.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
ABSTRACT	iii
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF CODE LISTINGS	x
CHAPTER I. INTRODUCTION: INTERACTIVE NARRATIVE. Interactive Narrative Domains and Examples. Non-Digital Examples. Traditional Narratives. RPGs. CYOA and Gamebooks. Improv. Digital Examples. Computer games. Digital Media. Story Generation. Early Visions. Interactive Drama.	1 1 1 1 1 2 2 2 3 3 3 4 4 4 4
Challenges	7
	۵۵ ۱۵
The Structure of Narrative	
Evolution of the Current Poetics	10
Aristotle	10
Smiley	11
Laurei	13
Problems within the Current Poetics	
A Reconstructed Poetics	16
Medium	16
Object	18
Manner	20
Placement of User Interaction	22
Application to System Architectures	22
Parallels to Traditional Narratology	27
Conclusion: A New Poetics of Interactive Narrative	
	29 20
Answeight	29 20
ι ιεγιαγ	

Propp and Story Morphology	33
Chatman	36
Conclusion	37
Unity	37
Completeness	38
Morphology	38
Content	39
Exceptions	39
Conclusion	39
CHAPTER III. MARI INSPIKE ARCHITECTURE	41
Marlinspike Design	41
Story World: Characters and Objects	42
World-level Events: Verbs and Deeds	43
Story-Level Events: Actions	44
Story-Level Events: Scenes	46
Story Context: Triggers	49
Narrative Unity: Threads	51
Scene Selection Details	54
Marlinspike Foundations	56
Theoretical Foundations	56
Previous Work	58
Conclusion	
	60
Game Engine	00
Story World	00 62
Characters	02 62
Verbe	02 64
Actions	04 65
Scenes	03
Tutorial	00
Conclusion	73 74
CHAPTER V. IMPLEMENTATION: LESSONS FROM THE TRENCHES	
Inform as a Game Engine	
Consequences	76
Inform's Simplicity	77
Outcome	81
Drama Manager versus the World	82
State Management	82
Attempted Actions	83
Narration	84

Characters	85
PC	85
NPCs	85
Scenes	88
Reactions	89
Reincorporation as Scene Selection Criterion	90
Failure: Immediate Reactions	91
Failure: Advancing the Story	92
Weakness: Authorial Burden	93
Success: Explicit Story Structure	95
Resulting Stories	100
Stories as Branching Trees	100
Authorial Conception	102
Authorial Control	102
Resulting Stories	103
Conclusion	105
CHAPTER VI. EVALUATION	107
Claims to be Tested	107
Narrative	107
Interactivity	107
User Experience	108
Hypotheses	108
Experimental Setup	109
Reincorporation	109
Experimental Process	110
Pilot Study	111
CHAPTER VIL EVALUATION RESULTS	113
Participant Recruitment	113
Participant Backgrounds	
Demographics	
Computer Experience	
Digital Narrative Game Experience.	118
Non-digital Interactive Narrative Experience	121
Incomplete Participants	122
Game Data	122
Time Spent and Commands Entered	122
Verbs, Actions, and Scenes Used	123
Internal Story Structure	130
Player Agency	133
Participant Responses	135
Story Structure	136
User Agency	138
World-level Agency	138
Story-level Agency	140

Satisfaction	143
Open-ended Responses	143
Single most memorable or noteworthy event	144
Least Enjoyable Aspect of the Game	144
Most Enjoyable Aspect of the Game	145
Other Questions	146
Discussion	146
Hypotheses	146
Story Structure	147
Agency	148
Narrative	149
CHAPTER VIII. CONCLUSION	151
Contributions	151
Future Work	152
Theory	152
Marlinspike	153
APPENDIX A: EXAMPLE SCENE: PC_UNCONSCIOUS	156
APPENDIX B: REINCORPORATION IN DEMETER: BLOOD IN THE SKY	163
Scenes	163
Reactions	165
Casts and Recasts	166
Silent Hooks	166
Narration-only Hooks	166
APPENDIX C: PARTICIPANT RECRUITMENT EMAIL	167
APPENDIX D: EVALUATION SURVEY: BACKGROUND	169
Consent Form	169
Your Background	172
APPENDIX E: EVALUATION SURVEY: USER RESPONSE	178
Story Structure	178
User Agency	180
Satisfaction	183
Thank You!	184
APPENDIX G: OPEN-ENDED RESPONSES	185
WORKS CITED	200

LIST OF TABLES

Table 1. Oz system components in terms of the poetics levels	24
Table 2. Facade system components in terms of the poetics levels	25
Table 3. Virtual Storyteller system components in terms of the poetics levels	26
Table 4. Spierling et al.'s proposed architecture in terms of the poetics levels.	27
Table 5. An example subset of Propp's functions	34
Table 6. State variables of passenger NPCs in Demeter	63
Table 7. Verbs used in Demeter	65
Table 8. Actions used in Demeter	66
Table 9. Scenes and scene components in Demeter	68
Table 10. Reactions and one reaction component* in Demeter	72
Table 11. Participant gender distribution	.114
Table 12. Participant age distribution	.114
Table 13. Participant education distribution	.115
Table 14. Comfort using a computer	.116
Table 15. Average computer use	.116
Table 16. Comfort reading text on a screen	.117
Table 17. Familiarity with a command line interface	.117
Table 18. Current digital game experience in terms of time spent	.118
Table 19. Previous digital game experience in terms of total games played	.119
Table 20. Familiarity with digital game genres	.120
Table 21. Familiarity with non-digital game genres	.121
Table 22. Verb use	.124
Table 23. Action use	.126
Table 24. Scene Use	.128
Table 25. System measures of story length	.130
Table 26. System measures of story unity	.132
Table 27. System measure of world-level agency	.133
Table 28. System measures of player action import	.133
Table 29. System measures of story-level agency	.134
Table 30. Responses regarding story structure	.136
Table 31. Responses regarding irrelevant story events	.137
Table 32. Responses regarding positive world-level agency	.138
Table 33. Responses regarding negative world-level agency	.139
Table 34. Responses regarding positive story-level agency	.140
Table 35. Responses regarding negative story-level agency	.141
Table 36. Responses regarding story-level agency for a memorable event	.142
Table 37. Responses regarding game satisfaction	.143
Table 38. Most memorable or noteworthy event	.186
Table 39. Least enjoyable aspect of the game	.191
Table 40. Most enjoyable aspect of the game	.196

LIST OF FIGURES

Figure 1: Smiley's Model	12
Figure 2: Laurel's Model	13
Figure 3: Mateas's Model	14
Figure 4: Simplified model of an art object instantiated in a medium	16
Figure 5: Model demonstrating the formal and material causes	17
Figure 6: A reconstructed neo-Aristotelian poetics for interactive drama	19
Figure 7: A neo-Aristotelian poetics for interactive narrative	21
Figure 8: The Oz interactive drama architecture	23
Figure 9: Facade Interactive Drama Architecture	24
Figure 10: The Virtual Storyteller Architecture	25
Figure 11: Spierling et al.'s Architecture	26
Figure 12: Chatman's model of narrative	28
Figure 13: Chatman's model of narrative with terms translated	28
Figure 14: Freytag's Pyramidal Model	32
Figure 15: An overview of the Marlinspike architecture	42
Figure 16: Marlinspike starts with a virtual world of objects and characters	42
Figure 17: The player directs his player character in the world using verbs	44
Figure 18: Verb-based deeds are then cast into action-based events	45
Figure 19: The drama manager replies to actions by playing scenes	47
Figure 20: Necessity and reincorporation are two views of one relationship	51
Figure 21: Two kinds of necessity in Marlinspike	52
Figure 22: A simple thread demonstrating the two types of reincorporation	52
Figure 23: Thread map demonstrating a fork and a splice	53
Figure 24: The drama manager selects scenes so as to form threads	54
Figure 25: Partial poetics model of story	56
Figure 26: A screenshot of the Demeter game	61
Figure 27: Relationship of Demeter game content to Marlinspike	62
Figure 28: Story thread resulting from GoParty returning to gondola	94
Figure 29: Thread leading to GoParty_Rentry is left hanging	95
Figure 30: Hooking GoParty_Reentry back into the GoParty_Returns thread	.95
Figure 31: Game output resulting from pushing Count Vladescu	96

LIST OF CODE LISTINGS

Listing 1. Example pseudo-code definition of an INSULT action	.46
Listing 2: Example pseudo-code definition of a NPC_Is_Insulted scene	.48
Listing 3: Example pseudo-code definition of a recasting story state trigger	.50
Listing 4: Example of Inform code used to report that the members of the curre	ent
GoParty enter the room	.77
Listing 5: System view of two events: a Push deed followed by a	
Discussion_Starts scene	.97

CHAPTER I. INTRODUCTION: INTERACTIVE NARRATIVE

Interactive Narrative Domains and Examples

This dissertation concerns interactive drama, which is a kind of interactive narrative. An *interactive narrative* is essentially a story that changes in response to its reader. The field of interactive narrative can be found at the intersection of narrative art forms, games, and artificial intelligence, with a number of examples already existing. Yet interactive narrative is a relatively new endeavor; most true examples of it have existed for less than fifty years. The advent of the digital computer—a medium that excels at both processing and interactivity—has greatly increased the incidence of interactive narrative.

Non-Digital Examples

Traditional Narratives

Many traditional narrative formats—literature, film, theater—allow for some very basic interaction. For example, a reader can control how fast she turns the pages of a novel. A viewer can control the volume of a DVD. An audience member can, through her reactions, impact the actors presenting a drama. But real interactive narrative intends to offer the user more significant impact than this.

RPGs

One of the best examples of interactive narrative is *roleplaying games* (RPGs). These come in two general forms. In table-top roleplaying games—such as the classic *Dungeons and Dragons* (2003)—players sit around a table, roll dice, and describe the action of the resulting story to each other.

In live-action roleplaying games—such as White Wolf's *Mind's Eye Theatre* (Woodworth 2005)—players tend to be more active, to the point of acting-out at least parts of the action. A somewhat atypical example of live-action roleplaying is the dinner mystery game. In such a game, each guest is given a role to play, and, by acting in character, they solve a fictional murder mystery together by the end of the evening.

In either form, one player assumes the role of the *game master*, who acts as a kind of referee. The game master directs the story by describing the details of the

game world to the other players and then responding to what the others players choose to do as characters in that world. The flexibility of the story then depends on the creativity of the players and responding adaptability of the game master.

The game rules constrain the action somewhat by outlining possible game actions and their effects. Dice are often used to determine the likelihood of success for complex actions. Roleplaying systems with a lot of detailed rules tend to become more strategic in nature, like the war-gaming and strategic miniatures that preceded modern roleplaying games. Systems that are particularly light on rules become more free-form and dynamic, producing collaborative storytelling experiences more reminiscent of childhood *pretending* play.

CYOA and Gamebooks

Roleplaying games are live events, guided in part by a human director. On the other hand, the *Choose Your Own Adventure* (*CYOA*) series of books (1979-1998) is a good example of an interactive narrative that has been completely encoded into an artifact that is then later read by the user. In this series, each book offers a branching storyline. The reader reads a few pages and is then offered a choice to make. Each choice includes a page number that the reader then turns to if he wishes to make that choice. *CYOA* is the most well-known "brand" of this kind of story, though similarly-structured works exist by other publishers.

A variation on branching storybooks is *gamebooks*. The *Lone Wolf* (Dever 1984-1998) and *Way of the Tiger* (1985-1989) series are good examples of this format. In a gamebook, the reader picks a number of skills or equipment items for their character at the start of the book. Besides making choices as in a *CYOA*, certain options are only available if the reader has a certain skill or item. Gamebooks also tend to involve combat sessions where the reader rolls a die to determine the outcome. Combat success is also impacted by the character's skills and current health. In this way, gamebooks are a bit more like table-top roleplaying games as initial character creation and chance have an impact on the story possibilities.

Gamebooks tend to be published as a series. Because of this, each book in the series tends to have only one successful ending—unlike a *CYOA* book in which multiple successful conclusions are usually possible.

Improv

Improv theater also offers an example of interactive narrative. When improv actors are performing together for their own entertainment, each exerts an

influence—but not directorial control—over the outcome of the story. Improv actors may also elicit suggestions from an external audience, thereby granting the audience a chance to influence the direction of the story.

Digital Examples

Computer games

Computer games include a number of narrative forms. For instance, *interactive fiction* games—though frequently puzzle-based—can still offer a simple story structure. Because it is text-based, interactive fiction supports experiments with language and literary form.

A successor to early interactive fiction was the *adventure game*, as represented by Sierra's *King's Quest* (Williams 1984-1992) and *Space Quest* (Crowe & Murphy 1986-1991) series. These graphical games also presented a linear story revealed through a series of puzzles to solve. Occasionally, these games would provide multiple possible endings—such as in Dynamix's *Rise of the Dragon* (1990).

Adventure games eventually evolved into *computer roleplaying games*, which have a greater focus on character abilities, controlling parties of characters, and combat over puzzles. These games have also tended to be more free-form, providing a large simulated world that can be explored. When the game includes a storyline or quests, they can be followed in various orders or completely ignored by the player. Good examples of this are the *Elder Scrolls* series (1994-2006) and *Grand Theft Auto III* (2001).

First-person shooter games—such as *Doom* (1993), *Quake* (1996), and *Half-Life* (1998)—and their third-person variants—such as *Tomb Raider* (1996)—offer detailed control of a character in a fictional world. However, despite a basic context-providing storyline, there are rarely significant story choices to be made during gameplay itself.

With the introduction of networked gaming and shared virtual spaces, *massively multiplayer online roleplaying games* (MMORPGs), such as *World of Warcraft* (2004), have introduced a new genre. Here, though presenting a massive simulated world and a number of possible quests, some of the more interesting narratives emerge from interactions between human players and the formation of communities within the virtual space.

Finally, some users claim that detailed *simulation games* such as *The Sims* (Wright 2000) can be attributed with emergent narrative properties. Although *The*

Sims is not intentionally designed to generate stories, story-like events occur as the autonomous characters follow their various inclinations.

Digital Media

Along with the multimedia capabilities of digital media comes the potential for interactivity. Yet many interactive digital works are so story-focused that they do not seem to quite qualify as games. For example, Romp.com's adult-themed series *Booty Call* (Metter & McManus 1999) was a graphical version of a *CYOA* book. Implemented in Flash, each episode presented a number of animated segments. At the end of each segment, the viewer could select which choice the main character, Jake, should pursue. From a single starting point, each play through an episode revealed a single path through a branching tree of possibilities.

Hypertext fiction also presents a kind of interactive narrative. These works can be completely textual or include various multimedia. In some hypertext works, the path the reader takes though a series of successive choices determine the underlying story—just as in a *CYOA* book. As an example of even greater interactivity, some sites allow readers to become writers, extending storylines when they reach the end of a branch. In other hypertext works, the underlying story is static. The links the reader follows simply changes the order in which a single story is revealed.

Hypertext fiction is an example of a larger *digital literature* movement, which uses the processing, multimedia, and interactivity capabilities of the digital medium to produce novel literary works. While digital literature nearly always includes some textual component, not all examples are narratives nor are all interactive.

Story Generation

While not interactive, *story generation* projects also inform interactive narrative. Both endeavors fall under the larger field of *narrative intelligence*—building artificial intelligence systems that can either understand or generate new stories. However, an interactive user fundamentally changes the problem. Many story generation systems build complete stories at once in a way that does not tolerate new inputs partway through the generation process.

Early Visions

Computers have an incredible potential to change our experience of narrative. Brenda Laurel and Janet Murray are two of the earliest authors explore this potential in detail, describing their visions of how computers and narrative might intersect. Their work has had a significant impact on later interactive narrative work.

Brenda Laurel (1991) suggests using narrative as a model for designing humancomputer interaction (HCI). She characterizes traditional software design as providing an interface metaphor to an application's underlying functionality. However, such interface metaphors often fall short in that they provide a quick initial view of the application that can then fail to explain or accurately characterize all of the application's functionality.

Laurel argues that designers should instead conceptualize applications as providing a virtual theater-like space. This space is occupied by both the virtual agents of the application and the manipulatable objects of the application. These objects should afford direct manipulation by the user. That is, an object's representation within the application should be continuously available. It should support physical actions instead of requiring an arcane command syntax, and those actions should be reversible. Finally, the object should provide immediate feedback about its current state. Application designers can then focus on crafting the "action" that occurs in this virtual application space, with special consideration for the will of the user as an actor.

While this narrative-based approach may not have gained much sway in the broader HCI field, Laurel's described architecture is extremely relevant to an interactive narrative application that actually intends to provide a narrative experience to an interacting user.

One of the most useful insights that I took from Laurel is abandoning the idea of an audience or a passive user. Instead, in an interactive narrative, the audience is up on the stage, interacting with the actors. Furthermore, the audience is not doing this in their street-clothes (figuratively speaking); instead, they are seeking to immerse themselves the story, to garb themselves as characters in that story.

If the players become actors, then we must also abandon the idea of providing a pre-constructed story for them to interact with. Instead, the author provides a setting, rules of interaction, and creative agent-actors around the player-actor, and then starts the action. Thus, the player-as-actor becomes as instrumental in constructing the finished story as any other agent controlled by the system. This means we need to focus on computational approaches to generating the narrative at run-time. Also, since the player-as-actor is unlikely to be skilled at acting, the other agent-actors will need to compensate, much as improv actors might act around an audience member called up to participate in their skit. The pioneers in interactive drama—Joseph Bates (1992) and the Oz Project—began exactly this way with live performance experiments.

Laurel goes on to describe the structure of narrative, building upon Aristotle's *Poetics*, which I will explore further in Chapter II. She also explores the nature of interactivity, describing it as a continuum along three axes: frequency (how often does the player get to interact?), range (how many choices are available?), and significance (do the player's choices matter? Do they really affect the outcome?). In short, as she characterizes it later, the goal of true interactivity is to make the player feel like she is an active participant when using the system.

Six years later, Janet Murray (1997) explored the effect of interactivity and computation on narrative experience. Murray's focus is less on design and more on the resulting user experience. Her central question is whether technology will extend and heighten narrative—as *Star Trek*'s imaginary holodeck extends the literary worlds of Sherlock Holmes, Jane Eyre, and Hamlet—or whether it will produce an addictive mind-numbing virtual reality—as portrayed by such as authors as Plato, Aldous Huxley, Ray Bradbury, and William Gibson.

Murray argues that digital environments possess four essential properties. They are procedural in that they are generated and governed by algorithms and rules. They are participatory in that they are intended for a human interactor. They are spatial both in that they provide a virtual space to explore and that they are usually presented using representations of at least two dimensions. Finally, they are encyclopedic in that they possess great capacity for information, especially when the digital environment supports expansion by users.

Murray then explores some of the new narrative experiences made possible by technology so far. Stories are more likely to be multiform. That is, if the narrative changes due to the user's participation, there is not only one authoritative version of a tale anymore. This can be foreign to us nowadays, but multiform stories are reminiscent of the story variations of the bardic oral tradition. Technology has allowed for more immersive narrative experience, from the simply visual IMAX and 3D movies to ride-the-movies amusement rides in which the audience physically explores a familiar narrative world in a different context. Online, fans participate in stories through the extra encyclopedic information available through some TV show websites. The potential is there for fans participating in online narrative games related to a running TV show to then indirectly affect the plot of the show. Computer games are often narrative-based, although they are still exploring and establishing their own unique media conventions. Work continues on believable autonomous characters and conversational agents such as Eliza.

Informed by some of these different developments, Murray describes three aesthetics of an interactive narrative medium. The first is immersion, which is the feeling of being transported to an elaborately simulated place and surrounded by another reality. This feeling of presence in the world of the narrative dominates our attention such that we forget our mundane reality for a while. Agency is the ability to affect real change in the narrative. This goes beyond mere interactivity, such as typing a command or twitching a joystick. The last aesthetic is transformation, both of the system and of the player. The underlying narrative may change in response to the user. Multiple viewpoints may be possible of the same underlying events, changing the experience of those events. Users may assume different roles within the narrative. Finally, the resulting experience may have a significant effect on the player.

Together, Laurel and Murray provide a vision of the positive potential that both technology and interactivity can have on narrative. In essence, through the active participation of the end user, it is possible to fundamentally change the experience of narrative. Although interactive and multiform narratives have existed in the past, new technologies can provide new forms of narrative experience as artifacts to be experienced separately from their original human author.

Interactive Drama

While many interactive narrative forms are possible, this dissertation is most concerned with a single type: the interactive drama.

In an *interactive drama*, a human player assumes the role of a single character in a story. Through this character, the player then interacts with objects and other system-controlled characters in a virtual story world—much like an actor on a stage. The interactive drama system then responds to the player in order to produce a story. It does this primarily by directing the behavior of the system-controlled characters. The resulting story should be well-formed and it should be significantly affected by the actions of the player.

Thus, an interactive drama experience is much like that of a modern roleplaying computer game. The difference is that, rather than providing only an open world for the player to explore or else a fairly rigid preset storyline, the events of the story can change significantly in response to the player. To achieve this, an interactive drama system generates the story procedurally at runtime.

Interactive drama also goes by many other names, such as interactive story, digital storytelling, cyberdrama, or ractive.

Challenges

A successful interactive drama system must first resolve a number of conceptual tensions. The first tension is between authorial and player control. If the author has control of the story, then the player will likely to restricted to making only minor or preset choices allowed by the author. But if story is driven solely by the player, then there is no way to ensure that the resulting outcome will even be story-like, let alone a good story.

Related to this is world simulation versus a plotted story. In a world simulation, the world supports many diverse actions. The system's character agents can independently follow their own programmed inclinations; it is easier to produce consistent, believable characters this way. However, in a simulated world, there is no resulting story. Or, if there is a story, it is *emergent* from the rules of the world—how each character and object works and how those behaviors then interact.

On the other hand, a plotted story requires specific player actions to advance. Where it offers choices, those choices are usually between a handful of explicit story paths. If character agents are autonomous, they must be subservient the demands of the story. Thus, the story can be said to be *directed*.

Aside from issues of control is the issue of knowledge and artificial intelligence. Computers are ignorant. They must be provided with explicit knowledge about how the world works. They must also be instructed on all the subtleties of human psychology in order to produce a believable character. They need to know the rules of natural language if they are to generate character speech or textual event narration at runtime. And they must be told about narrative structure and what makes for a good story over a bad story.

Frankly, this general knowledge challenge is insurmountable at this time. However, it remains an intriguing artificial intelligence problem. All humans share narratives with each other, usually starting at a very early age.¹ Many of the TV shows or movies we watch seem to follow fairly simple, predictable narrative rules. Yet, on closer examination, we find that narrative production involves some incredibly complex and intuitive processes.

So the challenge is to begin chipping away at this complexity, breaking the problem into tiny, solvable chunks. Much of this involves limiting the knowledge required to only the specific context of the current interactive drama. That is, the system doesn't need to know how the world, characters, or all stories work in general, but only what actions these particular virtual objects support, what internal states these certain agents have, and the rules by which to combine specific small pre-authored components to form a workable story.

Motivations

There are a number of reasons to create a true digital interactive drama system.

¹ Whether most of these early narratives are *good* narratives is another question. Most of the stories we read and films we watch are produced by skilled and/or highly-paid professionals. This level of expectation certainly raises the bar for any story-generating AI. Mimicking the general story-generating skills of a five-year-old would be impressive from an AI standpoint, but very few people would likely value the results on par with literature or film.

The first reason is that doing so requires a degree of automated narrative intelligence. Humans tell stories easily and nearly constantly. And yet—partly due to the large amount of real-world knowledge required—getting a computer to understand or generate stories proves incredibly difficult. Such narrative intelligence currently remains a challenging problem for the artificial intelligence community.

Secondly, interactive dramas are potentially very lucrative. Computer games are already a multi-billion dollar industry. Yet most of the games that purport to be story-based are too linear or restrictive, or else the story serves only as a context for combat or puzzle-solving. If games really were capable of adaptive story production—particularly if also including believable characters and complex social interactions—it could lead to a whole new market of interest to people who don't currently play computer games.

Aside from entertainment, there are serious or practical applications, such as a teaching or training tool. Currently, virtual training simulations can be used to train people to handle or explore complicated situations. For example, software already exists that is used to train soldiers how to interact with civilians in order to achieve certain mission goals. If such software could be extended to produce well-formed stories, it may make such learning experiences more memorable.

But the most important reason to pursue interactive drama is that it would be a new form of human expression. Traditional narratives let us explore the Other. While we might empathize greatly with a character in a film or novel, that character makes his own choices and we, as audience, simply follow along, slightly removed. Interactive dramas, on the other hand, would allow exploration of the Self. Whether following our own inclinations or those of a character role we assume, we would be able to make our *own* choices in the story. We would then be able to explore the consequences of those choices as the story unfolds.

Exactly what those consequences might be depends on the author of the interactive drama. Instead of writing single—or even branching—storylines, authors of interactive dramas will write the rules that produce a number of possible stories. This is will be a challenging creative form—writing a world or system of potentials, rather than a concrete tale.

So interactive drama proves to be an interesting AI challenge, potentially both lucrative and educational. Yet most importantly, it may provide a new experience for users and a new creative form for authors.

CHAPTER II. THEORY: A POETICS OF INTERACTIVE NARRATIVE

The Structure of Narrative

Before we can make a narrative interactive, we must first understand what a narrative is. Furthermore, our model must include a sense of how interaction affects that narrative.

Within the interactive drama community, such a model already exists. Aristotle's *Poetics* has served as the basis for understanding interactive drama since Brenda Laurel proposed her neo-Aristotelian model in 1991. Laurel's model is based largely on the work of Sam Smiley (1971). Michael Mateas (2004) has recently extended Laurel's model to include an explanation of how user interaction fits into the model.

Evolution of the Current Poetics

Mateas's poetics has accumulated a number of valuable additions during its development through four authors. However, the obscuring of key Aristotelian features—such as the distinction between object, manner, and medium—has lead to certain tensions. Here, I will trace the evolution of the current poetics in order to examine its strengths. I will then propose an overhauled model that includes most of these benefits while eliminating much of the internal strain.

Aristotle

In the fourth century B.C.E., Aristotle laid the foundations of narrative theory in his *Poetics*. Although he focuses primarily on describing the nature of tragic drama, he does refer to other art forms such as epic poetry, comedy, dithyrambic poetry, music, dancing, and painting. He claims that all these forms of "imitation" differ from each other in three defining respects: their objects, medium, and manner (which is sometimes called mode).

Objects. The object "imitated" in drama is "men in action" (Aristotle 1961, Chapter I). Tragedy and comedy can be distinguished by the character of the men represented and the nature of the action. The men can be portrayed as "better" or "worse" than they are in real life; the action may or may not be serious, unified, and complete. But the important aspect is "men in action". Fergusson, in his introduction (Aristotle 1961), explains that our understanding of Aristotle's *action* should be in light of his writings on ethics. Action here means *praxis*—an active, rational "movement of spirit", directed outwards. It is action arising from thought, focused to some end. The motivation of a character is essential to this sort of action.

Aristotle thus explains that the three objects of dramatic action are Plot (that is, the "arrangement of the incidents"), and the Character and Thought of its agents (Chapter VI).

Medium. Art can represent objects through a variety of different media—color and form, or the voice, or rhythm and harmony (Chapter I). Tragedy, specifically, is conveyed through Diction and Song (Chapter VI). That is, actors speak and sing in order to convey the action to the spectators.

Manner. Within the same medium, there may be different manners of presentation. For instance, in poetry conveyed through the media of speech and song, the events can either be narrated through a personality, narrated by the poet himself, or enacted as if the characters were "living and moving before us" (Chapter III). This is a distinction between epic and tragedy—epic is narrated, while tragedy is enacted. Aristotle calls this enactment of tragedy the Spectacle (Chapter VI).

Aristotle lists these six parts in terms of their order of importance to tragedy: Plot, Character, Thought, Diction, Song, Spectacle (Chapter VI). The first three are the objects—we understand the Plot of the action in part because we understand the Character and Thought of the characters. The action is presented through the media of Diction and Song, Diction being the more important for Aristotle. He lists the manner of Spectacle as the least essential to judging tragedy. Though the special effects of the stage may have a certain emotional appeal, they are the least connected with the art of poetry: "For the power of tragedy, we may be sure, is felt even apart from representation and actors" (Chapter VI).

Smiley

In *Playwriting: The Structure of Action*, Sam Smiley (1971) explores the process of playwriting using Aristotle's model as a framework. In the first chapter, in arguing that fine arts are artificial (that is, manufactured) objects, he explores Aristotle's four causes for coming into being of an artificial product.

For those unfamiliar with Aristotle's causes, the material cause of an object is the substance of its construction. The material cause of a house is the wood and concrete used to construct it. The formal cause is the form of the object. For a house, this would correspond roughly to its blueprint design. The efficient cause is the process that constructs the object. This would be the construction workers

who build the house. The final cause is the end to which the object is constructed. A house is usually constructed to provide shelter.

Smiley very briefly presents Aristotle's six parts of drama as connected by formal and material causes. (Although the four causes are an Aristotelian concept, Aristotle himself does not state such causes between the six parts in *Poetics*.) Smiley presents them in the same order as Aristotle—Plot, Character, Thought, Diction, Sounds, and Spectacle—and contends that each element dictates the form of those below, while each provides the material for the element above.



Figure 1: Smiley's Model (1971, p.11)

For Smiley, Plot is constructed in terms of the actions of the Characters. The material from which we build Character is Thought.² Thought is itself constructed from words, or Diction. Diction is made up of Sounds. (Note the change here from Aristotle's Song.) Spectacle—"the physical actions that accompany the words" (Smiley 1971, p.12)—is the most basic material of all.

A playwright holds the formal order to be most important. A certain Plot dictates the qualities required of the Characters. Based on these qualities, the characters will espouse certain Thoughts, expressed in certain Diction, and so on. In contrast to the playwright, the actors and production team tend to construct the play working in the material order, beginning with the Spectacle.

Note that Aristotle's distinction between object, medium, and manner has been ignored here. We will soon discover that a number of tensions were introduced by this omission, as these few paragraphs of Smiley's become the foundation of the modern poetics of interactive drama.

² For Smiley, Thought includes all the internal experiences of a character—emotions, qualities, and ideas. This is more inclusive than Aristotle's notion that only rational thought provides adequate motivation for a character's actions.

Laurel

Brenda Laurel (1991) begins with Smiley's model, renaming some of the elements to be Action, Character, Thought, Language, Melody, and Spectacle. She describes these elements first in terms of drama, and then expands their meanings to be suitable for all human-computer activities (such as computer-based interactive narratives).

Starting at the lowest levels, Laurel begins by defining Spectacle as "everything that is seen" and Melody as "everything that is heard". However, this does not fit cleanly into the causal hierarchy since Spectacle does not form the basis of Melody. Also, this emerging neo-Aristotelian model does not seem to allow for visual signals to travel "up" the hierarchy to become the basis of Language and the understanding of the drama.

So Laurel renames Spectacle to Enactment and redefines it to mean all the sensory dimensions of the represented action—visual, auditory, tactile, and any others. From these sensations, the user constructs Patterns. Language now does not mean only spoken human language, but any "selection and arrangement of signs, including verbal, visual, auditory, and other nonverbal phenomena when used semiotically" (Laurel 1991, p.50). Thought and Character remain largely unchanged, though for Laurel they may arise from computer-based, rather than solely human, origins.



Figure 2: Laurel's Model (1991, p. 51)

Although Laurel has overhauled the bottom half of the hierarchy in an attempt to fit the demands of the causal connections, we shall see that a number of inadequacies still remain.

Mateas

Michael Mateas (2004) follows Laurel's model, both in the terms used and the material and formal causes. He adds to the model Janet Murray's (1997) notion of *agency*, which Mateas defines as "the feeling of empowerment that comes from being able to take actions in the world whose effects relate to the player's intention" (Mateas 2004, p.21).



Figure 3: Mateas's Model (2004, p. 24)

In an interactive drama, the story is enacted with the player taking the role of one of the characters. To support interaction at this Character level, Mateas adds two new causal chains—a Material for Action and a User Intention. When a user is interacting within a virtual world, the objects and the characters in that world afford certain user actions (from below). In turn, the story provides some narrative constraints, or at least direction (from above). When the user acts upon other characters in the story, her intention becomes a formal cause in much the same way the requirements of the action shape characters in traditional narratives. "A player will experience agency when there is a balance between the material and formal constraints" (Mateas 2004, p.25)

Problems within the Current Poetics

We can thus see that Aristotle's model has come a long way through these additions and reformulations. However, due to the introduction of material and formal causes, a number of omissions and tensions have been introduced.

First of all, we have lost Aristotle's sense of manner. Rather than differentiating between whether a narrative is enacted or presented, Spectacle has come to mean "all that is experienced by the audience."

Secondly, we have lost the idea that the medium is variable, yet still specific. When defining medium, Aristotle admits that medium may be color, harmony, rhythm, etc. (Chapter I). Only in describing tragedy (and other drama such as epic) does he limit himself to Diction and Song. Building on this, Smiley implies that dramas are presented through Diction and Sound, which are primarily auditory channels.

However, in arguing that dramas can be computer-based, Laurel attempted to regain the flexibility of different possible media. In order to allow for visual signs, Sound and Diction was rather ungracefully expanded to Pattern and Language. We now speak only generally of Patterns, rather than specifically of medium-specific modes. Patterns must then be assembled into something as well-defined as a Language in order to serve as the basis of Thought, Character, and Action.

Most importantly, the causal hierarchy implies sequential and exclusive links between the levels. That is, it seems *only* the level directly below should form the basis for the level above. For example, we certainly construct our understanding of a Character in terms of her Thoughts, which are understood in terms of her spoken Language. However, her physical features, expressions, gestures, costume, and theme music also contribute to our understanding of who a character is. Yet these attributes seem to serve as the material for Character without conveying Thought or using Language.

Mateas runs into this problem when he describes interaction with objects as existing "somewhere between spectacle and pattern" (2004, p.25). Yet what affordances are granted by the raw sensory experiences of Spectacle? What sort of constraints are provided by Patterns such as a purple jacket and an ominous musical chord? Yet it does not seem right to move objects to the level of Character in this model, as objects are not assembled from Language-encoded Thought.

Aristotle does not mention setting or props, probably due to the fact that plays of his time had limited scenery. However, objects in the world play an increasingly important aspect of computer-based interactive drama since they are often the means through which the player can affect the action.

A Reconstructed Poetics

Aristotle provides us the basis for describing practically any art form in terms of its object(s), medium, and manner, while exploring tragedy as a specific example. Smiley gives us the idea of formal and material causes between these elements. Laurel explores this process that Smiley only sketches, and expands this framework to describe computer-based drama. Mateas tackles the problem of how interaction and the experience of agency can fit into this model. It is possible to keep all of these contributions yet still eliminate many of the tensions introduced during this model's evolution.

Returning to Aristotle, we can say that an *object* of "imitation" is presented through some *medium*. We can think of this medium as our experience of a "text", whether this be reading a script, watching a movie, or playing a narrative game. The object of imitation does not formally dictate the choice of a certain medium as a whole—a story could be presented as either a novel or as a play. However, the object does formally dictate its construction within a specific, chosen medium—a story's dialog is written within quotes in a novel or spoken by the actors in a play. As an audience, we construct a sense of the object from our experience of its instantiation in a particular medium.



Figure 4: Simplified model of an art object instantiated in a particular medium.

Medium

Our experience of the medium can be described at different levels of detail. At its most basic, our experience of a medium may utilize a number of *sensory channels*—the visual, auditory, tactile, etc. This raw sensory experience corresponds to Laurel's (and Mateas's) definition of Enactment.

At a higher level, as Laurel suggests, we discern patterns based on this sensory experience. From various sounds, we may differentiate music or speech. From our visual experience, we may differentiate text, diagrams, photographs,

animation, or live action. We might call these differentiated sensory patterns the *modalities*³ of the medium. They are essentially what Robert Stam refers to as "tracks". As Stam describes it: "The novel has a single material of expression, the written word, whereas the film has at least five tracks: moving photographic image, phonetic sound, music, noises, and written materials" (Stam 2000, p. 59).

We may also want to consider that, also as Laurel describes, there are certain *conventions* (something like a proto-language) that develop for these different modalities. For instance, a shot-reverse-shot with a fade can signify a reminiscent flashback in film. Comic books use different "word balloon" conventions to show whether a character is speaking, whispering, or thinking.

The specific sensations, modalities, and conventions depend on the particular medium used. Each level provides the material necessary for constructing those above it, while formally constraining those below it. This is a different, broader notion of medium than Aristotle's, which would correspond only to what I am calling the modality.



Figure 5: Model demonstrating the formal and material causes at work within any particular medium.

This reformulation of medium opens the way for a media-specific analysis of works, as called for by N. Katherine Hayles (2002). Whether a story is conveyed as a live performance, a film, or a novel, we should be able to explore the particular details of its material embodiment in a medium and how that specific embodiment affects our conception of the work as a whole.

³ Although Aristotle's manner is sometimes translated as *mode*, that is not what I mean here. Modalities, as defined here, essentially correspond to such parts as Aristotle described as comprising the medium—spoken language, musical rhythm, color and form.

In an interactive medium, the medium also provides interface controls that affect the imitated objects.⁴ When experiencing a drama, the users move through the stages of material causality: from their sensory experience, they discern separate modalities, each with their various conventions for relaying the objects of drama. When supporting user inputs, the drama system must make this same transition. The system may allow for various channels for input, such as haptic or auditory. Haptic input might involve different modalities, such as movement of the mouse or the pressing of keyboard keys. Mouse use has a number of conventions concerning the difference between left-clicking, right-clicking, and doubleclicking.

When discussing interaction, we are mostly concerned with the user's agency within the narrative (object) context—how the user can affect the world of the story. However, it is useful to remember that both the user's understanding of the story world and her attempts to control it must pass through the medium. Thus, the medium serves as an interface to the object. It presents a "system image" to the user by which the user makes sense of the represented object (Norman 2002).

Object

The object of drama is "characters-in-action". This *action*⁵ has *characters* as its material cause; in turn, the action determines what sort of characters are needed to produce it. As held by Aristotle, a character's motivation, or *thought*⁶, is essential to understanding that character. However, as discussed earlier, it is not the only material from which characters are formed. They have a number of other physical attributes, and often thought can only be inferred by the audience from these outward appearances. While essential to character, thought does not cleanly fit within the exclusive formal/material cause hierarchy.

The notion of *setting* is missing from Aristotle. Yet, a story's action is partly constructed in terms of where things happen and what objects are used. This is

- 4 Object-affecting interface controls are only present in an interactive system that internally models the objects of the story—such as the characters and events—and then intends for the user to affect those object representations. As we will see in the complete reconstructed poetics, some systems may intend only that the user interact with the manner (discourse) of the narrative—such as changing camera angles or reading the events of a hypertext story in a user-selected order. But, regardless of which narrative level is being meaningfully affected by the interactions, all such user interactions must pass through the controls provided by the medium (and its components levels).
- 5 I will use the term *action*, rather than Aristotle's *plot*, to refer the events of the story world. This is to avoid confusion with the term *plot* as used by narratologists, which refers instead to the discourse or telling of the events, rather than the events themselves. *Action* here is essentially narratology's concept of *story*.
- 6 In this reformulation of the poetics, I mean *thought* more as Smiley did—including any mood, emotion, idea, or internal state of a character—rather than only as Aristotle's meaning of rational intention.

particularly true in an interactive drama, in which the user assumes the role of a character and, through this character, interacts in a virtual story world. Although some of this interaction means affecting other characters, the user often spends time manipulating objects and his character's current location. We might refer to the characters and setting together as the *story-world*.

The user's actions at the story-world level serve as partial material for furthering the action, while the narrative context of the action so far provides some constraints on the user. This is just as Mateas describes *agency*, though in this reformulated model, the world's objects are placed within the same narrative context as characters. Like characters, modelled story objects often have an internal *state* that must be inferred by the user from the objects' outward appearances (as conveyed through the medium). Though we are usually most concerned with the affordances for interaction offered by the story-world, it is helpful to remember that the medium itself must also successfully afford the interaction controls needed to affect those virtual objects.



Figure 6: A reconstructed neo-Aristotelian poetics for interactive drama.

This concludes my restructuring of the current poetics of Laurel and Mateas. Although I have changed some of the relationships and labels, I have tried to maintain their basic concepts—particularly the formal and material causes, the description of "patterns" and "languages" at work in a medium, and the mechanism of user agency. Most importantly, I have reinstated the Aristotelian distinction between medium and object.

Manner

So far, I have limited myself to those elements that have been carried through the evolution of this poetics to Mateas. However, the notion of manner was dropped relatively early in this development. Since the focus has been on describing interactive *drama*, this has been easy to ignore. We can simply assume that, as the player is assuming the role of a character, the manner is one of enactment.

Aristotle's definition of manner is very brief:

For the medium being the same, and the objects the same, the poet may imitate by narration—in which case he can either take another personality as Homer does, or speak in his own person, unchanged —or he may present all his characters as living and moving before us (Aristotle 1961, Chapter III).

We can see here that the manner is how the story is presented, but it is independent of both the medium and the object. The main distinction Aristotle makes is between narrated and enacted manners. For example, the play *Romeo and Juliet* has an enacted manner: it presents the characters "as living and moving before us". However, this "enacted" play can still be embodied in different media: as a script, as a live performance, as a film.

If the same *Romeo and Juliet* story was instead written as a novel told from Juliet's nurse's point of view, then the story would have a narrated manner. This novel could be made into a film, as film can present both enacted and narrated manners (as explored in more detail below). Thus, the same object (the story—including events, characters, and settings—of Romeo and Juliet's ill-fated romance) and the same medium (film) can have a different manner (enacted or narrated).

Aristotle also distinguishes between two kinds of narration—an omniscient narration verses a limited, character-filtered narration. In either form, we find our experience of the action is provided only through the particular point of view of the narrator. This narrator may be unreliable; the filtering character may be fallible. The narration may be very overt, in which the narrator constantly evaluates or comments on the events and characters of the story.

Narration is not itself an event in the story it conveys, even when the narrator is a character in the story. This is essentially the difference narratologists make between the *story*—the chronological events of the action—and the *discourse*—the telling or presentation of those events (Chatman 1978). The discourse conveys the story to us, but it may comment on that story, focus our attention on certain events over others, omit events, foreshadow or flashback to previous events, provide "backstory" information, etc. There is usually a difference between the discourse and story timeframes—it might take an hour to read about

the events that happen to the characters in seconds, or a ninety minute film might show story events that occur years apart.

Although this difference between the story and the discourse is less obvious for enacted narratives, the distinction can still be made (Chatman 1990). For example, film can present things from a particular point of view. A narrator can be established through such conventions as using voice-overs, point-of-view shots, and having the narrating character present in all scenes. Two different directors can tell the same story in different manners using the same film medium depending on how they use scene cuts, pacing, staging, lighting, etc in order to reveal, highlight, or comment upon the action. Although easily overlooked, the discourse—how the story is presented—is clearly important. And it is, in fact, present in all forms of narrative. Mark Stephen Meadows (2003) goes so far as to argue that the perspective from which the events are relayed is even more essential to narrative than the events themselves.

	<u>Object</u>			
\wedge	Action			
	Character Setting	3		
ances)		ate	Forma	
Afford	<u>Manner</u>		I Cau	
nse (/	Discourse		se (C	
	<u>Medium</u>		onstr	
ateri	Medium conventions		aints)	
Σ	Modalities			
	Sensory channels			,

Figure 7: A neo-Aristotelian poetics for interactive narrative.

We have seen that Aristotle's manner is, essentially, narratology's discourse. This discourse is what we experience by materially constructing the conventions of the medium. And from the material of the discourse, we construct the world and events of the story based on the perspective we are provided. In reverse, the events and the world of the story constrain the form of the discourse, which in turn formally determines how it is embodied in a particular medium.

Placement of User Interaction

In an interactive drama, where the user assumes the role of a character, the level in the model at which user action occurs is clearly meant to be that of the storyworld. The player interacts by affecting the setting and other characters. The story-world provides the material for what interactions are possible; the user's actions should then become part of the action. However, other kinds of interactive narratives have user interaction at levels other than the story-world.

At its most basic level, most recorded narratives offer some control over their medium—particularly its timing. For example, the reader of a book can skim parts of the text, reread others, or put the book down and come back to it later. A museum-goer can glance at a painting as they walk by or study it for half an hour.

Some narratives offer the user control over the details of their discourse or presentation. For instance, the user might be able to control the camera viewpoint or might select different hypertext links, changing the order in which the underlying story is experienced.

Or the user might be able to specify the kind of high-order action she would like to see, either as an interactive "director" or as input to a story generator, which would then determine the details of the characters and setting.

So it could be argued be that a DVD player, a hypertext novella, an interactive drama game, and a story generator are all interactive narratives. They differ only at the level at which user action is intended to occur. If this is the only sort of interaction afforded and supported by the narrative, the user may still feel some sense of agency as long as those affordances and constraints are balanced. However, the user's interactions become more significant—that is, they have a greater impact on the action of the story—the higher the level at which those interactions occur.

Application to System Architectures

This reformulated poetics more closely mirrors many existing interactive drama system architectures than its predecessor. Such architectures tend to share the same (generally implied) medium of computer software. Within this software medium, interactive drama systems tend to include one or more separate modules for handling action, character, setting, and manner, though often by different names.

Action—the events of the story—is usually guided or produced by an agent called a *director*, *drama manager*, *game master*, *narrator*, *scene manager*, or *story generator*. Characters include *agents* or *cast members* under the control of the drama system. An *avatar* or *player character* is the character under the control of the user. The setting—those locations and props of the dramatic world —might be collectively called the *story world*, *environment*, *stage*, or *theater*. Finally, the manner, or discourse, layer of the system handles the details of presentation to the user, as well as receiving interaction from the user. This component may be called the *narrator*, *presenter*, *renderer*, *theater*, or *interface*.

Certain terms, such as *narrator* and *theater*, appear under more than one category. Additionally, some projects subdivide or combine certain elements depending on their needs.

With this proposed poetics framework as a foundation, we can examine existing interactive drama architectures for similarities.

Oz. The Oz Project has followed a practically identical model (Mateas 1997). This is also the model used by Chris Fairclough's (2005) OPIATE project.



Figure 8: The Oz interactive drama architecture (Mateas 1997).

These system components map to the poetics levels as follows:

Poetics	Oz Project
Action	Drama Manager
Story World	World (including Characters)
Manner	Presentation

Table 1. Oz system components in terms of the poetics levels.

Facade. Not all architectures map as cleanly to the new poetics as does the Oz architecture. For instance, Mateas and Stern's *Facade* uses the following architecture:



Figure 9: Facade Interactive Drama Architecture (Mateas 2002, p.49).

At first glance, it may seem that Facade's Natural Language Processing module provides the manner layer. However, this processing is done only for textual input, not output. When a player enters a text string, the system translates this surface text into a discourse act.⁷ These discourse acts have nothing to do with

⁷ In the *Facade* architecture, textual input is a *modality* of the medium, and each complete "surface text" command is a *convention* of the medium.

manner or discourse in terms of the poetics; rather, they are events in the story world that formally prompt character reactions.

Facade breaks action management into beat selection and modeling the story so far.

Poetics	Facade
Action	Drama Manager + Story Memory
Story World	Story World (including Characters)
Manner	[not shown]

Table 2. Facade system components in terms of the poetics levels.

Virtual Storyteller. The Virtual Storyteller project aims at developing an animated character that narrates generated stories.



Figure 10: The Virtual Storyteller Architecture (Theune et al. 2003).

Though they mention characters' knowledge of their "virtual environment", they do not include this virtual world in their model. The characters' actions are directed by the "director" to produce a story. This project's "narrator" then rearranges the events of the story, and the "presenter" relates them to the user. This places these two components at the level of manner.

Poetics	Virtual Storyteller
Action	Director
Story World	Characters; ["virtual environment" (setting) not shown]
Manner	Narrator + Presenter

Table 3. Virtual Storyteller system components in terms of the poetics levels.

Spierling et al. Spierling et al. (2002) propose an architecture comprised of four hierarchical levels. The output of each level provides input for the one below it—much as the poetics describes formal cause working downward through the levels.



Figure 11: Spierling et al.'s Architecture (Spierling et al. 2002).

Spierling et al. have divided story management into two levels: the management of the overall general (functional/morphological) structure and the construction of specific, instantiating scenes. The Character Conservation Engines also hints at a model of setting to support stage direction.
Table 4.	Spierling et al.'s proposed a	architecture	components	in terms	of the poetics
		levels.			

Poetics	Spierling et al.
Action	Story Engine + Scene Action Engine
Story World	Character Conversation Engines
Manner	Actor Avatar Engines

As shown here, the revised poetics provides a good overview of interactive narrative elements. Each project here uses different terms and different divisions between architecture components. Often a project may include multiple components for a single level of the poetics. This generally indicates what aspects the project is most interested in exploring. For instance, Facade is focused on story management and so has two components at the action level. On the other hand, the Virtual Storyteller is particularly interested in the narration of stories and so they have two components at the manner level. Similarly, a project may also combine two elements into a single architecture component or fail to mention an element in their overview if it plays a marginal role in that project's particular focus.

Overall, it seems the revised poetics can at least inform the design of system architectures—which is not something that can be easily claimed of the previous version.

Parallels to Traditional Narratology

Besides applying to cutting-edge interactive narrative systems, the new poetics also shares striking similarities to the definitions found in established narrative theory. For instance, Seymour Chatman gives the following conceptual division of a narrative text:



[diagram abridged].

These concepts match those of the proposed poetics, with the translation of terms shown as follows:



Figure 13: Chatman's model of narrative with terms translated to those of the new poetics.

This simple translation shows how interactive narratives share much of their structure with traditional narrative forms, and so they can be understood in the same terms. While this translation allows us to apply the bulk of narratology to interactive narrative, it also offers something to narratology in return—specifically, the notion of formal and material causes at work between the different levels, which are the foundations of a definition of interactor agency.

Conclusion: A New Poetics of Interactive Narrative

The intent here has not been to return to Aristotle, but to clarify the existing poetics model based on its evolution through four authors. To do this, it is important to return to Aristotle's distinction between the object, medium, and manner. A story (or any creative work) is always embodied in some particular

medium. Renaming Laurel's bottom three levels, I have clarified the different aspects inherent to all such media. I have added setting to the world of the story, as action does not progress in terms of character alone, especially in an interactive drama. I have also resurrected manner—that is, the presentation of the story—within the model.

Applying Mateas's definition of user action and agency at different levels, this new model can apply to other forms of interactive narrative besides dramas. The model also mirrors many existing interactive drama system architectures much more closely than the previous poetics model, thereby connecting theory to design. Finally, this model has evolved striking similarities to a standard model of narratology. While providing the new poetics a strong credibility, it also allows for the application of traditional narratology to interactive narratives.

The Structure of Action

Now that I have established a framework for understanding the structure of narrative, we can examine the details of its components. My aim is to direct a story that incorporates a user's actions. Therefore, it is the structure of a story's action that is of the most interest to us here. Among others, the important features of a story's action have been examined by Aristotle, Freytag, Propp, and Chatman.

Aristotle

Besides laying out six formal components of tragic narrative, Aristotle explores the requirements for each component. In examining the action, which he calls *Plot*, Aristotle states that it should be whole, complete, and of a certain magnitude. More specifically:

plot, being an imitation of an action, must imitate one action and that a whole, the structural union of the parts being such that, if any one of them is displaced or removed, the whole will be disjointed and disturbed. For a thing whose presence or absence makes no visible difference is not an organic part of the whole (Aristotle 1961, Chapter VIII).

Aristotle continually stresses this unity of the action. The events comprising the action must be connected to one another by necessary or probable cause to make a single whole. By the nature of these causal connections, a story has a beginning, middle, and end:

A beginning is that which does not itself follow anything by causal necessity, but after which something naturally is or comes to be. An end, on the contrary, is that which itself naturally follows some other thing, either by

necessity, or as a rule, but has nothing following it. A middle is that which follows something as some other thing follows it (Aristotle 1961, Chapter VII).

A story is complete when it includes all these necessary events; it does not start or end haphazardly.

Aristotle also held that a plot should be of a certain magnitude. It should be at least long enough to admit a change in fortune (Chapter VII). Generally, he considered larger, longer plots to be more beautiful, but only so long as they did not exceed the audience's memory. Once a plot becomes too long or complex to hold all in mind at once, the audience will lose any sense of its unity (Chapter VII).

A tragic plot can be divided into two parts. The events before the change in fortune are the Complication; those after are the Unraveling (or Denouement) (Chapter XVIII).

Aristotle held that the worst plots are episodic, where episodes succeed each other without necessary or probable causes. He stressed that the unity of a single protagonist throughout does not guarantee a unity of action (Chapter VIII). This low opinion of episodic narratives seems to have survived to today, as we generally consider TV shows and serial fiction to be of lower quality or less "literary" than movies and novels. However, even these episodic narratives tend to exhibit Aristotle's unity within an episode.

Aristotle describes additional requirements for a good plot, but most of them are specific to tragedy: the Recognition, the Reversal of Situation, and the Scene of Suffering, as well as the structural parts in relation to the choric song.

Yet despite Aristotle's focus on tragedy, he does leave us with some rules applicable to any well-formed story. It should concern itself with a single, unified action. It should be long enough to admit some change in state. It should have a beginning, a middle, and an end.

Freytag

Gustav Freytag, a 19th century German novelist and playwright, further explores the requirements of a well-formed drama. His focus—while wider than Aristotle's concern with classical Greek tragedy—is still limited to "serious" drama. Though Freytag follows Aristotle's lead quite closely and still favors tragedy over other dramatic forms, he does offer some additional insights.

Freytag holds that a drama is based on a *Idea* of the author's. This central Idea provides the unity of the drama, as it should serve to structure the action and determine the significance of the characters.

Similar to Aristotle's concept of *praxis*, Freytag holds that a character's emotions, thoughts and motivations are essential to serious drama. The emotions or actions themselves are not as interesting as how a character's emotions serve to bring about a will to action. A character's motivation can arise from within, or it can be produced though external influences upon the character.

The action should be unified, with a clear beginning and end. The end should bring a termination to any strife within the play. The events should follow each other as necessary or probable. The playwright must do more than simply show the events: he must make the events believable by exploring the characters' motivations and reasons, which should be consistent and credible.

Yet, for all this unity, Freytag does admit that occasional episodes not completely essential to the central plot or Idea may highlight or clarify a character, provide an interesting contrast to the main action, or otherwise enhance the overall effect of the play. If done correctly, these ornamental embellishments cannot again be easily "unclasped" from the main work.

Freytag is best known for his pyramidal depiction of dramatic structure. He begins by defining two states of the action: the *play* and the *counterplay*. During the *play*, the hero is predominately proactive, working outwards, striving, turning a desire into action. During the *counterplay*, external forces or opponents are affecting and directing the hero; the hero is primarily passive, his motivations and actions arising in response to outside forces rather than from within. A serious drama will contain a both a *play* and a *counterplay*, though either one can come first. The point where one becomes the other—when the passive hero finally resolves to action, or else when the active hero begins to be subjected to the circumstances his actions have wrought—is the *climax*. This definition implies that a serious drama will always contain some sort of struggle or conflict involving the hero.

Including the *climax*, Freytag defines five "parts" of drama, as well as three scenic effects, or "crises".



Figure 14: Freytag's Pyramidal Model (1895, p. 115).

These components occur in the following order:

The *Introduction* (*a*) explains the background of the drama by establishing time and place, noting the nationality and life relations of the hero, and briefly characterizing the environment. This may be presented as a narrator's call for attention (as is more common in older dramas) or as a short scene of action itself.

The *Exciting Force* is a scenic effect that occurs between the *Introduction* and the *Rising Movement*. It may be a whole scene, or only a few words. It marks when the volition arises in the hero that will lead to the action of the *play*; or, if the *counterplay* occurs first, it is when external forces resolve to affect the hero.

The *Rising Movement* (*b*) includes those events that further the action, introduce all major characters, and awaken the audience's interest.

As previously defined, the *Climax* (*c*) is the moment when the *play* becomes the *counterplay*, or vice versa. It should be inseparably connected to the previous action.

The *Tragic Force (or Moment)* is a scenic effect that may not occur in all dramas. Closely tied to the climax, it marks the beginning of counterplay.

The *Return* (*d*) begins to resolve the action. It should not introduce new characters or material, but build on what has already been established.

The Force (or Moment) of Final Suspense is a scenic effect that may not occur in all dramas. It seals the conclusion of the drama such that the audience feels "the compelling force of what has proceeded", for the Catastrophe should not come as a surprise.

The *Catastrophe* (e) completes the action. It should be brief, and provide a fitting end for the hero.

Freytag uses two dimensions for his diagram. Presumably, the horizontal axis is story time. Freytag does not define the vertical axis nor does he specify what exactly is "rising", "returning", or "falling" through the plot.

Besides exploring structure, Freytag also puts forth strong opinions on the content of serious drama which may not apply to many modern narratives. For instance, the hero's "force and worth shall exceed the measure of the average man" (Freytag 1895, p.63). The action should not be based on lamentable or common motives—such as thieving, cowardice or stupidity—leading to dishonest actions. The details of serious drama should not contradict reality. Modern narratives frequently violate these rules.

Propp and Story Morphology

Vladimir Propp authored *Morphology of the Folktale* in 1927, although it was thirty years until it was translated into English (Propp 1968). In it, Propp explores how to classify folktales. Most previous attempts at classification were based on the contents of tales—such as a tale's general category (fantastic, everyday life, or animal tale), a tale's themes, or a tale's motifs (such as the presence of a dragon, witch, or magic ring). Instead of content, Propp looked at each tale's structure—specifically, the effects of the actions of the characters and agents.

For example, consider these parts of three tales:

- The tsar gives the hero a great eagle, which bears the hero to a new kingdom.
- An old man gives the hero a horse, which the hero rides to a new kingdom.
- A sorcerer gives the hero a ring, which magically transports the hero to a new kingdom.

Rather than classify these depending on whether they include magical items, or a tsar, or whether the hero is an animal, Propp recognized the function present in them all: a "donor" is providing the hero with a special item, which is then followed by the spatial transference of the hero to a new land.

As Propp defines it, a function "is understood as an act of a character, defined from the point of view of its significance for the course of the action" (Propp 1968, p21). These functions "constitute the fundamental components of a tale", and they are "independent of how or by whom they are fulfilled" (Propp 1968, p.21). That is, while the specific details provide the color unique to each tale, the

underlying structures are constant. Furthermore, the number of known functions is limited; and, though not all functions are present in all tales, those functions that do appear always occur in the same sequence (Propp 1968).

A function can be seen as a genus, with a number of specific events (and even variation on those events) serving as an example of that function. Indeed, Propp provides a number of example "species" events for each "genus" function. However, the same events can fill different functions depending on their place in the story. For instance, a man may marry a widow with two children, thereby setting up the action of the story. Or the hero may receive the hand of the princess, thereby achieving his reward and ending the story. Here, the event of marriage is filling different functions at the beginning and the end of the tale.

Propp studied one hundred Russian folktales from the Aarne-Thompson index, and discovered 31 functions and a handful of other morphological features, such as character roles. As illustration, here are some of the more prevalent functions:

Symbol	Function	Description
α	Initial Situation	Introduction to hero by name or status, enumeration of family members, etc. (Though this a morphological element, it is not considered a function.)
β	Absentation	A member of the family leaves home.
Y	Interdiction	The hero is forbidden to do something.
δ	Violation	The hero violates the interdiction.
3	Reconnaissance	The villain attempts to gain information
ζ	Delivery	The villain succeeds in learning something about his victim
A	Villainy	The villain causes harm, injury, or misfortune to a family member. (This serves as the complication, ending the "prepatory part" and beginning the actual movement of the tale.)
а	Lack	The hero or a family member lacks something. (Serves as an alternative to A .)

Table 5. An example subset of Propp's functions

Symbol	Function	Description
В	Mediation	The hero is made aware of the misfortune or lack. (Distinguishes between seeker/voluntary heroes and victimized/involuntary heroes.)
С	Counteraction	A seeking hero agrees to go.
1	Departure	The hero leaves home.
D	First Function of the Donor	The hero encounters a donor (who greets or otherwise tests the hero).
E	Hero's Reaction	The hero reacts to the donor.
F	Provision or Receipt of Magical Agent	The hero receives some item, animal, or other assistance from the donor.
G	Spatial Transference	The hero is guided or transferred in the direction of the object of his search.
н	Struggle	The hero and villain engage in combat or competition.
I	Victory	The villain is defeated.
к	Liquidation of Misfortune or Lack	The initial villainy is undone, or the initial lack is fulfilled. This function is paired with A or a and forms the peak of the narrative.
\downarrow	Return	The hero returns home.
w	Wedding	The hero marries, ascends the throne, or receives some other reward.

Table 5. (Continued) An example subset of Propp's functions

Propp comments that the functions all "belong to a single axis" (1968, p64); that is, they follow each other sequentially in a tale. Furthermore, "one function develops out of another with logical and artistic necessity" (Propp 1968, p64). A number of the functions come in pairs. For instance, an Interdiction (γ) is always Violated (δ). As mentioned, the details of a Liquidation (**K**) depend on the nature of the initial Villainy (**A**) or Lack (**a**). Other logical groupings are noticeable, such as the complication (**ABC** \uparrow) and the interaction with the donor (**DEF**).

Propp explains other variations within functions. For instance, sometimes they have a negative form, such as a command rather than an interdiction (γ), which must then be fulfilled rather than violated. Or the hero may react negatively (**E**) to

the donor's request (**D**). Trebling of a function or sequence is common—the hero meeting three donors before receiving the agent, or three heroes meeting a donor before the last hero is successful.

As it provides specific classes of events that linearly comprise a well-formed tale, Propp has long been a inspiration for automatic story generation and interactive narrative systems. However, Propp's model applies to a small set of simple tales —Russian folktales. His functions do not always apply cleanly to fairy tales from other cultures. It is certainly not a universal model of narrative structure. The hope is that similar rules of structure will be found for other genres of tales. However, despite the work of other structuralists, such Tzvetan Todorov's analysis of the *Decameron* stories (Chatman 1978), Propp remains the most widely cited model in interactive narrative work to date.

Chatman

In examining the details of literary theory and narrative structure, Seymour Chatman (1978) explores the nature of a narrative's *events*, which corresponds to our meaning of *action*.

Chatman notes that a story's events are "radically correlative, enchaining, entailing", forming a sequence that is "not simply linear but causative" (1978, p. 45). As we have seen, this has been a view held since Aristotle. Chatman suggests that readers will often infer a causality between events, even if such a relationship is not explicitly stated in the narrative itself.

More recent critics have denied such a strict causal view of narrative. A more relaxed view is that later events are simply *contingent* upon earlier events. That is, the later events depend on earlier events for their existence or occurrence, even if those earlier events did not specifically cause the later events.

However, Chatman notes that not all narratives are concerned primarily with events, changes, or consequences. In a modern plot of revelation, the point is to simply reveal a state of affairs or to explore the details of a character.

The connections between events play a part in a story's verisimilitude. Part of a story's "believability" stems from how early events lead to later events. That is, to what degree does the story contain outrageous coincidences or completely unforeshadowed solutions. But another part of a story's verisimilitude involves the characters—particularly whether their reactions and motivations are understandable. However, it should be noted that verisimilitude is largely a matter of convention established by other texts in the same genre. For example, shooting a cheater over a game of cards is relatively normal behavior in a western, requiring little explanation. The same act is less usual in a Victorian

society novel, which means the killer's motivations may need to be presented in more detail for a reader to accept the action.

Chatman also argues that not all events are equal—some are more important than others. These important events, which he calls *kernels*, are those narrative moments where the course of events is decided, where one path is chosen from the various possibilities. It is these kernels that are connected by causality or contingency; as such, these kernels cannot be removed without destroying the logic of the story.

However, there also exist minor events, which Chatman calls *satellites*. Satellites do not entail any choice, but serve to flesh out the consequences and details of the kernels. Therefore, satellites could be removed from the story and leave the logic intact, although the resulting story would be impoverished.

These kernels and satellites define a microstructure of events. However, we can also discern a macrostructure—a general overarching story structure—which allows us to group stories together based on structural similarities. The work of Propp and Todorov are examples of this. But Chatman points out the limits of this approach. These macrostructures are not generic or universal, but very specific to a narrow, particular genre—such as Russian folktales or *Decameron* stories. These are simple, well-structured tales, and their macrostructure does not apply to more general stories.

Nor can we classify stories simply by indexing their kernel events, as an event can only be understood in terms of its greater story context. "A killing may not be a murder but an act of mercy, or a sacrifice, or a patriotic deed, or an accident, or one or more of a dozen other things. No battery of preestablished categories can characterize it independently of and prior to a reading of the whole" (Chatman 1978, p. 94).

Conclusion

Based on this review of four authors, we can now draw some general conclusions concerning the structure of narrative *action*.

Unity

First of all, the action should be a single, unified *whole*. This is either because it concerns a single action (Aristotle 1961) or revolves around a single central Idea of the author's (Freytag 1895).

Yet the action is made up of a series of separate events. What unifies these events is that they are connected by necessary or probable cause (Aristotle

1961, Freytag 1895), "logical or artistic necessity" (Propp 1968), or some other connective contingency (Chatman 1978). Speaking generally, we might simply call this *necessity*. An event is necessary to the tale if its displacement or removal would leave the whole "disjointed and disturbed" (Aristotle 1961).

An essential part of necessity is characters' motivation or *praxis*—how their emotions or thoughts lead to a will for action (Aristotle 1961, Freytag 1895). Believability of character motivations and the credibility of necessary connections contribute to the verisimilitude of the tale (Freytag 1895, Chatman 1978).

Completeness

Besides from being a unified whole, a story must be *complete*. Specifically, it must have a beginning, middle, and end. This usually implies some change of state—an initial condition, some change or problem, and then a final condition. This change of state often corresponds to a change of fortune for the protagonist (Aristotle 1961, Freytag 1895).

But more than a simple change, stories often involve some conflict concerning the protagonist. Freytag defines his climax in terms of the *play* and the *counterplay*, which refer to the protagonist actively working towards some end or else being the subject of external forces. In order to be complete, this conflict must be resolved by the end of the story.

The length and detail of a story—Aristotle's "certain magnitude"—varies greatly. Size has an impact on completeness, however, since generally there are more events and details whose various effects must be resolved before the conclusion of a long tale.

Morphology

Every story also has a macrostructure or formal morphology. At its simplest, story events occur during either the beginning, middle, or end of the tale (Aristotle 1961), or else within the bounds of Freytag's eight parts and crises. According to the conventions of a certain genre, we might determine a more detailed morphology, as Propp did for Russian folktales.

Chatman's point concerning an event's greater story context is also important to consider here. For instance, we may have an event in the world of the story, such as a killing. But it is only in terms of the story context that we can determine the importance or meaning of this event—whether it is a murder, a sacrifice, or act of mercy (Chatman 1978). Furthermore, these different story-contextualized events can fill different functions in advancing the story. For example, within Propp's

functions, a murder might serve as an initial villainy, or the hero might murder the donor in order to receive the magical agent.

This example reveals three views of a story's action. We have the world-level: the event itself in the story world. We have the story-level: what that event means in terms of the greater story context. And we have the morphological-level: what abstract function, if any, the event serves in advancing the story⁸

Content

A story does not concern only form, but content as well. Indeed it is the content the quality of the characters, the details of the world, the specific flavor of the events—that makes each narrative truly unique.

Exceptions

Finally, we must remember that not all narratives adhere strictly to these rules. Aristotle admits the existence of episodic plots that are neither unified nor complete. Chatman points out the existence of "antistories"—narratives that deny any linear sequence of events—as well as revelatory stories, where the focus is on the existents of the world rather than on the events. And even within a relatively well-structured story, some events do not need to be strictly necessary, as both Freytag's "ornamental episodes" and Chatman's satellites illustrate.

Still, if a story is well-formed, it should demonstrate these basic formal features: a unity of events through necessity, forming a complete whole with a beginning, middle, and end, and exhibiting a general macrostructure story-form.

Conclusion

We now have a working model of narrative sufficient for my purposes here. A narrative is more than just a series of events. Instead, it involves action, characters, setting, manner, and medium.

The action should be both unified and complete. Character and setting are essential material for this action, because "one cannot account for events without recognizing the existence of things causing or being affected by those events" (Chatman 1978, p.34).

This story-world and its events are then presented or narrated from a certain point of view. And this presentation is encoded in some medium. Whether told

⁸ These three views of the same event are simply different levels of abstraction. This means they are not connected in terms of material and formal causes.

live in spoken words or encoded in a static artifact to be viewed later, the medium affects and constrains the transmission of the story.

A narrative becomes interactive when it offers the audience some means of affecting one of these aspects. This interaction will offer the user a sense of agency when the formal and material causes involved are balanced.

CHAPTER III. MARLINSPIKE ARCHITECTURE

The Marlinspike architecture is designed to produce a *directed interactive drama*. An *interactive drama* is a system in which the player assumes the role of a character in an unfolding story and, through their actions within the virtual story world, can influence the outcome of that story. By *directed* I mean that the system has some form of centralized control over the story world. This centralized *drama manager* component works to direct the story, thus generating a well-formed, coherent plot that incorporates the user's actions. (This is in contrast to an *emergent* approach, in which the story is meant to emerge solely from the player's interactions with autonomous character agents or the static rules of the story world.)

The following section outlines the design of Marlinspike.⁹ We will then see how this builds upon the poetics discussed previously, as well as upon other existing interactive drama architectures.

Marlinspike Design

A Marlinspike drama takes place within a particular *story world*—a collection of simulated objects, including *locations*, *props*, and *characters*. The player dictates the actions of one of these characters (the *player character*, or PC). The other *non-player characters* (NPCs) are controlled by the drama manager through scenes.

The player interacts with the objects of the story world by selecting one of a number of *verbs* to perform. Verbs specify simple physical actions, such as taking, dropping, kissing, or talking. Based on the current story context, each verb is then translated into one or more *actions* that represent what that verb signifies within the current story. For example, a kiss may have different "meanings" depending on the context: it could be an attempt at romance, it could be done simply to break a magical enchantment, or it could be a Judas-like betrayal.

The Marlinspike *drama manager* (DM) then responds to user actions by selecting the next *scene* to play. Whenever possible, scenes are selected that refer back to or otherwise build upon earlier user actions. This *reincorporation* of earlier actions forms a story *thread* of connected events that makes earlier user actions narratively necessary to the finished story.

⁹ Many of the details of this chapter were presented in earlier works (Tomaszewski & Binsted 2008, 2009).



Figure 1: An overview of the Marlinspike architecture

While doing all of this, the drama manager deals with only the abstract story structure. All the specific content—including the world objects, characters, verbs, actions, and scenes—is defined separately by a particular implementing game.

The details of each component of the Marlinspike architecture follow below.

Story World: Characters and Objects

The objects of a Marlinspike story world are simply bundles of state and invoked behaviors. For example, props include such state as their current location in the world. They may also define their own specific responses to being affected by different verbs. The particular details of objects' state and behaviors are left to the implementing game.



Figure 2: Marlinspike starts with a virtual world of objects and characters

Similarly, non-player characters are not autonomous agents. Instead, NPCs are simply comprised of a number of game-specific attributes. Example attributes might include *morality* (how altruistic or self-interested that character is) and *affinity* (how much they like or dislike another particular character). These attributes may then affect how certain verbs are translated to actions and how the NPC behaves in different scenes. NPCs might also contain their own set of potential conversation responses, which can be used by scenes to customize the story's narration depending on which particular NPC has been selected to perform the events of that scene.

World-level Events: Verbs and Deeds

A Marlinspike game defines a number of *verbs*, which are used by the player to interact with the objects of the story world. As the current Marlinspike implementation is text-based, example player *commands* might include take tennis ball, go north, talk to Alice, or punch Fred. These commands illustrate the verbs Take, Go, Talk, and Attack (for which "punch" is a synonym).

When processing user input, the game system first determines whether the resulting command is possible. For example, the tennis ball might be nailed to the floor or Alice might not be in the room, making the commands take ball or talk to Alice impossible to complete.

Every valid command is then translated into one or more world-level events called a *deed*. A deed has a sentence-like structure that includes the verb performed, the direct object that was primarily affected by the verb, and an optional second object that was also affected by or involved in the deed. As an example, suppose the player's character is currently on the tennis courts and the player types give ball to Alice. If this deed is successful, it would be represented as: Give (ball, Alice).¹⁰

The physical outcome of deeds is usually presented to the player as a single sentence of narration, such as "Taken." or "You offer the tennis ball to Alice."

¹⁰ This is actually a simplification. Marlinspike deeds also contain the actor that performed the verb (usually the PC) and the location in which the deed occurred. So this deed would actually be represented as: (PC, Give, ball, Alice, TennisCourts). However, to simplify the examples over the next couple sections as we explore the process of casting deeds to actions, I will leave off the actor and locations from both deeds and events.



Figure 3: The player directs his player character to affect world objects through verbs, which produce specific deeds in the world.

Story-Level Events: Actions

Every world-level deed is then translated into a story-level *event* based on the current story context. Specifically, the player's deeds are represented as story-level *actions*. Action events have a structure like that of deeds that includes both the particular action and its affected objects or characters.¹¹

Through a processes called *casting*, every deed has a default translation to an action. For example, the verb Drop generally translates to the MANIPULATE action. This default translation may take the current state of the world or characters into account. So the deed Kiss(Alice) should translate to the event ROMANCE(Alice) only if Alice already has a high affinity for the player. If Alice detests the player, this same deed would be better translated to ASSAULT(Alice).

Occasionally, this default casting can be completely overridden by the story context. For instance, if Alice has been transformed into a frog, then Kiss(Alice) should perhaps instead become RESCUE(Alice).

¹¹ The current Marlinspike implementation actually uses the same data structure for both deeds and events. This event data structure looks like this: Event (actor, ACTION, verb, directObject, secondObj, location). If an event contains a verb value but is missing an action, then it is a deed. If an event contains an action value, it may or may not also include a verb. If the event started as a deed that was then cast into an action, it will have a verb. On the other hand, if the event is a recast sub-event (as explained later in this section), it will usually not have a verb. Scenes (discussed in the next section) also use this data structure, replacing the ACTION with the name of the scene played.



Figure 4: Verb-based deeds are then cast (and recast) into action-based events.

A deed can also be translated into more than one action. In this case, the extra action events form of a tree of sub-events called *recasts*. So, if the player has already established a girlfriend relationship with the character Betty, Betty might take offense at this kissing of Alice. This secondary effect would be appended as a recast, producing the following structure:

Actions have effects associated with them. Since only deeds change the "physical" state of the world, action effects are generally limited to character state changes. So RESCUE(Alice) will probably increase Alice's affinity for the player, and possibly increase the player character's morality value.

Unlike deeds, actions do not produce any narration. After each event has occurred, it (including all of its recasts) is appended to an *event history* list, which forms a transcript of story-level events that have occurred so far. Once recorded there, the DM can respond to the action with a scene.

Not every action the player performs will be equally significant. To represent this, every action also includes a pre-defined *import* value which suggests how dramatically exciting it is. For example, a MANIPULATE action (which represents such verbs as Take, Drop, and Push) would generally have a very low import, while a MURDER action would have a very high import. This import value is used by Marlinspike when prioritizing how to respond to various actions. This way, actions of high import receive more attention from the system than actions of low import.

Thus actions are fairly simple, as shown by the following pseudo-code definition of an INSULT action.

```
Listing 1. Example pseudo-code definition of an INSULT action.
Action INSULT {
    super.import = 5
    function do(event) {
        //Reduce insulted NPC's affinity for the actor
        affinity = (event.dirObj).affinityFor(event.actor)
        affinity.modify(-10)
    }
}
```

Story-Level Events: Scenes

A *scene* is a pre-authored bundle of narration and world-level manipulations to be made by the DM. Like an action, a scene is a story-level event. The difference between the two is essentially that action events are produced by the player while scene events are produced by the drama manager.

Scenes serve three purposes in Marlinspike. The first is to provide reactions to player actions. While verbs provide narration of their world-level effects, only scenes provide narration of NPC responses and other story-level effects. The second purpose of scenes is to advance the story by introducing new incidents and material. Finally, scenes can provide a story context that may affect later verb-to-action casting and action-to-action recasting (as previously described above).

Every scene has a list of *preconditions* that determines whether it can currently be performed and appended to the story-so-far. Preconditions might include the current story-world time, character locations, prop states, character attributes, or previous story events.

Besides preconditions that must be true for a particular scene to run, scenes can also include *hooks*, which are previous events that the scene can refer to or otherwise reincorporate if they have occurred. This aids in threading (described below).

Each scene belongs to one of three *functions*. *Beginning* scenes have no preconditions and are selected by the drama manager to start a new story. The bulk of scenes are *middle* scenes, which have preconditions and serve to advance the story in some way. They, in turn, usually satisfy the preconditions of

later scenes. *Ending* scenes also have preconditions, but provide a conclusion to the current story. A story will end as soon as the drama manager can meet the preconditions of an ending scene.

Like a paragraph in a novel, scenes vary in how much action they present. A scene may be only a single line of dialog delivered by an NPC in response to a question asked by the PC. On the other hand, a scene may be paragraphs long or summarize the passage of hours. Regardless of length, no user input—and thus no deeds or actions—are possible during a scene. Like actions, completed scenes are appended to the story event history.



Figure 5: The drama manager replies to actions by playing scenes.

Also like actions, scenes have an import value. This value indicates the narrative significance of the content that the scene just contributed to the story. Scenes also have an author-assigned *imperative* value that gives a certain weight to scenes during their selection process. (The details of this process are explained later in this chapter.) In brief, when other considerations are equal, scenes with higher imperative values will be chosen over those with lower imperatives.

Like actions, scenes have a very simple structure in Marlinspike. Each turn, every scene's canPlay function is polled to see whether that scene has its preconditions met. Then, once a scene that can play has been selected, its play function is called.

To illustrate this, here is a pseudo-code example of a very simple scene.

```
Listing 2: Example pseudo-code definition of a simple NPC_Is_Insulted scene.
MiddleScene NPC is Insulted {
 super.import = 3
 super.imperative = 2
 //internal state specific to this scene
 this.insult = null
 this.hook = null
 function canPlay() {
   // If all preconditions have been met, returns a list of
   // those precondition events and any optional hooks.
   // Otherwise, the scene can't currently play and so returns null.
   if EventHistory.current contains Event(any, INSULT, any,
                                           ofclass NPC, any, any):
     //precondition met: Some NPC was just insulted
     this.insult = EventHistory.current
     List reincs = new List()
     reincs.add(this.insult)
     this.hook = EventHistory.past contains Event(this.insult.actor,
                   INSULT, any, this.insult.dirObj, any, any)
     if this.hook:
       //hook found: This is not the first time this actor has
       //insulted the same NPC
       reincs.add(this.hook)
     return reincs //can play, reincorporating these events
   return null //no insult found, so can't play
 }
 function play() {
   // Plays this scene
   print this.insult.dirObj.name.uppercase()
   print ": "
   print random("How dare you!", "How rude!",
                this.insult.dirObj.insultedReply)
   if this.hook:
     print " Your continued petty belligerence is tiresome indeed."
   print lineEnd
 }
 function getTriggers() {
   //This scene does not add any triggers to the story context
   return null;
 }
}
```

As demonstrated by this pseudo-code example, although scenes are preauthored, they can be written in such a way that most of their specific details are filled in at run-time. For example, this scene will play for any NPC that is insulted, regardless of the location or the character that did the insulting (PC or another NPC). Also, this scene sometimes pulls text from the specific individual NPC that is replying in order to further customize the scene. This example also shows the use of a hook in that the output of the scene can change to refer to an earlier event when it exists in the event history. In practice, a repeatedly-insulted NPC might want to warn the insulter not to do that again. Such an interdiction would add to the story context, which brings us to triggers.

Story Context: Triggers

After a scene plays, its getTriggers function is called. The scene can then add to the story context by returning one or more triggers from this function. A *trigger* simply watches for the later occurrence of a certain deed or event. Upon finding a match, the trigger can then modify the casting of a matching deed or the recasting of a matching event.

For instance, in an interactive Bluebeard fairy tale, a scene could have Bluebeard tell the player not to open a certain closet. The scene then ends, but it would also provide a single trigger to watch for the Open(closet) deed. Should this deed occur, the trigger can then recast the default MANIPULATE(closet) event to include a DEFY(Bluebeard) sub-event. Now both the MANIPULATE and the DEFY actions can provide material for future scenes. Certainly such a fairy tale would include a scene that provides a response to DEFYing Bluebeard.

A deed is cast to an action by the first trigger that matches that deed. Casting triggers are processed in the order they were added to the story context, starting with the most recently added trigger. Each action event is then recast by all the triggers that match it. This means a single deed can produce a fairly sizable tree of sub-events that represent all the story-level interpretations of that deed.

Because casting and scene selection actually happens before a deed is allowed to complete, it is also possible for a casting trigger to interrupt a deed. For example, a player may try to open a certain door. However, a casting trigger could cast this opening deed to an ATTEMPT action and thereby prompt a scene in which a nearby NPC leaps forward to prevent the player from actually opening the door.

Sometimes a scene may need to establish a more complex context than can be managed with only one or two separate triggers. Or a number of scenes might all require that the same fairly complex state of affairs exists before they can play. In these situations, a story state can be used. A *story state* is a trigger that also represents that an important story-level state currently exists.

As an example of a story state serving as a precondition, a Discussion_Starts scene might activate a "discussion" story state trigger. This discussion state trigger may then recast certain events—such as the player character leaving the room or attacking another character—as ending the discussion state. But, the discussion state itself can also be a precondition for other discussion-related scenes that further or naturally conclude the discussion. This helps to simplify the authoring of the preconditions for these discussion-related scenes, as they do not need to perform all the tests needed to determine whether a discussion recently started and is still ongoing. Instead, they can simply check if the discussion state is active.

The following is a pseudo-code definition of such a discussion story state trigger. It has the same code structure as any other trigger, but it may also be used as a precondition for other scenes because it is a state trigger.

```
Listing 3: Example pseudo-code definition of a recasting story state trigger.
StateTrigger Discussion {
 // Set by the scene that originally provides this trigger, this
 // variable tells Marlinspike that any event recast by this trigger
 // reincorporates the given source event. It will later point to
 // the most recent Discusion Starts event
 super.source = null
 // Whether this trigger should be removed from the TriggerManager
 // as soon as it successfully recasts an event.
 super.expire = true
 function recast(event) {
   if event matches Event(PC, import > 4, any,
                          any, any, source.location):
     // Any significant action by the PC in the presence of the
     // discussion interrupts that discussion
     return new Event (PC, END STATE, null, this, null, PC.location)
   else if event matches Event(PC, TRAVEL, any,
                                any, any, source.location):
     // Any departure from the conversation by the PC interrupts it
     return new Event (PC, END STATE, null, this, null, PC.location)
   else if event matches Event(any, Discussion Concludes, any,
                               any, any, any):
     // A discussion may reach its natural conclusion through
     // the appropriate scene
     return new Event (event.actor, END STATE, null, this,
                      null, event.location)
 }
}
```

Similarly, a story state can also be used to represent certain roles filled by NPC characters. As an example, a scene might assign an NPC to the role of a Friend to the player character. This Friend role may then serve as a precondition or a hook for certain scenes. However, this role story state is essentially self-managing in that it recasts any conditions that would end the role—such as if the player severely affronts the Friend character.

Narrative Unity: Threads

It is not sufficient for an interactive drama system to simply provide believable responses to separate player actions. Such an approach may present a very believable virtual world, but it will frequently fail to produce a coherent story that includes the player's diverse actions in that world. Therefore, Marlinspike strives to weave both user actions and authored scenes together to produce a finished story that meets an Aristotelian definition of narrative unity.

More specifically, the events of a finished story can be connected in terms of necessity. That is, if one event must logically precede another, we can say that the first event is *necessary* for the second event to occur. Another way to look at this same relationship is that if a later event refers back to the contents of an earlier event, then the later event *reincorporates* the earlier event.



Figure 6: Necessity and reincorporation are two views of the same relationship between two events.

Marlinspike works in terms of reincorporation. It tries to select the next scene to play so as to reincorporate—that is, to narratively build upon—earlier events of the story. When successful, these reincorporations make those earlier events narratively necessary to the finished story.

There are two ways to form reincorporation connections between events in Marlinspike. The first is through a scene's preconditions and hooks: whenever a scene requires or refers to a previous event in its own narrated details, it is considered to be reincorporating that earlier event. The second form of reincorporation occurs through story context triggers: if an earlier scene set a trigger that is involved in casting or recasting the current action, then it means that the earlier scene has some direct relevance to the current action.



Figure 7: Two kinds of necessity in Marlinspike

We saw each of these connection methods in the earlier Bluebeard example. A scene may have Bluebeard instruct the player not to open a certain closet door. The scene then sets an appropriate trigger to enforce this context. If the player opens the closet, then this trigger produces the DEFY(Bluebeard) sub-event. When this happens, we can say either that the earlier interdiction scene was necessary for the DEFY(Bluebeard) action can now serve as a precondition for an enraged Bluebeard scene. By playing this Bluebeard's Wrath scene, the system builds upon the player's action and thus opening the closet becomes necessary to the finished story.



Figure 8: A simple thread demonstrating the two types of reincorporation.

A sequence of reincorporated events like this is called a *thread*. Threads weave events into a coherent storyline. Every story starts with a *beginning* scene. This initial thread can be continued by some action building on the context established by this first scene.

However, the player might also start a new thread by performing some unrelated action of high import. As mentioned earlier, every event—whether action or scene—has an *import* value that suggests how narratively significant that event is. The current Marlinspike implementation uses import values between 1 and 9. Any event with an import of 4 or higher is significant enough to start its own thread.¹²

¹² This is a rather arbitrary default value used in the current Marlinspike implementation that can be customized by the author of an implementing game.

The Marlinspike drama manager can respond to the player and advance the story by choosing the next scene. When a scene plays, it will thread all of the previous events that it can. (Because all of an action's recast sub-events are considered to be part of that event, the whole event is considered to be reincorporated if any of its sub-events is reincorporated.) If these reincorporated events include the last event of only one active thread, the scene simply *extends* that thread.¹³ If the events of more than one thread end can serve as preconditions or hooks for the scene, then the drama manager can effectively *splice* two (or more) threads together into one thread. If the system cannot currently extend the end of any existing thread, it may be forced to *fork* an earlier thread event, thereby creating a new thread which it will hopefully be able to splice back in later.

This threading behavior is the source of the *Marlinspike* name, which refers to the splicing and fancy rope work of marlinspike seamanship. This splicing approach also enables Marlinspike to reincorporate significant actions of the player into the story, even when those actions were not related to any preceding events.



Figure 9: Thread map demonstrating a fork (producing Thread 3 from Thread 1) and a splice (producing Thread 4 from Threads 2 and 3).

In the Figure above, SCENE₆ is still pending. If its thread (Thread 1) is not extended before the end of the story, then the story will be poorly-formed, as it contains unnecessary events. However, this evaluation is tempered by the rule that not every action or scene needs to be spliced into a thread, but only those of high import.

As this example also suggests, not every Marlinspike story will necessarily be perfectly formed. To operate successfully, the drama manager must be armed with sufficient scenes to reincorporate any player actions of high import. If the implementing game author does not provide these scenes, or if the player is particularly strident about starting new unrelated threads, the resulting stories are

¹³ Unlike an action, middle and ending scenes should never start a new thread. This is because, in accordance with the definitions given earlier, such scenes should be authored to have at least one story-level precondition. In the unusual absence of any other relevant precondition events, the beginning of the story can fill this requirement. Therefore, if any middle or ending scene can play, it will reincorporate at least one previous event.

not guaranteed to be well-formed. Instead, Marlinspike will do the best it can until it can reach an ending scene.



Figure 10: The drama manager selects scenes so as to form threads of events connected by narrative necessity.

Scene Selection Details

After each player deed, the Marlinspike drama manager gets a chance to respond by playing a scene. The details of how this next scene is selected are as follows.

The drama manager first polls each scene in its pool to see if the scene can currently play. If all of the scene's preconditions are currently met, the scene returns a list of all the previous events it would reincorporate if it was selected to play at this point. This list includes the scene's precondition events but also any hooks—optional events that the scene will be able to refer to in its narration given the current event history.

The drama manager then scores each scene that can play by summing the following values:

 The weight of all material (threaded or otherwise) that will be reincorporated by this scene.
 Material means all those events directly reincorporated by the selected scene, plus all the events recursively reincorporated by those events. The weight of this material equals the highest import value of any scene in this material plus a bonus for every four¹⁴ additional events in the material. This score gives a boost to scenes that extend a major thread or build upon the bulk of the story so far.

- The weight of the unique thread material that will be reincorporated if the scene is played.
 Material and weight here are the same as defined as above. Unique thread material means those events of the material that exist in one and only one thread. This score favors scenes that build on neglected or fringe story material. If this scene is also splicing this unique thread to a significant amount of total material, it will score well on both measures.
- The scene's author-specified *imperative* value. This lets the author fiddle slightly with the selection process in order to favor certain scenes in the case of ties or close scores.

As an example of this scoring process, let's look at the selection of $SCENE_9$ in the previous example.



The events reincorporated by SCENE9

SCENE₉ directly reincorporates ACTION₄ and ACTION₈. This means the scene extends both Thread 2 and Thread 3, and recursively reincorporates all of the events (material) of those threads: ACTION₄, SCENE₃, and ACTION₂ in Thread 2, and ACTION₈, SCENE₇, ACTION₅, and SCENE₁ in Thread 3. This total material includes seven events. The total material might also include any events reincorporated by SCENE₉ that are not in any thread (and thus not shown here), such as certain low import events.

Most of these reincorporated events exist in only one thread, except for $ACTION_5$ and $SCENE_1$, which are part of both Thread 1 and Thread 3. Therefore, the five remaining events— $ACTION_4$, $SCENE_3$, and $ACTION_2$ in Thread 2, and $ACTION_8$ and $SCENE_7$ in Thread 3—form the unique threaded material.

¹⁴ The +1 weight bonus for every four extra events is a rather arbitrary default value used in the current Marlinspike implementation. This setting can be customized.

The selection score for $SCENE_9$ would thus be the weight of its total reincorporated material + the weight of the unique threaded material + $SCENE_9$'s imperative value. If this total score is higher than than of any other scene that can play, $SCENE_9$ will be selected to play next.

In the prototype implementation of Marlinspike, author imperative and reincorporation can each be switched off. When imperative is off, scene imperative values are ignored while selecting the next scene. When reincorporation is switched off, neither the total reincorporated material nor unique threaded material scores are used during scene selection. In addition, when reincorporation is turned off, scenes should not *hook* any optional previous events when played. Thus, if both features are turned off, Marlinspike simply selects the next scene randomly from those scenes that have had their preconditions met.

Marlinspike Foundations

The Marlinspike design just described was informed by both theory and previous interactive drama systems.

Theoretical Foundations

Marlinspike is based on the neo-Aristotelian poetics discussed previously, which specifies that an interactive drama can be defined in terms of multiple levels. Its highest level—Action—represents the significant events of the story. These events materially rely upon the underlying level of the story world—comprised of Characters and Setting—while at the same time also formally specifying that story world's nature.



Figure 11: Partial poetics model of story

Marlinspike's architecture explicitly follows this model. It provides a virtual world filled with non-player characters (NPCs), locations, and props, all of which have internal state. As suggested by both Aristotle (1961) and Freytag (1895), the

specific behaviors of these NPCs within scenes depend on the state of their internal thoughts and motivations. Interactions within the virtual world materially produce the story events, or Action. The Marlinspike drama manager explicitly models the events of this Action level and then responds to influence its direction. The effects of previous events formally shape the state of the virtual world.

As a character in the story world, the player interacts through verbs and deeds. The alternative—letting players specify actions and events directly—would make the user more of a director than an actor in the drama. It is hoped that using verbs will increase the player's sense of immersive presence in the virtual world and of playing a character within the story, in accordance with the definition of an interactive drama. It also allows multiple possible ways to achieve the same story-level action. For example, the player might RESCUE(princess) by breaking down her door, picking the lock, or magically teleporting into her cell. This is somewhat similar to how Propp's different functions can be filled by the same event (Propp 1968). It also recalls Chatman's point that the meaning of an event cannot be characterized independently of the entire story context in which it occurs (1978).

Marlinspike scenes are also partly inspired by the *encounters* of tabletop roleplaying games (*Dungeons & Dragons* 2003). An encounter is an event structured in if/then terms: "If the players do this, then this will happen." Sometimes this is can be as simple as "if the players enter this room, this monster will attack them." Marlinspike's scene preconditions work much the same way. Then, just as all encounters are linked together to form an *adventure*, Marlinspike's scenes are strung together as threads.

These threads, when taken together, aim to meet the poetics definition of a story being unified and complete, such that no part can be removed without leaving the whole "disjointed and disturbed" (Aristotle 1961). Each thread is a series of events unified by necessity. Yet, making allowances for the fact that some events may be more important or key to a story—as suggested by Freytag's "ornamental" events and Chatman's distinction between kernel and satellite events—only actions of high import are required to be incorporated into a thread.¹⁵ Marlinspike stories are complete when they start with a *beginning* scene, followed by a number of connected *middle* scenes, and then an *ending* scene.

The way Marlinspike selects a next scene that builds on past events is inspired by Keith Johnstone's (1979) work in improvisational theatre. Specifically, he claims that if one focuses on the structure produced by reincorporating previous

¹⁵ As mentioned earlier, whether Marlinspike will be successful in pulling all threads together into a single unified thread depends heavily on its supply of scenes as provided by the implementing game. These scenes must be capable of sufficiently reincorporating earlier actions and scenes.

events, the content of stories tends to take care of itself. I am suggesting here that this is because reincorporating—referring back from the current event to earlier events—while improvising a story means that the events of the story will be linked by narrative necessity when considering the finished story from beginning to end.

Previous Work

The design of Marlinspike also owes much to certain existing interactive drama systems.

Marlinspike's actions and events are much like Chris Crawford's verbs and events as used in his Erasmatron system (Crawford 2004). However, unlike Crawford, I make the distinction between world-level and story-level events. Instead, Crawford only concerns himself with the story-level. (For instance, his verbs take only characters as subject or direct object.) Additionally, any character can be the subject of Crawford's verbs, while all Marlinspike deeds are playercentric. Finally, Crawford's system is largely decentralized, with the rules of system (specifically, inclination formulae) determining how characters respond to the player's actions. As such, his system has no significant model of the plot beyond an event history and certain global variables.

The process of stringing pre-authored scenes together at run-time based on scene preconditions has also been used, among others, by Grasbon and Braun in GEIST (Grasbon & Braun 2001; Spierling et al. 2002) and by Mateas and Stern in *Facade* (Mateas 2002). Neither approach includes user actions as atoms in the story model, however. Instead, user actions within a scene (or within a *beat*, as Mateas and Stern call their story atoms) simply determine the outcome of that scene.

Marlinspike uses a much more flexible story model than either Grasbon and Braun's Propp-based morphological approach or Mateas and Stern's tensionbased story arc model. While this hopefully means a system more adaptive to user action, it also means that Marlinspike dramas may not be as well-structured in terms of rising and falling tension.

Fairclough's (2005) OPIATE system includes a similar dynamic casting of characters into roles based on affinity for the player's character. However, OPIATE uses case-based reasoning to build complete storylines at once. If the player fails three times to cooperate with a storyline's required choices, that storyline would be put on hold and a new one generated. Instead, Marlinspike strives to incorporate any important user actions into a single (albeit multi-threaded) story-line generated piece-by-piece.

Building on my own previous work with a scene-based approach (Tomaszewski & Binsted 2007), the Marlinspike design includes both scenes and user-entered actions as equivalent story atoms. Additionally, in the process of reincorporating past actions when selecting the next scene, it is possible to reveal to the player exactly how their actions are impacting the story. For example, an NPC may refer to the previous event that resulted in their assumption of a role that is now relevant to the current scene. This is intended to offer the player a greater sense of story-level agency than existing systems.

Another step towards greater user agency is that Marlinspike always gives the player a chance to act before starting the next scene. This is meant to avoid long strings of non-interactive scenes, a problem that plagued an earlier design (Tomaszewski & Binsted 2007).

Conclusion

Marlinspike's design strives to better merge the strong story control of scenebased approaches with the high user agency of Crawford's verb-based approach. In this system, the player interacts as a character at the story-world level of verbs. The translation from these verbs to story-level actions can then be affected or overridden by the story context established by earlier story events. Marlinspike responds to user actions by selecting the next pre-authored scene that will further the story. The primary feature of Marlinspike is that, beyond just finding the next scene it can play, it strives to play scenes that actually make earlier user actions integral and necessary to the resulting plot. Thus, the system offers the user a wide range of possible actions at any point in the story, but also works to then incorporate those actions into its own model of the story.

CHAPTER IV. PROTOTYPE GAME: DEMETER

To demonstrate and evaluate Marlinspike in practice, I wrote a simple prototype game named *Demeter: Blood in the Sky*. Set in an alternate 1923, seven wealthy passengers—one of whom is the player's character—are enjoying a pleasant trans-Atlantic flight aboard the *Demeter*, a Zeppelin-class airship. The story begins on the dawn of the third day when the dying captain awakens the passengers. Speaking through the intercom system, he warns them with his dying breath that the crew has all been brutally slain! Told to lock themselves in the passenger gondola, the passengers must decide what to do as the *Demeter* drifts onward over the cold Atlantic below.

The *Demeter* game was designed to take about thirty minutes to play through for the first time.

Game Engine

The Marlinspike architecture could work in a mouse-based, graphical environment. It requires only that the world (as defined by the game engine) report all deeds performed by the player in that world to the drama manager. The drama manager also needs the ability to manipulate the characters and objects in the world in order to produce the content of scenes.

However, in order to simplify the implementation of the prototype game, I chose to use a text-based approach for *Demeter*. Specifically, I used the interactive fiction system Inform (2010) to define the story world objects, to process user input, and to display output. Therefore, the experience of playing *Demeter* is much like that of other interactive fiction games: the player types in commands in natural language and the system responds with text describing the results of her actions.

Demeter.ulx - Glulxe 0.4.3

Dining room

Dining room

This is the main meeting area for passengers. A large dining table dominates the area, surrounded by a number of sturdy chairs. A couch stretches across the far end of the room; it faces out the huge window that forms the front of the passenger gondola. A hallway [E] leads back to the passenger rooms.

You can see Count Dragos Vladescu, Mrs Irene Winters, Major Jonathan Winters, Ms Hannah Evenworth, Miss Miriam Vanderley, Mr Elijah Roman, a dining table (on which is a large knife), some chairs and a couch here.

> kiss Miriam

You press your lips to Miss Miriam Vanderley's.

Miss Miriam Vanderley glances towards Ms Hannah Evenworth. Miss Miriam Vanderley pulls away from you. VANDERLEY: "That's sweet, but I'm afraid I'm not available for a romantic relationship right now."

ROMAN: "I think that, right now, there are more pressing matters that concern us all."

ROMAN: "Something dangerous is happening. Obviously we need to face the situation and deal with it. But I think we should be careful and not just go bumbling around in unfamiliar territory without a little more information first. Still, I think bold action will see us through this successfully."

> talk to Roman

Figure 1: A screenshot of Demeter running in the Gargoyle Inform/Glulx interpreter.

To avoid potential communication problems between the Marlinspike drama manager and the story world, Marlinspike was also implemented in Inform. As mentioned in the previous chapter, Marlinspike provides only a framework, leaving all details of game content to the implementing game. This means all specific world objects, characters, verbs, actions, scenes, and triggers are defined by *Demeter*. Marlinspike consists of the drama manager and the model of the resulting story. The resulting Inform program then requires a virtual machine named Glulx to run. There are a number of possible Glulx interpreters, including Gargoyle and Zag. All of these relationships are shown in the following figure.



Figure 2: Relationship of Demeter game content to Marlinspike within the Glulx virtual machine.

Story World

The events of the story are confined to the *Demeter* airship. This includes eighteen total locations or "rooms": ten in the passenger gondola and eight within the Zeppelin body. There are a number of scenery objects that the player can interact with in these locations, including doors, bunks, sinks, toilets, engines, and gas bags. There are other objects, such as furniture and corpses, that the player can move around. Finally, there is a small number of props that the user can also pick up and carry around, including a hammer, knife, chair legs, and sacks of food.

Characters

There are seven non-player characters (NPCs): six other passengers and a revenant—the undead creature that killed the crew. Each passenger NPC includes a number of variables that represent their internal state.
NPC Variable	Description
affinity	How this character feels about a particular topic, from love (+100) to hate (-100). Every character has an separate affinity for each of the other characters (including the PC and revenant) as well as for the idea of leaving the passenger gondola.
belief	The degree to which the character accepts the supernatural nature of their foe. Can vary from acceptance of the supernatural (+1), uncertainty (0), or a mental break and denial that there is any danger at all (-1).
evidence	The level of evidence this character has witnessed or has been told regarding the existence of the revenant. Values include 0 (none), 1 (Captain's message only), 2 (crew corpses seen), 3 (revenant sighted), 4 (witnessed a revenant attack), and 5 (witnessed revenant's demise).
health	Reduced from 3 (healthy) as the character is injured. At 0, the character falls unconscious. At -1, the character dies.
morality	A scale that indicates this character's altruism (+100) verses total self-interest (-100).
physical	This character's physical abilities, such as strength, dexterity, fortitude, and combat prowess (+1, 0, or -1). This affects how much damage they deal when attacking another and how much damage they might take in return.
propriety	How conscious this character is of social convention and manners. Mostly affects how they address others, such as whether they use only first names, etc (+1, 0, or -1).
relationship	Some NPCs have pre-existing relationships. Such a relationship will influence how this NPC reacts to the other NPC in the relationship in some situations, rather than reacting solely based on current affinity levels.
will	This character's willpower, courage, or mental focus (+1, 0, or -1). This variable affects how a character responds to increased evidence. Strong-willed characters become more determined while weak-willed characters will become more fearful of the revenant and of leaving the passenger gondola. These changes are reflected in changed affinities and belief values.

Each NPC also has a preferred plan. There are nine possible plans selected based on the NPC's current affinity for the revenant and for leaving the passenger gondola. The plans include: facing the revenant, escaping the Zeppelin altogether, hunting the revenant, self-preservation, exploring cautiously for supplies and more information, gathering weapons, doing nothing, defending the passenger gondola, or luring the revenant into a trap in the passenger gondola.

The execution of these plans may then be modified by other NPC states. For example, an NPC with a plan to face the revenant believes the revenant is too powerful to be killed but that it should still be sought. For an NPC of high morality, this will manifest as a sense of brave futility. On the other hand, an NPC of low morality with this plan will seek the revenant in hopes of joining it or somehow negotiating their own survival (possibly at a cost to others). Similarly, if an NPC has a plan to do nothing, her current belief state will determine what that NPC says while suggesting the futility of all other plans.

These plans are not complex models. That is, they are *not* a series of steps towards an objective. Rather, they are simply flags used to customize the content of certain scenes. For example, if an NPC with a plan to gather weapons is in the same location as an available weapon, the Pursues_Plan scene could play to have that NPC pick up the weapon.

Aside from those states listed in the table above, NPCs have other states that are less story-relevant. For example, each NPC has a first and last name, a current location, and might be carrying an item. NPCs also have a limited number of individual behaviors, such as how they respond to some questions.

Overall, NPCs are fairly simple in *Demeter*. This is because the focus of this prototype game was on the evaluation of the effect of the Marlinspike drama manager on story structure rather than on the presentation of believable characters.

Verbs

The standard Inform library provides a number of pre-defined verbs. A game author can then define more. *Demeter* uses twenty-seven different verbs, most of which are standard to Inform.

Table 2. Verbs used in Demete

Verbs			
Ask Attack Close Drop Eat Enter Examine	Give Go Insert Kill Kiss Knock Lock	Look Open Push PushDir PutOn Rape Take	Talk Tell Touch Unlock Use Wait

These are the only verbs that produce deeds that are reported to the Marlinspike drama manager. Because there are additional verbs defined in the Inform library, the game will recognize additional verbs. However, these extra verbs will produce only a bland response and the world will remain unaffected. For example, if a player tries to sleep, the Inform response is "You aren't feeling especially drowsy." If the player types jump, the reply is "You jump on the spot, fruitlessly."

In addition, Demeter defines some verbs that simply map to those in the list above. For example, Excrete will produce a Use(toilet) deed if a toilet is present, or else gives the message "Your socialization is just too strong to do that here. You'll have to find a toilet first." (This verb is important in the pre-game tutorial, described below.)

Similarly, the Talk verb provides a menu of options that produce valid Ask and Tell deeds. I used a menu for conversations in the hopes of increasing the affordances of the system. Without it, players would have to guess the appropriate keywords for supported conversation topics while having no idea which topics or even how many topics are supported. The required "ask *npc* about *topic*" syntax is not obvious either. However, it is still possible to use the Ask and Tell verbs directly without going through the Talk menu for those players familiar with other interactive fiction games.

Finally, all verbs have multiple synonyms. For example, the commands kiss, hug, and embrace all map to the same Kiss verb.

Actions

The player character's world-level deeds are then represented at the story-level using actions. *Demeter* has twenty-six such actions.

Action	Description	Import
ACQUIRE	To take or otherwise come into possession of an item.	4
AFFRONT	To make another character think less of you. (DEFY, HINDER, OFFEND, ASSAULT, BATTERY, and HARASS are also affronting actions.)	
ASSAULT	To physically injure, sexually harass, or significantly threaten someone (such as with a weapon).	6
ASSIST	To help another, such as by providing them with valuable resources.	4
ATTEMPT	To try to perform an action, but to be interrupted in some way so that it does not complete.	3
BATTERYa	The height of physical attacks and depravity: murder, attempted murder, etc.	8
CHANGE_STATE	To significantly change the details of an existing story state.	0
CONVERSE	To ask, tell, or otherwise verbally interact with another character.	3
DAMAGE	To disable, break, or destroy an inanimate object.	4
DEFY	To violate or oppose some NPC's previously expressed interdiction, request, or desire.	5
DELAY	To be inactive for a period of time, generally with the purpose of allowing more exciting events to unfold.	3
END_STATE	To end an existing story state.	0
ENDEAR	To make another character think more highly of you. (ASSIST and ROMANCE are also endearing actions.)	3
EXAMINE	To look at or examine a physical object in a non- invasive manner.	1

Table 3. Actions used in Demeter

a This is the only action name that is clearly a noun where all the others names are verbs. (*Verb* as in part-of-speech, not world-level-behavior.) I originally thought of this action in terms of "assault and battery". The more correct form—BATTER—made me think of raw cupcakes. So this action is still called BATTERY in the code, and so I am representing that accordingly here.

Action	Description	Import
HARASS	Unwanted physical contact, significant taunting, or threatening that does no physical damage or serious lasting psychological harm.	3
HINDER	To obstruct another's goals.	4
INTERACT	To communicate in some neutral way other than verbal conversation.	3
LOSE	To lose possession of an ACQUIREd item, such as when dropping it or giving it away.	2
MANIPULATE	To physically interact with an inanimate object in a normal manner, such as by picking it up, putting it down, turning it on, using it, etc.	2
OFFEND	To insult, argue vehemently against, order around in an insulting or abrasive manner, etc.	3
OPPOSE	To express or demonstrate opposition to someone regarding another NPC or plan.	3
ROMANCE	To woo someone, seeking their favor.	5
START_STATE	To start a new story state.	0
SUPPORT	To express or demonstrate support to someone for another NPC or plan.	3
TRAVEL	To move from place to place or otherwise change physical orientation.	2
USE	To use or apply some possessed or otherwise handy item to some story-significant purpose or end.	2

In accordance with the Marlinspike design, actions do not change the state of world objects; that is the job of verbs. However, actions can change the states of NPCs. Indeed, most of the *Demeter* actions affect NPC affinities. The ENDEAR and AFFRONT groups of actions directly change the feelings of the recipient for the actor. The HARASS, ASSAULT, and BATTERY actions can additionally affect the affinities of witnesses.

The SUPPORT and OPPOSE actions are more subtle. Many different events in the story can be recast as supporting or opposing a plan or individual. Witnesses of a SUPPORT action then change their own opinions of either the supporter or the

supported topic depending on how they feel about both. For example, if a witness has a high affinity for the supporter of some topic but is rather ambivalent about the topic itself, the witness will be encouraged to favor the supported topic a little more. On the other hand, if an actor that the witness doesn't feel strongly about supports a topic that the witness despises, then the witness will think less of the supporter. OPPOSE rules works similarly but in reverse.

Scenes

Demeter has twenty-six possible scenes plus two *scene components*. Scene components were used in authoring *Demeter* to break some of the more complicated scenes into more manageable pieces. They cannot play on their own as scenes; instead, they are only played as part of certain other scenes. Scene output varies from a line or two of NPC conversation to three screens of text. However, the average scene produces one or two paragraphs of text.

Scene	Description
A_Long_Night	A fast-forward through the night, during which a surviving revenant will attempt to assault the passenger gondola.
Awaiting_GoParty	What happens among the stay party (when the PC stays) in the absence of the GoParty.
Captains_Message	The beginning of the story, where the captain issues his warning about something being aboard the ship.
Discussion_Concludes	An NPC summarizes the discussion and thus establishes the GoParty story state.
Discussion_Continues	A random NPC who has not already contributed to the discussion throws in her two cents.
Discussion_Curtailed	An NPC curtails a PC-started discussion during the GoParty stage.
Discussion_Interrupted	Played when a discussion state ends before a Discussion_Concludes can play.

Table 4.	Scenes	and	scene	components*	in	Demeter.

Scene	Description
Discussion_Offscreen	A quick discussion when PC is offscreen in order to form the first GoParty.
Discussion_Query	An NPC asks the player for her opinion regarding the current discussion.
Discussion_Starts	An NPC, chosen at random, opens a discussion about what the best plan of action would be.
Evidence_Revealed	An NPC witnesses evidence of the revenant or that the command gondola is irrevocably sealed.
GoParty_Departs	Marks the initial departure of the GoParty from the passenger gondola.
GoParty_Eviction	The PC gets kicked out of the GoParty for being violent to other NPCs.
GoParty_Moves_Along	Moves the GoParty to a new location.
GoParty_Offscreen	The GoParty moves along offscreen, revealing various evidence and possibly encountering the revenant (if possible in that part of the airship).
GoParty_Reentry	The GoParty and/or PC needs someone in the passenger gondola to unlock the hatch. Those in the gondola may refuse the request, thus betraying the GoParty.
GoParty_Reports*	As part of GoParty_Returns, a member of the GoParty reports what it saw since GoParty_Departs.
GoParty_Requests_Follow	If a GoParty leaves the PC behind, it will request the PC to follow along on the following turn (or as soon as it can play after that).
GoParty_Returns	Marks the return of GoParty through the hatch into the passenger gondola.
Impromptu_GoParty	A GoParty that forms quickly based on action rather than through discussion.

 Table 4. (Continued) Scenes and scene components* in Demeter.

Scene	Description
Landfall	The Zeppelin reaches the US shoreline, which ends the story.
NPC_Proposes_Plan*	A discussion scene component in which an NPC will lobby for a certain course of action to everyone present.
NPCs_React	A special middle scene that handles all the possible reactions that various bystander NPCs might care to perform this turn.
PC_Dies	The PC dies from injuries, a Zeppelin crash, or leaping from the Observation Deck. This ends the story.
PC_Unconscious	The PC succumbs to injuries that prove non-fatal. The PC will wake up in various locations depending on the situation.
Pursues_Plan	An NPC attempts to fulfill one of the nine possible plans.
Revenant_Acts	Depending on various factors, the revenant may not act, be foreshadowed, enter the location, attack, or flee.
Waiting_through_the_Day	The day passes because no one can think of anything else to do.

Table 4. (Continued) Scenes and scene components* in Demeter.

Demeter has only one beginning scene (Captains_Message) and two ending scenes (Landfall or PC_Dies). The story ends as soon as either of these two ending scenes can play.

Demeter uses only two story state triggers. One marks that a discussion is currently under way. The other marks that an exploration "GoParty" is currently active, as well as tracking which characters are currently in that party.

Due to the use of these two story states to group scenes and due to the preconditions of the various scenes, a general story structure of *Discussion - Exploration - Waiting* tends to emerge in *Demeter* stories. However, depending on the actions of the player, some of these story stages can be skipped in any particular story. For example, if the PC immediately tries to leave the passenger

gondola, he can circumvent all discussion by the NPCs. Similarly, exploration can be skipped if neither the PC nor any NPCs are motivated to go. Or exploration can happen offscreen in a scene or two if the PC does not join or follow the exploration party. Finally, the passage of time through the day and final night of the trans-Atlantic flight can be quickly summarized in the ending Landfall scene if the revenant is slain before this point in the story.

So, similar to Propp's (1968) folktale functions, higher order "functions" do emerge over multiple *Demeter* sessions. However, this higher level structure is only implicitly defined within the system itself. It is also very flexible, since any single part of this structure is generally optional. That is, except for the beginning Captains_Message scene, it is possible to play the game in such a way as to avoid any one particular scene.

Even when the same scene plays in different stories, the content can vary significantly depending on the world state and story context that exist at the time the scene runs. For example, PC_Unconscious plays as soon as the PC has sustained significant injuries. The most recent injury can come from either the revenant or another passenger, and this determines how the scene plays.

If the PC's injury came from the revenant and another passenger NPC is present at the time, the NPC might rescue the PC. Whether the NPC is motivated to rescue the PC depends on both the NPC's morality and her affinity for the PC. The rescuing NPC may also need to be healthy enough to survive an attack from the revenant while performing the rescue. If rescued, the PC will wake in his bunk and the rescuing NPC will briefly summarize what just happened. If the PC is not rescued from the revenant—either because no other passenger is present, survives the revenant, or cares to perform a rescue—this scene becomes PC_Dies as the revenant moves in for the kill.

On the other hand, if the PC falls unconscious due to an injury from another passenger while in the passenger gondola, the PC will wake up disarmed and locked out on the Observation Deck. If the PC was instead up in the body of the Zeppelin at the time, he may be left where he fell or he may be dragged back to the Observation Deck depending on the motivation of any present NPCs to rescue the PC from potential danger.

This is a fairly typical example of the internal complexity and variation possible from a single scene in *Demeter*. As further illustration, the complete Inform code for this scene is available in Appendix A.

NPCs_React is a special scene that deserves explanation. As discussed in more detail in the following chapter, I discovered that NPCs must respond to certain events immediately. The current Marlinspike design does not accommodate this well, and so I wrote a special scene to handle any NPC reactions to the current

action event. NPCs_React can play as a scene in its own right, but it is also used as a component at the start of nearly every other scene. When it plays, NPCs_React can play a combination of one or more of the following thirteen reactions (and one reaction component).

Reaction	Description
NPC_Annoyed	The given NPC expresses displeasure or annoyance.
NPC_Defends	The given NPC reacts to being physically attacked.
NPC_Defends_Other	Similar to NPC_Defends, but does so on behalf of another NPC.
NPC_Defied	The given NPC expresses disappointment that the actress did something that the NPC explicitly asked her not to, usually in an NPC_Interdicts.
NPC_Interdicts*	A reaction component in which an NPC forbids another character from performing some specific future action.
NPC_Observes_Destruction	The NPC observes a successful attack on an object.
NPC_Offered_Item_By	In reply to an ASSIST[Give], the NPC either accepts or refuses the offered item.
NPC_Outraged	The given NPC expresses either incredulity, great rage (by screaming, cursing, etc), pushes, or attacks the offending character, depending on affinity for the affronting character.
NPC_Rebuffs	The given NPC expresses disinterest in a physical/sexual advance.
NPC_Replies	The given NPC replies to a neutral statement made by another character.

Tahle 5	Reactions and	d one reaction	o component*	in Demeter
Table 5.	Reactions and	u one reaction	i component	III Demeter.

Reaction	Description
NPC_Replies_To_Plan_Action	The given NPC expresses support or opposition for a non-verbal action made by another character because the action supports a plan the NPC favors or opposes.
NPC_Replies_To_Opinion	The given NPC replies (possibly non-verbally) to a verbal statement of opinion.
NPC_Wooed	An NPC replies to a pleasing kiss, touch, gesture, etc.
Revenant_Attacks	Trumps normal NPC reactions for the revenant. The revenant's only reaction is to attack.

Table 5. (Continued) Reactions and one reaction component* in Demeter

Because *Demeter* was designed to evaluate Marlinspike's reincorporation feature, *Demeter* scenes are written to respect whether Marlinspike's reincorporation feature is currently on or off. When it is off, *Demeter* scenes do not include any of their optional *hooks* when reporting what events they would reincorporate if played. They also do not include any reference to these hook events when they play. For a full list of all the specific scene changes that result from turning off reincorporation, see Appendix B.

Tutorial

The *Demeter* also includes a pre-game tutorial scenario. In this scenario, the PC wakes up in the middle of the night and has to use the restroom. Through a series of in-game prompts and instructions, the tutorial teaches the player the most essential verbs, how to navigate from room-to-room, and how to interact with objects and other characters.

The tutorial uses the Marlinspike drama manager in the same way as the main story. The tutorial is comprised of eighteen scenes. Most of these scenes are very simple: they just prod the user in a certain direction or provide a couple paragraphs of instruction. However, one scene—Miriam_Acts—that controls how an NPC responds to what the PC has done, has over thirty possible permutations.

Conclusion

This completes a brief overview of *Demeter: Blood in the Sky*. The full game can be played online (Tomaszewski 2011). Also available at this online location is the full documentation and source code for both the Marlinspike and *Demeter* implementations.

CHAPTER V. IMPLEMENTATION: LESSONS FROM THE TRENCHES

I originally planned to spend three to six months implementing the prototype version of the Marlinspike interactive drama system and the short text-based game, *Demeter*, that used it. In the end, this implementation took twenty-six months, from November 2008 to January 2011. It resulted in approximately 23,000 lines of Inform code and 9,000 lines of documentation. Only about 3,500 lines of code (15%) and 2,500 lines of documentation (28%) went into Marlinspike; the remainder went into Demeter.

As an example of "expressive AI" (Mateas 2002), this dissertation work as a whole involves theory-building, engineering, art, and scientific evaluation. The theory-building in Chapters II and III propose the means by which a reincorporation-based approach can produce a story structure that conforms to a neo-Aristotelian poetics while still offering the player a sense of user agency. Chapters VI and VII describe an empirical evaluation of these theoretical claims.

But, for the two years it took to implement Marlinspike and *Demeter*, I set science aside. Instead, it was an endeavor of both engineering and art. As an engineer, I saw the construction of my theoretical design into a concrete working system. As an artist, I then authored a particular user experience using that system. This final user experience arises from the descriptions of the virtual world, the nature of its characters, the tone of the narration, the specific user actions supported, the range of potential stories, and the choices made by the player that produce a single concrete narrative within that space.

One of the joys of such craftsmanship is that, as a creator, I can directly determine if my work is successful. As an engineer, I can test whether the system performs as I intended. As an artist, I can step back, tilt my head to one side, and see if the work before me mirrors my original vision of it. I don't need an audience or a controlled study to tell me whether I'm meeting my personal design goals.

This chapter explores the personal lessons I learned from the implementation itself: the stumbling blocks along the way, the compromises made, and the disparities between what I'd imagined and what I actually produced.

Inform as a Game Engine

Initial Rationale

I decided to use the interactive fiction system Inform (2010) to implement the story world. At the time, this seemed like a good decision for a number of reasons.

It is very easy to define virtual world objects in Inform using its boiler-plate world model. Its libraries already include a number of default verb definitions to interact with those world objects. Inform also includes an input parser to handle text input from the user. This meant I would not have to deal with any natural language processing myself. All of these Inform features can easily be customized or extended to support game-specific requirements.

Created in 1993, Inform has already been used by other authors to create hundreds of games. It is debugged, stable, and still has an active community. Inform games run using a virtual Z-Machine architecture that has been ported to a impressively-wide array of hardware platforms, including some hand-held devices. More generally, interactive fiction games have a thirty-year history, so it is a well-known, established game genre.

I also had some prior experience using Inform, but I have never worked with a graphics engine. So I felt that a text-based approach would be much faster and easier to develop than a graphical one, especially for a simple prototype game. In short, interactive fiction seemed like a good place to start for interactive narrative.

Consequences

This decision to Inform as a game engine then had a number of consequences.

Since this implementation was a prototype, I wasn't certain yet to what extent I could separate the implementation of the story level—which includes the drama manager, scenes, and internal character states—from the implementation of the world level. If I separated them, they would effectively become two distinct programs. Then the drama manager would not have direct access to the state of world objects. Instead, the drama manager would have to send queries across some sort of bridge between the two programs. But it would not be technically easy to connect such a bridge to the Z-Machine containing the world implementation. This may have been workable in a controlled environment, but I also wanted users to be able to play the game online from their own computers. So, instead of using some sort of communication bridge between two separate programs, I decided it would be safer to implement the story level using Inform as well.

Inform's Simplicity

Inform is a very simple, low-level, C-like programming language. It has no dynamic memory allocation, so an author must manually specify the maximum number of each type of object he might use during the course of the game. This restriction forced minor changes in the design of the event and action data structures. Also, because Marlinspike may potentially require a high number of event objects, the *Demeter* game exceeded the memory limitations of the Z-Machine architecture. Thankfully, it is possible to compile Inform games to an alternate virtual machine architecture named Glulx (Plotkin 2011). However, this switch to Glulx limited the choice of possible interpreters I could use to deliver the game online.

There is no garbage collection in Inform, so object use must be very carefully managed to prevent memory leaks or bugs resulting from accidentally reusing the same object in memory from two different code locations.

Inform has no advanced or dynamic data structures, such as hashes or lists. So, before even starting on Marlinspike, the first task I undertook was writing an array-based linked list implementation. Inform does not have floating point numbers either, which is why character affinities were written to vary from -100 to 100 rather than from -1.0 to 1.0.

It is also not possible to easily build or change strings of characters in memory in Inform. This means any output cannot be assembled and then manipulated before being printed. Instead, all output must be printed directly in linear order. Because of this, a fair amount of work went into handling simple things like printing "I" or "We" when a speaker is part of a group or adding an "s" to verbs depending on the number of character involved.

Listing 4: Example of Inform code used to report that the members of the current GoParty enter the room.

```
DST_GoParty.detail.prettyPrint();
print " enter";
if (DST_GoParty.detail.size() == 1) {
    print "s";
}
print " from ";
...
```

This makes narration very tedious to produce. It is not uncommon for a sentence of scene narration to take 5 to 10 lines to produce, and, occasionally, even more than that.

Another string-related headache was controlling the number of blank lines between paragraphs. Because the output of a scene can vary so much, certain paragraphs of a scene's text may or not print based on the current world and story contexts. Accurately printing one and only one line-break between each of a series of optional paragraphs is trickier than it first sounds, especially when any duplicate blank lines cannot be removed after assembling the output but before printing it.

Related to output, Inform will not display any of the output printed during a turn until the end of that turn. This means that if a bug causes an infinite loop at any point during the turn, there is no output displayed at all. Instead, the game simply hangs. The harried author then knows that an infinite loop was just activated somewhere within 23,000 lines of code but has no output to indicate where. And so the author must iteratively repeat the situation that caused the lockup while logically paring down the code that executed that turn until the bug is eventually located.

There exist a number of third-party extensions that do overcome some of these Inform shortcomings. However, as an author, I was often hesitant to use them. There is always some question as to how tested and supported these extra libraries are. If an extension fails, then I would be forced to puzzle through its code to debug it. Frequently, rather than risking that headache, I chose to do without the extension or write a quick work-around myself.

This worry is not unfounded. I used parts of the ORLibrary set of extensions (Fisher 2011), most notably to produce the menu of conversation options displayed when the PC talks to another character. The ORLibrary's menu code needed to be tweaked a couple times to run properly in different interpreters, such as Quixe (Plotkin 2011). The Zag interpreter (Zeppieri 2005) also had a very small bug that would cause lockups when the talk menu was used.

I do not mean to disparage the work of these fine programmers! Every piece of software has bugs that must be worked out. I am grateful for their work and the fact that their code was freely available so I could tweak it to meet my needs. My point is simply that the existence of extensions is not an automatic time-saver. Also, when libraries are part of the core language, they are generally more likely to be debugged and maintained than those maintained by third parties.

Yet admittedly, even if Inform did have an extensive core library of general data structures—like variable-length lists—Marlinspike would still need certain custom variations of these. For example, Marlinspike would greatly benefit from a robust searching mechanism to find matching objects in a list. Scenes need this functionality to easily find events that match their preconditions and hooks within the event history list. It is also frequently necessary to grab all NPCs that meet a certain criteria, such as those NPCs that are currently conscious, in a certain location, armed, and/or having a certain affinity or plan preference.

In writing Marlinspike and *Demeter*, this was another constant tension: whether to spend time implementing better custom tools and libraries or to just to press on without them. Given that this was meant to be a quick prototype and that later implementations would not be written in Inform, I generally tended towards the latter option. That said, I did write a number of useful helper methods for common tasks, though these tended to be rather ad hoc.

In short, using Inform meant starting with a simple language that required a lot of attention be diverted to nitty-gritty technical details. Then any advanced programming features I wanted had to be built from the ground up.

Interactive Fiction Affordances

Aside from impacting the implementation of Marlinspike, choosing Inform as a game engine also determined the game's interface. Since the user perceives the Inform-defined objects through a textual description of the world and then enters commands handled by Inform's input parser, Inform determined the medium and constrained the manner of the finished game. Obviously, using Inform meant that *Demeter* would be an interactive fiction (IF) game.

Although interactive fiction has a 30+ year history, it is no longer widely popular today. Even the general command line mode of interacting with a computer through entering typed input and reading text responses is becoming increasingly rare. This means interactive fiction may be completely foreign to modern players.

Yet this shouldn't be a problem if the interface is still fairly easy to use. After all, every computer user usually knows how to read and type. However, in my own experience and based on watching a few people new to IF struggle through alpha versions of *Demeter*, the affordances of the interactive fiction medium are far from ideal.

The first step for the player is making sense of the world. Although each room is described in text, it is frequently hard to know which objects described there actually support interactions. Even when the exits of each room are explicitly listed in the room's description, it still requires a fair degree of visualization and memory to build a cognitive map of the virtual space. Simply examining an object for more information about it requires an explicit command on the part of the user, such as look at chair.

Once the player determines what is present in the world, then the next step is interacting with it. The player must know what verbs are possible in general. Traditional interactive fiction games generally contained a lot of puzzles for the player to solve. In this context, part of the challenge of the game is determining what is even possible. Because each verb can have multiple synonyms, it is usually possible to produce a valid command after a few naive attempts.

However, in the context of an interactive drama, where I want world-level interactions to be quick and simple so that players can focus on the story-level effects, this trial-and-error approach to learning verbs becomes a distracting and frustrating hurdle.

Some players also assume that, if they see a word in the game's output, then that word should be valid input. For example, if they read that an NPC just smiled at them, they assume that they can smile in return. However, in Inform, the output displayed is separate from the input accepted by the parser. It is up to the game author to make sure that every noun and adjective used to describe a world object in all of its possible states can also be used by the user to refer to that object in input. Adding support for new verbs is no small task either, since their grammar and their effects on various objects must then be defined. Frequently, it is easier as an author to not support a verb than to go through this work for the limited return of supporting a verb few players are likely to use anyway.

To compound this problem of forming valid input, not all verb-object combinations are valid. That is, even if the user has found a valid verb and is correctly referring to a world object, the resulting deed may still not be valid. For example, if an author added a Smile verb, it would still probably not be useful to smile at a door or a table. But refusal text must still authored to handle these situations.

So there are many obstacles to producing valid input in IF: recognizing which objects can be interacted with, determining how to refer to those objects in the way the author intended, knowing which verbs those objects support, and then combining object and verb into a syntactically valid command. In *Demeter*, there is a further disconnect here since even a valid deed in the world may not be reported to the Marlinspike drama manager. As discussed previously, this is because the Inform library provides bland results for a number of verbs—such as Jump and Sleep—that the Demeter game does not in fact use. Thus, the user may even produce valid deeds that still do not affect the world or the story.

Most of these input affordances could be overcome by doing away with the command line and using a menu-based approach instead. For example, the names of world objects that support interactions could be highlighted in the description of a room. Clicking on these objects would then bring up a menu of verbs that that object supports. Using this approach, it would be obvious exactly which objects can be interacted with and what actions are possible with each. It would also be impossible for the user to enter an unparsable command or invalid deed.

This menu-based approach is not revolutionary. Chris Crawford (2004) has recommended the use of a graphical inverse parser that uses contextual menus to help users build valid commands. Emily Short, an active interactive fiction

author, has lamented the problems arising from command line interface and suggested essentially the same possible solution (2010). Graphical games, such as *Neverwinter Nights* (2002) and the *The Sims* (Wright 2000), have long used radial menus to display what actions are possible to perform on an object.

However, such a major overhauling of Inform's input system was beyond the scope of this implementation. I did at least use menus for conversations in order to make it clear what conversation topics were supported by the game.

Another issue stemming from the use of Inform is its turn-based structure. With its proactive NPCs, an interactive drama can support the sensation of being caught up and carried along in a story, being able to influence it while not completely controlling it. I had exactly this experience in playing Mateas and Stern's *Facade*; I had not experienced this feeling in a game before. Yet, at the end of each turn in *Demeter*, the empty waiting prompt implicitly says to the player: "Do something!" This makes it difficult to support passivity as a valid option. I did try to make waiting a simple thing to do: just hitting Enter at the prompt will result in a wait command in *Demeter*. It also would have been possible to use a time limit after which the game would advance automatically. However, I did not feel this was a valid option in a text-based game given people's varying reading and typing speeds.

Outcome

Overall, using Inform was a fair choice, especially for a prototype system. Inform did a fine job at what it was designed to do. Defining world objects and verbs proved simple (though that was the simplest part of the project to begin with). Affordances of the interface aside, Inform's parser did a good job handling input. Producing reliable output was still troublesome, though, and some of Inform's other quirks made debugging a headache. Because Inform is such a simple low-level language, it was a hindrance in implementing Marlinspike and all of the scenes.

I have since learned that some of these issues could have been overcome using TADS3 (2009). However, at the time I started implementing, TADS3 was still fairly new. Also, TADS still does not have a web-based interpreter option, which was necessary to run an online evaluation study.

A text-based approach to interactive drama still seems worthwhile. This is especially true given the proliferation of hand-held mobile devices these days; text-based games could work well in this environment. Many of the IF affordance issues could be overcome with a object-based menu input system. This would even be an aid in a mobile environment where typing is more cumbersome.

Drama Manager versus the World

The Marlinspike design assumes that there are actions performed by the player character (PC) and scenes played by the drama manager (DM). This PC-centric notion of actions grew from the way Inform works, where the default actor for all deeds is the player. Inform's narration generally assumes this. Thus, in the initial design, deeds and actions would come only from the player; all other world state changes and NPC manipulations would be performed within scenes. This approach quickly became untenable.

State Management

The reason for this is that scenes became a nightmare of state management. For example, if one NPC supports a plan during the course of a scene, that scene would then be obligated to update the affinities of all listening NPCs that were affected by that proposal. This leads to duplicated code between all the scenes in which an NPC supports a plan.

Yet there already exists a SUPPORT action that correctly produces of all these same effects when the PC proposes a plan. Therefore, I very quickly broadened the design so that any character—not just the PC—could be the actor of an action event. However, NPC actions still only occur within the bounds of scenes. Besides reusing the code that produces the action's effect, an NPC's action event can also be added as a sub-event of the scene itself. This is useful in providing a more detailed system view of each internal effect of a scene.

Aside from character state management, scenes also had to update world-level states. The most challenging example of this was directing NPC motion. For example, when the GoParty departs, other NPCs in the gondola come to see them leave. Similarly, if the GoParty is moving without the PC present, it will go to certain rooms in the Zeppelin and then return. Performing this motion is very tricky, though. First, only those NPCs that are conscious and active should be moved. If any doors intervene, they need to be currently unlocked or at least unlockable from the side the NPC is on. The door must then be unlocked and opened (so that the NPC does not appear to teleport out of locked rooms), but only if necessary. Then the NPC needs to move to her intended destination. But this should be narrated differently if the NPC is leaving from the PC's location, passing through the PC's location, or arriving at the PC's location. If the NPC never passes the PC during its motion, then her behavior is completely offscreen and should not be narrated at all. If more than one NPC is arriving at a time, it reads better to list this as a single event, rather than narrating each arrival separately. In short, there is a great deal of attention required to tiny details if the author wants to present a consistent and believable world. Consistency is

important, since if one scene leaves the world in an illogical state, it can indirectly affect the correct behavior of later scenes.

A partial solution to this headache is to implement well-defined deeds for NPCs. (*Demeter* did this only partially.) This way, error-checking and execution of NPC deeds in the world can be centralized and controlled in a single code location. It would then be possible to define things like a path-finding script from a source room to a destination room in terms of these deeds. To do this, the performance of these NPC deeds should then be disconnected from their narration so that they can occur silently offscreen when necessary.

A further advantage of having NPCs perform all actions through deeds is that scenes could be authored explicitly in these terms. Every NPC deed would then be explicitly modelled in the resulting story structure as a part of that scene. These deeds could then be recast by the same mechanism that applies to player actions. This is an important next step in the development of Marlinspike in order to ease the complexity of the authorial burden. Such an approach would also open the way to system generation of scenes as scripts of deeds.

Attempted Actions

The drama manager is given a chance to deny any deed performed by the player. This was necessary for such scenes as Impromptu_GoParty and GoParty_Departs. Should the player try to leave the gondola before a GoParty has been formed, the player's Go deed is prevented in the world. This deed is then cast as an ATTEMPT instead of its normal TRAVEL action. Then one of these two scenes plays to explain why the player did not successfully leave the room: because other NPCs entered the room to either join the PC or see him off.

In practice, interruptions of the player by the drama manager are rare. And supporting them means the connection between the world and the drama manager is fairly complex. Before any deed executes in the world, the world first needs to determine whether the deed would be successful. But it must then pause before actually completing the deed to check whether the drama manager wishes to prevent it. The complexity of this approach is a major obstacle to implementing the drama manager as a separate program that communicates with the story world through only a simple software bridge. Another approach is needed—either a way to author stories with no player deed interruptions or else a way to perform those interruptions at a world level rather than through the drama manager.

Related to this, I learned that sometimes even non-interrupted failures are important actions. In the current implementation, only successful (or rather, about-to-be-successful) deeds are reported to the drama manager. This means certain actions—such as trying to open a locked door—are not reported by

default, since the world knows that the deed is going to fail (even if the user does not). But this creates a disconnect between the user's input (and thus the user's conception of the story) and the story-level representation of the story. Also, attempts are often important to the story too: In *Demeter*, attempting to open the command gondola hatch was cast as supporting a certain kind of plan and thus prompted replies from nearby NPCs—even though the underlying Open (hatch) deed was not successful. This change to report deed failures as well as successes requires a bit more complexity—the world would need to report not only the deed but also whether it is about to succeed or fail—but it would be worthwhile in the next version of Marlinspike.

Narration

As suggested in relation to some of the limitations described above, providing accurate, well-formatted, readable narration in this implementation was not particularly easy.

First of all, there is a great deal of detail modelled by the system that is not easy to narrate in text that is explicitly descriptive yet not tedious. This is particularly true when narrating character affinities and story structure connections, which I will discuss in more detail later in this chapter. As an author, this difficulty leads to a feeling of opacity: I have modelled all of these details, but many of them are not apparent to the player.

This is partly due to an artistic choice in how I narrated the game. As author, I wanted a kind of limited, rather than omniscient, narration that portrays the world as the player would experience it if actually there. This was meant to heighten the tension, as appropriate for a horror game. For example, as a player, you may hear a hatch open somewhere offscreen, but you wouldn't know, based on the sound alone, which hatch it was or who—or what!—opened it. An alternative would be to explicitly state these details: "You hear Elijah Roman open the hatch to the passenger gondola." This is certainly more explicit, but it may strain credibility if you cannot see either the hatch or Elijah Roman from your current location. (Also, since these objects are currently offscreen, you cannot refer to them in any typed commands, which leads back to the issue of affordances discussed earlier.)

This explicitness in narration could be extended to also explaining causality, such as "The revenant enters through the door you left unlocked" or "Because he wants to hunt the revenant, John Winters proposes that a search party be formed." This is a tricky balance to strike as author. Such explicitness reduces the opacity I was just lamenting, but it also violates the advice that good narration should show, not tell, the reader what is happening. There does not seem to be a clear answer to this issue. Finally, narration sometimes needs to be disconnected from the events it describes. As mentioned, frequently NPCs are operating offscreen. These offscreen deeds are often not narrated at all, but sometimes they are narrated in different ways. For example, in *Demeter*, when an offscreen NPC is attacked by the revenant, the PC will hear the NPC scream. He may also hear the fall of a body or the opening or closing of a hatch. This is quite different narration than would be given for these same deeds if the PC was present to witness them directly. Thus, if NPC deeds are used as a future means of authoring NPC behavior, the narration of those deeds will need to be separately customizable based on the location and condition of the PC.

Characters

In order to focus on the story, I kept the character models fairly simple in *Demeter*.

PC

The PC is nameless and genderless. This was to simplify and shorten the game: it was not necessary to ask the player to specify PC details or to explain the nature of a pre-authored PC to the player. Instead, players could project onto the PC whatever personality they cared to. This choice did make certain scenes difficult to author, as it is hard to refer to someone when you do not know either their name or gender.

The PC should be modelled in more detail in the future, though. Roleplaying is an important part of interactive drama. Just as in traditional roleplaying games, a player's choices when constructing a character can give strong indications as to what sort of game they want to play. Players can also provide background material—such as PC history or previous relationships—that could be used to pull that character into the current story.

NPCs

I wanted a social milieu in *Demeter*. The horror films that inspired the story are as much about the personal interactions of the characters as about the evil they face. To that end, I wanted NPCs to have conflicting goals and arguments but also to allow the PC to influence those goals and opinions.

The internal state of each NPC is modelled using less than twenty variables, even when counting an NPC's affinity for each other character as a separate variable. Since most player actions simply serve to change NPC affinities, I was not particularly constrained by this model at all. In fact, the possibilities of even this simple model were not fully utilized.

Instead, I found that I was constantly challenged to make what NPC states I did have apparent to the player. Explicitly stating certain positions—as in "I like your plan!"—can be rather jarring. But only hinting at stances—such as changing how an NPC refers to the PC based on her affinity for the PC—can be too subtle to be noticed. The long discussion sequence in *Demeter* was meant to demonstrate the range of NPC opinions, as well as their disagreements with each other. During this sequence, the NPCs even change each other's opinions, and the PC can do the same. But this sequence generally just proved to be boring especially with the more exciting threat of the revenant on hold in the meantime.

Besides the difficulty of narrating what an NPC's current state is, it is also difficult to indicate when it has been changed and to what degree. This stems largely from the problems already discussed about how best to narrate the full details of the system's internal view of the story events.

Some non-textual channel might be a better approach to this problem. In real life, we often read people's opinions based not only on what they say, but also on subtle tone, facial cues, or body language. For example, different colors or fonts could be used when an NPC speaks to the PC in order to represent different degrees of affinity for the PC. In a graphical system, a bar graph could briefly pop-up over the character's head and animate the change in position resulting from the current event. Such approaches may sound jarringly overt at first, but the player cannot be expected to engage in social manipulation without some clear indication that his actions are having an effect. In lieu of clear NPC facial expressions, tone of voice, and body language, some other means of succinctly expressing this is needed.

So one side of NPCs is making their internal state clear; the other is affording clear ways to affect them. This is another problem with *Demeter*: the interactions with NPCs are fairly limited. It is possible to give items to NPCs, but it is not possible to ask them for an item they have. (This is simply because I did not define this behavior for the NPCs.) It is possible to ask them for their opinions on various topics, but these topics are limited to other characters and their preferred plan. (This is due to the underlying NPC model, as these are the only opinions NPCs actually have.) It is possible to kiss, shove, and attack NPCs, but most players do not try these things. Of course, players must recognize all these actions are even possible—a task which suffers the problems of Inform's poor affordances discussed earlier.

Indeed, many player skip direct NPC interactions altogether, choosing instead to pursue the revenant alone instead. This is a perfectly acceptable choice, and *Demeter* does support this behavior while still making it a social action. NPCs that care to will join the PC unasked in the exploration of the Zeppelin. Also,

there are several actions that the PC can perform that will affect NPCs positively or negatively based on their opinions. For example, picking up a weapon shows SUPPORT for the plan of gathering weapons. This in turn will change the affinity of witness NPCs based on whether they support this plan too. What seems to be missing most in all of this is that NPCs' corresponding reactions are relatively simple and hollow.

This leads to the larger problem that *Demeter* NPCs have a rather "cookie-cutter" feeling to them. My focus in this implementation was on story and events, not on characters, and that choice can be felt in the finished game. But producing unique NPCs is a heavy authoring burden. For example, whenever an NPC makes a statement in *Demeter*, this text is usually randomly selected from three to five alternative ways of saying the same thing. This is so players do not see the same text repeated by the same NPC over and over again in repeated scenes. Since sometimes a scene may include a series of two or three strings of text flow properly is not always easy. After completing this much effort for one NPC, I was loath to multiply the effort by six to cover the remaining NPCs. Therefore, NPCs all pull from the same pool of possible text. But this is precisely what makes the NPCs feel like they are all the same.

One technique to help alleviate this would implementing a way to "cross-off" text that has been randomly selected for a particular response. This would prevent immediately repeated statements both within NPC and across NPCs, though text would still repeat after all options have been "crossed off" and the list reset.

Providing unique history for each NPC that the PC could explore in conversation would also help distinguish NPCs from each other. However, as mentioned, many players do not talk to NPCs. So having NPCs interject unique comments during the course of action events would be an even better approach. There is a little bit of this in *Demeter*. For example, the NPC Elijah Roman has seen a revenant before, which he mentions when first witnessing it or reporting seeing one upon returning to the passenger gondola. More of this would be good, though this just means another increase in the already high authorial burden.

So, in conclusion, I tried to build a game that included a high degree of social interaction and manipulation. However, since NPC states and state changes are not very apparent and the possible social interaction are not always obvious or fun to perform, this aspect of the game was not realized very well. Yet characters and their motivations are essential to a good narrative, just as Aristotle and Freytag pointed out centuries ago. So, even when focusing on story, believable NPCs will be necessary in a good interactive drama.

To this end, it does not seem that more complex models of NPCs are needed. Instead, better ways of portraying those states and motivations are needed. However, this is going to remain a terribly tricky problem when portraying humanlike characters. If such characters convincingly behaved as humans in a wide variety of situations, they would effectively be passing a form of the Turing test, a long-sought yet still-elusive standard of AI performance.

Scenes

Marlinspike leaves the details of scene content to the implementing game. As shown in Chapter III, a Marlinspike scene is little more than two functions: canPlay and play.

I initially considered a design in which a scene listed its preconditions as event objects to be matched. However, some scenes may have special requirements, such as requiring that a matching event occur only within the most recent three events. Additionally, some scenes have preconditions that are not event-based, such as world or character state values. Since not all world state changes have a corresponding event in the event history, this approach would not work.¹⁶ So I went with the simpler canPlay approach: Each scene is polled through a simple function call. The scene is then free to do whatever checking it needs to do within this canPlay function. This simplified Marlinspike, but left a lot of work to the game author.

Similarly, each scene is played by calling a single play method. These monolithic play methods often became very complex to author.

As far as Marlinspike is concerned, scenes are single events of no duration, much like an action. However, this is not true in practice of longer scenes, which might be comprised of multiple sub-events and even other sub-scenes. The first problem that arises from this is that any NPC action events used to represent parts of the scene's details are not executed until after the scene is complete. For example, if an NPC affronts another NPC at the beginning of the scene, the effect of that affront won't take place until after the end of the scene.

¹⁶ This is something I would like to see changed in the next version of Marlinspike. A number of actions used in *Demeter*—such as the different kinds of affronts and assaults —exist simply to indicate different degrees of change to character affinities, rather than to mark significantly different kinds of story actions in themselves. A better approach would have been to have action subclasses for different kinds of state changes, such as WorldStateChange or AffinityChange. Then each different kind of action—such DEFY or ASSAULT or INSULT—could use a separate AffinityChange sub-event to represent the resulting effect of this action. In this way, every significant change of state in the game would be discretely represented in the event history as an event object. This would simplify authoring certain scene preconditions, as it is often desirable to search for effects rather than specific actions. For example, an author may want to know what event had the greatest negative affinity impact on a particular NPC, regardless of the nature of the causing action.

Also, sometimes scenes should take longer than a single turn to perform. For example, in *Demeter*, the exploration of a GoParty should perhaps take a number of turns equal to the number of deeds required of the NPCs to actually perform the exploration. Instead, their exploration happens in a single turn. On the other hand, sometimes it is desirable to be able to skip quickly through a number of scenes. For example, if the PC loses unconsciousness and wakes up again a couple hours later, or even if they simply choose to Wait, it would be nice to be able to "fast-forward" through any GoParty exploration in single turn.

Finally, any world-state changes made within an early part of a scene can actually violate the preconditions of later parts of the scene. For example, it is possible for a violent PC to be knocked unconscious by an NPC as part of a reaction at the start of a scene. However, since the preconditions are not automatically checked again before continuing the scene, the scene continues as if the PC were still conscious and aware of the action.

All these problems result from the conflict between Marlinspike's view that a scene is a single atomic event and an implementing author's need to have more than one event occur within a scene. This is something that needs to be addressed in future versions of Marlinspike.

Reactions

A component of nearly every *Demeter* scene is NPC reactions, which brings us to handling NPC reactions in general. When the player interacts with an inanimate world object, it responds immediately. If a player opens a door, it opens. If the player passes through that open door, she arrives at the other side. If she picks up an item, the items moves into her inventory. In certain story-specific situations, it is possible for Marlinspike to interrupt such deeds before they complete. But otherwise an action in the world leads to an immediate result. In short, a player deed is atomic.

NPCs exist in the world and so are subject to player deeds just like props and other world objects. However, the results of affecting an NPC is not atomic. Instead, the player affects the NPC, and then the NPC responds as a separate event. This is because the response of an NPC is not a single, predictable success-or-failure outcome, as is the case with most objects. For example, if you slap an NPC, there is a number of possible responses: The NPC could stare at you in disbelief, or strike you back, or start to cry, or turn and run from the room. And this action may prompt other NPCs in the room to also react. For example, a different NPC may come to the victim's defense, either verbally or physically. Or perhaps the bystander NPC may even side with the PC and starts slapping the victim NPC too! But all of these reactions need to be coordinated in some way. If the bystander NPC decides to slap the victim too, but the victim flees the room before this can happen, then the bystander's selected reaction cannot be played

after all. Also, because NPC behaviors are story-level events, they need to be managed by the drama manager to ensure the best fit in the ongoing story. This is especially true in the current design, where NPCs can only respond through played scenes.

I long struggled with the problem of coordinating NPC reactions in the design of Marlinspike. I considered some sort of reaction sub-system, but it was not clear how it would best be implemented. For example, NPCs should be given a chance to react after every deed. However, sometimes those reactions are so significant that immediately segueing into another scene that is not directly related would be jarring. That is, if reactions are significant, the player should be given a chance to respond first. Also, sometimes NPCs will want to react to each other during a scene.

Since I saw no clean, obvious solution to this problem that would work for all situations in all games, I left the decision to the implementing game. *Demeter* solves this problem by using a fairly complex NPCs_React scene. As described in Chapter IV, this scene manages thirteen possible NPC reactions. It determines which NPCs will react based on who was most affected by the current action. It also determines how many reactions will be played based on the import of the current action and the import of the resulting reactions. For example, if the PC is violent to an NPC, this is a significant action that warrants a high degree of reaction from bystanders. However, if an NPC decides to attack the PC in return, this is also a significant reaction. This means any remaining NPCs will tend to react to a lesser degree, such as by calling from the sidelines for an end to the violence.

NPCs_React does a decent job of coordinating NPC reactions. It determines which reaction is most appropriate for a given NPC as well as coordinating reactions between multiple NPCs. It can be called at the start of most other scenes to provide an appropriate degree of reaction to the current event before continuing the story. It can also be called within a scene to cause NPCs to react to each other. However, implementing NPCs_React side-stepped another problem that was more serious: NPCs need to react immediately to certain actions. But the initial Marlinspike design that selected the next scene based solely on reincorporation measures did not accommodate this well.

Reincorporation as Scene Selection Criterion

The "best choice" of the next scene involves much more than just the amount of material it reincorporates. This was reaffirmed a number of times during the development of *Demeter*. The design of Marlinspike evolved slightly during this implementation in an attempt to respond to some of these deficiencies.

Failure: Immediate Reactions

The first failure of Marlinspike's reincorporation-based selection process was in failing to provide immediate NPC reactions. The original Marlinspike scene selection criteria¹⁷ included the import of the current event as a bonus if the selected scene would reincorporate it. This effectively provided a bonus to scenes that continued the current action, and that bonus would be greater if the current action were particularly significant. This was intended to handle NPC reactions, which were originally designed to play like any other scene.

However, then situations like this would arise: Imagine the PC is currently in a long conversation with a number of NPCs, and the PC suddenly slaps one of the NPCs. Suppose at least two scenes could then play: one that continues the conversation or another that has the NPC reply to the slap. If the conversation is long enough, the material reincorporated by continuing the conversation would overwhelm the bonus of immediately replying to the slap. So the slap would be completely ignored and the conversation would continue.

Yet ignoring the slap—even if it might be replied to in some way later in the story —is not a valid option! It has to be acknowledged immediately, even if only in passing. Otherwise, the NPCs become completely unbelievable. Also, the player loses a high degree of world-level agency, since many interactions with NPCs would have no apparent effect.

Thus the NPCs_React scene evolved to fill two roles. The first was to coordinate NPC reactions, as described previously. But the second was to provide immediate reactions. This was done by checking if NPCs_React can play at the beginning of nearly every scene. If it can play and is responding to a current action of high import, most scenes will then choose not to play. If NPCs_React can play but in response to an action of low import, then the scene will first play NPCs_React as a component and then proceed with its own content, possibly with a small amount of bridge material.

For example, let's reconsider the earlier "conversational slap" example. Because it now checks NPCs_React first, the scene that continues the conversation would no longer be able to play in this situation because NPCs_React is responding to a significant action. Instead, the conversation would stop while the NPCs express outrage at the PC's behavior. On the other hand, if the PC had offered some food to one of the NPCs instead of slapping him, then NPCs React would be able to

¹⁷ The current scene selection process was described in Chapter III. Briefly, each scene that can play next is assigned a score equal to the scene's imperative value plus the weight of both the total material and the unique threaded material that would be reincorporated if the scene was selected to play at this point. The scene with the highest resulting score is selected to play next. The original Marlinspike design did not include an imperative value in this selection process. Instead, it included the import of the current event if the scene were able to reincorporate it.

play in response to a low import event. In this case, the conversation-continuing scene would still be able to play. But it would first include the results of NPCs_React—having the NPC accept the PC's proffered gift—before continuing the conversation. It could even include a bridge statement of some sort, such as having one of the other NPCs mention that the PC should pay attention closer to the conversation.

It is important to note that this change improved the responsiveness to player actions by ensuring they are reincorporated immediately. Because this can often happen as part of a scene that then continues the story, this effectively pulls what the player has done into the story line. But the difference from the previous design is that this reincorporation work is now being done by the scenes themselves deferring to each other, rather than through some mechanism of Marlinspike.¹⁸ This is less than ideal.

Given this superior method of responding to user actions in a timely manner, I dropped the import of the current event from Marlinspike's scene selection algorithm.

Failure: Advancing the Story

Reactions aside, another irritation as an author was that Marlinspike would frequently fail to select the "best" next scene from those available. In short, Marlinspike was working correctly, but there was more to consider than just reincorporation.

An example from *Demeter:* Suppose that the PC is a member of the GoParty exploring the Zeppelin for signs of what happened to the crew. The PC leads the party into a new room. On the floor of this room is the bloody corpse of a crew member and a stout hammer. One of the NPCs in the party has the plan of arming himself. Three scenes are then possible at this point. One of the NPCs could exclaim at the sight of a dead body. The NPC that wants a weapon could pick up the stout hammer. Or one of the NPCs could lead the group from the room.

As an author, I feel that the NPCs should acknowledge the corpse first, then arm themselves, and finally leave. However, in an early version of *Demeter*, they would simply leave. This is because leaving continued the thread of their ongoing

¹⁸ Regarding the later evaluation of Marlinspike, this sort of intra-scene reincorporation is not disabled when the reincorporation feature of Marlinspike is turned off. Although the scenes could have been written to be disabled this way, I feared that the resulting behavior of oblivious NPCs would appear too obviously "broken." For purposes of evaluation, Marlinspike's behavior when reincorporation is switched off was meant to be comparable to other scene-based systems that do not explicitly use reincorporation as a scene-selection mechanism.

exploration—which was already fairly long. Noticing the corpse or picking up the hammer, on the other hand, didn't reincorporate any events that had gone before, other than perhaps the Captain's message at the beginning of the story. Since the GoParty exploration thread ultimately stemmed from that same starting event, the ongoing exploration thread still had greater weight.

Similar situations occur at other points in the story. Upon finding the revenant, the GoParty continues their exploration. Upon being attacked by the revenant during the middle of the night, the NPCs resume an earlier interrupted conversation.

In all these situations, there is an urgency to respond to current conditions or else a need to advance the story. Selecting the next scene by only looking at what has happened previously cannot handle these situations.

In the initial Marlinspike design, when scenes were being selected to play and tied for the highest score, Marlinspike broke the tie by taking the scene that was last added to the scene manager. As author, I began to "abuse" this behavior by loading scenes in a particular order in order to prioritize certain scenes in the case of ties. But this was a brittle, error-prone approach. And sometimes scenes would differ by only one point and inferior scenes would still get chosen. So I decided to make this feature of author-preference explicit. Import—which is essentially a measure of how important it is to reincorporate an event *after* it has played—was not useful here as a means to break the near-ties. So I added an imperative measure to scenes and to the scene selection process.

Imperative is admittedly a very vague concept that should be examined more closely in future work. But practically-speaking it allows me as author to grant a little bonus to certain scenes when their selection scores would otherwise be very similar. However, because imperative is only part of the scene selection score, it can be overruled by the system in cases where one scene is clearly a better choice based on the normal reincorporation rules.¹⁹

Weakness: Authorial Burden

Authoring Marlinspike scenes is not an easy task. Their monolithic play functions can become twisted logic puzzles of complex conditional tests needed to handle myriad contexts and situations. But even authoring the canPlay functions is not simple. Choosing which events a scene will reincorporate also determines how it will be scored and selected. This selection of scenes will then determine the soundness of the resulting story structure.

¹⁹ In order to evaluate Marlinspike's reincorporation approach on its own merit, this imperative feature can be switched off in the prototype implementation. For the most part, disabling it makes little difference in the resulting stories other than correcting the handful of problem situations described above.

So the first burden on the author is ensure that there are sufficient scenes to reply to all possible significant player actions in a variety of contexts.

The next burden is explicitly listing the preconditions for each scene. This includes various world requirements, such as whether the PC is currently conscious, that NPCs are present and capable of acting, that certain objects are present, that certain doors are unlocked and open, or that certain locations are logically reachable from the current one. Failing to explicitly list each precondition can lead to program errors and crashes when implicit preconditions or assumed states are not met in some fringe situation.

In addition, under the current design, every middle scene is required to have at least one event prerequisite so that it extends some part of the story. Sometimes this means world-level requirements become story-level requirements. For example, encountering the revenant for the first time reincorporates traveling to that location. However, using such events as preconditions can occasionally provide surprising scene selection choices—such as when arriving at a location is also part of a hefty GoParty exploration thread.

Just as significant player actions need to be sufficiently reincorporated by other scenes, scenes must also remember to reincorporate earlier significant scenes. If this does not happen then the resulting internal story structure is not always clean.

Here is an example of this from *Demeter:* When a GoParty returns to the passenger gondola, it will usually report what it experienced in the Zeppelin during its exploration. GoParty_Returns requires as a prerequisite that the GoParty just moved into the gondola stairwell from outside the passenger gondola.

```
GoParty_Moves_Along(Ladderway) ---> GoParty_Moves_Along(Stairwell) --> GoParty_Returns
|-- GoParty_Reports
```

Figure 1: Story thread resulting from GoParty returning to gondola and reporting.

However, sometimes the GoParty comes back to find the passenger gondola locked from the inside. If someone knocks on the hatch, those inside the passenger gondola must decide whether to open the hatch and let the GoParty back in. Depending on the NPCs' states at this time, they can choose to betray the GoParty by refusing to open the hatch. Since this is a fairly significant event, the scene has an import high enough that it will start its own thread.

Figure 2: Thread leading to GoParty_Rentry is left hanging.

This requires that the GoParty_Reentry scene be explicitly reincorporated by the end of the story in order for the resulting story structure to be unified. However, GoParty_Returns did not originally refer back to GoParty_Reentry, which meant GoParty_Reentry was often let hanging. This is very hard to notice in the game output because, to the audience, it is implicitly included in the flow of events: the GoParty came back, knocked, were let in, and gave a report. But it falls upon the author to make sure that these same implicit connections between events are also explicitly modelled in the system's view of the story.

Therefore, I had to add a hook from GoParty_Returns to GoParty_Reentry. To make this hook explicit in the resulting output, I also had to have a returning NPC thank the opener of the hatch before reporting.



Figure 3: Hooking GoParty_Reentry back into the GoParty_Returns thread.

Due primarily to time constraints, there are number of similar situations left like this in *Demeter*. This means that most stories, even with a willing player, do not result in only a single story thread by the end of the game. There will be a handful of extra threads—often with only one or two events of unique unreincorporated material each. This sort of story structure could be cleaned up with a bit more tedious effort by the author.

Success: Explicit Story Structure

Due to problems like the one just described, the system story structure may not always perfectly match the audience's conception of the story. However, such a detailed, explicit story structure is still one of major strengths of Marlinspike.

First of all, through the process of casting and recasting, the system is capable of a detailed view of the various components and effects of different events. For example, consider the *Demeter* view of one action and its subsequent scene: The player shoves Count Vladescu again during an ongoing conversation.

First, here is a screenshot of the resulting output:

🚣 Zag	x
File Edit	
Dining Room Moves: 6	
> push count You give Count Dragos Vladescu a hearty shove.	
VLADESCU (to YOU): "I already asked you to stop doing that!" VLADESCU (to YOU): "You continue to disappoint me. You didn't listen when I asked you earlier not to harass me, either."	
ROMAN (to YOU): "That wasn't even amusing the first time you did it. I really don't think you need to repeat it again."	
MAJOR WINTERS: "We're getting sidetracked from our discussion here"	
MAJOR WINTERS: "Something very strange is going on aboard this ship. And I think we're obligated to go find out exactly what that is. Staying here earns us nothing! Let's go."	
VANDERLEY (to MAJOR WINTERS): "I agree."	
EVENWORTH (to MAJOR WINTERS): "Perhaps I'm overlooking something here, but I just can't agree with you."	
MRS WINTERS scowls disapprovingly.	
>	-

Figure 4: Game output resulting from pushing Count Vladescu.

The following is the system view of this same turn.

Listing 5: System view of two events: a Push deed followed by a Discussion Starts scene. |- 11=MEvent 76(HARASS<3> [Push], Count Dragos Vladescu, , Dining Room, $\{-8\}$) <5> |-|- 11.0=MEvent 77(DEFY<5>, Count Dragos Vladescu, , Dining Room, {-10}) [reinc:1]>[8.0.0] |-|- 11.1=MEvent 78 (AFFRONT<3>, Mrs Irene Winters, MEvent 76, Dining Room, {-1}) |-|- 11.2=MEvent 79(AFFRONT<3>, Major Jonathan Winters, Mevent 76, Dining Room, $\{-1\}$) |-|- 11.3=MEvent 80 (AFFRONT<3>, Ms Hannah Evenworth, MEvent 76, Dining Room, {-1}) |-|- 11.4=MEvent 81(AFFRONT<3>, Miss Miriam Vanderley, MEvent 76, Dining Room, $\{-1\}$) |-|- 11.5=MEvent 82 (AFFRONT<3>, Mr Elijah Roman, MEvent 76, Dining Room, $\{-1\}$) |- 12=MEvent 83(Major Jonathan Winters: (Discussion Starts)<6>, , Dining Room) [reinc:6]>[0, 11.0, 10.0, 11.5, 10.1, 4] <6> |-|- 12.0=MEvent_84(nothing: (NPCs_React)<3>, _, _, Dining Room, {6}) |-|-|- 12.0.0=MEvent 85 (Count Dragos Vladescu: (NPC Defied) <3>, you, MEvent 77, Dining Room) |-|-|- 12.0.1=MEvent 86(Mr Elijah Roman: (NPC Annoyed)<3>, you, MEvent 82, Dining Room) |-|-|-|- 12.0.1.0=MEvent 87(Mr Elijah Roman: (NPC Interdicts)<4>, you, MEvent 82, Dining Room, {365433}) |-|- 12.1=MEvent 88 (Major Jonathan Winters: START STATE<0>, (DST Discussion), leaving the passenger gondola, Dining Room) |-|- 12.2=MEvent 89(Major Jonathan Winters: (NPC Proposes Plan)<3> [Tell], leaving the passenger gondola, HUNT, Dining Room) |-|-|- 12.2.0=MEvent 90 (Major Jonathan Winters: SUPPORT<3> [Tell], Count Dragos Vladescu, leaving the passenger gondola, Dining Room, {1}) |-|-|-| 12.2.0.0=MEvent 103 (Major Jonathan Winters: AFFRONT<3>, Count Dragos Vladescu, _, Dining Room, {-1}) |-|-|- 12.2.1=MEvent 91 (Major Jonathan Winters: SUPPORT<3> [Tell], Mrs Irene Winters, leaving the passenger gondola, Dining Room, {3}) |-|-|- 12.2.1.0=MEvent 102(leaving the passenger gondola: ENDEAR<3>, Mrs Irene Winters, _, Dining Room, {1}) |-|-|- 12.2.2=MEvent 92 (Major Jonathan Winters: SUPPORT<3> [Tell], Ms Hannah Evenworth, leaving the passenger gondola, Dining Room, {5}) |-|-|-| 12.2.2.0=MEvent_101(Major Jonathan Winters: AFFRONT<3>, Ms Hannah Evenworth, , Dining Room, {-4}) |-|-|- 12.2.3=MEvent 93(Major Jonathan Winters: SUPPORT<3> [Tell], Miss Miriam Vanderley, leaving the passenger gondola, Dining Room, {1})

Listing 5: (Continued) System view of two events: a Push deed followed by a Discussion_Starts scene.

```
|-|-|- 12.2.3.0=MEvent 100 (Major Jonathan Winters: ENDEAR<3>,
                              Miss Miriam Vanderley, , Dining Room, {1})
|-|-|- 12.2.4=MEvent 94 (Major Jonathan Winters: SUPPORT<3> [Tell],
                         Mr Elijah Roman, leaving the passenger gondola,
                         Dining Room, {6})
|-|-|-|- 12.2.4.0=MEvent 99(leaving the passenger gondola: ENDEAR<3>,
                             Mr Elijah Roman, _, Dining Room, {5})
|-|- 12.3=MEvent_95(nothing: (NPCs_React)<3>, _, _, Dining Room, {6})
|-|-|- 12.3.0=MEvent 96(Miss Miriam Vanderley: (NPC Replies To Opinion)
                          <2> [Tell], Major Jonathan Winters, leaving the
                          passenger gondola, Dining Room, {1})
|-|-|- 12.3.1=MEvent 97 (Ms Hannah Evenworth: (NPC Replies To Opinion)
                         <2> [Tell], Major Jonathan Winters, leaving the
                          passenger gondola, Dining Room, {-1})
|-|-|- 12.3.2=MEvent 98 (Mrs Irene Winters: (NPC Replies To Opinion) <2>
                          [Tell], Major Jonathan Winters, leaving the
                          passenger gondola, Dining Room, {-2})
```

First of all, pushing Count Vladescu was the 11th event of the story, as shown by the 11 at the start of the first line above. This was cast as a HARASS action, which had a negative affinity impact (-8) on the Count. This wasn't the first time the player pushed the Count in this story though, so this push was also a DEFY (line 11.0) of the Count's earlier request that the player stop being belligerent. This sort of antisocial behavior was also slightly AFFRONTing to all the other NPCs that witnessed it (lines 11.1 to 11.5).

The system then responded by resuming an earlier interrupted discussion. This scene as a whole reincorporates a number of earlier events: the start of the story, various previous discussion events, and the defying of the Count. This scene has a number of components. First, NPCs_React plays (12.0). This has the Count express his annoyance at being defied (12.0.0). Elijah Roman also chimes in (12.0.1), warning the player not to bother the Count any more (12.0.1.0).

Then Major Jonathan Winters starts the conversation again (12.1) by proposing a plan (12.2): leaving the passenger gondola in order to hunt the revenant (although at this point in the story, the NPCs do not know yet that it is a revenant they face). This public support for leaving the passenger gondola is equivalent to suggesting the idea to each person individually (12.2.0 to 12.2.4). This has different effects on each listening NPC. For example, Count Vladescu and Ms Evenworth disagree, and so they were AFFRONTed by Major Winters's support of such an idea (12.2.0.0 and 12.2.2.0). They now think slightly less of him. On the other hand, Mrs Winters and Mr Roman think more highly of Major Winters than they do of either leaving or staying in the gondola. So they were swayed by his speech to think more highly of leaving the gondola (12.2.1.0 and 12.2.4.0). Finally, Miss Vanderley already strongly supported leaving the gondola, so
hearing Major Winters support it too means she thinks more highly of him (12.2.3.0).

After Major Winters' proposal, NPCs_React plays again to provide some responses (12.3). It would be tedious if every NPC responded every time a proposal was made, so NPC_Reacts has only some those most affected reply. Here, those NPCS with the strongest feelings about the proposal to leave the gondola reply. Note that, in this case, this is not the same as those NPCs with the greatest affinity change. For example, Mrs Winters was convinced just a little to consider leaving, but she still feels very strongly overall that it is a bad idea. Similarly, Elijah Roman's opinion was shifted quite a bit, but, overall, he's still on the fence and so does not reply.

Aside from this complete view of the internals of each event, the events are also connected by reincorporation links. These often express different kinds of causality between events.

This detailed system view is not intrinsically valuable, especially since much of its richness is not narrated effectively. However, a structure like this could serve as a very important foundation for future interactive drama systems. The full details of the past story are available in a machine-readable format that could be used by NPCs explaining their motivation for current actions or when summarizing or narrating offscreen action.

This latter use occurs in *Demeter's* GoParty_Reports scene. It searches through the details of the event history to discover the details of what the GoParty accomplished: what evidence it encountered, who was attacked and how many times, whether the revenant was slain, whether the PC wandered off or made a nuisance of himself, etc. Because it mines the event history for these details, they can be reported regardless of whether they were all generated by a single GoParty_Offsceen event or by a long, multi-event exploration lead by the player.

Still, there is a small problem to be worked out of this representation. Because reincorporation links between events are built by Marlinspike based on the list returned from the root scene's canPlay function, all reincorporations for scenes are incorrectly represented as coming from the root event, rather than from the specific sub-event that is actually performing the reincorporation. For instance, in the event tree given above, Count Vladescu's NPC_Defied reaction is actually what is reincorporating the PC's DEFY action (11.0). However, this reincorporation is listed as if Discussion_Starts is responsible for it. Like many of Marlinspike's other problems, this stems from Marlinspike's expectation that scenes will be atomic, without such component events.

Resulting Stories

So far I have covered many of low-level details involved in Marlinspike and *Demeter*. But what of the stories produced by the finished system? This section explores the nature of those complete stories, especially in relation to how they are authored.

Stories as Branching Trees

A story is primarily a series of events, and so we often think of a story as a sequential, linear, path-like thing.²⁰ And so, when we begin to conceptualize an interactive story—or any story in which choices could have lead to a different series of events—we think of it as a branching tree of possible story paths. There is a single path through this tree structure taken by the actual story, but at each point that a significant choice is made, we can also trace the other potential, untravelled story paths stretching away in different directions.

An interactive story can be explicitly authored this way as a series of choices that produce a branching tree of narrative paths. The *Choose Your Own Adventure* book series did exactly this. However, this approach does not scale well. If each choice were only binary, and a story includes only ten such choices that each lead to a truly unique outcome, it would be necessary to author 2¹⁰, or 1024 different story paths!

The scene-based approach used by Marlinspike was meant to overcome the authorial burden of this explicit branching-tree approach. If scenes are modular and can be mixed-and-matched at run-time to produce a story, then the author does not need to compose so many complete stories.

But, even when the story is not generated by explicitly using a branching tree, I find I still tend to think of the resulting structure in a branching-tree way: "What would have happened if the player had made a different choice or performed a different action at this point?" And so elements of this mental framework persisted while authoring *Demeter*.

First of all, the body of most scenes is a fairly complex series of if statements. Most scenes are so internally complex as it to take several days (about 15 hours of coding) to author and that long again to adequately test all the possible branches.

²⁰ Part of this linear sense of story comes from its narration, as it is possible to alternate narration between two or more concurrent series of story events. But, at least in traditional narrative forms, the narration of these two series is linear. By linear, I simply mean sequential and "path-like", but not necessarily straight or even unidirectional, since the course of a narration can jump back and forth in time.

Authoring *Demeter*'s tutorial further highlighted this. Because most of the tutorial scenes provide a paragraph or two of instruction and then prompt the user to perform a particular action, the bulk of them took only about 20 minutes each. So scenes are not necessarily complex.

The exception to this in the tutorial was Miriam_Acts, a scene in which an NPC responds to what the player did. The point of this scene is that Miriam will try to reclaim her lost novel and then return to bed. There are four different ways this scene can be prompted to play: the PC waits, leaves, says goodnight, or does some action other than looking around or conversing with Miriam. Then Miriam will try to get her book. There are five possible states there: she already has it, the PC has it, it is somewhere she can get it (with a special case for finding it in the toilet), or it is gone (actually flushed down the toilet). Additionally, for any case where the book does exist, it could now be soggy from being in the toilet (three more possibilities). And, if the player has it, he can refuse to give it up when asked (1 more possibility). So this single scene that results in a paragraph or so of text took about 6 hours to write, and over 30 different permutations of its output are possible.

This scene is a better example of a typical scene. It it is also a reminder that many potentially interesting story paths—such as what happens if you put someone else's book into the toilet—are frequently not explored by users. Of the 40 or so of the game sessions I have seen so far, no player has ever tried this on their own.

So the branching structure remains within scenes. Between scenes, the structure is only implicit. Scene preconditions force a necessary order between certain scenes. First of all, beginning, middle, and ending scenes impose simple limitations on where they can play in a story. Other scenes have logical relationships that require one scene come before another. For example, it is not possible to conclude a discussion before starting one. The GoParty cannot depart the passenger gondola until after it has been formed. Other restrictions are emergent. Since the player starts in the passenger gondola, and the revenant does not enter the gondola until nightfall, it is not possible to immediately encounter the revenant within the first few events of the story.

As discussed when describing *Demeter*, these different scene preconditions do produce a general *Discussion - Exploration - Waiting* story structure. But any particular scene or even one of these stages can be skipped. And it is possible to explore before any discussion occurs.

So one of Marlinspike's design goals was successfully met: The story was not produced as a simple branching tree of story paths. Instead, it is dynamically built at runtime. But the branching tree structure is still evident within scenes. And, as author, I still think of potential stories in this way. For example, when testing all

the possible contexts in which a particular scene might play, I still find myself trying to imagine all the possible story paths that could lead to the scene.

Authorial Conception

So there is a mental switch required when authoring an interactive drama. An interactive drama author is definitely not writing a single story, with a single path for the player to follow. Nor is he even writing multiple interwoven stories, where the player's choices lead down different branching paths. Instead, the author is crafting a story space. This is not an easy mental switch to make.

Authoring a "story space" first means that a wide variety of verbs and actions need to be modelled. The author has some control here on what sort of actions he wants to support, though. For example, if the author defines no violent actions, then it will simply not be possible for the player to act violently in the game. However, this can be frustrating for those players that want to try those avenues. Granting such a wide range of options means the PC can then be heroic or indecisive or psychotic or apathetic. This is overwhelming at first as an author, since all these different choices and play styles then need to be supported in the story space in such a way that they can still lead to interesting stories. The means to do this is by authoring scenes that can build on the diverse range of player actions.

Admittedly, not all of these resulting stories may be of equal quality or length. But the focus of interactive drama needs to be on supporting significant player choices and then producing interesting narrative consequences stemming from those actions. To achieve this, the author needs to abandon the notion of any particular story path or required player behavior. This is what I mean by authoring a story space rather one or more story paths. There should be no "best story" in such a space that the player needs to be directed towards.

Authorial Control

Although an interactive drama author may need to surrender the notion of a "best story", that does not mean he should also need to surrender all quality control! The author provides the player with potential choices through the actions he defines, and he provides the consequences of those actions through the scenes. It is then the job of the system to provide the best possible consequence to each action in the author's absence. That is, given the author-provided material it has on hand, the system should make the "best decision" in responding to the player at each point in the story.

Implementing Marlinspike and *Demeter* has shown me that reincorporation is part of this "best decision" as it weaves the consequences into a story. But, as

the addition of an imperative value to scene selection shows, reincorporation is definitely not the only factor. As an author, after suffering through the burden of authoring all of the world, characters, verbs, actions, and scenes and then handing the reins over to the drama manager to weave these together in response to a player, it is very disappointing to see the system making poor decisions with the material I have given it. The player is free to act in any way afforded by the system, but the drama manager should be subservient to the author's will!

Discovering what factors besides reincorporation should go into selecting the next scene is important future work. Import of the current action is certainly one aspect. Urgency is another, as many actions need to be responded to immediately. And, when import and urgency of the current action are low, it may be time to introduce new material that advances the story.

Resulting Stories

This finally brings us to the qualities of complete *Demeter* stories: complexity, novelty, and well-worn paths.

First of all, testing complete *Demeter* stories was a nightmare. This arises from the dozens of ways a single scene can play combined with the myriad possible world and story contexts in which that scene could play. Also, many scenes rely on randomness and the contents of the event history to make certain choices. This makes it very difficult to produce reliable or useful unit tests for scenes. Thus, the most reliable way of testing scenes was to quickly play through multiple games up to that scene, possibly after some tweaking of the initial world state such as characters' initial locations and affinities.

In the process of testing with different initial world states, I would occasionally encounter fringe cases and story contexts that I am not sure are possible to reach through normal play. For example, NPCs with certain plans that result from fearing the revenant will not join the GoParty nor will they come to the rescue of someone attacked in the passenger gondola by the revenant. So I am not sure if they can ever encounter the revenant at all. But I still authored their responses should it happen. Similarly, scenes (should) still work if there are no NPCs left alive, or if there is only one. Yet the chance of this happening—of the PC going on a homicidal killing spree—is fairly low.

Since I authored all the content displayed by the game (except for some of Inform's standard verb responses), I do not often encounter any novelty in the game's output. But, given the combination of complexity, randomness, unexpected player choices, and fringe cases I have since forgotten I authored, I am still surprised occasionally, especially by certain emergent combinations. For example, although I wrote the reactions of an NPC to eventually attack a violent

PC, I was surprised and amused to see the NPCs gang up and beat a particularly violent PC to death.²¹ Arising from the independent behavior of the NPCs and the revenant, the revenant will pick off members of a divided GoParty. The most memorable surprise for me was learning that it is possible to seduce a married man: Due to his relationship with and high affinity for Irene Winters, Jonathan Winters refuses most romantic advances. But, as part of waiting through the night, he may then still show up later at the player's door with romance on his mind.

In general, novel or unexpected behavior is not desirable as an author. Instead, it is extremely nerve-wracking! It means I have overlooked some combination of scenes or states or have neglected to make explicit some precondition. This is worrisome because most overlooked combinations do not lead to amusing behaviors like those described above. Most of the time they produce illogical behavior or outright errors that generate garbage output or game crashes. So there is a strong authorial tendency to try to constrain these "surprises" as much as possible.

This inability to exhaustively test combined with discovering unexpected combinations means that, even as both engineer and author, the system still feels somewhat opaque to me. I still don't have a clear conception of the full and exact range of story possibilities. What I desire is a map of all the possible game states. However, Marlinspike was designed to go beyond the branching tree approach required to provide just such a precise mapping of all possible of story paths. So there is a definite tension here between a desire for authorial certainty and the system's complexity and flexibility.

Despite all of the potential fringe cases, most game sessions adhere to a few, well-worn story paths—both in terms of the scenes played and the paths through those scenes. The initial affinities and other states of NPCs prompt those NPCs to make certain consistent choices through the game. Players also tend to make the same choices. In *Demeter*, they want to be active: They usually march directly up into the Zeppelin and wander back and forth until they encounter the revenant and then try to kill it. This is fine, but it means most of the other story paths go unexplored. For example, it is possible to woo NPCs or be beaten into submission by them. It is possible to get a well-formed story by doing nothing at all. Some choices are extremely rare. For example, if you stay in the passenger gondola while the GoParty explores and if the other NPCs with you are evenly divided on whether to open the hatch again, you can become responsible for the choice of whether to let the GoParty back in or else betray them to the predations of the revenant.

These fringe story paths still need to be authored so that there is a breadth of potential story and in order to support the full range of possible player actions. So

²¹ I have since modified this so that the PC is only beaten to unconsciousness.

the lesson here is that an author may feel more fulfilled if he can recognize what the most common story path will be. He can then focus a bit more of his attention there by providing more interesting choices along that path. Doing this would in fact divide this path into other less well-travelled paths. Note that this is not necessarily easy, since these "paths" are only emergent features of the finished system at work!

Also, sometimes emergent behaviors need to be tweaked with a eye to the resulting story. In *Demeter*, the revenant randomly decides whether to act and what to do based on its current health and the number of other characters currently present. That is, if it is injured or if the PC is not alone, the revenant is less likely to appear or to attack. At the time I wrote this, I was favoring the simulated "reality" of how a revenant might actually behave: Just because the PC arrives does not guarantee a sighting. However, favoring the verisimilitude of the simulation over the story leads to a long lull in most stories as players are forced to wander back and forth along the length of the Zeppelin until the revenant finally appears. It seems now that a concession should have been made to the demands of the narrative: the revenant should appear after a certain threshold number of wandering scenes.

Conclusion

Implementing Marlinspike and *Demeter* thus posed a number of challenges. As a summary of those challenges, this implementation project was characterized by simplicity, complexity, opacity, brittleness, and moderate success.

Simplicity: In retrospect, Inform was too simple a language to comfortably implement Marlinspike and all of the scenes. In focusing only on story structure, Marlinspike was also too simple, leaving many thorny problems for the scenes of the implementing game to handle. While a reusable interactive drama framework must be agnostic about the content of any particular story, greater structural support for the inner contents of scenes is still needed.

Complexity: There is a great deal of content required of the author—from world object definitions, to characters and their motivations, to the scenes themselves —and all of these components then interact with each other. Much of the state management performed by scenes in this implementation was rather ad hoc, rather than through previously well-defined means such as NPC deeds. The resulting story structures and contexts are difficult to conceptualize, and there was little support for testing the resulting stories.

Opacity: For the players, the affordances of interactive fiction make it difficult to know what it is possible to do in the world, as well as how to then issue the commands to do it. For the author, the complexity can lead to moments of

unexpected, and frequently unpleasant, surprises. In general, much of the internal richness of the NPCs and event structure goes unnarrated.

Brittleness: Even minor errors by the finished system—such as subject-verb disagreements—are obvious and jarring. Yet correct performance is generally subtle and unremarkable. Combined with the complexity and opacity, this can leave an author with a sense of lurking game failures that will be both sudden and severe.

Yet, for all the problems and remaining warts, the project was still a success. Although perhaps simple and not as wonderfully varied as I had once envisioned, Marlinspike/*Demeter* does produce a wide range of stories with the material it has to work with. It translates verbs into actions based on the current story context. In doing so, it also builds a very detailed model of the story that includes both player actions and system scenes. With a few small improvements, this basic architecture could serve very well as a solid foundation for further interactive drama development. Reincorporation also proved useful, especially in the way that recasts and preconditions form logical links between different events in the system's story model. Finally, given all the complexity, bugs in the final implementation are relatively rare.

But these are only my impressions as engineer and author. The next two chapters will provide the results of an evaluation of *Demeter* by other players, focusing in particular on the effects of Marlinspike's reincorporation feature. Then, in the final chapter of this dissertation, I will outline some future work that is informed by both that evaluation and the lessons learned here.

CHAPTER VI. EVALUATION

Claims to be Tested

The key feature of Marlinspike is its threading behavior. Because it works important user actions into the unfolding story, *reincorporation should lead to a better interactive narrative experience*. In evaluating this claim, we need to look at three variables: *narrative*, *interactivity*, and *user experience*.

Narrative

In an interactive drama, the Action level of narrative—specifically, the story's event structure—is of particular importance. The poetics outlined earlier provides a working definition of narrative structure: a unified, complete series of events with a beginning, middle and end, in which all events are connected by necessity such that none can be removed without leaving the whole disjointed or disturbed.

Admittedly, Marlinspike applies a slightly relaxed version of this definition in that only user actions of high *import* need to be made necessary to the unfolding story. Still, because the system tracks all the events—whether scenes or actions —and the threads of necessity between them, it can provide a number of measures of how well-formed the resulting story structure is.

The system measures of well-formed story structure include:

- The existence of beginning, middle, and end scenes in a single main thread, which reveals whether the system was even able to form a complete story.
- The story's length in terms of events, which reveals whether a particular session resulted in an unusually short or long tale.
- The number of additional threads, which shows the degree to which the system failed to produce a unified story structure.

Interactivity

Interactivity is more than the extent or frequency to which the player can make inputs to the system. Rather, I am interested here in the degree to which the player can exert agency by bringing about significant and meaningful changes in the system. The player's world-level agency is important, as this represents the degree to which the player can successfully move her character through the story-world, manipulate objects, or affect other characters. However, in determining the success of reincorporation, the player's story-level agency—the degree to which her actions significantly impact the later events of the story—is of even greater interest.

The system measures of story-level agency include:

- The variety of the actions used, which illustrates how exploratory or varied the player's choices were.
- The average import of actions, which indicates how active the player was in general at the story level.
- The percent of actions of high import that are threaded into the main story thread, which indicates the frequency that the player contributed significantly to the story.

User Experience

The purpose of an interactive drama is to provide a novel and meaningful experience to the user. Indeed, research has shown that interacting participants in a live-action interactive drama reported the plot to be an intense and significant experience, even when a passive audience found the same drama slow and poorly-formed (Kelso, Weyhrauch, & Bates 1992). Another significant aspect of an interactive drama is its replay value—that the player can experience the same story-world again but experience different narrative consequences by making different choices.

While it is hoped that Marlinspike will provide a satisfying overall experience, there are a great many factors involved, such as the story's subject matter, the characters, the story's genre, the dialog, the presentation style, the text-based medium, the interface mechanisms, the player's mood and surroundings, etc. Yet, with regard to reincorporation effects, it is possible to simply ask whether the user feels that the story is better-formed and that they experience greater story-level agency when reincorporation is used.

The measures for the player's experience of story structure and story-level agency will be the player's responses to a questionnaire administered after playing a session of the *Demeter* game.

Hypotheses

Therefore, Marlinspike's reincorporation feature should result in a better-formed story structure and greater story-level user agency. Furthermore, players should be able to detect these differences and report a corresponding improvement in experience.

Thus, there are four specific hypotheses to be tested:

- H_1 : Reincorporation produces a better-formed internal story structure.
- H₂: Reincorporation provides greater user agency at the story-level.
- *H*₃: Reincorporation produces higher user ratings of story coherence and structure.
- H_4 : Reincorporation produces higher user ratings of story-level agency.

Experimental Setup

Reincorporation

The Marlinspike prototype was implemented to allow both author imperative and reincorporation to be turned on and off. Turning off reincorporation has two effects.

The first change is to the scene selection process. As described in Chapter III, when Marlinspike wishes to play a scene, it first establishes the set of scenes that have their preconditions met. From this set of possible scenes, it then normally selects the scene that will splice the most unreincorporated material into the story. When reincorporation is turned off, this second stage of the selection process is skipped. That is, Marlinspike does not apply any of its normal scene weightings. Instead, it simply selects the next scene randomly from those that can currently play.

The second change is to how scenes play. As described previously, many scenes include hooks that can optionally refer back to certain earlier events. When reincorporation is turned off, these hooks are ignored. Without these hooks, fewer earlier events will be reincorporated in both the system's internal model of the story and in the resulting scene narration. The specific *Demeter* scene hooks affected by this change are listed in Appendix B.

Even when the general reincorporation feature is turned off, Marlinspike still forms threads based solely on the reincorporation of each scene's precondition events. This means Marlinspike can still report the same internal measures of story structure and agency as when reincorporation is turned on.

Reincorporation is significant as most existing scene-based interactive drama architectures do not use it. Instead, they simply determine which scenes can be played, and then select one at random—as Marlinspike does with threading turned off. For instance, my previous system design, Eudaemon, worked precisely this way (Tomaszewski & Binsted 2007). In particular, scene

preconditions included the player's current location, recent actions, and whether the scene filled the next Propp function in the story. The perceived limitations of the Eudaemon system were due to adhering to Propp's requirements, not to the random selection of next scenes.

GEIST, which shares much is common with Eudaemon, also works in a manner very similar to that of Marlinspike when reincorporation is turned off (Grasbon & Braun 2001).

Experimental Process

The experiment to test the effect of Marlinspike's reincorporation feature was executed in the following way. Participants were recruited by email and then participated online. Participants were randomly assigned to one of two groups. They completed a background survey and then played the *Demeter* game twice, once with reincorporation on and once with it off. The two groups differed in whether they played first with reincorporation on or with it off. After each play session, they completed an identical response survey.

Here are the details of this process.

First, each interested participant clicked a link provided in a recruitment email message. (See Appendix C for a copy of this email message.) This link directed participants to a webpage where they were assigned an anonymous participant ID number and thus assigned to one of two experimental groups. Even ID numbers were assigned to Group 0 and odd ID numbers were assigned to Group 1. Since IDs were assigned consecutively, this is effectively a random assignment to groups.

Participants were then sent to an initial online survey hosted by SurveyGizmo.com. They were required to give informed consent and provide background information, which included the participant's gender, age group, education level, previous computer experience, computer game experience, and roleplaying game experience. (See Appendix D for a full copy of this background survey.)

Each participant was then sent back to the game web server that initially assigned the participant IDs. There they could play the *Demeter* game online in their web browser. The game was executed in a modified version of the Javabased Zag interpreter (Zeppieri 2005). The first game session started with a short in-game tutorial that had to be completed before the game itself started. This was used to help introduce participants to the text-based game format and reduce learning effects between the two game sessions.

When the game window was closed, the modified version of the Zag interpreter sent back to the game web server a transcript of all input entered during the game. By comparing the recorded timestamp for when a given participant arrived at the game page and the timestamp for when the game sent back its transcript, it was possible to track how long participants spent on each game.

After completing the first game session, participants clicked a link on the game page to be taken back to SurveyGizmo.com to complete a response survey. This three-page survey asked questions concerning their experience in terms of story structure and user agency. (See Appendix E for a full copy of this response survey.)

At the end of the response survey, participants were directed again to the game server for a second game session. They then completed the same response survey regarding this second play session.

Those participants in Group 0 (even IDs) first played the game with reincorporation on and then with it off. Group 1 (odd IDs) played first with reincorporation off and then with it on. This was used to control for the effects of initial exposure to the system.

The author imperative option used by Marlinspike in selecting scenes was also turned off for all games in the evaluation. This is because the intent of this study was to examine only the effect of reincorporation as compared to selecting possible scenes randomly.

The experiment ran for two weeks. After all data was collected, I replayed each recorded game transcript to review the different game sessions. I also collected certain game data concerning the system's internal view of the structure of each finished story.

Pilot Study

Before running the final study, I first alpha-tested the game in its final online form by asking a handful of close friends and dissertation committee members to play the game and report any bugs. I asked these testers to play the game four times: twice in its normal mode, once with imperative turned off, and once with both imperative and reincorporation turned off.

After fixing the bugs revealed by these alpha tests, I ran a pilot study of the complete evaluation study pipeline, including assignment to groups and the two online surveys. I asked all of my dissertation committee members as well as some family and friends to complete this pilot study. This pilot also served as a beta test of the game.

Twelve pilot study participants completed the background survey. Of those, nine completed the first game session response survey. From those nine, I received six complete responses, with three in each group.

The pilot revealed a few more game bugs. The most significant issue involved getting the modified Zag interpreter to start correctly on Mac computers. The surveys were deemed sufficient and were largely unchanged. However, in order to help participants move through the evaluation steps correctly, the instructions to participants were clarified. Also, the two game session webpages were more clearly differentiated by using different background colors.

The pilot data set collected was too small to show any trends. In addition, I assigned the games to the groups incorrectly so that the first group played both games with reincorporation on and the second group played both games with reincorporation off. This was corrected for the final study.

CHAPTER VII. EVALUATION RESULTS

Participant Recruitment

As previously described, I recruited participants by email. I sent a recruitment email message to a number of lists in the following departments at the University of Hawai'i—Mānoa: Communications and Information Science, Information and Computer Science, and Academy for Creative Media. Most of these lists were student lists—both graduate and undergraduate—though a few of them also included faculty members. These departments were chosen on the assumption that their list subscribers would likely be interested in an interactive drama system. This was an issue because, in most cases, it was necessary to convince a list moderator to forward the email to the list.

The original email included a request for people to forward it on to others that might be interested. Based on responses collected from participants, this request also reached students second-hand in the Library and Information Science, Information Technology Management, and Engineering departments. By my best estimate, this recruitment email reached over 500 list subscribers.

From this email recruitment, 102 clicks were received in return. Of these, 53 completed the initial background survey, 35 made it through the first game survey, and 29 made it through the second game survey.

However, not all 29 of these complete responses were usable. One participant played two complete games before answering the first survey. Two others quit halfway through both games. Due to unresolved technical issues, two were missing game transcripts for their first game. This left 24 participants that played two complete recorded games each.

This selection process means that the population represented by this study is those people who are interested enough in interactive drama to take a couple hours out of their busy schedules to give it a try and who are perseverant enough to continue the experience through two games.

Participant Backgrounds

The technique used to randomly assign participants to groups resulted in somewhat uneven groups. Of the 24 participants, 15 were assigned to Group 0, while the remaining 9 participants were assigned to Group 1. There also proved to be some significant differences between the two groups with regard to previous gaming experiences.

The following is a summary of the background information for the 24 participants used in this evaluation.

Demographics

Here are the distributions of gender, age, and education for all 24 participants combined, and then for each of the two separate groups.

Gender	Total	Group 0	Group 1
Male	15	9	6
Female	9	6	3

Table 1. Participant gender distribution

Age	Total	Group 0	Group 1
45-54	1	1	0
35-44	5	3	2
25-34	9	7	2
18-24	9	4	5

Table 2. Participant age distribution

Highest Education Level completed	Total	Group 0	Group 1
Doctoral / Professional	4	3	1
Graduate School	5	4	1
University / College	8	4	4
Community College	1	1	0
Some college, no degree	4	3	1
Graduated high school	2	0	2

Table 3. Participant education distribution

Gender was fairly evenly divided between the groups, though men outnumbered women in all cases. On average, Group 0 was older than Group 1 and more highly educated, though not significantly so. The majority of participants had completed college or a higher level of education.

Computer Experience

The remainder of the background data is presented here using a standard format for each question. This format includes a total mean score for each response, a distribution of all response scores, and separate means for the two groups. When these means are significantly different from each other (p < .05), they are given in bold and the resulting *p*-value is given in the discussion below.

The questions in this section aim to measure participants' general comfort level on a computer, particularly with the sort of reading and input methods required to use the game.

Possible responses for this first question included Strongly Agree (4), Agree (3), Neutral (2), Disagree (1), or Strongly Disagree (0).

Survey Question	Mean		spo	nse	Co	unt	Mean:	Mean:
		SA	Α	N	D	SD	Group 0	Group 1
"I feel comfortable using a computer."	3.79	19	5	0	0	0	3.87	3.67

Table 4. Comfort using a computer	Table 4	4. Comfor	t using a	computer
-----------------------------------	---------	-----------	-----------	----------

The following question uses the following scale: six or more hours (4), three to five hours (3), one to two hours (2), less than one hour (1), rarely or never (0).

Survey Question	Mean	Re	spo	nse	Со	unt	Mean:	Mean:
		6+	3+	1+	1-	0	Group 0	Group 1
"On average, I use a computer for the following number of hours each day"	3.58	14	10	0	0	0	3.60	3.56

Table 5. Average computer use

The following question allowed the following responses: "I regularly read text on a computer screen without trouble" (2), "I dislike reading text on a computer screen, but I can tolerate it for short periods" (1), or "I avoid reading text on a computer screen whenever possible" (0).

Survey Question M		Respo	onse C	ount	Mean:	Mean:	
Survey Question	wean	No Trouble	Dislike	Avoid	Group 0	Group 1	
How do you feel about reading text on a screen like the one you are currently using?	1.92	22	2	0	1.93	1.89	

Table 6. Comfort reading text on a screen

The following question on command line interfaces allowed these responses: "I use a command line at least once a week for many different tasks" (4), "I use a command line at least once a week for a small number of tasks" (3), "I use a command line a few times a year, or I have regularly used a command line in the past" (2), "I rarely use a command line" (1), or "I have never used a command line (or, I do not know what a command line interface is)" (0).

		F	Respon	Mean:	Mean:			
Survey Question	Mean	Weekly/ Many Tasks	Weekly/ Few Tasks	Occ.	Rare	Never	Group 0	Group 1
How often do you use a command line interface?	2.29	5	4	10	3	2	2.33	2.22

Table 7	'. Familiarit	v with a	command	line	interface
1 0010 1	·········	<i>y 11111</i> a	oominana		

From these results, we can see that all participants were comfortable using a computer and did so for multiple hours a day. There was little complaint about reading text on a screen. All but 2 participants were familiar with a command line interface, and most had some degree of experience using one. There were no significant differences between the two groups on any of these measures.

Digital Narrative Game Experience

These questions explore participants' time spent play digital games using the following scale: six or more hours (4), three to five hours (3), one to two hours (2), less than one hour (1), rarely or never (0).

Survey Question	Mean	Re	espo	onse	Οοι	Mean:	Mean:	
	mourr	6+	3+	1+	1-	0	Group 0	Group 1
"On average, I spend the following number of hours per day playing online or computer games on a personal computer (PC):"	1.25	0	6	5	2	11	1.33	1.11
"On average, I spend the following number of hours per day playing digital games on platforms other than a personal computer (such as on a gaming console, mobile phone, hand- held device, arcade game, etc.):"	1.08	1	1	5	9	8	1.33	0.67

Table 0. Current aigital game experience in terms of time spent	Table	8.	Current	digital	game	experience	in	terms	of time	spent
---	-------	----	---------	---------	------	------------	----	-------	---------	-------

It is interesting to see that those participants who play PC games generally spend more time on them than those who play console or other digital games.

Some people may not currently spend much time playing games but may still have played a great deal in the past. The following questions attempt to measure this, as well as the general breadth of participants' digital game experience. They use the following scale: Over 100 games (5), 50 to 100 (4), 20 to 50 (3), 5 to 20 (2), 1 to 5 (1), or none (0).

Survey Question	Mean	Response Count						Mean:	Mean:
		100+	50+	20+	5+	1+	0	Group 0	Group 1
"I have played approximately the following number of different games using a personal computer:"	2.92	5	2	7	6	4	0	3.4	2.11
"I have played approximately the following number of digital games on platforms other than a personal computer (such as on a gaming console, mobile phone, hand-held device, arcade game, etc.):"	3.04	7	2	2	11	2	0	3.60	2.11

Table 9. Previous digital game experience in terms of total games played

So every participant had played at least one digital game before. However, those in Group 0 had played significantly more computer games, Mann-Whitney U = 31.5, n1 = 15, n2 = 9, p = 0.03, and significantly more games on other digital platforms, Mann-Whitney U = 29, n1 = 15, n2 = 9, p = 0.02, than Group 1. (Mann-Whitney was used to determine significance here because the response scale was not a continuous scale.)

These questions measure the participants' familiarity with various digital game genres using the following scale:

- I have played a number of games in this genre (4)
- I have played at least one game in this genre for many hours (3)
- I have briefly played or watched someone else play a game like this (2)
- I know what this is, but I have never played one (1)
- I do not know what this is (0)

Survey			Res	ponse	se Count Me			Mean [.]
Question	Mean	Many games	One game	Wit- nessed	Heard of it	Unfamiliar	Group 0	Group 1
Text adventure / interactive fiction	2.08	5	5	5	5	4	2.67	1.11
Adventure games	2.38	9	3	4	4	4	3.00	1.33
Computer or console roleplaying games	3.08	12	4	6	2	0	3.33	2.67
MUDs	1.13	3	2	1	7	11	1.47	0.56
MMORPGs	2.29	7	1	9	6	1	2.67	1.67
Social simulations	2.58	5	10	4	4	1	2.80	2.22
First-person shooters	2.96	9	7	7	0	1	3.20	2.56

Table 10. Familiarity with digital game genres

Probably due to having played more games in general, Group 0 was noticeably more familiar with every game genre than Group 1. However, only two of these differences between means were statistically significant: interactive fiction, t(22) = 3.05, p = .01, and adventure games, t(22) = 2.93, p = .01.

Non-digital Interactive Narrative Experience

These questions measure the participants' familiarity with various traditional interactive narrative game genres using the following scale:

- I have played a number of games in this genre (4)
- I have played at least one game in this genre for many hours (3)
- I have briefly played or watched someone else play a game like this (2)
- I know what this is, but I have never played one (1)
- I do not know what this is (0)

Survey			Response Count				Mean:	Mean:
Question	Mean	Many games	One game	Wit- nessed	Heard of it	Unfamiliar	Group 0	Group 1
Branching novels and gamebooks	2.54	8	6	3	5	2	2.8	2.11
Table-top roleplaying games	2.13	5	6	2	9	2	2.6	1.33
Live-action roleplaying games	1.58	4	2	2	12	4	1.73	1.33
Improv	1.88	5	2	5	9	3	2.07	1.56

Table 11. Familiarity with non-digital game genres

Again, Group 0 is noticeably more familiar with these game genres. However the difference was only significant for table-top roleplaying games, t(22) = 2.43, p = .02.

Therefore, the random assignment to groups was not fully successful in producing equivalent groups. Group 0 was larger and its members had played significantly more digital games than those in Group 1, as well as being more familiar with interactive fiction, adventure computer games, and table-top roleplaying games.

Incomplete Participants

As mentioned when describing recruitment, 53 participants finished the initial background survey, although only 24 of those went on to complete all other stages of the study. On average, the backgrounds of the 29 "incomplete" participants did not significantly differ from those that completed the study, with the following exceptions.

The incompletes had less experience with interactive fiction games (1.21) than those that completed the study as a whole (2.67), t(51) = 2.28, p = .03, although it is important to note that their mean score is still higher than the Group 1 mean (1.11). The incompletes also had less experience on average with social simulation games such as *The Sims* (1.97) than those that completed the study (2.58), t(51)=2.05, p = .05. However, I do not believe either of these differences are practically significant in explaining why certain participants did not complete the study.

Game Data

I replayed the 48 game transcripts of the 24 complete participants and gathered various measures of Marlinspike's story threads and event history structure at the end of each game. (Due to a programming bug, two of these game sessions were replayable but it was not possible to gather the subsequent story data. Therefore, the only data available for those two games is the time spent and the number of commands entered.) This game data summarizes the nature of the games played and the system's performance in terms of its own measures—story structure completeness and unity.

Time Spent and Commands Entered

The mean time spent on a game session was 21.5 minutes (SD = 16.3 minutes), with a max of 94.1 minutes and min of 5.5 minutes. (These measures were calculated after dropping an extreme outlier: a game session that lasted over 5 hours. This participant likely took a long break during the game.)

Participants spent significantly more time on the first game (M = 28.1 minutes) than on the second game (M = 15.2 minutes), but this is to be expected since the first game also included a introductory tutorial session while the second game did not. Although it was not possible to separately measure the time spent on the tutorial from the time spent on the game that followed it, it was possible to determine which commands were typed during the tutorial and which were typed during the game session proper.

The mean number of commands entered during the tutorial was 38.4 lines (*SD* = 18.6 lines), with a max of 87 lines and a min of 20 lines.

Disregarding the tutorial, the mean number of commands for the two game sessions as 75.2 lines (SD = 37.3 lines), with a max of 175 lines and a min of 27 lines. There was no significant difference between the mean number of commands entered for the first game (79.8 lines) compared to the second game (70.63 lines), t(21) = 1.26, p = .22. The difference between those sessions played with reincorporation on (65.79 lines) and those played with reincorporation off (84.67) was more noticeable, although also not significant, t(21) = 1.69, p = .11.

Dividing the total number of commands entered during a game session by the time spent on that session provides a measure of the player's speed of play in commands per minute (cpm). The mean speed of participants was 5.23 cpm (*SD* = 1.93 cpm), with a max of 9.85 cpm and a min of 1.86 cpm. There was a significant difference between mean speed during the first game (4.48 cpm) and the second game (5.94 cpm), t(23) = 1.00, p = .003. All of these speed measures include the tutorial as part of the first game session.

Thus, on average, participants spent about 30 minutes playing the tutorial and first game, followed by 15 minutes playing the second game. Although their second game was not significantly shorter in terms of commands entered, they played significantly faster. This greater speed during the second game makes sense since players were now familiar with the world and could skim many of the text descriptions of rooms, objects, and even some of the more common events.

Verbs, Actions, and Scenes Used

It is interesting to see how the potential range of *Demeter* verbs, actions, and scenes were utilized in practice. The following tables show the number of games in which each verb, action, and scene was used in the 46 games for which data was available, the maximum number of times each was used within a single game, and the mean number of uses for each in those games in which it occurred at least once.

Verbs	Games Used	Max Count In One Game	Mean Count in Games Used
Ask	0 (0%)	0	0
Attack	16 (35%)	4	1.63
Close	13 (28%)	4	1.62
Drop	4 (9%)	4	2
Eat	3 (7%)	1	1
Enter	44 (96%)	29	10.52
Examine	35 (76%)	33	8.11
Give	2 (4%)	2	1.5
Go	45 (98%)	32	10.71
Insert	0 (0%)	0	0
Kill	4 (9%)	1	1
Kiss	2 (4%)	1	1
Knock	3 (7%)	1	1
Lock	1 (2%)	2	2
Look	46 (100%)	33	7.46
Open	43 (93%)	10	5.14
Push	1 (2%)	1	1
PushDir	2 (4%)	4	3.5
PutOn	0 (0%)	0	0
Rape	0 (0%)	0	0
Take	33 (72%)	7	2.36
Talk	36 (78%)	25	9.72
Tell	0 (0%)	0	0

Table 12. Verb use

Verbs	Games Used	Max Count In One Game	Mean Count in Games Used
Touch	0 (0%)	0	0
Unlock	27 (59%)	5	1.59
Use	0 (0%)	0	0
Wait	43 (93%)	41	11.21

Table 12. (Continued) Verb use

These measures reflect only those verbs used directly by the player. For example, players were instructed in the tutorial to use the Talk verb to converse with other characters. This provided a menu of options which then produced appropriate Ask and Tell deeds behind the scenes. So Ask and Tell were still used internally by most games, but they were not used directly by any of the players.

The mean number of unique verbs used during a single game was 8.74 verbs (SD = 2.37 verbs), with a max of 14 unique verbs and a min of 2 verbs. From the mean uses above, we can see that the most common player deeds involved waiting or moving between rooms, followed by talking to other characters and examining the world.

Actions	Games Used	Max Count In One Game	Mean Count in Games Used
ACQUIRE	33 (72%)	10	3.39
AFFRONT	44 (96%)	29	10.68
ASSAULT	8 (17%)	4	1.75
ASSIST	3 (7%)	5	3
ATTEMPT	32 (70%)	10	3.59
BATTERY	11 (24%)	2	1.36
CHANGE_STATE	7 (15%)	2	1.14
CONVERSE	45 (98%)	53	15.87
DAMAGE	2 (4%)	1	1
DEFY	0 (0%)	0	0
DELAY	45 (98%)	41	12.71
END_STATE	37 (80%)	7	2.89
ENDEAR	46 (100%)	37	12.41
EXAMINE	46 (100%)	49	16.43
HARASS	2 (4%)	1	1
HINDER	0 (0%)	0	0
INTERACT	0 (0%)	0	0
LOSE	6 (13%)	5	2.83
MANIPULATE	43 (93%)	20	7.93
OFFEND	31 (67%)	24	11.13
OPPOSE	41 (89%)	37	12.12
ROMANCE	1 (2%)	1	1
START_STATE	28 (61%)	8	2.57

Table 13. Action use

Actions	Games Used	Max Count In One Game	Mean Count in Games Used
SUPPORT	45 (98%)	67	21.13
TRAVEL	46 (100%)	100	40.37
USE	0 (0%)	0	0

Table 13. (Continued) Action use

The mean number of unique actions resulting from PC deeds during a single game was 13.1 actions (SD = 2.25 actions), with a max of 17 unique actions and a min of 8 actions. All games involved some influencing of NPC opinions. For example, all games had at least one ENDEAR player action, which would increase the affinity of the recipient NPC for the PC.

Scenes	Games Used	Max Count In One Game	Mean Count in Games Used
A_Long_Night	29 (63%)	1	1
Awaiting_GoParty	1 (2%)	1	1
Captains_Message	46 (100%)	1	1
Discussion_Concludes	24 (52%)	2	1.17
Discussion_Continues	39 (85%)	13	3.95
Discussion_Curtailed	9 (20%)	3	1.44
Discussion_Interrupted	31 (67%)	4	1.94
Discussion_Offscreen	17 (37%)	2	1.29
Discussion_Query	13 (28%)	2	1.23
Discussion_Starts	43 (93%)	8	2.65
Evidence_Revealed	46 (100%)	8	3.67
GoParty_Departs	30 (65%)	2	1.07
GoParty_Eviction	0 (0%)	0	0
GoParty_Moves_Along	46 (100%)	54	18
GoParty_Offscreen	18 (39%)	6	3
GoParty_Reentry	15 (33%)	1	1
GoParty_Reports	24 (52%)	3	1.25
GoParty_Requests_Follow	23 (50%)	10	2.17
GoParty_Returns	29 (63%)	3	1.24
Impromptu_GoParty	23 (50%)	3	1.57
Landfall	37 (80%)	1	1
NPC_Annoyed	3 (7%)	1	1
NPC_Defends	3 (7%)	4	2.33

Table 14. Scene Use

Scenes	Games Used	Max Count In One Game	Mean Count in Games Used
NPC_Defends_Other	3 (7%)	4	2.33
NPC_Defied	0 (0%)	0	0
NPC_Interdicts	3 (7%)	2	1.67
NPC_Observes_Destruction	2 (4%)	1	1
NPC_Offered_Item_By	2 (4%)	2	1.5
NPC_Outraged	3 (7%)	5	2.67
NPC_Proposes_Plan	43 (93%)	20	6.23
NPC_Rebuffs	1 (2%)	1	1
NPC_Replies	17 (37%)	3	1.41
NPC_Replies_To_Opinion	45 (98%)	63	18.8
NPC_Replies_To_Plan_Action	17 (37%)	4	2
NPC_Wooed	0 (0%)	0	0
NPCs_React	45 (98%)	32	10.76
PC_Dies	9 (20%)	1	1
PC_Unconscious	1 (2%)	1	1
Pursues_Plan	46 (100%)	12	5.33
Revenant_Acts	46 (100%)	16	5.04
Revenant_Attacks	27 (59%)	5	1.93
Waiting_through_the_Day	30 (65%)	1	1

Table 14. (Continued) Scene Use

This list of scenes also includes all reactions and components. These scene results hint at some of the under-utilized story paths. For example, most of the NPC_* scenes are reactions, and these were played in very few games. Most of the exceptions to this—such as NPC_Proposes_Plan and NPC_Replies_To_Opinion—stem from discussions where NPCs respond to each other. In general, participants did not interact with NPCs non-verbally.

Participants were also very active—only one game contained Awaiting_GoParty in which a PC did not form, join, or follow an exploration GoParty into the Zeppelin. Of the two possible ending scenes, the "successful" ending, Landfall, occurred four times more often than PC_Dies.

The mean number of unique scenes (including reactions and components) played during a single game was 19.3 scenes (SD = 3.5 scenes), with a max of 26 unique scenes and a min of 9 scenes.

No significant differences in the number of unique verbs, actions, or scenes used per game were found between the two experimental groups, between first and second games, or when reincorporation was on or off. Thus, the range of content used within games did not vary systematically across the different conditions.

Internal Story Structure

All 46 game sessions were complete stories. That is, each contained a single "main" story thread of events that connected the beginning scene to an ending scene with one or more middles scenes between them. (This is not always the case with *Demeter*. For example, two of the games played by the incomplete participants contained starting and ending events that were not connected by a single thread.)

The following table shows various measures of story length.

Measure	Mean	SD	Max	Min	Reinc ON	Reinc OFF	Reinc Significance
Root Events	84.7	34.21	173	40	72.55	95.83	<i>t</i> (21) = 2.39, <i>p</i> = 0.03
Root Scenes	35.54	14.39	71	17	30.41	40.25	t(21) = 2.41, p = 0.03
Player Deeds	61.43	28.89	149	23	52.36	69.75	t(21) = 2.06, p = 0.05
Total Events	349.54	162.78	820	131	298.18	396.63	t(21) = 2.10, p = 0.05

Table 15.	System	measures	of	story	length
-----------	--------	----------	----	-------	--------

As described previously, events are represented in Marlinspike by a tree-like structure of recasts and sub-events. These sub-events represent components of a scene or other interpretations of an action. The top-most event in this tree

structure is the root event that represents either the initial player deed before recasts or the parent scene that includes the various components.

Therefore, the number of Root Events indicates how many whole events occurred during the story. Total Events also includes a count of all the recasts and other sub-events that were components of those root events. Root Scenes is a count of those root events that were scenes. Player Deeds is a count of those player commands that resulted in a deed reported to the drama manager and thus represented as an action at the story level.²²

These four measures are very closely related. Since a story is formed by player actions and resulting scene responses, these counts will correlate with the number of root events, as will the number of total events.

Games were significantly shorter in all measures when reincorporation was on versus when it was off. This was unexpected, but it makes sense in retrospect. In certain situations—such as during a discussion—irrelevant scenes—such as Pursues_Plan—are randomly selected when reincorporation is off. Thus, when reincorporation is off, a discussion may be punctuated by occasional moanings from an NPC who is pursing the plan of inaction. In contrast, only discussion scenes tend to play during a discussion sequence when reincorporation is on. This is one example of how certain parts of the story can be extended by less relevant scenes when reincorporation is off. Since the player is prompted to act after each scene, these extra scenes will also increase the number of player deeds required to complete the story.

²² Occasionally, in *Demeter*, more than one deed will be combined in a single action root event. This is because certain verbs—such as Look and Examine—do not always warrant a drama manager response. So it is possible to have two Examines followed by a Talk combined under a single action root event. For this reason, Root Scenes + Player Deeds does not always equal the number of Root Events.

The next table shows a number of related measures of the stories' unity.

Measure	Mean	SD	Max	Min	Reinc ON	Reinc OFF	Reinc Significance
Main Thread Size	20.93	12.96	67	6	28.64	13.88	<i>t</i> (21) = 4.83, <i>p</i> < 0.001
Main Thread Weight	13.74	3.3	25	9	15.59	12.04	<i>t</i> (21) = 4.32, <i>p</i> < 0.001
Threads Spliced	2.09	1.95	8	0	3.23	1.04	<i>t</i> (21) = 3.85, <i>p</i> < 0.001
Extra Threads	9.04	5.59	25	2	5.45	12.33	<i>t</i> (21) = 5.14, <i>p</i> < 0.001
Unthreaded Unique Weight	41.15	27.49	124	9	22.82	57.96	<i>t</i> (21) = 5.37, <i>p</i> < 0.001

Table 16. System measures of story unity

Main Thread Size is the number of root events in the thread that contains the ending scene of the story. As mentioned above, all these main threads also contained the beginning scene. The Main Thread Weight is equal to the import of the event with the highest import in the main thread, plus 1 for every four events in the main thread. Thus, Main Thread Weight is closely related to the length of the main thread.

Threads Spliced is the number of times during the story that two threads were successfully combined to form a single thread. Extra Threads are the number of threads besides the main thread that existed at the end of the story. Ideally, this value would be 0. However, as described in Chapter V, there are reasons stemming from how scene preconditions are authored that many short threads can be left unreincorporated even with a fairly passive or cooperative player. Unthreaded Unique Weight is the sum of the weight of the unique material for each Extra Thread. Thus, this measure is closely correlated to the number of extra threads, but it also reflects the import of the events of that unthreaded material.

As shown above by the reincorporation means, turning reincorporation on made an extremely significant difference in the internal structure of the finished story. On average, twice as many root events were tied into the main thread, and three times as many threads were spliced together. There were also fewer than half the number of extra threads at the end of the story when reincorporation was used. Thus, reincorporation successfully produced a much more unified story in terms of the system's own measures.

There was no significant difference in either story length or story unity resulting from play order.

Player Agency

The first measure of player agency is world-level agency: the percentage of input attempts that successfully produced deeds at the story level.

Measure	Mean	SD	Max	Min	Play 1 Mean	Play 2 Mean	Play Significance
Percent of inputs that produced a deed	80.5%	10.7	100%	50%	76.6%	84.0%	t(21) = 3.00, p = 0.01

Table 17. System measure of world-level agency

While there was a wide variation between the participants, the mean percentage here indicates that 1 of every 5 commands attempted by players overall resulted in an error or otherwise failed to affect the world or story. The mean level of world agency significantly increased between the two games sessions, however.

The first requirement for story-level user agency is that the user perform actions of significant import.

Measure	Mean	SD	Мах	Min
Mean Action Import	2.46	0.16	2.89	2.12
Significant Action Count	3.19	3.15	12	0

Table 18. System measures of player action import

Mean Action Import is the average import of all of the player actions. Thus, it indicates the average import of the story-level effects of the player's deeds. Significant Action Count is simply the number of player actions with an import of 4 or higher. These are actions that would be significant enough to start their own threads if not already relevant to an existing thread.

These values show that the bulk of the players' actions were of fairly low import along the lines of traveling, exploring, manipulating objects, and interacting with NPCs in a mild manner. This is not surprising, since these are exactly the most common deeds performed by players, as shown above.

On average, players produced only 3 events of significant import per story. So the demand placed on Marlinspike to reincorporate all significant player events is not particularly high in an average *Demeter* game.

Neither of these two measures of player action import differed significantly between the first game and the second game or when reincorporation was used or not. This suggests that players did not significantly vary the kinds of actions they performed between different game sessions or in response to reincorporation.

Marlinspike's task is to then reincorporate user actions—particularly those of high import—into the finished story structure.

Measure	Mean	SD	Max	Min	Reinc ON	Reinc OFF	Reinc Significance
% of Actions in Threads	55.0%	11.8pp	76.1%	30.8%	59.2%	51.2%	<i>t</i> (21) = 2.05, <i>p</i> = 0.053
% of Actions in Main Thread	24.5%	16.2pp	63.1%	4.4%	36.8%	13.2%	<i>t</i> (21) = 7.02, <i>p</i> < 0.001
Count of Significant Actions in Main Thread	0.76	1.45	7	0	1.41	0.17	t(21) = 2.92, p = 0.008
% of Significant Actions in Main Thread	24.3%	35.8pp	100%	0%	43.6%	3.9%	<i>t</i> (21) = 3.97, <i>p</i> < 0.001

Table 19. System measures of story-level agency

These results show that Marlinspike was significantly more successful in this task when its reincorporation feature was used. First, a slightly higher percentage of all the players' actions were reincorporated into a thread when reincorporation was used. This difference is on the border of statistical significance.

However, the real goal for Marlinspike is to reincorporate user actions into the main story thread that connects the beginning and end of the story. When the reincorporation feature was used, the percentage of user actions reincorporated into the main thread increased threefold. For significant actions, this was an elevenfold increase.
While this performance improvement was both practically and statistically significant, it should be noted that many player actions were still not affecting the main story thread. On average, only about one third of all player actions and fewer than half of all significant player actions were made necessary to the main story thread in *Demeter*, even when reincorporation was used. So there is still some room for improvement here.

There was no significant difference between any of these means due to either play order or experimental group.

Participant Responses

The following is a summary of participant responses.

I looked primarily at the difference between responses given after playing *Demeter* with reincorporation on versus those responses given after playing with reincorporation off. I also tested for any order effects—whether responses were different after playing the first game session versus after the second game session—and for any group effects—whether one group answered differently than the other. I also looked for interaction effects. For example, did reincorporation perhaps only produce a significant effect when it occurred on the first or on the second play?²³

Overall, none of these differences were statistically significant (p < 0.05).²⁴ So the fact that the two groups were significantly different in their previous game experience did not produce any significant differences in their responses here. The one statistically significant finding was a difference in the frequency of input errors reported after the first and second game sessions. This finding is examined more closely below.

Results are presented here for each survey question. These results include a mean response score, a distribution of response scores, and total number of responses (*n*) for each question. (In the original survey, it was possible to answer N/A for some questions. Any N/A responses are not included in the count of total responses or in the means for that question.) The results also include the separate means for those responses given after playing with the reincorporation feature either on or off, although these two means are never significantly different.

²³ Actually, finding that reincorporation only has an effect on the first or second play would be the same as finding that reincorporation has a different effect depending on the participants' group.

²⁴ Or even p < 0.1, for that matter.

Story Structure

Participants rated the following statements as Strongly Agree (4), Agree (3), Neutral (2), Disagree (1), or Strongly Disagree (0).

Survey Question	Mean		Res C	spo ou	nse nt	•	n	Mean: Reinc ON	Mean: Reinc OFF	
		SA	Α	Ν	D	SD				
The events of the game had a story-like structure.	3.27	21	21	4	2	0	48	3.25	3.29	
The game session had a clear beginning, middle, and end.	2.48	10	15	15	4	4	48	2.67	2.29	
The events of the game were logically related to each other.	2.94	15	21	7	4	1	48	3.04	2.83	
Earlier events led to later events in a coherent and understandable way.	2.73	12	18	12	5	1	48	2.83	2.63	
The other characters' actions seemed to be consistent with their apparent goals and personalities.	2.70	9	21	11	3	2	46	2.78	2.61	

Overall, ratings were favorable. That the experience was story-like received the highest rating of any response in the survey.

The rating for the story's completeness—having a beginning, middle, and end was fairly low with a relatively wide distribution of responses. This may have been due to the fact that not everyone finds the possible *Demeter* story endings satisfactory. If the PC does not die, then the Landfall scene intentionally ends the story without answering the question of whether the passengers will be able to safely get down from the Zeppelin.

Survey Question	Resp Co	oonse ount	Resp Co Rein	oonse unt: c ON	Response Count: Reinc OFF	
	Yes	No	Yes	No	Yes	No
Were there any important events that seemed irrelevant to the main storyline?	4	44	2	22	2	22

Table 21. Responses regarding irrelevant story events

In agreement with the solid rating given above for the story's logical structure, only about 10% of responses indicated the story contained irrelevant events. All of the reports of irrelevant events came from Group 0, and two of the reports came from the same participant after both games. The following are the explanations given by the participants:

- The presence of the revenant seemed strangely irrelevant. I sort of assumed that we needed to dispose of him before he killed all the passengers, but he didn't seem to attack anyone. [#44, play 1, reinc ON, group 0]
- I didn't figure out the main storyline, so I have no idea [#50, play 1, reinc ON, group 0]
- I don't think it was so important that my character needed to go to the bathroom, by the time I found that WC I think I would have already peed myself. Also, why the book on the sink? It didn't help that I took it other than someone wanted it back. [#72, play 1, reinc ON, group 0] [This comment refers to the events of the tutorial, not the game itself, and so it was not counted above.]
- the way it ended just seemed weird, like not really an ending at all ... [#44, play 2, reinc OFF, group 0]
- In this game the screams would be distant and the direction undetermined. When I tried to follow the action I could find no results of the distant events that had been described to me. Most notably the screams on the other side of the hatch didn't result in the death of any characters other than the ones that had already passed in the previous game. Those characters seem more like set pieces than interactive events/characters. [#100, play 2, reinc OFF, group 0]

User Agency

World-level Agency

These questions aim to measure a positive sense of world-level agency. Participants rated the following statements as Strongly Agree (4), Agree (3), Neutral (2), Disagree (1), or Strongly Disagree (0).

Survey Question	Mean	F	Resp Co	oo our	ns nt	e	n	Mean: Reinc ON	Mean: Reinc OFF	
		SA	Α	Ν	D	SD				
I knew what actions were possible to perform within the game.	2.77	15	20	3	7	3	48	2.46	3.08	
I was able to construct commands that the game understood.	3.00	13	26	5	4	0	48	2.83	3.17	
I was sufficiently able to direct my character's actions in the game world, such as move from place to place, manipulate objects, talk to other characters, etc.	2.90	14	22	6	5	1	48	2.88	2.92	

Table 22. Responses regarding positive world-level agency

Participants were generally able to successfully control their characters in the world. However, given that the focus of an interactive drama is on story-level interactions, world-level input should fade into the background of the experience. These scores do not quite reflect such an ease of input. There was also a small secondary group that did not feel they knew what actions were even possible in the game. This is noticeable in the distribution of scores for the first question here.

The difference between reincorporation and non-reincorporation means for the first question in this section is comparatively large. However, as mentioned, responses to this question are not normally distributed. Using a robust test of statistical significance, Mann-Whitney U = 373, n1 = n2 = 24, p = 0.08, reveals that this finding is not in fact significant.

These measures all showed significant correlation with the system measure of world agency—the percent of commands that translated to valid deeds—that was

described earlier. Knowing what actions were possible, r(44) = 0.45, p = .002, and being able to sufficiently direct one's character, r(44) = 0.46, p = .001, correlated more highly with this system measure than being able to construct valid commands, r(44) = 0.36, p = .01. Presumably, players were capable of constructing some valid commands while still producing many invalid ones.

The next set of questions explores how frequently participants encountered different kinds of world-level agency problems. Responses include Never (0), Rarely (1), Occasionally (2), Frequently (3), Most of the time (4).

Survey Question	Mean		Res C	spo ou	nse nt)	n	Mean: Reinc ON	Mean: Reinc OFF	
		N	R	0	F	М				
I entered a command that caused an error message or that the game obviously did not understand.	1.71	6	15	15	11	1	48	1.88	1.54	
I entered a command that the game seemed to understand but that did not have the effect I intended in the story world.	1.32	14	10	17	6	0	47	1.29	1.35	

Table 23. Responses regarding negative world-level agency

It seems the majority of participants encountered some occasional trouble entering commands, but generally not excessively so.

There was a significant difference between the mean reported frequency of error messages depending on play order. The average response was 1.96 after the first game and 1.46 after the second game, t(23) = 2.63, p = 0.02. The meaning of this is fairly obvious: After learning what commands are possible during the first game, players are less likely to report entering invalid commands during the second game. This finding matches that found earlier in the game data by looking at the percentage of input commands that produced valid deeds.

As would be expected, both of these player response measures exhibited a significant negative correlation with the system measure of world agency. Reports of entering a command that caused an error, r(44) = -0.68, p < .001, corresponded more highly than reports of entering a command that did not have the intended effect, r(43) = 0.45, p = .002.

Story-level Agency

These questions examine the participants' sense of story-level agency. Participants rated the following statements as Strongly Agree (4), Agree (3), Neutral (2), Disagree (1), or Strongly Disagree (0).

Survey Question	Mean		Res C	spo Coui	nse nt	ļ	n	Mean: Reinc ON	Mean: Reinc OFF	
		SA	Α	Ν	D	SD				
My actions seemed to have a significant impact on the course of the story.	2.21	8	12	14	10	4	48	2.25	2.17	
I believe the story would have been different had I performed different actions.	2.83	14	19	8	7	0	48	2.96	2.71	
I believe the story would have been better had I performed different actions.	2.63	11	16	14	6	1	48	2.71	2.54	

Tahle	24	Resnonses	regarding	nositive	story-level	agency
Iable	24.	Responses	regarunng	positive	Sloi y-level	agency

On average, participants seemed fairly neutral regarding their impact on the story. However, note the wide distribution of opinions in this area, with almost as many responses indicating no significant impact as those that did. There was a more positive consensus that player actions were influencing the course of the story in at least some way, however.

Feeling that one's actions had a significant impact on the story correlated very strongly with the world-level agency responses regarding knowing what actions were possible, r(46) = 0.47, p = .001, and being able to sufficiently direct one's character's actions in the world, r(46) = 0.58, p < .001. This suggests that world-level agency is indeed a prerequisite for story-level agency. At the very least, those that feel high agency at one level are more likely to feel high agency at the other. However, reports that one is able to construct valid commands did not significantly correlate with having a significant impact on the story, confirming that there is more required for a sense of agency—especially at the story-level—than simply providing successful inputs to the system.

The next set of questions explores how frequently participants encountered story-level agency problems. Responses include Never (0), Rarely (1), Occasionally (2), Frequently (3), Most of the time (4).

Survey Question	Mean		Re: C	spo Cour	nse nt	•	n	Mean: Reinc ON	Mean: Reinc OFF
		N	R	0	F	Μ			
I entered a command that did something significant in the story world.	2.04	4	12	15	10	6	47	2.13	1.96
I entered a command that did something significant in the story world, but this action then failed to influence the other characters or subsequent events to the degree that I think it should have.	1.40	5	20	14	4	0	43	1.32	1.48

Table 25. Responses regarding negative story-level agency

Participants felt they performed significant actions in the world slightly more often than they reported producing input errors, which is good. In the context of an interactive drama, not every action performed will be significant. So an average rating of only occasional significant actions is not too disappointing, although it could still be improved. Also, most significant world effects were perceived to successfully affect the story (though, in retrospect, this second question could have been worded a little more clearly).

None of the question responses for either world or story-level agency corresponded significantly with the system measures of story agency. Responses indicating that one's actions seemed to have a significant impact on the course of the story did show the strongest correlation with both Mean Action Import, r(46) = -0.24, p = .1, and Significant Action Count, r(46) = 0.24, p = .1. But these are not significant. Furthermore, performing actions with a higher mean import overall actually has a *negative* correlation with a sense of agency. Unlike the system measures of world-agency, the system measures of story-agency do not match the experience of users. Simply performing high-import actions was insufficient to grant a sense of story agency.

The next few questions measure participants' perceptions of their own story-level agency in a specific context. Participants were asked to identify the single most memorable or notable event of the session they just played. (However, as shown in the following section, a handful of participants did not describe an event in answer to this question. For example, some described the mood of the game as a whole as the most notable "event".)

They were then asked the following questions in regards to that event. Possible responses include Definitely (3), Very likely (2), Likely (1), Neutral / I don't know (0), Unlikely (-1), Very unlikely (-2), and Definitely not (-3).

Survey Question	Mean	F	Res	por	ise	C	oun	t	n	Mean:	Mean: Reinc OFF	
		D	VL	L	N	U	VU	DN		Reinc ON		
This event occurred as the direct result of an action I performed in the game.	0.85	19	0	7	7	9	4	2	48	1.08	0.63	
If I played the game again, I could cause this event to happen again.	1.72	21	4	13	7	1	1	0	47	1.74	1.71	
If I played the game again, I could avoid this event or prevent it from happening again.	0.91	14	3	10	11	4	4	1	47	0.78	1.04	

Table 26. Responses regarding story-level agency in relation to each participant's most
memorable event

Looking at the distribution of responses, there seems to be two groups of participants here: those that felt a strong, clear ability to control the story and those that felt they only had a slight influence on the story. However, closer inspection showed that over 75% of those that answered Definitely to each of these three questions were referring to a memorable action they performed. Other participants were instead replying to these questions in regards to a memorable event that occurred to them. Thus, these questions were not as useful a measure of story-level agency as intended.

Satisfaction

Although a great many factors go into producing a satisfying game, I was interested, as author, in roughly how enjoyable this experience was as a whole. Possible responses include Strongly Agree (4), Agree (3), Neutral (2), Disagree (1), or Strongly Disagree (0).

Survey Question	Mean	Re	spo	nse	Со	unt	n	Mean:	Mean: Reinc OFF	
		SA	Α	N	D	SD		Reinc ON		
I enjoyed playing this game.	2.54	12	12	15	8	1	48	2.5	2.58	

	Table 27.	Responses	regarding	game	satisfaction
--	-----------	-----------	-----------	------	--------------

Responses were very stable between the participants' two game sessions, r(22) = .83, p < .001. Four participants dropped their rating by 1 point on the second game, two participants increased their rating by 1 point, and one participant increased by 2 points.

As author, I think a **1/2 rating is pretty fair review for this game. That is about what I would give it too. I am glad that 25% of the participants enjoyed it very much and that the one person that despised the game increased his rating to simply disliking it on the second play.

Interestingly, satisfaction correlated with a number of other responses. However, the most significant correlations (r(46) > 0.50, p < .001) were with responses regarding whether events were logically related to each other, r = 0.53, whether earlier events led to later events in a coherent and understandable way, r = 0.65, whether the NPC's actions seemed to be consistent with their apparent goals and personalities, r = 0.68, and whether the player's actions had a significant impact on the course of the story, r = 0.57. Although no claim of causation can be made from this, satisfaction is at least correlated with a sense of agency within a well-formed story with believable characters. Thus, it is encouraging to learn that those who enjoyed the game tended to have exactly the experience *Demeter* was designed to provide.

Open-ended Responses

Participants were also asked a number of open-ended questions. A number of common themes in these results were tagged and are summarized here. The full list of all responses and their corresponding tags are available in Appendix G.

Single most memorable or noteworthy event

Of the 48 total responses, 41 responses identified some sort of event in answer to this question. Of the remaining 7 responses, 5 referred instead only to the descriptions or the pervasive mood of the game, and 2 felt that there was nothing memorable.

Of the 41 events reported, 17 referred to either some action the participant performed or a direct consequence of that action. The other 24 responses reported events of the game that happened to them.

19 of the 41 reported events involved the revenant somehow—either hearing it offscreen, encountering it for the first time, killing it, or being killed by it. This is fitting, since the revenant represents the central conflict of the story.

11 of the 41 events mentioned some aspect of the passenger NPCs. 7 of these concerned some autonomous action on part of an NPC, while 2 of them commented on the flatness or unresponsiveness of NPCs. The remaining 2 found simply interacting with them in some way—either in one-on-one conversation or attacking them—to be most memorable.

16 of the 48 responses mentioned some aspect of the mood of the story, including noteworthy scenery objects or well-written descriptions. 5 responses involved a sense of frustration or powerlessness. 4 referred to differences between the two game sessions as being most memorable, though one of these despaired that there was no perceivable difference. Only 2 mentioned the text-based input mode—one was gratified to see his command successfully converted to a deed, and other mentioned some minor input problems as why later events unfolded as they did. Finally, 3 referred to events of the tutorial as being most memorable.

Least Enjoyable Aspect of the Game

5 responses did not list any least-enjoyable aspect (including one 1 response that jokingly wished the game were longer). Most of these empty responses were from second-play sessions.

The most common complaint was the text-based input, as mentioned in 12 of the 48 responses. Closely related to this were 10 responses that stated the available actions and options of what to do in the game were not clear. 2 of these 10 responses mentioned unclear options only in reference to NPCs.

4 responses stated that reading so much text was the least enjoyable aspect. 8 responses complained of the difficulty in visualizing the story world and getting oriented in that virtual world. 3 of these orientation-related complaints were not

spatially-oriented—2 complained that the time between scenes was not predictable, and 1 found the story itself disorienting.

4 responses stated that they wished they were provided with a clearer objective of what they were meant to accomplish in the game. 7 complained that they could not actively direct the story, could not exert the influence they wanted to, or that their actions had no apparent consequences.

Finally, 7 responses reported that the resolution or ending of the story was the least satisfactory part of the game.

Most Enjoyable Aspect of the Game

Of the 48 responses, 7 did not list any most-enjoyable aspect.

12 responses found some aspect of player agency to be the most enjoyable part of the game. This ranged from the flexibility of the input system (1 response), mastering the commands needed (1 response), accomplishing tasks in the world (4 responses), the wide range of potential actions (2 responses), influencing or actively controlling the outcome of the story (3 responses), and simply "making things happen" (1 response).

Closely related to agency, 5 responses enjoyed simply exploring the world, including the potential reactions of NPCs. 5 responses also indicated that having an objective—attempting to accomplish some task or solve some problem—as the most enjoyable aspect.

5 responses found the story itself enjoyable, with 2 more responses enjoying either the autonomy of the NPCs or the way the story unfolded with no active participation required of the player.

4 responses mentioned the NPCs in a more neutral context—2 responses found interacting with the NPCs enjoyable, while the other 2 enjoyed the dialog or character believability.

6 responses mentioned enjoying the Zeppelin setting; 5 enjoyed the general mood; and 4 mentioned the descriptions of the world. 1 response enjoyed reading text.

Finally, 4 responses found variations between the two game sessions to be the most enjoyable aspect.

Other Questions

There were other optional open-ended questions in the response survey. While interesting to the author, the responses there did not noticeably cover any themes not already mentioned in one of the three questions above.

Discussion

Based on the empirical data presented here, it appears that Marlinspike's reincorporation feature worked correctly but that it had no discernible effect on players. In addition, *Demeter* successfully provided a story-like experience while still offering most players a sense of agency in that story.

Hypotheses

• H_1 : Reincorporation produces a better-formed internal story structure.

This hypothesis was supported by the fact that using reincorporation resulted in more concise stories with a greater number of events in the main story thread, more threads spliced together during the story, and fewer extra threads left over at the end of the story.

• H_2 : Reincorporation provides greater user agency at the story-level.

This hypothesis was supported by the fact that significantly more user actions are made necessary to the main story thread when reincorporation is used.

• *H*₃: Reincorporation produces higher user ratings of story coherence and structure.

Because user ratings of story structure did not significantly vary between the two reincorporation conditions, there was no support for this hypothesis.

• H_{4} : Reincorporation produces higher user ratings of story-level agency.

Because user ratings of agency did not significantly vary between the two reincorporation conditions, there was no support for this hypothesis.

Story Structure

Although there were clear significant differences in the system's internal representation of the story when reincorporation was used, there was no corresponding difference on any participant response measure. There are a number for possible reasons for this.

First of all, before any scene can play, all of its preconditions must be met. This means any scene that can play at a given point in the story is already going to be fairly relevant to the story. This is especially true given the limited number of possible scenes in *Demeter*, all of which are related to the same relatively small space of potential stories. Thus, when a reincorporation weighting is then used to select between the two or three scenes that could currently play, any improvement in selection will be subtle.

The other effect of reincorporation is narrating all hooks. This too is a subtle effect resulting in only an extra sentence or two for most scenes. These hooks make a big difference to the system's view of the story by providing explicitly-modelled connections between events. However, as suggested by Chatman (1978), audiences will often infer causality between events even if such a connection is not explicitly narrated. So explicitly narrating certain connections between scenes may add nothing significant to players' mental model of the story.

Because most reincorporation effects would be subtle for players, perhaps a different experimental instrument would have been useful here. For example, it may have been more informative to ask users about their experience of specific instances of reincorporation within a game session or to have them summarize their perceived story structure afterward. Asking about specific reincorporation instances would at least clarify whether players even noticed reincorporation effects in the story. It may also have been useful to ask users to explicitly rate how the two game experiences differed, rather on assuming any difference in experience can be measured with a rather coarse 5-point Likert scale. A larger sample size may also have helped, although effect sizes for most of the survey measures were fairly small.

In addition, while very convenient, a post-game survey as used here does have a number of inherent limitations. First of all, the users' ratings are summative of the complete experience after the fact. A survey assumes participants can reliably recall all of the salient details of their experience at the time of the survey. Since the same survey was used in both cases, there may be a priming effect at work that would lead participants to give the same responses on the second survey as they did on the first.

Different methods may have circumvented some of these limitations. Concurrent measures—such as using periodic in-game pop-up questions or having players

think aloud—would measure their experience as it unfolds. However, such measures are not without their own drawbacks as they might also break players' sense of immersion in the story.

Agency

Beginning with the medium, the text-based input was the least-liked aspect of the game. Both as measured by the system metrics and by participants' responses, the command error rate was fairly high—about 1 in 5 commands. Participant responses correlated significantly with the system measure of world-level agency. A number of participants were also unclear as to what actions could be performed. Although the error-rate generally decreased while play speed increased on the second play, these two problems of input and clearly afforded deeds remained substantial obstacles to world-level agency, which is a precursor to story-level agency.

On average, story-level agency was not rated very highly. Opinions varied widely on this, though. Actions performed by players produced about 35% of the most memorable events. Some sense of agency was the most enjoyable aspect of the game for 25% of participants—more popular than any other aspect of the game. However, feelings of inadequate agency was the least enjoyable aspect of the game for about 15% of participants. The system measures also suggested that *Demeter* could perhaps provide more regularly used actions of high import.

According to the poetics model, players should experience agency at each narrative level when material affordance for action are balanced by formal narrative constraints (Mateas 2004). Obstacles to world-level agency obviously limit the material affordances that are then available for story-level action. This is suggested by the strong correlation found between world-level and story-level agencies.

However, it was interesting to note that a number of participants also complained that they were not provided sufficient formal constraints. That is, 4 participants noted that the lack of a provided objective—that they didn't know what they were "supposed" to do in the game—was the least enjoyable aspect of the game. While my goal in developing *Demeter* was to provide an open world in which any action leads to a story, this is a reminder to all interactive drama developers that they may want to still provide at least an initial objective for players. However, the data here only showed that a lack of story-supplied objectives decreased enjoyment of the interactive drama in some cases. There was insufficient data to show that this decreased enjoyment was actually due to a lessened sense of story-level agency.

Participants' objectives may also provide a different interpretation of the participant complaint that potential actions were not apparent. In *Demeter*, most

players head fairly quickly up into the Zeppelin in order to investigate the death of the crew. Because the revenant may not appear immediately, players tend to wander back and forth along the length of the Zeppelin. If players are focused on this "find the killer" objective, then there are few obvious actions to perform in the Zeppelin to directly achieve this. In particular, none of the objects there readily afford a direct solution to this problem.

While it is a concern that agency may be limited in relation to the player's objectives (or lack thereof), it still seems that the greater problem for *Demeter* is that possible actions must be recalled rather than examined as a list of options. This system opacity is compounded by the fact that it is not immediately clear which objects are interactive. For example, if the player's goal is to find the killer, they could look at the corpses for clues or up into the gas bags for hiding places. Look was the first verb taught in the tutorial, yet few players try doing these things. The player will occasionally smell a strange smell or see the revenant scramble away up the girders of the Zeppelin. Yet only two players tried to smell something and none tried to climb the girders. This is probably because most players simply did not think to try these things because they were not obviously possible.

Therefore, although significantly more player actions were made necessary to the internal story structure when reincorporation was used, players did not report a corresponding difference in their sense of agency. This is may be due to the somewhat limited world-level agency achieved in *Demeter*. The limitations of a post-game survey experimental design, as discussed above in relation to story structure, may also be at work here too.

Narrative

Reincorporation effects aside, player responses were generally positive that the game experience was story-like overall. Responses were mildly positive that the story was well-formed with few to no irrelevant events. The story's ambiguous "cliffhanger" ending caused some complaints, though. However, it was interesting to find that, beyond the events of the Action level, all other narrative levels also captured players' attention.

First of all, characters are important, although they did not make or break the game. That is, the NPCs were rarely mentioned as either the most or the least enjoyable part of the game. The autonomy of NPCs made for some memorable events, but their general two-dimensionality and lack of believability in certain contexts was also a cause for some complaints.

The Zeppelin setting and some scenery items were both memorable and enjoyable for many participants.

The descriptions of both objects and events—that is, the narration and the suspenseful mood that it produced—was well-received. This is a reminder that manner was indeed an important level to reinstate in the poetics model of narrative.

The text-based medium was something of a barrier, though. Reading was disliked by some. It also likely contributed to the difficulty some participants experienced in visualizing and orienting themselves in the virtual world.

Thus, in conclusion, reincorporation was internally useful to the system in producing the story. Overall, the result was a successful interactive drama with a decent story structure. However, agency was somewhat impaired by the text-based interface. This means that Marlinspike's success at converting world-level deeds into story-influencing actions was likely occluded by the fact that many players did not have the material affordances necessary to experience agency at either level.

CHAPTER VIII. CONCLUSION

Although examples of interactive narrative exist in such forms as roleplaying games and improv theater, the development of robust computer-based interactive narratives is a relatively new endeavor. This development holds great potential, both in producing a new form of creative human expression and as a challenging exploration of narrative AI.

Contributions

The work presented here has made the following contributions to the development of a working interactive drama.

To begin, I reviewed the existing theoretical model of what comprises an interactive narrative. Although the foundations of this poetics had already been laid, it needed to be closely examined and then reformulated to correct the tensions introduced during its evolution. The overhauled poetics model proposed here more closely mirrors traditional narratology, as well as better informing system design.

I then proposed that Johnstone's (1979) improv reincorporation technique effectively builds connections of narrative necessity between events in the resulting story. This means the finished story will be well-formed because all of its events will be essential to that story, such that they could not have been left out without breaking the narrative unity.

Guided by this reincorporation insight, a solid poetics theory, and the successes of previous scene-based interactive drama systems, I developed the Marlinspike architecture. This design offers a number of interesting new features.

First is the translation of a player's world-level deed to story-level actions. This allows the system to provide multiple, explicit story-level interpretations of the same physical deed. This translation can be customized by the specific context of the story in which it occurs. This narrows the scope of this tricky AI problem of interpretation. Not all possible interpretations of a deed need to be considered but only those relevant to the current story context.

Secondly, by explicitly treating player actions as story atoms on par with the system's scenes, the player is no longer subservient to the author's scenes. This has the potential to increase player story-level agency. In most other scene-based approaches, players can only act within the bounds of scenes. In Marlinspike, the player can perform any legal deed at any time. This design implicitly mandates that the author consider player actions to be as significant to

the story as his own scenes. This focus on the player's contributions to the story encourages the author to then build upon what the player has done. This provides an important conceptual balance of power between the player and author. As Brenda Laurel (1991) suggested, the key to an interactive drama is not to author a story and then try to make it interactive. Instead, the author and the audience should be active in the same space, building the story together. Marlinspike's design reflects this ideal.

Finally, Marlinspike explicitly models the story structure, complete with all action recasts and scene elements. The detail of this representation of events and the causal connections between them was only partly utilized in *Demeter*. However, such a model could be useful in a more sophisticated system. For example, NPCs could use this structure to trace the causality at work between events or to coherently summarize an earlier series of events.

The implementation of Marlinspike and the interactive drama, *Demeter: Blood in the Sky*, provided many practical lessons beyond the effects of reincorporation of the structure of story events. That work also suggested many improvements that could make Marlinspike a more robust interactive drama architecture.

The empirical evaluation of Marlinspike/Demeter showed that reincorporation was indeed important to forming an internal system story model of unified events. Like the lessons learned from the implementation, the evaluation results also highlighted that a successful interactive drama will be affected by all aspects of narrative—including its characters, setting, and manner—and not only by its events. The study also revealed that, while a text-based interactive drama can be successful, more attention first needs to be given to improving its affordances in order to provide adequate world-level agency.

Future Work

Theory

Reincorporation did lead to a better internal story structure, but this was not reflected in the reported experience of end users. However, as discussed previously, there are some alternate hypotheses that could explain this lack of effect. Regarding story coherence, the improvements from reincorporation were fairly subtle in *Demeter*—perhaps too subtle to be measured with the particular survey instruments used here. Regarding agency, story-level agency effects were potentially occluded by the world-level agency problems causes by the affordances of the text-based interactive fiction medium.

Thus, further work should be done to confirm the claims that reincorporation can really improve the story quality and player agency of an interactive drama. One way to do this would be to repeat the study with an improved interface, more extreme differences in reincorporation settings, and/or different measures of player experience. Another would be to run a study using a simple interactive narrative system in which players make explicit story-level choices, as in a hypertext novel or *Choose Your Own Adventure* story. For a short story, this would be a much simpler system to author. Any positive reincorporation effects found when the player interacts directly at the story level would very likely—though not necessarily—apply to an interactive drama in which the player interacts at the world level.

Marlinspike

Although using reincorporation as the primarily criterion for scene selection proved somewhat disappointing, I feel that Marlinspike still generally presents a solid architecture for further interactive drama research and development. This is mostly due to the contributions just discussed revolving around its verb-action architecture and its explicit story model. Therefore, I plan to continue development of Marlinspike.

The first step here is to re-implement it in a more robust language. In particular, I would like to explore using a scripting language to script parts of the system (such as scene contents) and a logic language to handle such things as recast rules.

This rewrite would provide an opportunity to fix a few internal weak points in the architecture. Some of these are merely small, technical irritants. For example, the event at EventHistory.current should equal EventHistory.indexOf(EventHistory.size()) for looping purposes. But most changes would be important feature additions. One needed feature is a robust searching architecture to quickly form lists of events and NPCs that match certain criteria. Changes in world and characters states should be separately represented from the actions that caused them. It should be possible to determine which characters were present at any given past event.

More generally, the connection between the virtual world and the drama manager needs to be clarified and tweaked. I would like to find a way to avoid the need for the drama manager to explicitly approve every player deed. This will likely require a design in which the author may preempt certain deeds from within the world, rather than having the drama manager deny them after the fact. Also, even attempted deeds should be reported to the drama manager by default, since these often represent important events too. And it needs to be possible to execute NPC deeds offscreen in the world. Once the bridge between the world and story levels has been clarified, I would like to be able to plug the drama manager into different front-end game engines so the same system could be used to manage text-based, 2D, or 3D interactive dramas.

Due to the relative ease of authoring and the positive feedback regarding *Demeter*'s descriptions and narration, I believe that text-based interactive dramas are indeed viable, especially for research and development games. However, a pleasant menu-based interface that indicates which objects support interaction and in what ways needs to be developed. This may also be a valuable contribution to the general interactive fiction community.

NPCs need more attention in the Marlinspike design. I believe NPCs can be made more believable, not by increasing their internal complexity, but by focusing on expressing their existing state and offering more fluid ways of interacting with them to change that state. If continuing with a text-based medium, exploring some non-textual means at hinting at NPC states and state changes would likely be valuable. Authoring each NPC separately, rather than pulling from a common pool of responses, would also help.

Continuing a process already started during the implementation of *Demeter*, the monolithic design of scenes needs to be broken down. Scenes should be authored in terms of their component parts. This suggests a deed-based approach, where NPC behavior is specified in terms of deeds with well-defined effects. One advantage of this would be a clearer, more detailed event history. Another is that NPCs and players would leverage the same casting and recasting rules to actions. Finally, this could lead to a scene-authoring approach that involves scripts of NPC deeds, or even more sophisticated planning for NPCs to reach author-specified story goals. Or NPCs could become more autonomous, suggesting their own potential actions and reactions, with the drama manager simply coordinating them within the current scene. In short, rather than remaining as story atoms, I see scenes becoming a framework for a particular segment of the story. Gradually, scenes may even atrophy away, leaving only the more general architecture of deeds, actions, recasts, and story goals.

The scene selection process also needs to be re-examined. Reincorporation is important, but other factors are at work here, such as the need for immediate reaction to some events, as well as a need to move the story forward with new material that is not always directly related to what came before.

In general, I envision a much more modular system where each component drama manager, scene selection process, front-end game engine, etc—can be independently swapped out to test different approaches. Eventually, authoring tools specific to such an interactive drama system would also be useful. However, it seems premature to design authoring tools while the underlying drama system itself is still evolving. This project has also clarified my vision of what an interactive drama architecture should be. I see interactive drama as a new creative medium that explores the narrative effects of choices made by an active, roleplaying user. But, ultimately, this experience is authored by a human. This is a long, complicated task. The interactive drama architecture should not then attempt to usurp the role of that author. Instead, it should serve as a tool to extend the reach of the human author, responding to players in the author's absence. It allows the author to craft an experience he would otherwise be unable to provide.

I believe that interactive drama systems should gradually provide the author with more and more tools to work his trade. Artificial intelligence components will be developed to handle more than just selecting the next pre-authored component. Eventually, the author will be able to delegate to the system more and more low-level authoring tasks—such as managing NPC plan executions or giving NPCs the ability to coherently summarize previous events. The author will then be free to focus on his higher level vision.

Marlinspike and the work presented here is but one step in this direction.

APPENDIX A: EXAMPLE SCENE: PC_UNCONSCIOUS

Chapter IV describes the scene PC_Unconscious as an example of a scene's internal complexity. As a longer example, the following is the original Inform code for this scene. Lines beginning with ! or !! are comments.

```
!! Scene: PC Unconscious
1.1
!! The PC succumbs to injuries that prove non-fatal.
11
!! > MEvent(attacker: PC Unconscious[?], PC, rescuer?, ?)
!!
!! The PC will recover consciousness in various states depending
!! on who knocked her unconscious in the first place. The detail
!! will indicate the nature the event: +1 is an active rescuing, 0 is
!! neutral (no rescue possible), -1 is an imprisonment, and -2 is a
!! betrayal (left behind). The LOOK event contained within this event
!! will give the awakening location.
11
!! Pre:
!! * An Attack on PC
!! * PC is incapacitated at 0 health.
!! * (But usually invoked directly as an interruption by
!!
   <DT PC Falls Inactive>)
!!
!! Play:
!! First, heals PC back to 1 health and advances the <DemeterTime> by
!! one.
!!
!! If attacker was a passenger NPC:
!! That NPC disarms the PC (if armed).
11
!! If PC was in the gondola or observation deck, the PC will then wake
!! on the locked observation deck.
1.1
!! Otherwise, depends on surrounding NPCs. If at least one has a
!! positive _average(morality, affinity(PC))_, this "rescuer" will have
!! dragged the PC back to obs deck. However, if no one cares enough
!! about the PC, she'll wake up where she fell (probably locked out:
!! any GoParty will advance silently until it has returned).
11
!! If attacker was Revenant:
!! If no NPCs around, this becomes <PC Dies>. Otherwise, the revenant
!! gets another offscreen <Revenant Acts>. If there is still an NPC
!! standing, then will check affinity/morality and proceed accordingly,
!! as described above. If a rescuer is found, PC will wake in gondola
!! in their bunk. Rescuer will be there to narrate what happened.
1.1
!! Import 4; Imperative 9.
!!
```

```
MMiddleScene PC Unconscious
  with
    import 4,
    imperative 9,
    cause nothing,
    canPlay [;
      if (DCharacter__PC.health == 0 &&
          DCharacter__PC has incapacitated) {
        self.cause = MEventHistory.find(0, nothing, nothing, DV ATTACK,
                                          DCharacter PC);
        if (self.cause) {
         return MList new(self.cause);
        }
      }
    ],
    !! Method: getRescuer(victim, bystanders)
    11
    !! Looks through the list of <DNPC> bystanders for the NPC with
    !! the highest average(morality, affinity(victim)) . Returns
    !! said NPC, or nothing if no one has an average > 0.
    !!
    getRescuer [ victim bystanders
      i len npc moral aff highest highestNPC;
      highest = 0; !needs to be at least higher than this
      highestNPC = nothing;
      len = bystanders.size();
      for (i = 0 : i < len : i++) {
        npc = bystanders.get(i);
        moral = npc.morality.getValue();
        aff = (npc.affinities-->victim.affIndex).getValue();
        aff = (moral + aff) / 2;
       if (aff > highest) {
         highest = aff;
         highestNPC = npc;
        }
      }
     return highestNPC;
    ],
    !PLAY
    play [event
      witnesses i sub npc wakeIn disarmedOf hooks deckers;
      event.actress = self.cause.actress;
      event.dirObj = DCharacter PC;
      witnesses = DNPC GetActive();
```

```
if (event.actress == Revenant) {
  if (witnesses.size()) {
    npc = witnesses.get(random(witnesses.size()) - 1);
    !revenant gets another action while everyone else responds
   event.add(Revenant Attacks.playAt(event.loc, true, npc));
    if (npc.health <= 0) {
     !that npc just fell too
     witnesses.remove(witnesses.indexOf(npc));
   }
  ļ
  if (witnesses.size()) {
   !still someone standing that might save you
   event.secondObj = self.getRescuer(DCharacter PC, witnesses);
  }else {
   !left alone with Revenant: PC Dies
   MList.destroy(witnesses);
   sub = MEvent new(DCharacter PC, PC Dies, nothing,
                       nothing, self.cause, event.loc);
   hooks = Ending Hooks.getHooks();
    sub.predicate.play(sub);
    if (hooks) {
     !manual/lazy/play-time reinc
      sub.reincorporates.join(hooks); !destroys hooks
    }
    event.add(sub);
    event.detail = 0;
   return;
  }
  if (event.secondObj) {
   !will be dragged back to safety. Even if already in gondola.
   wakeIn = YourBottomBunk;
   event.detail = 1;
  }else {
   !will be left behind
   wakeIn = event.loc;
   event.detail = -2;
  }
}else {
  !attacked by an NPC
  !remove any weapons from player
  ! (but can't remove from objectloop)
  disarmedOf = MList.create();
  objectloop (i in player) {
   if (i ofclass DWeapon) {
     disarmedOf.add(i);
   }
  }
  for (i = 0 : i < disarmedOf.size() : i++) {
   move disarmedOf.get(i) to event.actress;
   event.add(MEvent new(event.actress, DA Manipulate, DV TAKE,
                disarmedOf.get(i), DCharacter PC, event.loc));
  }
```

```
event.secondObj = self.getRescuer(DCharacter PC, witnesses);
  if (event.secondObj || event.loc ofclass DGondolaLocation) {
    !"rescuer" or gondola NPC will drag you to obs deck
    wakeIn = Observation Deck;
    event.detail = -1;
  }else {
   !will be left behind
    wakeIn = event.loc;
    event.detail = -2;
  }
}
!remove any revenant (whether fell to NPC or revenant)
remove revenant;
!Fast-forward any go party
if (MTriggerManager.hasState(DST GoParty) &&
     DNPC SelectActive(DST GoParty.detail)) {
  !still someone active to lead GoParty
  !PC just dropped
  if (DST GoParty.detail.contains(DCharacter PC)){
    event.add (MEvent new (nothing, DA ChangeState, nothing,
                DST GoParty, DV DROP, event.loc,
                DCharacter PC));
    DST GoParty.detail.remove(DST GoParty.detail.indexOf(
                                DCharacter PC));
  }
  !make sure we can get in (no betrayal w/ unconscious PC)
  give hatch to Gondola ~locked;
  sub = MEvent new(nothing, GoParty Offscreen, nothing,
                    nothing, nothing, event.loc);
  GoParty Offscreen.reacts = nothing;
  GoParty Offscreen.play(sub, true);
  event.add(sub);
!heal/revive and advance time
print "Then everything goes black.^";
new line;
new line;
DCharacter PC.damage(-1); !prints waking message
DemeterTime++;
if (DemeterTime > 2) {
  DemeterTime = -1;
}
!In the meantime, regardless of where PC is going to wake up,
! move any NPCs in from deck
if (door to Observation Deck hasnt locked ||
    DNPC GetActiveNPCs(DGondolaLocation, true)) {
  !can get in, or someone to let them in
  deckers = MDramaManager.getPresent(true, Observation Deck);
```

```
for (i = 0 : i < deckers.size() : i++) {
    !Even if no one around to help, uncons. NPC wake long enough
    ! to drag themselves in
   move deckers.get(i) to deckers.get(i).getPassengerRoom();
  }
 MList.destroy(deckers);
}
!now, finally narrate waking
PlayerTo(wakeIn); !will produce a look at new location
if (wakeIn == Observation Deck) {
  !regardless of whether dragged, left after NPC attack,
  ! or fell to rev:
  give door to Observation Deck ~open;
  if (door to Observation Deck has lockable) {
   give door to Observation Deck locked;
  }
  if (event.loc ~= Observation Deck) {
    !was dragged here
    new line;
    print "Aside from your other injuries, your clothes are
      rumpled and pulled askew, your wrists are bruised, and your
      shoulders are sore. In short, it feels like you were
      dragged out here, though not very gently.^";
  ļ
  if (disarmedOf.size()) {
    if (event.loc ~= Observation Deck) {
      new line;
      print "It also ";
    }else {
      new line;
      print "It ";
    }
    print "seems that someone has relieved you of ";
    if (disarmedOf.size() > 1) {
     print "all your potential weapons.^";
    }else {
      print "your ", (name) disarmedOf.get(0), ".^";
    }
  }
}
new line;
print "You are not sure how much time passed while you were
  unconscious";
if (~~(wakeIn == Ladderway || wakeIn == WC ||
       wakeIn ofclass DZeppelinLocation)) {
  print ", but a glance at the sky ";
  if (wakeIn == Observation Deck) {
   print "all around you";
  }else {
    print "through the window here";
  }
  print " shows that it is now ";
```

```
switch (DemeterTime) {
    0: print "night";
    1: print "dawn";
    2: print "late morning";
    -1: print "sunset";
  }
}
print ".^";
if (wakeIn == YourBottomBunk) {
  event.add(MEvent__new(DCharacter__PC, DA_Travel, DV_GO,
                        Your Room, event.secondObj, event.loc));
  move event.secondObj to Your Room;
  new line;
  print (name) event.secondObj, " leans over you.^";
  event.secondObj.speakTo(DCharacter PC);
  print "You're awake! I was worried there for a while...~^";
  !a brief manual report (too hard to use GoParty Reports
  ! properly here, since the GoParty fast forward happens within
  ! this scene).
  event.secondObj.speakTo(DCharacter PC);
  print "After you fell, that thing also attacked ";
  npc = event.get(0).dirObj;
  if (npc == event.secondObj) {
   print "me";
  }else {
    event.secondObj.printAddressFor(npc);
  print " while I tried to drag you clear.";
  if (npc == event.secondObj) {
   print "~^";
   print (name) npc, " shows you ", (my) npc, " wounds.^";
   npc.speakTo(DCharacter PC);
    print "Then, I don't know why, but it suddenly turned
      and took off! We were both lucky, I guess.~^";
  }else {
    print " I yelled, and for some reason, that suddenly scared
      the thing off. ";
    if (npc.health <= 0) {
      print "But... it wasn't enough to save ", (name) npc, "! ";
      if (npc.health == 0) {
        !expired
        npc.damage(1, true);
        event.get(0).verb = DV KILL;
      }
     print (CI) npc, "'s dead!~^";
    }else {
      event.secondObj.printAddressFor(npc);
     print " was injured, but ", (I) npc, " is still alive.~^";
    }
  }
}
!Otherwise, will wake where fell, which is pretty self
!explanatory
```

```
MList.destroy(witnesses);
if (disarmedOf) {
    MList.destroy(disarmedOf);
    }
],
;
```

APPENDIX B: REINCORPORATION IN DEMETER: BLOOD IN THE SKY

One of the goals of the game *Demeter* was to test whether the technique of reincorporation leads to better-formed stories and greater user agency. Therefore, it is handy to know exactly how the behavior of *Demeter* scenes changes when reincorporation is turned on. These changes include both what previous events the scene may optionally hook into the story, as well as the resulting changes in the scene's narration.

Scenes

A_Long_Night - Hooks a previous NPC_Wooed, if that lover NPC is still active with a high affinity for the PC and the PC is not in the Zeppelin. The lover will visit the PC during the night (possibly freeing him from the Observation Deck).

Discussion_Starts - Refers back to the last Discussion_Interrupted if that discussion has not yet been concluded. Provides a bridge comment to highlight that the NPC speaker is resuming the earlier discussion.

Discussion_Continues - Each major speaker refers back to the last speaker in either agreement or disagreement, thus connecting the different arguments of a discussion.

Discussion_Curtailed - Refers back to the formation of the current GoParty, whether formed by discussion including the PC or in direct response to the PC's actions. Suggests that since the PC was instrumental in forming the GoParty, he should know that the time for discussion has passed.

Discussion_Offscreen - Refers back to any Discussion_Interrupted, as per Discussion_Starts. This hook may be silent (unnarrated) if the PC refuses to move to join the discussion.

Evidence_Revealed - Will only play to have an NPC comment on another's attempt to open the locked hatch to the command gondola if reincorporation is on.

Ending Hooks - These are all used in various ways by the two ending scenes, Landfall and PC_Dies.

- 1) Any attacks directed at the PC (from NPCs or revenant)
- 2) Any killings of NPCs or revenant by the PC
- 3) Any killings of NPCs by the revenant
- 4) The death of the revenant

5) Successful NPC_Wooeds (each with no NPC_Rebuffs sub-events)6) All Evidence_Revealed scenes that provided evidence that is >= the PC's current evidence level

7) The most recent PC_Unconscious

8) Any GoParty_Reentry that lead to a betrayal of the GoParty.

9) Any subsequent GoParty_Returns that would have ended that betrayal.

GoParty_Departs - The end of the last GoParty. The NPC that calls for the GoParty to pause for a farewell will instead wish them better luck than the named members of the last GoParty (or, if the departing actor was part of the previous GoParty, better luck than she did on the previous trip).

GoParty_Eviction - This whole scene only plays if reincorporation is on; otherwise, only the normal NPCs_React is used to reply to the prerequisite antisocial behavior. A PC that is violent to other passengers will be dropped from the GoParty by its other upset members. A harassing PC is simply warned and will then be evicted on a second offense.

GoParty_Moves_Along - If the PC leads the GoParty into a cul-de-sac room in the gondola, the other members will question why.

GoParty_Reentry - 1) A previous GoParty_Reentry betrayal that doesn't have a GoParty end state after it. This means the same GoParty is still betrayed, so the locked out GoParty will be more verbally insistent about opening the hatch. (Only produces this narration if the PC is present.) If PC is present, will also mention any death of the revenant (in narration only).

2) If PC is on the gondola side of the hatch and the NPCs there tie on the vote to open the hatch, will preface asking the PC for a decision by referring to the PC's previous stance on leaving the gondola, on the revenant, and any harassment or assault of one of the locked out NPCs. (Due to hacky work-around, these hooks are not included from canPlay() but are reincorporated at play time.)

GoParty_Reports (as a component of GoParty_Returns) - Reports references to the following:

1) All Evidence_Revealed events

2) Drop of any members from the GoParty, including disregard by the PC of a GoParty request to follow.

3) Any violence to the revenant, including by the PC. If the PC slew the revenant, the reporting actor will thank the PC explicitly.

4) Additional actual sightings or attacks by the revenant beyond the Evidence_Revealed events

5) PC violence (Assault or Battery) to NPCs, excluding the revenant (only if PC is still in the GoParty, though).

These hooks may not be narrated if the PC is not present at the GoParty's return. (When reincorporation is off, this scene is little more than a vague "We're back, and things are bad.")

GoParty_Returns - Refers to a recent non-betrayal GoParty_Reentry (someone opened the hatch), though this is only narrated if the PC is already in the gondola stairwell as the GoParty returns.

Landfall - In addition to the general Ending Hooks:

- 1) PC damage to one or both engines.
- 2) PC damage to one of the gas bags.

3) Any acquisition of food by the PC (if the PC still has the food).

Works all these references into the Landfall narration in subtle ways.

PC_Dies - Includes all Ending Hooks, generally narrated as reflective last thoughts of the PC as he dies.

Revenant_Acts - If the revenant decides to attack, it will choose to attack whoever assaulted it last (if that person is still alive and present).

Waiting_through_the_Day - Hooks the acquisition of food for all characters that possess food. If characters have food, they will share if. If PC has some, the player will be asked whether he wants to share.

Reactions

Most of the reincorporation changes in NPC reactions involve giving and then enforcing an interdiction. This allows NPCs to grow increasingly agitated with the PC's behavior. Without reincorporation, NPCs tend to react the same way to repeated affronts--as if each one were the first time.

NPC_Annoyed - If this is the second Annoyed reaction by the same NPC to the same sort of deed, will issue an interdiction against a third instance of the behavior.

NPC_Defends_Other - The defended NPC will thank the defender. Also, if a defender issues an interdiction against such behavior, will actually set the appropriate recast triggers to enforce the recasting of a violation into a DEFY action.

NPC_Defied - If this is not the first time this NPC has been defied, she will refer back to the earlier NPC_Defied.

NPC_Rebuffs - Actually sets the appropriate recast triggers to enforce the recasting of a violation of the rebuff interdiction into a DEFY.

Casts and Recasts

These triggers are only active when reincorporation is on as they provide greater levels of response to the player's actions.

DT_PCStartsDiscussion - A Tell to all present will start a full discussion state, prompting a greater level of response from present NPCs.

DT_PC_Leads_GoParty_Departure - If a PC leads the GoParty from the gondola, the motion will be cast to an ATTEMPT and GoParty_Departs played to see the party off.

Silent Hooks

These hooks provide silent structural reincorporations but do not actually change the narration of the game.

Discussion_Starts - Hooks the most recent end of a GoParty.

NPC_Rebuffs - Hooks any previous rebuff by the same NPC of the same suitor in order to form a rebuff thread.

NPC_Wooed - Hooks any previous rebuff by the same NPC of the same suitor in order to form a wooing thread.

Narration-only Hooks

This "hook" provides additional narration when reincorporation is on, but doesn't change the story structure or any reincorporation connections. This can be seen as intra-scene reincorporation.

Discussion_Concludes - When a DST_GoParty is formed, one of the newly assigned party members (if different from the concluding speaker) will agree with the party formation. If there is another member of the party, this agreeing NPC will then prod the party to start exploring.

APPENDIX C: PARTICIPANT RECRUITMENT EMAIL

As described in Chapter VII, the following email message was sent to various email lists in order to recruit participants.

Subject: Research participants needed: Play a new game today!

This is a request for volunteers to participate in a fun doctoral research study.

THE RESEARCH:

I am a PhD student in the Communications and Information Science program at the University of Hawaii. My dissertation work is in the field of interactive narrative. That means I'm trying to get a computer to generate a new story in response to a human who is playing the role of a character in that story.

I have developed a text-based prototype game in order to explore a particular approach to this problem. But now I need some live players to help me evaluate it!

THE GAME: "Demeter: Blood in the Sky" In an alternate 1923, you are one of seven passengers aboard the Zeppelin-class airship, Demeter. Your flight across the Atlantic is pleasantly routine... until a gruesome dawn suddenly changes everything.

THE STUDY: Anyone can play! You can participate online from any computer with a Java plugin installed (which you probably already have).

You'll answer a short survey about your gaming and computing background, and then play though a text-based game twice, answering another short survey after each game session. I need each participant to play twice because I am comparing two different versions of the game.

The full study will take about 90 to 120 minutes to complete.

To get started, click this link: http://tinker2.zach.tomaszewski.name/cgi-bin/demeter.py

The study will only run until 15 Feb, so act now!

WHY SHOULD I CARE? There are so many good reasons for you to participate:

- * HELP a desperate hungry grad student finally graduate.
- * EXPERIENCE something new: Both a new story and a new way of telling stories.
- * DISCOVER something old: Text-based adventure games and interactive fiction have a 30+ years history and are still being produced by a small but active community. Have you ever played one?
- * SUPPORT local research in Hawai'i: What goes around, comes around.
- * RELAX and play a game, because you could probably use a break right now anyway.

Yes, you can have all of this and MORE! Click the link above.

QUESTIONS?

If you have any questions or want to report any problems, you can contact me at: ztomasze@hawaii.edu

Feel free to forward this email to others you think might be interested. Thanks for your time!

--Zach Tomaszewski

APPENDIX D: EVALUATION SURVEY: BACKGROUND

Consent Form

You are about to participate in a research study regarding interactive narrative. If you choose to continue, you will be asked to:

- 1. Complete a short background survey.
- 2. Play a text-based game named Demeter: Blood in the Sky
- 3. Complete a short response survey about your experience of the game
- 4. Play the same game a second time with slightly different settings
- 5. Complete the same response survey again

The complete study should take you about 90 to 120 minutes to complete.

Your participation is anonymous, and you are free to quit at any time.

You will need the Java plug-in (including Java Web Start) installed on your current machine to play the game. (Most people already have this.)

Details:

RESEARCH:

This research project is being conducted as a component of a doctoral dissertation. The purpose of the project is to investigate the effects of different algorithms on the generation of an interactive narrative experience.

PARTICIPATION:

Your participation should take approximately 100 minutes. Your participation will consist of:

- filling out a form on background information about yourself (10 minutes)
- playing a short, text-based computer game (40 minutes)
- filling out a form regarding your experience of the game (10 minutes)
- playing the same text-based computer game again (running with different settings) (30 minutes)

filling out a form regarding your second experience of the game (10 minutes)

Participation in this research project is completely voluntary. You are free to withdraw your participation at any time during the duration of the project with no penalty or loss of benefit to which you would otherwise be entitled.

RISKS:

The investigator believes that, beyond the risks associated with normal computer use, there is little or no risk to participating in this research project.

BENEFITS:

Participating in this research may be of no direct benefit to you. It is believed, however, the results from this project will help improve the design of interactive narrative systems.

CONFIDENTIALITY:

Your name will not be associated with your responses on the questionnaire forms or with any other generated research data. Your participation in the project will be confidential to the extent allowed by law. Agencies with research oversight, such as the UH Committee on Human Studies, have the authority to review research records. Record of your participation will be destroyed at the end of the project.

QUESTIONS:

If you have any questions regarding this research project, please contact the investigator:

Zach Tomaszewski ztomasze@hawaii.edu / (808) 923-5372. http://zach.tomaszewski.name/argax/

If you have any questions regarding your rights as a research participant, please contact:

UH Committee on Human Studies 2540 Maile Way, Rm. 253 |University of Hawai'i Honolulu, HI 96822 Phone: (808) 956-5007

I have read and understand the above information, and I agree to participate in this research project.

[]Yes
How did you hear about this study?

- () An email list. Please specify: _____
- () A personal email.
- () Other. Please specify: _____

Is this your first time participating in this study or playing Demeter?

- () Yes, this is my first time.
- () No, I have played/participated in the past.

NEXT

Your Background

Please answer the following questions regarding your computer use and gamerelated experience.

"I feel comfortable using a computer."

- () Strongly agree
- () Agree
- () Neutral
- () Disagree
- () Strongly Disagree
- () N/A

"On average, I use a computer for the following number of hours each day:"

- () Six or more hours a day.
- () Three to five hours a day.
- () One to two hours a day.
- () Less than one hour a day.
- () I rarely or never use a computer.

"On average, I spend the following number of hours per day playing online or computer games on a personal computer (PC):"

- () Six or more hours a day.
- () Three to five hours a day.
- () One to two hours a day.
- () Less than one hour a day.
- () I rarely or never play computer games.

"On average, I spend the following number of hours per day playing digital games on platforms other than a personal computer (such as on a gaming console, mobile phone, hand-held device, arcade game, etc.):"

() Six or more hours a day.

() Three to five hours a day.

() One to two hours a day.

() Less than one hour a day.

() I rarely or never play digital games on a non-PC platform.

"I have played approximately the following number of different games using a personal computer:"

() Over 100.

() 50 to 100.

() 20 to 50.

() 5 to 20.

() 1 to 5.

() I have never played a game on a personal computer.

"I have played approximately the following number of digital games on platforms other than a personal computer (such as on a gaming console, mobile phone, hand-held device, arcade game, etc.):"

() Over 100.

() 50 to 100.

() 20 to 50.

() 5 to 20.

() 1 to 5.

() I have never played a digital game on a non-PC platform.

	I have played a number of games in this genre	I have played at least one game in this genre for many hours	I have briefly played or watched someone else play a game like this	I know what this is, but I have never played one	l do not know what this is
Text adventure / interactive fiction (Zork, Adventure, games by Infocom or Legend, modern IF, etc.)	()	()	()	()	()
Adventure games (King's Quest, Space Quest, Monkey Island; AGS games, etc.)	()	()	()	()	()
Computer or console roleplaying games (Diablo, Final Fantasy, Grand Theft Auto, etc)	()	()	()	()	()
MUDs and/or MOOs	()	()	()	()	()
MMORPGs (World of Warcraft, Everquest, etc.)	()	()	()	()	()
Social simulations (The Sims, etc.)	()	()	()	()	()
First-person shooters (Doom, Quake, Half-Life; Tomb Raider, etc)	()	()	()	()	()

Please indicated how familiar you are with each of the following computer game genres.

Please indicated how familiar you are with each of the following traditional roleplaying game genres.

	I have played a number of games in this genre	I have played at least one game in this genre for many hours	I have briefly played or watched someone else play a game like this	I know what this is, but I have never played one	l do not know what this is
Branching novels and gamebooks (Choose Your Own Adventure series, etc.)	()	()	()	()	()
Table-top roleplaying games (Dungeons and Dragons, GURPS, etc.)	()	()	()	()	()
Live-action roleplaying games (White Wolf's Mind's Eye Theater, etc; dinner mystery games)	()	()	()	()	()
Improv (Improvisational theater; roleplaying training exercises, etc.)	()	()	()	()	()

Your gender:

() Male

() Female

Your age category:

- () under 18
- () 18-24
- () 25-34
- () 35-44
- () 45-54
- () 55-64
- () 65-74
- () 75+

The highest level of education you have completed:

- () Doctoral / Professional (PhD, MD, etc)
- () Graduate School (Masters, etc.)
- () University / College (Bachelors, etc.)
- () Community College (Associates, etc)
- () Some college, no degree
- () Graduated high school or equivalent
- () 12th grade or less (none of the above)

How do you feel about reading text on a screen like the one you are currently using?

() I regularly read text on a computer screen without trouble.

() I dislike reading text on a computer screen, but I can tolerate it for short periods.

() I avoid reading text on a computer screen whenever possible.

How often do you use a command line interface?

() I use a command line at least once a week for many different tasks.

() I use a command line at least once a week for a small number of tasks.

() I use a command line a few times a year, or I have regularly used a command line in the past.

() I rarely use a command line.

() I have never used a command line (or, I do not know what a command line interface is).

Thank you! After you click Submit, you should be redirected to another webpage where you can play the game.

SUBMIT

APPENDIX E: EVALUATION SURVEY: USER RESPONSE

Story Structure

Please answer the following questions about the game session of Demeter that you just finished playing.

What was the single most memorable or noteworthy event from the game session you just played?

Please briefly describe the event. What made it memorable?

Concerning the events and characters...

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	N/A
The events of the game had a story-like structure.	()	()	()	()	()	()
The game session had a clear beginning, middle, and end.	()	()	()	()	()	()
The events of the game were logically related to each other.	()	()	()	()	()	()
Earlier events led to later events in a coherent and understandable way.	()	()	()	()	()	()
The other characters' actions seemed to be consistent with their apparent goals and personalities.	()	()	()	()	()	()

Were there any important events that seemed irrelevant to the main storyline?

() Yes

() No

If you answered Yes to the previous question:

Which events seemed irrelevant?

[Optional] Please share any other comments you have regarding the game's story structure.

You may also add here any clarifications regarding your answers above.

.....

NEXT

User Agency

Concerning your interactions with the game...

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	N/A
I knew what actions were possible to perform within the game.	()	()	()	()	()	()
I was able to construct commands that the game understood.	()	()	()	()	()	()
I was sufficiently able to direct my character's actions in the game world, such as move from place to place, manipulate objects, talk to other characters, etc.	()	()	()	()	()	()
My actions seemed to have a significant impact on the course of the story.	()	()	()	()	()	()
I believe the story would have been different had I performed different actions.	()	()	()	()	()	()
I believe the story would have been better had I performed different actions.	()	()	()	()	()	()

	Never	Rarely	Occasionally	Frequently	Most of the time	N/A
I entered a command that caused an error message or that the game obviously did not understand.	()	()	()	()	()	()
I entered a command that the game seemed to understand but that did not have the effect I intended in the story world.	()	()	()	()	()	()
I entered a command that did something significant in the story world.	()	()	()	()	()	()
I entered a command that did something significant in the story world, but this action then failed to influence the other characters or subsequent events to the degree that I think it should have.	()	()	()	()	()	()

How frequently did you encounter the following situations?

Were there any actions you performed that had unexpected, surprising, or amusing results?

If so, please briefly describe one or more examples. Although unexpected, were the results still believable consequences of the action you performed?

On the previous page, you described the most memorable event of the game as follows:

[User's answer to "What was the single most memorable or noteworthy event from the game session you just played?" inserted here.]

Based on your impressions of the game so far, how likely do you guess the following to be?

	Defin- itely	Very likely	Likely	Neutral / I don't know	Un- likely	Very un- likely	Defin- itely not	N/A
This event occurred as the direct result of an action I performed in the game.	()	()	()	()	()	()	()	()
If I played the game again, I could cause this event to happen again.	()	()	()	()	()	()	()	()
If I played the game again, I could avoid this event or prevent it from happening again.	()	()	()	()	()	()	()	()

[Optional] Please share any other comments you have regarding your ability to direct your character's actions in the game.

You may also add here any clarifications regarding your answers above.

NEXT

Satisfaction

Concerning your overall game experience...

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	N/A
I enjoyed playing this game.	()	()	()	()	()	()

What was the LEAST enjoyable aspect of this game?

What was the MOST enjoyable aspect of this game?

[Optional] Anything else?

If you have any other comments to share about the game or this survey that didn't seem to fit elsewhere, please enter them here.

If this is your first time through this survey, you will be taken back to the game to play a second time after hitting Submit.

SUBMIT

Thank You!

You've made to the end!

Thank you so much for your time! I really appreciate your help in completing my research.

You have been playing a version of *Demeter* in which certain features were disabled for research purposes. The normal/full version is a little better (though not by much). To learn more about the research involved, visit the <u>Argax Project</u> [*link*].

If you want to receive a short summary of the final results of this study, you can <u>enter your email address</u> [*link*]. (To preserve your anonymity, this is handled elsewhere so that your email address is not associated with your responses here.)

APPENDIX G: OPEN-ENDED RESPONSES

From the 24 participants that completed all components of the study, the following are the 48 raw responses to three of the open-ended questions of the response survey. Each response is labeled with the participant's ID (#), whether the response was from the participant's first or second play session (P), whether the participant was in group 0 or 1 (G), and whether reincorporation was on or off during the corresponding game session (R). In addition, responses are tagged with common themes found across more than one response. See Chapter VII for a summary of these responses and the resulting conclusions.

#	Ρ	G	R	Most memorable or noteworthy event	Tags
1	1	1	0	Interesting plot - mystery, danger, murder	mood
1	2	1	1	exciting story line - danger, suspense, scary	mood
2	1	0	1	the major being killed	event
2	2	0	0	the beast hammering on my door, as i did not know if it would pound through the door or not	event, revenant(unseen)
8	1	0	1	The ending was very memorable, as my character died by being attacked by a revenant.	event, revenant(killed by)
8	2	0	0	nothing, at least i didn't die this time	none, frustration, difference
20	1	0	1	Using the restroom. I typed pee, and it recognized it as a command.	event, action, interface
20	2	0	0	Killing the revenant. Description, detail, and mood settings made the action feel as if I were playing an original Resident Evil game.	event, action, revenant(killing it), mood(description)
21	1	1	0	near the end of the game when your player was struck with fear while listening to the noises at night. I could vividly imaging this scene, and it was scary!	event, revenant (unseen), mood
21	2	1	1	this time around I got worked! it was memorable getting killed.	event, revenant(killed by), difference
44	1	0	1	kissing miss miriam. It was the one off the cuff thing I tried that worked; not that it had any larger narrative effect	event, action

Table 1.	Most	memorable	or noteworthy event
----------	------	-----------	---------------------

#	Ρ	G	R	Most memorable or noteworthy event	Tags
44	2	0	0	my failure to achieve anything. I got the knife and knew that roman had a hammer, but I couldn't get into the crates or the command module, and I didn't seem to have any way to get Roman to agree to a plan to do that. I had this idea that I might have been able to kill the revenant with the knife, open the hammer with the crates, get other interesting stuff, and possibly help us descend at the end by curring the gas bags, but nothing really happenend again.	event, action, frustration, no consequence, npcs(flat)
49	1	1	0	The sudden appearance of the revenant was the most memorable moment. Even though I expected something to happen, the fact that the revenant did not appear until after going through the entire vessel a few times caught me off guard.	event, revenant(meeting)
49	2	1	1	Finally killing the revenant, after it managed to escape on every previous encounter with it.	event, action, revenant(killing it)
50	1	0	1	plummeting to my death	event, action
50	2	0	0	Sense of danger at animal/bad guy lurking	mood
52	1	0	1	The most memorable event was the Revenant (spelling?) showing up suddenly after looking at all of the dead characters. It was unexpected, to say the least. I thought I was looking for a murderer on board.	event, revenant(meeting), mood(scenery)
52	2	0	0	Because I chose to wait, I wasn't the leader. This time it was the others who were leading me.	event, npcs(auto)
57	1	1	0	The revenant appearing and it's subsequent death. It was memorable because of all of the action associated with it, and because it seemed particularly significant to the story.	event, revenant(meeting, killing it)

Table 1. (Continued) Most memorable or noteworthy event

#	Ρ	G	R	Most memorable or noteworthy event	Tags
57	2	1	1	Getting attacked by the revenant.	event, revenant(attacked by)
58	1	0	1	The description of the revenant. The descriptions of everything are very well done.	mood(descriptions)
58	2	0	0	descriptions	mood(descriptions)
59	1	1	0	the scream my character heard. but there was no option to investigate it.	event, revenant(unseen), frustration
59	2	1	1	the faint knocking. because I had the option to do something or not	event, option
61	1	1	0	seeing the undead that would be shocking	event, revenant(meeting)
61	2	1	1	dying from revenant	event, revenant(killed by)
63	1	1	0	Talking to miss miriam during the tutorial. It was memorable because the choices available when talking to her were more diverse and interesting than at any other part in the story.	event, action, tutorial, options, npcs(flat)
63	2	1	1	hitting the revenant with the hammer a second time. It was memorable because it made no difference whether I hit it a second time with a hammer or without. I still died.	event, action, frustration, difference(none)
64	1	0	1	I attacked the Revenant and apparently I died. I liked the description of my death.	event, action, revenant(killed by), mood(description)
64	2	0	0	Defeating the Revenant. I died the last game, so it was good to survive this time.	event, action, revenant(killing it), difference

#	Ρ	G	R	Most memorable or noteworthy event	Tags
72	1	0	1	going into the dining room and having no exit before I found the wc. On the full version: the Captain's message of danger with no explanation.	event, tutorial, mood
72	2	0	0	the sky seemed to clear out, there was no scary blackness out that window	event, mood(scenery)
74	1	0	1	Lingering in a room while following Major Winters and having them call back towards me to follow them. I've never seen that in a text-based game before.	event, npcs (auto)
74	2	0	0	Didn't grab the knife right away, so it felt uneasy.	event, action, mood
76	1	0	1	One of the characters used the stout hammer to kill the zombie creature. I think it was the written dialogue that made it memorable.	event, npcs (auto), mood(descriptions)
76	2	0	0	Nothing, really.	none
80	1	0	1	Being followed from the meeting room, with some passengers joining and some disagreeing, which makes me want to see if that is scripted or interactive	event, npcs (auto)
80	2	0	0	Having NPCs take charge and leave while I was dithering with other stuff.	event, npcs (auto)
84	1	0	1	The appearance of the revenant in front of your character because it put your character in jeopardy.	event, revenant(meeting)
84	2	0	0	discovering the bodies of the zeppelin crew.	event, mood(scenery)
95	1	1	0	If you went back to your room after the night the captain dies, the description of something pounding on the door down the hallway got me excited.	event, revenant(unseen), mood(description)

Table 1. (Continued) Most memorable or noteworthy event

#	Ρ	G	R	Most memorable or noteworthy event	Tags
95	2	1	1	When the revenant first appeared and me stabbing it with the large knife	event, action, revenant(meeting, attacking it)
96	1	0	1	The one on one character interaction in the tutorial was memorable because unlike addressing all the people at once, it felt like I interacting with the characters more.	event, action, tutorial, npcs
96	2	0	0	Exploring outside the passenger gondola, because I was hoping out find out what the creature was.	event, action
97	1	1	0	attacking someone	event, action, npcs
97	2	1	1	breaking the chair, because a person in the story did it	event, action, npcs (auto)
100	1	0	1	When I ventured up with Major Winters and Roman into the upper deck and they left me in a room alone. I was trying to put the food in cupboards into the sacks so that I could carry them down, but I couldn't figure out the commands before they had left. When I was reminded of the creatures running in the shadows I quickly fled to find them.	event, action, mood, revenant, interface, frustration
100	2	0	0	When Major Winters and Mrs Winters were separated by a closed hatch and we had heard a scream from the other side, Mrs Winters demanded the hatch opened. When I followed afterwards, however, there was no consequence that I could tell came directly from the previous event. However, it was memorable to see the characters interaction to a fluid event.	event, npcs (auto), no consequence

#	Ρ	G	R	Least enjoyable aspect of the game	Tags
1	1	1	0	Limited ability to do much beyond talking to others and the speech/topics were limited - little control over events	options, story control
1	2	1	1	Not being to do what I wanted	options
2	1	0	1	it was a little frustrating trying to find commands that worked with the environment	input
2	2	0	0	limited options	options
8	1	0	1	I didn't really enjoy the whole game, but I think that the lack of background information about what this thing was going to be about and the short tutorial was very made this game even more unenjoyable	
8	2	0	0		none
20	1	0	1	Having to visualize character and world orientation and reading.	orientation, reading
20	2	0	0	Reading.	reading
21	1	1	0	So far in the game I just followed around the other characters, by hitting enter and seeing where they went (as we explored), I felt as though I wasn't forced to lead or figure out anything for myself with the exception of going to the bathroom.	objective
21	2	1	1	I was not clear on the commands to use my items	input
44	1	0	1	the fact that there were not many objects around to interact with and I was kind of assuming I had to find the right kinds of objects to suceed. I was unclear about my objective. Just stay alive? Romance some other characters? Show myself to be more competent than some other characters? Get to the bottom of relationships between those characters?	options (objects), objective
44	2	0	0	frustration being unable to suggest courses of action to other characters, and general sense that I was missing something important	options (npc), story control, objective

#	Ρ	G	R	Least enjoyable aspect of the game	Tags
49	1	1	0	Having to deal with the other characters with no observable benefits was the least enjoyable part of the game. I think (although I am not certain) that talking to the other passengers "unlocks" certain other events later on such as the revenant attack(s), going to sleep, and such, but there is no immediate evidence to suggest that this is the case.	consequences (npc)
49	2	1	1	Although character interaction is much improved from the first version, it still seems a bit awkward, particularly in conversations.	input
50	1	0	1	Not being able to understand what was going on as far as the mechanics of interaction	input
50	2	0	0	Actions felt constrained, couldn't go after the mysterious noises	options
52	1	0	1	Not understanding the commands at first made it hard to progress through the story. Additionally, I didn't know how I could make the story progress. I think I eventually figured out that if I used "wait" then things would happen more quickly.	input, story control
52	2	0	0	The fact that the monster wasn't killed this time.	resolution
57	1	1	0		none
57	2	1	1	Getting killed so quickly	resolution
58	1	0	1	Couldn't do all the actions I wanted to.	input, options
58	2	0	0		none
59	1	1	0	though I agreed to explore the zeppelin, it seem to have no effect on the rest of the story. also, the most memorable event of the story (the scream) had no opportunity for a response, whereas there were moments when there seemed to be no reason I that had to make a decision.	consequences

Table 2. (Continued) Least enjoyable aspect of the game

#	Ρ	G	R	Least enjoyable aspect of the game	Tags
59	2	1	1	it is not clear on what I'm capable of accomplishing within the game. there a prompts for me to do something without any indication of what I can choose to do. it is very frustrating!	options
61	1	1	0	hard to manuever and say what I wanted in talk mode	input, orientation, options (npc)
61	2	1	1	can't control character	input, options
63	1	1	0	The narrative was confusing and disorienting. The story was boring.	orientation (narrative)
63	2	1	1	The fact that the revenant kills the main character whether or not he/she fights which makes me feel that being able to control the actions of the main character is futile since the outcome is inevitably the same.	resolution, story control (agency)
64	1	0	1	Minor: Typing. I kept misspelling "examine".	input
64	2	0	0	Reading, but that's because I'm spoiled by graphics and voice acting in games these days.	reading
72	1	0	1	trying to remember the commands to the areas. E and W seemed to be the general choice but I don't understand the connection. My room was S, don't know why. Also, the same commands kept popping up for different areas. E command was to go to more than one area during the game.	input
72	2	0	0	having to scroll up to see what command went to where. I really didn't know what my quest was in this game to die?	input, objective
74	1	0	1	Having to read, at time, a lot of text.	reading
74	2	0	0	The creepiness of navigating a zeppelin full of corpses. The game ended before I could carry out my murderous plans.	

#	Ρ	G	R	Least enjoyable aspect of the game	Tags
76	1	0	1	I kept forgetting where I had been or I kept getting turned around because I couldn't see a layout of the environment.	orientation
76	2	0	0		none
80	1	0	1	Setting	
80	2	0	0	Incomplete	resolution
84	1	0	1	No resolution of the revenant.	resolution
84	2	0	0	lack of drama	
95	1	1	0	Wish it was longer :D	none(?)
95	2	1	1	I do wish the revenant was harder to kill	
96	1	0	1	The concept of time- how fast it goes, how long WAITing takes, how I would choose WAIT and the next thing that happened was dawn and End of Session.	orientation (time)
96	2	0	0	Not really getting anywhere even though I kept retracing steps and LOOKing at everything.	resolution, story control
97	1	1	0	not remembering where i was in context of everything else	orientation
97	2	1	1	hard to know/remember where everyone was	orientation
100	1	0	1	Struggling to complete actions that I thought were applicable commands to the situation but lacked the finesse to communicate them to the game quickly, without error.	input

Table 2. (Continued) Least enjoyable aspect of the game

#	Ρ	G	R	Least enjoyable aspect of the game	Tags
100	2	0	0	Events happened very quickly and unfortunately the murders seem to have little to no consequence to the player character. Even when left alone nothing threatens me but instead makes me feel a lot of tension. The game ended very abruptly while I was still above deck and had no awareness that the night had passed. It was much shorter as well. It was almost as if the Non-Player Characters were exacting the story elements with greater effect in the 2nd game while I was elsewhere.	resolution, orientation (time)

Table 2. (Continued) Least enjoyable aspect of the game

#	Ρ	G	R	Most enjoyable aspect of the game	Tags
1	1	1	0	Interesting storyline	story
1	2	1	1	The storyline	story
2	1	0	1	mastering the commands for moving around	agency (mastery)
2	2	0	0	controlling the story	agency (story control)
8	1	0	1	nothing	none
8	2	0	0		none
20	1	0	1	Interesting story and dialogue.	story, npcs
20	2	0	0	Story. Likened the investigation and exploration to an early 90s horror game.	story, exploration
21	1	1	0		none
21	2	1	1	very fun this time around, unlike last round i led the exploration group and got myself into trouble. Because i failed i really want to play again.	variation
44	1	0	1	Nice setting	setting
44	2	0	0	general setting and context was good. I like the 1920's and zeppelins. Descriptions were nice. got some sense of getting further second time around, but still this general frustration with being unable to actually make significant progress in the game, or see the characters interacting in significant ways	setting, variation
49	1	1	0	Simply searching around the airship for clues was enjoyable for me; I think that part of this was nostalgia, while knowing that there was a problem that I could solve (assuming that I survived) was another component.	exploration, objective
49	2	1	1	Problem solving remains the most enjoyable part of the game, especially knowing that the revenant can in fact be killed.	objective

#	Ρ	G	R	Most enjoyable aspect of the game	Tags
50	1	0	1	figuring out how to jump off the balcony, and do other things I feel the designer(s) didn't want me to do	agency (world)
50	2	0	0	Sense of mood, some of the descriptions	descriptions, mood
52	1	0	1	Making things happen was the most enjoyable aspect of the game.	agency
52	2	0	0	Not being the leader this time around. Also, I got one additional investigator this time around.	variation
57	1	1	0		none
57	2	1	1	Interacting with the other characters and getting to explore the setting.	exploration, npc (interacting with)
58	1	0	1	Descriptions	descriptions
58	2	0	0		none
59	1	1	0	the alternate timeline and using zeppelins as passenger vehicles	setting
59	2	1	1	riding on a zeppelin.	setting
61	1	1	0	choosing a story line to follow	agency (story)
61	2	1	1	interesting locations described	descriptions, setting
63	1	1	0	Talking to Miss Miriam about the Vampire book.	npc (interacting with)
63	2	1	1	Being able to type "hit revenant with hammer" and the character doing exactly that, he hit the revenant with the stout hammer.	agency (world)

Table 3. (Co	ontinued) Most	enjoyable	aspect of	the game
--------------	----------------	-----------	-----------	----------

#	Ρ	G	R	Most enjoyable aspect of the game	Tags
64	1	0	1	Why I like any other RPG, I can influence the outcome of the story	agency (story)
64	2	0	0	The degree of freedom	options
72	1	0	1	The chaotic event in the middle of the game wondering why there had been a masacre.	story, mood
72	2	0	0	reading. Everyone should read.	reading
74	1	0	1	Trying commands and seeing what happens.	agency (world)
74	2	0	0	Knowing I was about to stab the other passengers eventually.	objective (personal)
76	1	0	1	The commands were nicely done. It was great that you could type in different versions of that command and the game would recognize it, like I for look.	agency (input)
76	2	0	0		none
80	1	0	1	Intrigue	mood
80	2	0	0	Mystery	mood
84	1	0	1	Discovering what happened to the crew (the cause of their deaths).	objective
84	2	0	0	exploration	exploration
95	1	1	0	Very descriptive, very believable (character wise), love the setting	setting, npc, descriptions
95	2	1	1	Definitely the atmosphere of the game, creepy, something is hunting you down, no place to really run too	mood
96	1	0	1	Finding things that I could actually interact with like the food or knife.	agency (world)
96	2	0	0	Hoping that searching everything would turn up something.	exploration, objective

	Table 3.	(Continued)	Most enjoyable	aspect of	the game
--	----------	-------------	----------------	-----------	----------

#	Ρ	G	R	Most enjoyable aspect of the game	Tags
97	1	1	0	unlimited options in what i can do	options
97	2	1	1		none
100	1	0	1	Watching the story unfold even if I didn't do anything. The continual presence of events occurring around me (such as scratching at a door or other characters leaving me in a room alone) built a lot of tension.	story autonomy
100	2	0	0	Seeing a slightly different version the second time around was the most enjoyable part of the play through. My memorable moment when Mrs Winters was worried about her husband brought unexpected realism to her character. I think this was in large part because her concern was voiced suddenly as if coming from her own motivation and not in a cookie cutter response to my actions. The forcing of events and responses from NPCs in the world definitely create a livelier environment.	variation, story autonomy (npc)

Table 3.	(Continued)	Most enjoyable	aspect of the	game
----------	-------------	----------------	---------------	------

WORKS CITED

- Aristotle. *Poetics.* Trans. S. H. Butcher. Ed. Francis Fergusson. New York: Hill and Wang, 1961.
- Bates, Joseph. "Virtual Reality, Art, and Entertainment." *PRESENCE: The Journal of Teleoperators and Virtual Environments* 1.1 (1992): 133-138. http://www.cs.cmu.edu/afs/cs.cmu.edu/misc/mosaic/common/omega/Web/Groups/oz/papers/presence_1_1.ps
- Chatman, Seymour. *Story and Discourse: Narrative Structure in Fiction and Film.* Ithaca, NY: Cornell University Press, 1978.
- Chatman, Seymour. *Coming to Terms: The Rhetoric of Narrative in Fiction and Film.* Ithaca, NY: Cornell University Press, 1990.
- Choose Your Own Adventure series. 185 volumes. New York: Bantam Books, 1979 1998.
- Crawford, Chris. *Chris Crawford on Interactive Storytelling.* Berkeley, CA: New Riders, 2004.
- Crowe, Mark, and Scott Murphy. *Space Quest I IV*. Sierra Entertainment, 1986 1991.
- Dungeons & Dragons: Dungeon Master's Guide: Core Rulebook II v3.5. Renton, WA: Wizards of the Coast, 2003.
- Dever, Joe. *Lone Wolf* series. 28 volumes. New York: Berkley Books, 1984 1998.
- Doom. id Software, 1993.
- Elder Scrolls I IV. Bethesda Game Studios, 1994 2006.
- Fairclough, Chris R. "Story Games and the OPIATE System: Using Case-Based Planning for Structuring Plots with an Expert Story Director Agent and Enacting them in a Socially Simulated Game World." PhD Thesis. TCD-CS-2005-59. University of Dublin, Trinity College, 2005. http://www.cs.tcd.ie/publications/tech-reports/reports.05/TCD-CS-2005-59.pdf
- Fisher, Jim. ORLibrary. http://www.onyxring.com/orlibrary.aspx Last accessed: 04 Apr 2011.

Freytag, Gustav. Technique of the Drama. Translated: Elias J. MacEwan. 6th German ed. Chicago: S.C. Griggs & Company, 1895.

Grand Theft Auto III. Rockstar Games, 2001.

- Grasbon, Dieter, and Norbert Braun. "A Morphological Approach to Interactive Storytelling." Proceedings of cast01//Living in Mixed Realities, 2001. http://netzspannung.org/version1/extensions/cast01- proceedings/pdf/by name/Grasbon.pdf>
- Half-Life. Valve Software, 1998.
- Hayles, N. Katherine. Writing Machines. Cambridge, MA: MIT Press, 2002.
- Inform. V 6.31N. 22 Apr 2010. < http://www.inform-fiction.org/inform6.html> Last accessed: 04 Mar 2011.
- Johnstone, Keith. Impro: Improvisation and the Theatre. New York: Routledge, 1979.
- Kelso, Margaret Thomas, Peter Weyhrauch, and Joseph Bates. "Dramatic Presence." PRESENCE: The Journal of Teleoperators and Virtual Environments. 2.1 (1992). Technical Report CMU-CS-92-195. <http://www.cs.cmu.edu/afs/cs.cmu.edu/misc/mosaic/common/omega/We b/Groups/oz/papers/CMU-CS-92-195.ps>

Laurel, Brenda. Computers As Theatre. Reading, MA: Addison-Wesley, 1991.

- Mateas, Michael. "An Oz-Centric Review of Interactive Drama and Believable Agents." Technical Report CMU-CS-97-156, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. June 1997. <http://www.cs.cmu.edu/afs/cs/misc/mosaic/common/omega/Web/Groups/ oz/papers/CMU-CS-97-156.html>
- Mateas, Michael. "Interactive Drama, Art, and Artificial Intelligence." PhD Thesis. Technical Report CMU-CS-02-206. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, December 2002. <http://www.cs.cmu.edu/afs/cs/project/oz/web/papers/CMU-CS-02-206.pdf>
- Mateas, Michael. "A Preliminary Poetics for Interactive Drama and Games." First Person: New Media as Story, Performance, and Game. Eds. Noah Wardrip-Fruin and Pat Harrigan. Cambridge, MA: MIT Press, 2004. Reprinted at:

<http://www.electronicbookreview.com/thread/firstperson/mateas>

- Meadows, Mark Stephen. *Pause & Effect: The Art of Interactive Narrative*. Indianapolis, IN: New Riders, 2003.
- Metter, Julian, and Cate McManus. *Booty Call*. 1999. http://www.romp.com (now defunct).
- Murray, Janet. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace.* New York: Free Press, 1997.
- *Neverwinter Nights*. BioWare, 2002.
- Norman, Donald A. *The Design of Everyday Things*. New York: Basic Books, 2002.
- Quake. id Software, 1996.
- Plotkin, Andrew. *Glulx: A 32-Bit Virtual Machine for IF*. http://www.eblong.com/zarf/glulx/ Last edited: 22 Jan 2011.
- Plotkin, Andrew. Quixe: a Glulx VM interpreter written in Javascript. http://www.eblong.com/zarf/glulx/quixe/ Last edited: 17 Feb 2011.
- Propp, Vladimir. *Morphology of the Folktale*. Translated: Laurance Scott. Ed. Louis A. Wagner. 2nd ed. Austin, TX: University of Texas Press, 1968.
- Rise of the Dragon. Dynamix, 1990.
- Short, Emily. "So, do we need this parser thing anyway?" <http://emshort.wordpress.com/2010/06/07/so-do-we-need-this-parserthing-anyway/> Posted: 07 Jun 2010.
- Smiley, Sam. *Playwriting: The Structure of Action*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1971.
- Smith, Mark, and Jamie Thomson. *The Way of the Tiger* series. 6 volumes. New York: Berkley Books, 1985 1989.
- Spierling, Ulrike, Dieter Grasbon, Norbert Braun, and Ido Iurgel. "Setting the Scene: Playing the Digital Director in Interactive Storytelling and Creation." *Computer & Graphics* 26 (2002): 31-44.
- Stam, Robert. "Beyond Fidelity: The Dialogics of Adaptation." *Film Adaptation*. Ed. James Naremore. New Brunswick: Rutgers University Press, 2000. 54-76.

TADS3. V 3.0.18. 05 May 2009. < http://www.tads.org/tads3.htm>

- Theune, Mariet, and Sander Faas, Anton Nijholt, Dirk Heylen. "The Virtual Storyteller: Story Creation by Intelligent Agents." 2003. http://www.home.cs.utwente.nl/~anijholt/artikelen/tidse2003.pdf
- Tomaszewski, Zach, and Kim Binsted. "The Limitations of a Propp-based Approach to Interactive Drama." *Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium.* Technical Report FS-07-05. Menlo Park, CA: AAAI Press, 2007. http://www.zach.tomaszewski.name/argax/pubs/2007-TomaszewskiBinsted-ProppLimitations.pdf
- Tomaszewski, Zach, and Kim Binsted. "Marlinspike: An Integrated Approach to Interactive Drama." *Integrating Technologies for Interactive Stories: Papers from the INTETAIN 2008 Workshop.* Playa del Carmen, Mexico. 7th Jan 2008. pp. 29-33.
- Tomaszewski, Zach, and Kim Binsted. "Demeter: An Implementation of the Marlinspike Interactive Drama System." *Intelligent Narrative Technologies II: Papers from the AAAI Spring Symposium.* Technical Report SS-09-06. Menlo Park, CA: AAAI Press, 2009. pp. 133-136.
- Tomaszewski, Zach. *Demeter: Blood in the Sky*. 2011. http://zach.tomaszewski.name/argax/games/demeter/
- Tomb Raider. Eidos Interactive, 1996.
- Williams, Roberta. *King's Quest I VI*. Sierra Entertainment, 1984 1992.
- Woodworth, Peter. *Mind's Eye Theatre Core Rulebook*. White Wolf Publishing, 2005.

World of Warcraft. Blizzard Entertainment, 2004.

Wright, Will. The Sims. Electronic Arts, 2000.

Zeppieri, Jon. Zag. V1.06. 2005. < http://wso.williams.edu/~jon/zag/>