

University of Alberta

**INTENTIONAL DIALOGUES:
LEVERAGING INTENT TO ENABLE THE EFFECTIVE REUSE OF
CONTENT**

by

Christopher M. Kerr

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

©Christopher M. Kerr
Fall 2009
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Examining Committee

Duane Szafron, Computing Science

Sean Gouglas, History

Vadim Bulitko, Computing Science

Abstract

The creation of rich, immersive game worlds is one of the major goals for designers of modern story-based games. The inclusion of unique and interesting dialogues for all of a game's non-player characters (NPCs), especially the secondary NPCs, does a great deal to increase the believability of the game world being created. Unfortunately, the creation of such dialogues can be very time consuming, and is often considered to be prohibitively expensive. In order to address this problem, I introduce the concept of an *Intentional Dialogue* that allows designers to efficiently and effectively create multiple variations of a dialogue while ensuring that the variations are appropriate to the NPCs that will be speaking them. I will also introduce several support structures for the *Intentional Dialogues* and a machine learning classifier that allows designers to quickly populate the system with both new and existing content.

Acknowledgements

I would first like to thank my supervisor, Dr. Duane Szafron, for the support and guidance he has provided during the course of my research. I would also like to thank the other members of my committee, Dr. Vadim Bulitko and Dr. Sean Gouglas, for their effort and the feedback they provided me with.

I would like to thank Jeff Siegel, whose research provided the basis for my own. I would like to thank my fellow graduate students within the ScriptEase research group: Neesha Desai, Richard Zhao, Marcus Trenton, Yi Yang, AmirAli Sharifi, and Maria Cutumisu for the support and advice they have provided over the course of the past year.

I would like to thank all the members of the ScriptEase implementation team, especially the current team of Jason Duncan, Joshua Friesen, and Robin Miller for their work in providing the researchers in the ScriptEase group with the tools necessary to perform our research.

I would like to thank my friends and family for all of the love and support they have shown me over course of my degrees. And I would like to thank Carla for both her love and support, as well as the disapproving looks she would send my way if I spent too much time procrastinating.

Finally, I would like to thank all of those not mentioned here, whether at the University of Alberta, the Department of Computing Science, or elsewhere, who supported me during my research.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Analysing the Problem	2
1.3	The Content Bottleneck	3
1.4	Finding a Pattern	5
1.5	A Focus on Intent	8
1.6	Chocolate or Vanilla? Choosing a Flavour	10
1.7	Summary	12
2	Related Work	13
2.1	The Problem: A Look at Industry Examples	13
2.1.1	Neverwinter Nights	13
2.1.2	Mass Effect	16
2.1.3	Oblivion	17
2.1.4	Fallout 3	19
2.1.5	Facade	19
2.1.6	GTA IV	20
2.2	How It's Made: Creating the Dialogue	21
2.2.1	Automated Dialogue Generation	21
2.2.2	Current Tools	22
2.2.3	Better Tools	25
2.3	Classifying Dialogue Lines	34
3	Intentional Dialogues	36
3.1	Overview	36
3.2	Intentional Dialogues	37
3.3	Intentional Dialogue Lines	45
3.3.1	The Grid	47
3.3.2	Extending the Functionality of Remarks	50
3.4	Intentional Dialogues in Action	51
4	Supporting Flavours	55
4.1	Overview	55
4.2	Collecting a Training Set	56
4.3	Selecting the Features	58
4.4	Evaluation	60
4.4.1	Results with Automatically Labelled Film Data	61
4.4.2	Results with Juried Game Data	62
4.4.3	Additional Results	63
5	Future Work and Conclusion	65
5.1	Future Work	65
5.2	Conclusion	66

Bibliography	68
A Game Dialogue Lines	70

List of Figures

1.1	A dialogue from the first chapter of Neverwinter Nights. . . .	6
1.2	A dialogue from the first chapter of Neverwinter Nights. . . .	7
1.3	A Directory intentional dialogue.	9
1.4	An intentional greeting line.	11
2.1	The reoccurring dialogue where Aribeth inquires as to the player's progress in Neverwinter Nights.	14
2.2	The identically structured dialogues of several companion characters in Neverwinter Nights.	15
2.3	An example of Mass Effect's synopsis based conversations. (taken from [8])	16
2.4	An example of Oblivion's wiki-based dialogue system.	18
2.5	A comparison of a dialogue in both the NWN editor (left) and as an abstracted graph (right). (from Siegel [26])	23
2.6	The <i>Placeable Use - Open Door</i> encounter pattern in the ScriptEase tool.	27
2.7	The adapted <i>Placeable Use - Open Door</i> pattern.	28
2.8	The creation of an <i>Explosion</i> action atom.	29
2.9	An <i>Exterminate</i> quest as viewed in the ScriptEase editor. . . .	30
2.10	A <i>Guard</i> role as viewed in the ScriptEase tool.	31
2.11	An example of a topic with multiple exchanges in the ScriptEase dialogue tool.	32
2.12	A dialogue as displayed in the Aurora toolset. Note the distance between sibling nodes near the root of the dialogue. (from Siegel [26])	33
2.13	A simple dialogue built in ScriptEase.	34
3.1	An Extortion intentional dialogue.	38
3.2	An Extortion intentional dialogue that includes a decision pattern.	40
3.3	A Bartender intentional dialogue.	41
3.4	An adapted Bartender intentional dialogue that includes non-intentional dialogue lines.	42
3.5	Prototype of an intentional dialogue collapsed into a single dialogue component.	43
3.6	Prototype of a <i>Hook</i> line being propagated up when an intentional dialogue is included within another dialogue.	44
3.7	A non-intentional dialogue that makes use of intentional dialogue lines.	46
3.8	The partitioning of an intentional dialogue line by sophistication (vertical) and disposition (horizontal).	48
3.9	Several mappings performed using a PC Charisma decision pattern for disposition and an NPC Intelligence decision pattern for sophistication.	49

3.10	The action block of a <i>Purchase Item</i> dialogue line pattern with the parameters unset.	49
3.11	A parameterized intentional dialogue line.	50
3.12	An excerpt from an extortion dialogue shown as a regular dialogue (left) and an intention dialogue (right).	52
4.1	The formula used to calculate <i>term frequency-inverse document frequency</i>	58
4.2	An illustration of the average-words-per-sentence feature. . . .	59
4.3	The formula's for Precision, Recall, and F-Measure.	60
4.4	The results of the binary-valued classifier trained on film dialogue lines. (from Kerr [15])	61
4.5	The results of the TF-IDF-valued classifier trained on game dialogue lines. (from Kerr [15])	62

Chapter 1

Introduction

1.1 Motivation

You recently arrived in Beregost while on your way south to Nashkel. You didn't really mean to stop here, but the town is in a tight spot and the people have asked for your help. It probably doesn't hurt that they've offered up a rather large sum of gold to anyone who is able to stop the bandit raids that have been plaguing them, either. Regardless, you've agreed to do what you can to stop the raids and have begun to prepare yourself for the task ahead.

The strangest thing has been happening, though. As you make your way around town, trying to solicit a bit more information from the townsfolk, everyone seems to be saying the same thing. You can understand that with the problems the town has been facing people would be a bit preoccupied, but this is more than that. People aren't just talking about the same things; they're literally speaking exactly the same words. It's as though the townsfolk are all speaking from the same script, and the whole thing is making you more than a little uneasy. Is there something else at work here? You've heard tales of people being spellbound by evil monsters or wicked sorcerers, but surely that couldn't happen to an entire town. Is it all some kind of trap? Or are the people of this town just that linguistically unimaginative? Even so, you can't help but question whether continuing to stay in Beregost is the best idea.

Now the above narrative may seem to describe the discovery of some sinister mind control plot, but it is simply the standard approach to populating the dialogues of secondary non-player characters (NPCs) in the vast majority

of modern story-based games. This specific example comes from the game *Baldur's Gate* [4], a best selling computer role-playing game from BioWare Corp. The game was released over a decade ago, and yet the same problems are just as prevalent in modern games as they were in those made 10 years ago.

1.2 Analysing the Problem

Why has there been so little progress in the quality and variety of dialogue in story-based games? This lack of progress becomes especially confusing when considering the focus of so many in the games industry on immersion [11]: the process of aiding the player's suspension of disbelief, which consequently leaves them feeling more involved in the game. Do players not notice the monotony of the dialogues? Or perhaps they simply don't pay any attention to the secondary characters and are therefore only exposed to the main, plot-driving conversations? If either of these were the case, it would certainly explain why the games industry has not put in the effort to provide unique and believable dialogues for the majority of NPCs. However, while there may be players who do not notice these inadequacies, there seems to be evidence that, in general, most game players do.

Perhaps the most infamous example is that of Bethesda Softworks' game *Oblivion* [28]. While Bethesda certainly deserves some praise for their attempts at creating NPCs that would behave in a life-like manner, there were certain aspects of their implementation that did not live up to their potential. As part of their behaviours, the NPCs in *Oblivion* would walk up to one another and participate in ambient conversations. These conversations were meant to provide the player with a greater sense of immersion while playing the game. However, due to the limited selection of topics for NPCs to converse about, the player would constantly overhear the game's characters describing their most recent encounter with a mud crab. Furthermore, *Oblivion*, as is the case with most modern games, was fully voice acted, and had a rather limited number of voice actors. Many characters were therefore voiced by the same actors,

and consequently, not only would players hear repetitive conversations, but they would hear the same voices speaking those conversations as well. This repetition led to the development of a fad in the game’s community where players would poke fun at the prevalence of mud crab discussions in the game. Even now, three years after the game’s release, numerous examples of this can be found on gaming forums and websites such as YouTube.

However, while *Oblivion* is one of the most infamous examples, it is by no means unique in its repetitive dialogue. Bethesda’s most recent game, *Fallout 3* [29], also suffered from many of the same problems described above. A limited number of voice actors and conversation topics could lead to a similar immersion-breaking amount of repetition in the game’s conversations. Nor are these deficiencies limited to the games of Bethesda, as a further example can be found in BioWare’s *Mass Effect* [6]. In order to avoid the problems with repetitive dialogue that were present in games like *Oblivion*, BioWare instead opted to not include any dialogue at all on the majority of secondary NPCs found in the game. While this does prevent the problem with repetition, it does so in a less than ideal manner that leaves the game world populated with flat, uninteresting characters that appear to be in some form of fugue state.

1.3 The Content Bottleneck

If this lack of progress in providing varied NPC dialogues is not the result of players being unable to notice the shortcomings then why has there been so little improvement? The answer to this question is perhaps somewhat more obvious than might be initially expected. The problem is simply that the creation of dialogue is very time consuming, and the majority of the industry has concluded that the creation of specific dialogues for the multitudes of secondary NPCs in a large game is not cost effective. As such, the player is presented with hordes of NPCs that all speak in exactly, or very near to, the same manner.

This problem is known as a content bottleneck and is not unique to the creation of game dialogue. More generally, this problem also presents itself in

the creation of all other aspects of NPC behaviour as well. Whether the task at hand is the creation of unique and context-sensitive NPC dialogue or realistic ambient behaviours for these same characters, the central problem remains that the current state-of-the-art in content generation largely limits designers to an all-or-nothing level of reuse. The results of this can clearly be observed in the previous examples from *Oblivion* [28] and *Mass Effect* [6]. In these games, NPCs will either share the same dialogue files, or the less important characters will simply lack any dialogue at all. However, the effects of the bottleneck are also visible in the taverns of games such as BioWare Corp.’s *Neverwinter Nights* [5] where the player may find NPCs staring blindly at a wall for hours on end. Given the effort required to create new content, what is needed is a means of allowing for the more extensive reuse of content, while also ensuring that the final result of such reuse produces characters that behave and speak in a realistic, believable manner.

These problems, the generation of dialogues and the behaviours, possess many shared elements, but the problems remain different enough to require separate solutions. Fortunately, while these similarities may not be great enough to enable the creation of a single, unified solution to the problem, they do suggest what avenue of research is most likely to solve the problem. As was mentioned previously, the problem of a content bottleneck is not unique to the creation of NPC dialogue, and neither is it unique to the more general problem of creating realistic NPC behaviour. A previous games-related example of a content bottleneck can be found by investigating the history of computer graphics. The rendering of a game’s characters was once just as significant a problem as the generation of dialogue is today. The addition of a new character to a game required the game designers to not only specify the appearance of a character, but also to write the code to handle the rendering of the character within the game engine. This process had to be repeated in its entirety for every character within a game, and so NPCs in games often found themselves having an inordinate number of twin brothers and sisters.

In computer graphics, the solution to this problem was found in the adoption of a data-driven content model. This approach separated code from con-

tent, and in doing so allowed for the general implementation of the rendering algorithms. By reducing the amount of programming necessary for the inclusion of new visual elements, the graphics community was able to reduce their content bottleneck. Furthermore, the adoption of the *triangle-model-texture map* abstraction for the representation of graphics content aided in the reduction of a second content bottleneck as well [23][10]. The separation of the structure of content (i.e. the model) from the surface appearance (i.e. the texture map) allowed designers to reuse a single model for multiple objects in such a way that the objects would all appear to be unique to the player. It was now possible to create a multitude of characters with varying, unique appearances simply by re-skinning (i.e. applying a new texture map to) a small number of underlying models.

Through the use of abstractions and modularization, the field of computer graphics was able to achieve truly impressive levels of reuse. It is this level of reuse that has played a major role in allowing modern games to possess the rich levels of visual detail present in the environments of titles such as CryTek's *Crysis* [7]. If dialogue in games is to reach a comparable level of richness, a similarly-themed abstraction must be adopted.

1.4 Finding a Pattern

The question then becomes how to develop such an abstraction. In computer graphics, the *triangle-model-texture map* abstraction was a result of the move from two-dimensional, sprite-based graphics to three-dimensional graphics. This change caused an increased demand for visual detail. The adoption of the *triangle-model-texture map* abstraction, and the increased level of content reuse that it provided, allowed games to deliver the desired increase. This reuse was possible because of the way that the abstraction's underlying polygonal meshes successfully abstracted the structure of a model away from its other elements. By applying differing textures to the same underlying mesh, it became possible to drastically reduce the amount of original content needed to create new game objects. Is there then an analogue in game dialogue to the



Figure 1.1: A dialogue from the first chapter of *Neverwinter Nights*.

underlying polygonal mesh of character models in computer graphics?

To investigate this, let us consider a few dialogue excerpts from *Neverwinter Nights* [5]. Figures 1.1 and 1.2 show a pair of dialogues from the first chapter of the game. Both dialogues are found on commoners that are hiding in their homes to avoid the chaos that has overrun the local streets. It is immediately evident that these dialogues are not exact duplicates of one another, but it may be the case that there is in fact a common element to them.

Let us first consider the dialogue from Figure 1.1. The dialogue begins with the NPC demanding to know why the player character (PC) has entered his home. The PC responds to this by asking whether the NPC would be willing to answer a few questions. The NPC agrees to do so and the conversation arrives at a central question hub. If we compare these first few lines with the beginning of the dialogue found in Figure 1.2, there are some striking

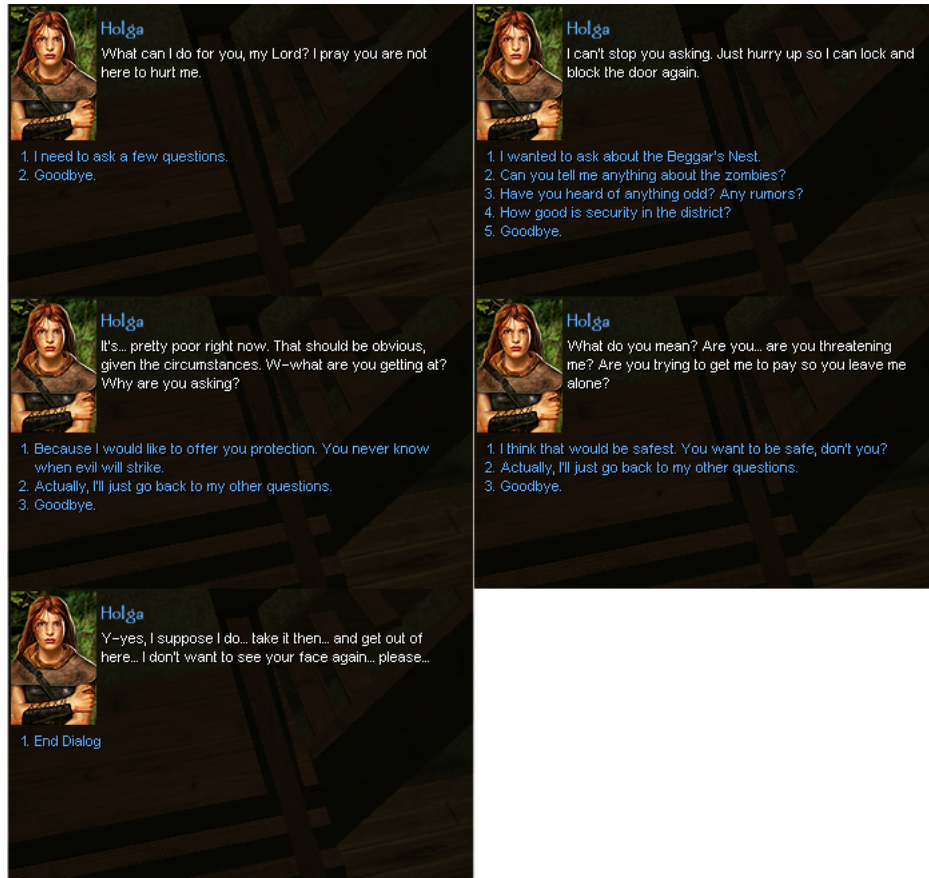


Figure 1.2: A dialogue from the first chapter of Neverwinter Nights.

similarities. Again, the dialogue begins with the NPC demanding to know why the player has entered his home, and just as before, the PC then responds by requesting that the NPC answer some questions. The NPC in the second conversation also agrees, and this brings us to a similar dialogue hub.

Returning to the dialogue in Figure 1.1, we see that one of the player's options is to begin an extortion attempt against the NPC. If the player chooses to do so, the NPC will respond with incredulity, at which point the player is given the option of continuing with the extortion or deciding against it. Assuming that the player decides to continue with the extortion attempt, the dialogue will then proceed to a second similar exchange of NPC protest and PC decision of whether to continue the attempt or not. If the player decides, once again, to continue the extortion attempt, the NPC will make a final protest and then give the PC their money. Afterward doing so, the NPC

will no longer be willing to talk. If, however, the player decides against the extortion attempt, the conversation will be routed back to the question hub with no harm done. Comparing this with the second dialogue, we see that the dialogues continue to have a shared intent. The phrasing differs, but here we see that the player is once again given the option of extorting money from the NPC. Also as before, the NPC will be shocked by the attempt, and then the player will be given two opportunities to choose whether to continue the attempt or to instead decide against it.

It is important to note that the similarities between these two dialogues are much more than just a shared structure. There are many dialogues throughout NWN that begin with an NPC greeting, followed by a PC response, and then the arrival at a central question hub. The important aspect of the dialogues in figures 1.1 and 1.2 is the shared intent. It is this shared intent that allows for the substitution of one dialogue for the other without changing the underlying meaning of the interaction. Without the shared intent, such substitutions would lead to nonsensical conversations, and so this intent provides the necessary leverage to enable reuse.

1.5 A Focus on Intent

How, then, can shared intent be leveraged in a manner that increases reuse, and therefore reduces the designer effort required to populate dialogues? In the previous section, two distinct levels of shared intent were discovered. Shared intent existed between individual dialogue lines and also at a larger scale where partial, or potentially even whole, dialogues shared an intent. As the larger-scale shared intent was the product of the shared intent between sequences of dialogue lines, it is perhaps best to take a bottom-up approach and begin with the individual lines.

Perhaps the most straightforward example of commonly-used dialogue lines that share an intent would be lines in which the player or NPC speaks a farewell. The vast majority of conversations in a game, especially those that are not themselves major plot elements, will end with the player choosing to

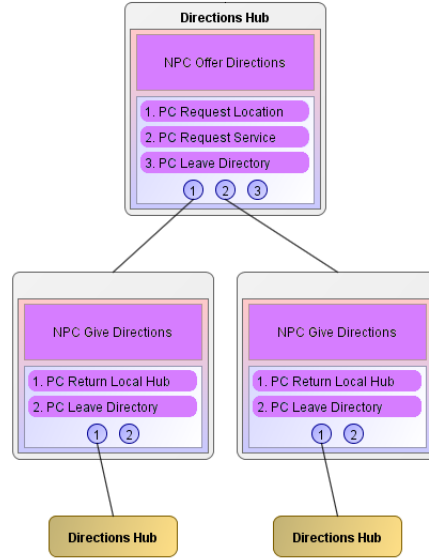


Figure 1.3: A Directory intentional dialogue.

end the dialogue. “Good bye,” “Farewell,” and “I should go now” are all examples of such lines, and what is common to these lines is their shared meaning. While obviously different lines, all of them have the same farewell intent. What is needed is an abstraction that represents this intent, and to this end I introduce the concept of an *Intentional Dialogue Line*.

An *Intentional Dialogue Line* is an abstract dialogue line that a designer can use in place of an actual line when creating a dialogue. The intentional line represents the set of dialogue lines that share its intent. It is replaced at game time with an actual dialogue line that is appropriate to the character speaking the line and to the context in which it is being spoken. Further examples of potential intentional dialogue lines can be found in PC and NPC greetings, PC requests to view the items in an NPC’s store, and PC requests to ask an NPC some questions. By using an intentional dialogue line, designers are required to know only the intent, instead of the full text, of what they want to be spoken when creating a conversation.

The intentional dialogue line captures the shared intent at the level of individual lines, but there remains the larger-scale shared intent found in whole, or partial, dialogues as well. This larger scale of shared intent is what was found in the extortion attempt dialogue examined in the previous section.

Therefore, I introduce an *Intentional Dialogue*, which is an abstract dialogue, or sub-dialogue, that represents a set of dialogues that share a similar intent. In addition to the extortion attempt dialogues discussed previously, there are a great deal of other examples easily found in games. One such dialogue, the Directory intentional dialogue, can be seen in Figure 1.3. By allowing designers to add whole intentional dialogues at once, the effort necessary to create conversations that contain these intentional dialogues is drastically reduced.

1.6 Chocolate or Vanilla? Choosing a Flavour

There are, however, two remaining challenges to overcome before a suitable implementation of intentional dialogues and dialogue lines is possible. First, there must be a means of associating actual dialogue lines with the intentional dialogue line that shares their intent. And secondly, a means of deciding which actual line is used in place of the intentional line at game time is also required. To solve both of these problems I present the *flavouring* system for intentional dialogue lines.

When creating an intentional dialogue line, the designer must specify the intent of the line as well as an arbitrary number of axes on which to label the actual dialogue lines associated with it. These axes rank the actual lines based on a specified criteria, and each axis has three distinct values (low, medium, and high). The default set of axes consists of one for the sophistication of a line and one for the disposition of a line. This set of axes was found to provide the best balance of power and simplicity, and is sufficient in the majority of situations. An example partitioning of an intentional greeting line using these default axes is shown in Figure 1.4.

This method of partitioning provides the necessary means of associating the actual lines with the intentional dialogue lines. However, the labelling can remain a time-consuming process. To expedite the labelling process, a machine learning classifier was developed. This classifier uses the support vector machine (SVM) [30] approach to automatically predict labels for the sophistication of dialogue lines. The feature vectors used by the SVM are a

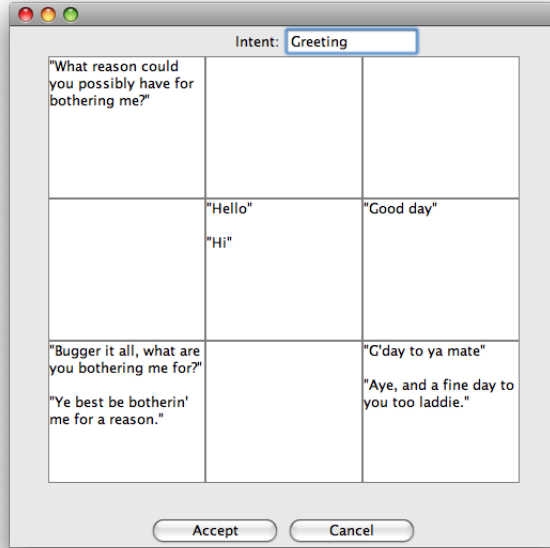


Figure 1.4: An intentional greeting line.

combination of bag-of-words [17] style features and average-words-per-sentence features. While only a classifier for sophistication was developed, a similar approach should be capable of performing the labelling for other properties as well.

With the actual dialogue lines now associated with the intentional lines, there remains the challenge of selecting a specific actual line at game time. It is important that the flavouring system selects a line that is appropriate to the character that is speaking the line, as well as the context in which the line is being spoken. For this task I have adapted the work done by Jeff Siegel [26] of the ScriptEase research group [19]. In his work, he developed *decision patterns* that provide a simple and straightforward way for designers to map arbitrary combinations of game-state values to conditionals that control the display of dialogue lines. By adapting these patterns to allow designers to use decision patterns to map game state to the bins, instead of to the individual lines, it becomes possible to rapidly specify the appropriate flavour for a given situation without any requirement for specific programming knowledge. The use of dialogue filters in this manner allows for a late binding of the actual dialogue lines to the intentional dialogue lines ensuring that the selected line

is appropriate to the current game state and satisfies the requirements laid out previously. Additionally, the use of decision patterns allows designers to reuse filters they have created previously. Consequently, common mappings, such as character intelligence to sophistication, need only be created once. This potential for reuse further reduces the effort required on the part of designers when using intentional dialogues.

1.7 Summary

This chapter briefly discussed the challenges faced by the games industry in regards to the efficient creation of dialogues for large numbers of non-player characters. To this end, I have introduced the concepts of an *Intentional Dialogue* and an *Intentional Dialogue Line*, as well as a *flavouring* system to support them. The remainder of this dissertation will consist of an in-depth discussion of the topics introduced in this chapter.

First, a review of the current state of dialogue creation tools and techniques, both in the industry and academia, will be performed. This will be followed by a more thorough look at the concepts of an *Intentional Dialogue* and an *Intentional Dialogue Line*. Next, there will be a discussion of the *flavouring* system and an analysis of the results obtained with the automatic sorting method. Finally, the paper will conclude by looking at potential avenues for future research.

Chapter 2

Related Work

2.1 The Problem: A Look at Industry Examples

Prior to developing a new method for creating dialogue, it is important to have a thorough understanding of the specific shortcomings that must be addressed. As it is the final product that a game’s players are exposed to, and not the development tools, I will begin by looking at the dialogue found in a number of existing games.

2.1.1 Neverwinter Nights

Neverwinter Nights (NWN) [5] was briefly mentioned in chapter 1 while discussing the lack of interesting behaviours for NPCs in most modern, story-based games. However, while the behaviour of the game’s NPCs certainly is lacking, I am instead interested in the quality of the game’s dialogues. Here I will break the dialogues of NWN into three separate categories.

The first category to consider is the main plot-driving dialogues that progress the story. Such dialogues are usually the most prominent in a game and are generally both unique and interesting. Consider the character of Aribeth, who is the PC’s main ally early in the game and the central antagonist for much of the remainder of the story. Aribeth’s dialogues are all unique to her character and act either to unveil new plot points or to remind the player of the task that must currently be completed in order to progress the story. However, while there is not the problem of numerous other characters all talking like Aribeth,



Figure 2.1: The reoccurring dialogue where Aribeth inquires as to the player’s progress in Neverwinter Nights.

she does get rather repetitive in her own right. Take the dialogue found in figure 2.1 for example. During the first chapter of NWN, the player is tasked with the recovery of four magical creatures. During this time, any attempt to converse with Aribeth will lead to this dialogue. While no other character in the game uses this dialogue, the fact that Aribeth asks the player about their progress in the exact same manner each time is itself rather immersion-breaking. This form of monotony I shall identify as *single-character repetition*.

The next category of conversations consists of those that belong to the companions that the player can enlist. The conversations generally have a few options that the player can use to control the NPC’s behaviour, an option to abandon the NPC and an option to inquire about the NPC’s past. As can be seen in figure 2.2 there is a great deal of shared structure, and some shared dialogue lines, between these conversations. The largest portions of these dialogues is found in the discussions about the NPCs’ pasts, and, much like the plot driving conversations of the first category, these portions of the dialogues are written uniquely for each character. Unfortunately, other than a bit of progression in the dialogue options about the companion’s past, the content of each companion’s dialogue is the same each time the player speaks with them. This is the same *single-character repetition* described previously



Figure 2.2: The identically structured dialogues of several companion characters in Neverwinter Nights.

and leads to a similar decline in the player's gaming experience.

Finally, the remainder of the conversations in the game are what I will call secondary conversations. These conversations are characterized by the fact that they are spoken by secondary NPCs and are not necessary for the progress of the main plot line. While some of these dialogues are unique to a single character, such as the extortion dialogues described in chapter 1 of this dissertation, a great deal of other secondary conversations are shared between multiple characters. This kind of *multi-character repetition* is perhaps even more distracting to players, as it can lead to hordes of NPCs who all speak in exactly the same manner. Although many of these dialogues will contain several variants to introduce some variety, there are a great deal of such secondary conversations that also produce *single-character repetition* as well.

There are therefore two related, but distinct, deficiencies present in the conversations of Neverwinter Nights that must be avoided. *Single-character repetition* is the most common and prevents characters from being dynamic and lifelike. It is unlikely that many people would use identical wording every



Figure 2.3: An example of Mass Effect’s synopsis based conversations. (taken from [8])

time they had a conversation, even if the conversations they had were on the same topic, and therefore the fact that NPCs do so seems unnatural. *Multi-character repetition* is less common but of at least as much concern. Whereas *single-character repetition* only causes a single character to sound fairly static, *multi-character repetition* causes a whole crowd of NPCs to all sound exactly the same. In both cases, however, the overall impact is the same: the repetition in dialogues reduces the player’s immersion in the game world and negatively impacts their game playing experience.

2.1.2 Mass Effect

Mass Effect [6] is another role-playing game developed by BioWare. However, the nature of the dialogue in Mass Effect is quite different from the dialogue found in NWN. The conversations in Mass Effect feature fully-voice-acted PCs and NPCs. To allow the conversations to flow more naturally, the player is presented with only a brief synopsis for each of the dialogue options available. Such a set of dialogue options can be seen in figure 2.3. After the player selects a dialogue option, the PC will speak the full line and the conversation will continue. The use of the synopsis lines allows for the dialogue to be fully

voice acted without the repetition of reading the full line and then hearing the full line. Regardless, while the presentation of the dialogue differs from other games, the underlying structure remains that of a dialogue tree, such as those found in NWN.

Two problems were identified in the dialogue of NWN, and the first, *single-character repetition*, is also clearly present in Mass Effect as well. While some characters do change what they say throughout the game as certain events occur, speaking with an NPC multiple times at any given point in the game will lead to a repeat of the same conversation. Once again, this repetition leads to the same static, unnatural characters that it did in NWN.

It is on its approach to the second problem that Mass Effect differs from NWN. Whereas NWN featured secondary characters who would all speak from the same scripts, the secondary characters in Mass Effect are largely mute. While this certainly avoids the problems caused by multiple characters who all sound the same, it does so in a manner that may leave the player rather dissatisfied. Instead of NPCs who all sound the same, the player is presented with NPCs who completely ignore the PC's presence. From the perspective of increasing the player's immersion, this is simply not a sufficient solution.

2.1.3 Oblivion

The game Oblivion [28], which was developed by Bethesda Softworks, takes a different approach to dialogue. Instead of structuring conversations around dialogue trees, as was done in NWN and Mass Effect, the conversations in Oblivion use a wiki-based approach. As can be seen in figure 2.4, the PC does not have actual dialogue lines. Instead, the player selects from a list of topics and the NPCs will then make a comment on the selected topic.

This approach to dialogue allows Oblivion to reduce the problem of *single-character repetition* present in the previously examined games. The conversation topics and NPCs responses are selected from a general pool of topics and responses. Furthermore, the topics have multiple responses ensuring that NPCs will not always reply in the exact same manner. Unfortunately, there are a relatively limited number of available responses for NPCs to choose from.



Figure 2.4: An example of Oblivion’s wiki-based dialogue system.

While the use of these topic pools and responses reduces the problem of *single-character repetition* it does so in a manner that exacerbates the problem of *multi-character repetition*. There is too little variation in the responses for the number of NPCs who are sharing them. This leads to characters who, while being dynamic and interesting in their own right, all sound as though they are copies of one another.

Consider the following example. There are three characters in NWN who each have a single-line dialogue. Each of these dialogues are unique, but the character will always speak the same dialogue. Compare this with a similar situation using Oblivion’s wiki-based system. There are, once again, three characters and three single line dialogues. However, this time the characters share the dialogues and select one to speak at random. Whereas in the NWN example, the result is three unique, but static, characters, the Oblivion example leads to three dynamic, but identical, characters.

In addition to the wiki-based conversations between the PC and NPCs, Oblivion also strongly features ambient NPC to NPC conversations. NPCs will approach one another and select a conversation topic from a shared pool. They will then exchange a series of remarks on the topic and then either select a new topic or end the conversation and move on. These conversations occur throughout the game world in an attempt to make the NPCs seem

more lifelike. However, as is the case with the PC to NPC conversations, there are too few topics for the NPCs to discuss. Consequently the resulting conversations become very repetitive.

2.1.4 Fallout 3

Also developed by Bethesda, Fallout 3 [29] did away with the wiki-based dialogue system of Oblivion and instead adopts the more traditional, dialogue-tree-based dialogues of games like NWN. Unfortunately, while this does reduce much of the *multi-character repetition* found in Oblivion, it does so in a manner that produces the same static, unnatural characters found in the previously examined games.

In addition to the *single-character repetition* found in the PC to NPC dialogues, Fallout 3 also features ambient conversations similar to those in Oblivion. While the variety of the conversations is somewhat improved, the player will still overhear the same conversations many times. This increase in the number of ambient conversations, in addition to the lack of shared PC to NPC dialogues, leads to a substantial decrease in the amount of *multi-character repetition* in comparison to what was found in Oblivion. This decrease in *multi-character repetition* comes with a substantial increase in the amount of *single-character repetition*, however. The result is dialogue that is similar to that found in NWN.

2.1.5 Facade

Façade [20] is an interactive drama developed by Michael Matheas and Andrew Stern. Unlike the other games that have been examined so far, Façade was not developed as a commercial game but was instead created as part of a research project. In Façade, the player is invited to a small dinner party with a couple of old friends. Unbeknownst to the player however, the friends, who are now married, have been going through a rough spot in their marriage. It is this marital conflict that drives the game's story.

The game is fairly short, requiring around 20 minutes per play-through, but is quite broad in scope. The story line, instead of being fixed, is made

up of a number of interchangeable set pieces that link into an overarching narrative. In addition to the dynamic approach to the plot, conversations in *Faade* are decidedly more dynamic than those found in most games. Instead of using a dialogue tree or wiki-based system, where player involvement in the conversation is restricted to predefined conversation nodes and player choice is limited to the list of options that the game presents, *Faade* allows players to type responses directly. The game will then attempt to match the intent of the player’s statement with one of the underlying dialogue options and have the NPCs respond accordingly.

Faade’s approach to dialogue is not without its flaws, however. Interpreting the intent of a player’s statement is not always possible. And while players are free to type any statement they wish, these statements must all be mapped to a more limited set of underlying dialogue options. This can lead to the game misunderstanding the player, or the NPCs acting as though they did not understand the player. Furthermore, the effort required to create the content modules for *Faade* is enormous. The creation of further *Faade*-like games is simply not practical until a number of content-generation bottlenecks have been alleviated [22] [21].

2.1.6 GTA IV

Grand Theft Auto IV (GTA4) [9] is a sandbox game developed by Rockstar Games. The game is quite a bit different from the others described here as it lacks player-driven dialogues. It does, however, have a large number of ambient NPC conversations similar to those found in *Oblivion* and *Fallout 3*. These conversations are notable for the variety of topics, as well as the variety of voice actors. The NPCs’ accents are appropriate to the specific boroughs of Liberty City that they reside in. Topics are relevant to the current state of the game, and, moving slightly beyond the dialogue itself, even the foreign language signs found on certain shops are properly translated. Unfortunately, this level of detail in the ambient conversations is not without its price. It has been reported that GTA4 had a development team of approximately 1000 people and a budget of around \$100 million [2]. It is simply not feasible for

many studios to invest such a large sum of money into the development of a game and so a more efficient solution is necessary.

2.2 How It's Made: Creating the Dialogue

It is important to have a clear picture of the current state of dialogue in story-based games. However, it is perhaps even more important, for those who wish to improve upon the current state, to understand the tools that are currently used to make that dialogue. At the coarsest level, there are three approaches to creating dialogue for games.

The first approach is to manually enter the dialogue. There are a variety of ways to accomplish this, but it may involve writing scripts that contain the dialogue to be displayed, writing scripts that read the dialogue from a file, or something similar. The common aspect to any manual approach is that the designer must write both the scripts to display the dialogue as well as the dialogue content itself. While this approach is not as widely used as it once was, it is still present. *Faade* is an example of a game that was developed using such an approach.

2.2.1 Automated Dialogue Generation

The second approach to creating dialogue for games is to use automatic generation. Such generation could involve the complete generation of text from some knowledge base, or it could be performed by some other process such as the paraphrasing of exemplar lines. With both approaches, designers are potentially able to obtain large quantities of dialogue for substantially less effort than would be required to create all of the dialogue manually.

Unfortunately, generated dialogue tends to suffer from certain drawbacks. As can be seen in the work of Kacmarcik [14], in order for current NLP techniques to reliably create convincing dialogue, the generated dialogue must be rather simplistic and quite rigidly structured. While it may be easier to achieve satisfactory results by paraphrasing existing dialogue lines, the current state-of-the-art in paraphrasing [18] is not currently able to achieve the qual-

ity necessary for practical use. Additionally, the adoption of either technique mentioned above presupposes the willingness of designers to relinquish some control over the dialogue in their games. As the reputations of game studios are heavily impacted by the quality of the writing in their games, it is not clear that designers would be willing to do so.

Greater success has been seen in the use of NLP techniques in relation to PC dialogue lines. As discussed previously, *Faade* has players type their own dialogue lines instead of selecting from a menu of choices. This has the benefit of eliminating the need to write explicit PC dialogue, but allowing players to choose their own dialogues requires that designers account for considerably more variety in PC responses. As PC lines must still be identified implicitly, and a greater number of NPC lines must be written, it seems clear that *Faade*’s approach does not reduce the amount of effort required to write dialogue. This increased dialogue complexity, in combination with the risk of parsing errors presented by allowing players to select their own responses, suggests that while such a freeform approach to dialogue is interesting, it is not currently practical for large scale games.

2.2.2 Current Tools

The third approach to creating dialogue is through the use of a specialized dialogue editor within a toolset. The presence of such an editor is not guaranteed by the presence of a more general toolset. The editor for the Unreal Engine [12], which is used in the game *Mass Effect*, is an example of a toolset that provides extensive tools for level building and scripting, but provides no dialogue editor. The presence of such an editor allows the designer to write game dialogue without needing to write large quantities of scripting code. While some scripting may still be required to decide which specific lines of a conversation are visible, there is no scripting required to control the display of the dialogue or the selection of dialogue options. This allows writers to create dialogue for games without needing to be capable of programming.

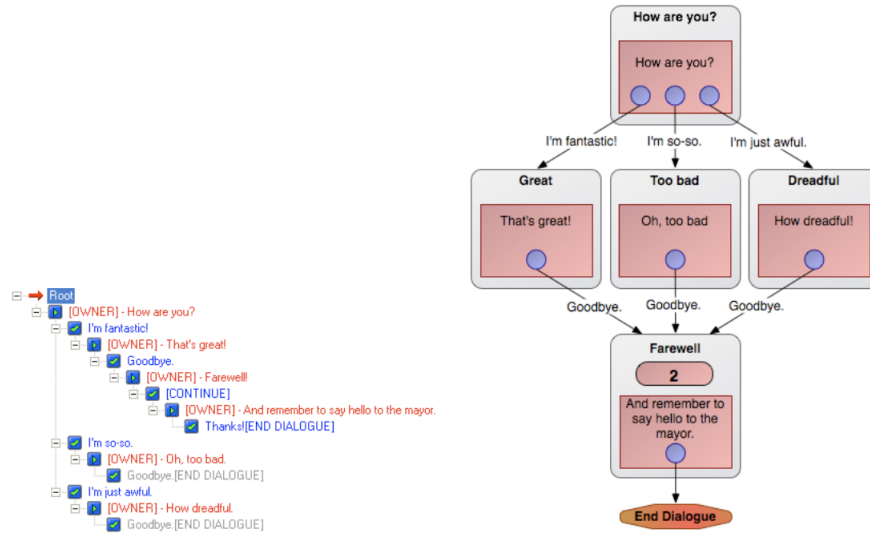


Figure 2.5: A comparison of a dialogue in both the NWN editor (left) and as an abstracted graph (right). (from Siegel [26])

Neverwinter Nights: The Aurora Toolset

Returning to the games addressed previously, when NWN was released, it included the toolset that was used to develop the game. This toolset, the Aurora Toolset, allows players to develop their own game modules that can then be released to the community at large.

Many of the tasks that can be performed in the toolset are quite straightforward. The placement of terrain, objects, and NPCs is performed through the use of selection menus and brushes. After selecting the type of object, NPC, or terrain that is desired, the designer then selects where in the game world it should appear. Terrain, buildings, and other large scale objects are made up of tiles, while smaller objects can be placed anywhere within the bounds of the currently existing tiles. Anyone familiar with an image editing program (e.g. Photoshop, Gimp, MS Paint, etc.) should be able to learn how to use this part of the toolset without much difficulty. However, other tasks, such as scripting and dialogue creation, are less intuitive.

Scripting in NWN is done in the NWScript language, which was developed specifically for NWN by BioWare. The language is C-like and relatively easy for someone with programming experience to learn. The language is not, how-

ever, easily accessible to non-programmers, due to their lack of experience with programming concepts. As NPC behaviours, in-game events, and dialogue flow are all controlled by scripts, this makes it difficult for non-programmers to make use of these portions of the toolset.

Dialogue creation in Aurora has two major drawbacks. The first, as described above, is that control of a dialogue’s flow is performed through the use of scripts. These scripts decide whether a line will be displayed, or not, at a given time. The second problem is that “tree-based” dialogues are rarely actually tree-based. Most tree-based dialogues are, in fact, more general graphs, as they allow cycle forming back-links and for children to have multiple parent nodes. Therefore, a dialogue editor, such as the one found in Aurora, that attempts to display dialogues as trees quickly becomes difficult to use as the size and complexity of a dialogue increases. This problem is illustrated in figure 2.5, which shows the same dialogue in both tree and graph form. While relatively easy to understand as a general graph, the symbolic links makes the dialogue difficult to follow when displayed as a tree.

Neverwinter Nights 2 Toolset

Obsidian Entertainment’s Neverwinter Nights 2 (NWN2) was, much like the original NWN, also released with the toolset that was used to develop the game. In general, the toolset for NWN2 is much more powerful than the Aurora toolset, but this comes at the cost of added complexity. The most drastic differences between the toolsets are found in the level editing tools. Whereas NWN is tile based, the levels in NWN2 are created using freeform brushes.

In relation to dialogue, however, the NWN2 toolset differs only slightly from the Aurora toolset. Both games use a tree-based representation of dialogue, and both games use the NWScript language to control the flow of the dialogues. In fact, the dialogue systems are so similar that it is possible to import NWN dialogues into the NWN2 toolset. While it is possible to attach larger numbers of scripts to dialogues and to pass parameters to these scripts, these changes are not to the benefit of non-technical users. Similar to other

aspects of the NWN2 toolset, power is emphasized over usability. Due to this, the NWN2 toolset suffers the same drawbacks as the Aurora toolset in regards to amount of necessary programming knowledge and easy to follow dialogues.

The Elder Scrolls Construction Kit

Oblivion is another game that had its toolkit released to the community. From a level-building perspective, the TES Construction Kit is much more akin to the NWN2 toolset than it is to NWN's Aurora toolset. There is, however, one major difference: the TES Construction Kit allows users to add new content to the existing game world of Oblivion, instead of allowing them to create new, unrelated game modules.

As was discussed previously, Oblivion features a wiki-based dialogue system instead of the more common tree-based dialogue system of games like NWN, NWN2, and Mass Effect. Dialogue is written by creating topics and then associating a set of responses with the topic. Scripting is then used to control which topics are available to an NPC at a given point in the game. While it may be possible to mimic tree-based dialogues in Oblivion through the use of large numbers of topics and heavy scripting, it is clearly not the intent of the included editor. Unfortunately, the lack of ability to easily create tree-based dialogues and the presence of the same power-before-usability design principles that were found in the NWN2 toolset leave the TES Construction Kit rather lacking as a model for future dialogue creation tools.

2.2.3 Better Tools

It is clear that the current state of the dialogue creation tools found throughout the games industry is lacking. There has, however, been a great deal of work done with the aim of improving these existing tools. It is almost certain that such work has been, and continues to be, done within the games industry. However, such work is not generally released publicly due to the competitive nature of the industry. Therefore, this section will focus on the work that has been performed within academia, as this work tends to be more freely available.

Wide Ruled

Wide Ruled [27] is an interactive-story authoring tool developed at the University of California, Santa Cruz. It is an extension of the Universe author-goal-based model for story generation [16], and attempts to provide users with an interface that is useable by non-programmers. While somewhat successful in this regard, user feedback suggests that many aspects of the tool continue to be difficult for non-expert users to make use of effectively [27]. Furthermore, Wide Ruled relies heavily on plot devices that may not be applicable when writing game dialogues in general. When playing a game generated by Wide Ruled, a player is able to interact with the story only when there are multiple valid plot fragments related to the PC that the engine must decide between. It may be possible to simulate traditional game dialogue through the use of large numbers of small plot fragments, but it is clear that this is not the intended use. As such, while Wide Ruled is interesting as a plot generation tool, it does not provide a good model for developing better dialogue tools.

ScriptEase

ScriptEase [19] is a tool that is being developed by the games group, of which I am a part, at the University of Alberta. ScriptEase allows non-programmers to create scripting code without any explicit programming through the use of generative design patterns. These patterns are divided into four categories: encounter, quest, behaviour, and dialogue patterns. The patterns are abstractions and allow for the separation of the specification and the implementation of game design tasks. In order to create an instance of an existing pattern, a user need only select a pattern and set its parameters. Once the parameters are set, ScriptEase will generate the scripting code necessary for the implementation of the pattern, and an instance of the desired pattern will have been created without any code being written by the user.

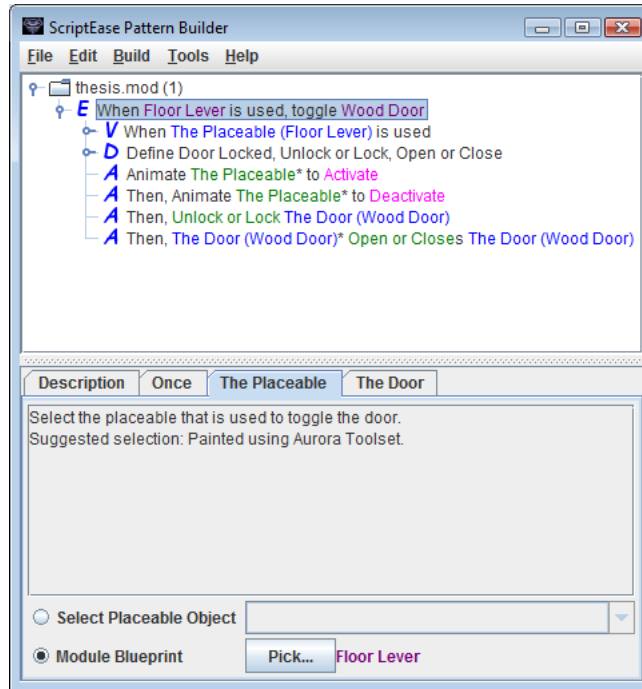


Figure 2.6: The *Placeable Use - Open Door* encounter pattern in the ScriptEase tool.

ScriptEase Encounters

The simplest patterns in ScriptEase are the encounter patterns. An encounter is a set of actions that is triggered by an in-game event. An encounter pattern consists of an event, and potentially several definition, condition, and action atoms. An atom is the lowest level ScriptEase component and makes up a single line of a pattern. An example of such a pattern is the *Placeable Use - Open Door* pattern. In this pattern, which can be seen in figure 2.6, when the specified placeable is used the associated door will open.

If the pattern catalogue does not contain a pattern that meets the designer's requirements, a pattern can be adapted by adding additional atoms. Consider a situation where the the designer wants an NPC butler to open the door for the PC when a bell is rung. While similar to the *Placeable Use - Open Door* pattern, the desired pattern differs in that the door should not open itself, but should instead be opened by the NPC. In order to achieve the desired result, the designer must adapt the existing pattern by adding a definition for the NPC and changing the parameters of the close door action atom so that the

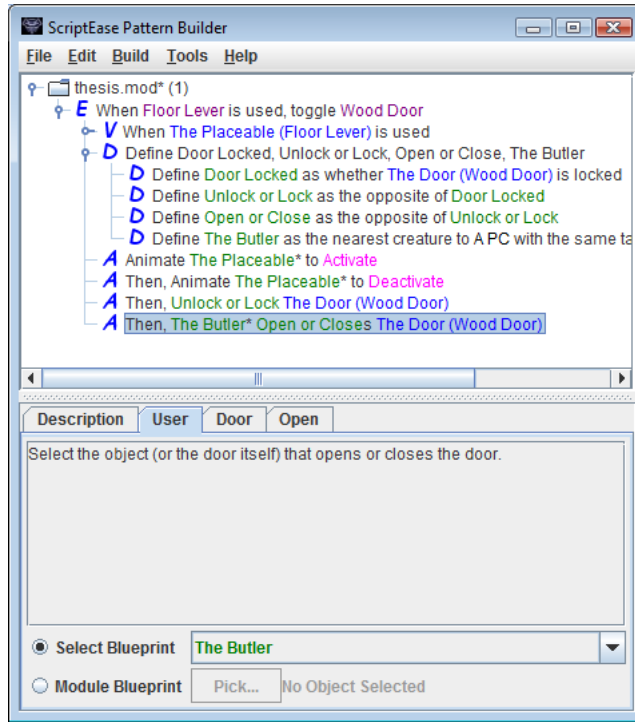


Figure 2.7: The adapted *Placeable Use - Open Door* pattern.

NPC closes the door. The result of such an adaptation can be seen in figure 2.7.

If no pattern can be suitably adapted, it is also possible to create entire patterns from scratch. To do so, the designer must select the desired event, select any necessary parameters, add the default atoms, and select the name and description for the new pattern. After the creation process is complete, the new pattern is used in the same manner as any other ScriptEase pattern.

The only encounter related task that a designer requires specific programming knowledge for is the creation of new atoms, though once an atom has been created it can be reused without any further programming required. If, for example, a designer desired an *Explosion* action atom, there is currently no such single atom in the ScriptEase pattern library. Figure 2.8 shows part of the atom creation process. Here, the designer specifies the necessary parameters for the atom and writes the actual scripting code. Much like creating custom patterns, once a new atom has been created, it is used in the same manner as any other atom.

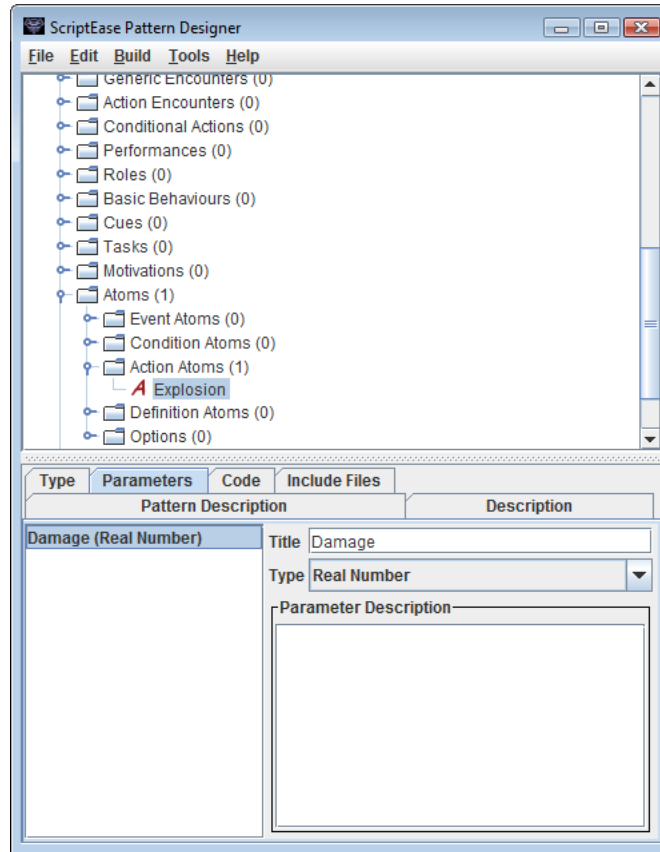


Figure 2.8: The creation of an *Explosion* action atom.

ScriptEase Quests

Additionally, ScriptEase also contains another group of patterns called quest patterns. Quest patterns are used to manage the state of quests. A quest pattern is composed of quest points. These quest points contain a set of conditions to satisfy the quest point; a journal entry to be displayed once the quest point is completed; and potentially a set of actions to be performed upon completion of the quest point. The quest points are then linked together and ScriptEase ensures that the quest points cannot be completed outside of the specified order. Figure 2.9 shows an *Exterminate* quest as displayed within the main ScriptEase tool. An exterminate quest requires the PC to kill a specified number of a certain type of creature.

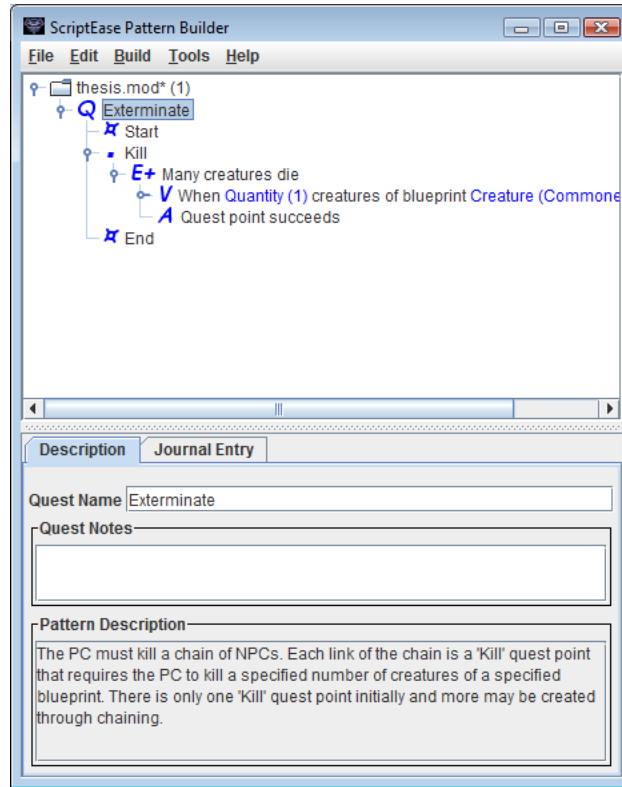


Figure 2.9: An *Exterminate* quest as viewed in the ScriptEase editor.

ScriptEase Behaviours

The third category of patterns in ScriptEase is behaviour patterns. These patterns are applied to NPCs and cause them to act in a certain manner. One of the simplest behaviour patterns would be the *Loiter* pattern, in which the specified NPC will wander about an area. Behaviour patterns are divided between latent and proactive behaviours, as well as between individual and collaborative behaviours. Latent behaviours are those that are triggered by an in-game event, whereas proactive behaviours are those that the NPC will choose to perform automatically. Much as the names suggest, independent behaviours are those that are performed by a single NPC, and collaborative behaviours require multiple NPCs. If the previously mentioned *Loiter* behaviour was set to occur automatically, it would be an example of an independent proactive behaviour. Multiple behaviours can be encapsulated within a single *role*. An example of a *Guard* role can be seen in figure 2.10. The *Guard* role is a combination of several independent proactive and independent latent behaviours

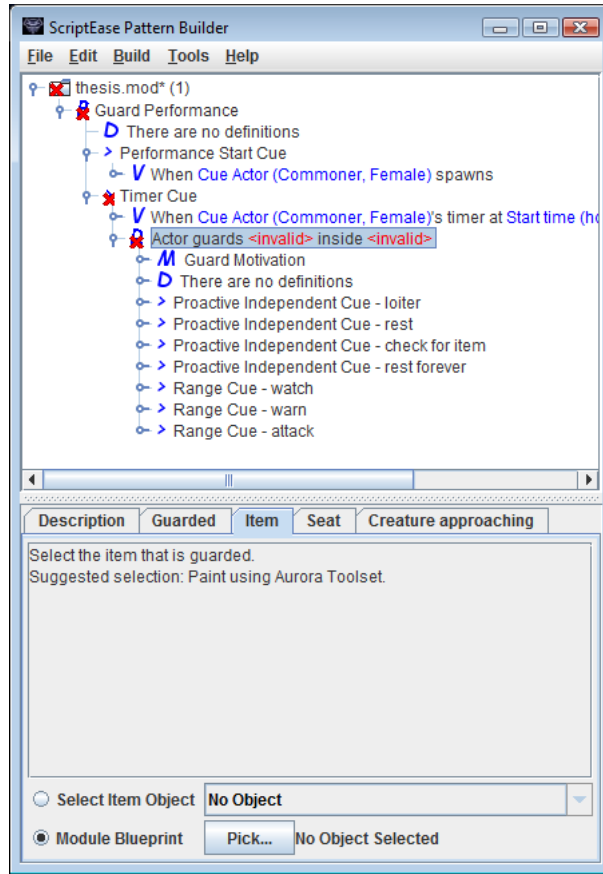


Figure 2.10: A *Guard* role as viewed in the ScriptEase tool.

that result in the specified NPC providing a detailed guard performance.

ScriptEase Dialogue

Finally, the ScriptEase patterns that are most relevant to this dissertation are the dialogue patterns. Dialogue patterns include both non-scripting structural patterns as well as scripting-related patterns such as decisions, choices and filters. Structural dialogue patterns differ from other ScriptEase patterns in that they do not involve the generation of scripts. These patterns, instead, generate the actual structure of the dialogues by dividing them into exchanges and topics. An exchange is a single NPC dialogue line and its associated PC responses. A topic is a series of exchanges where all but the last exchange have only one possible PC response, and this response must lead to the next exchange in the series. Figure 2.11 shows a dialogue that consists of a single topic with multiple exchanges.

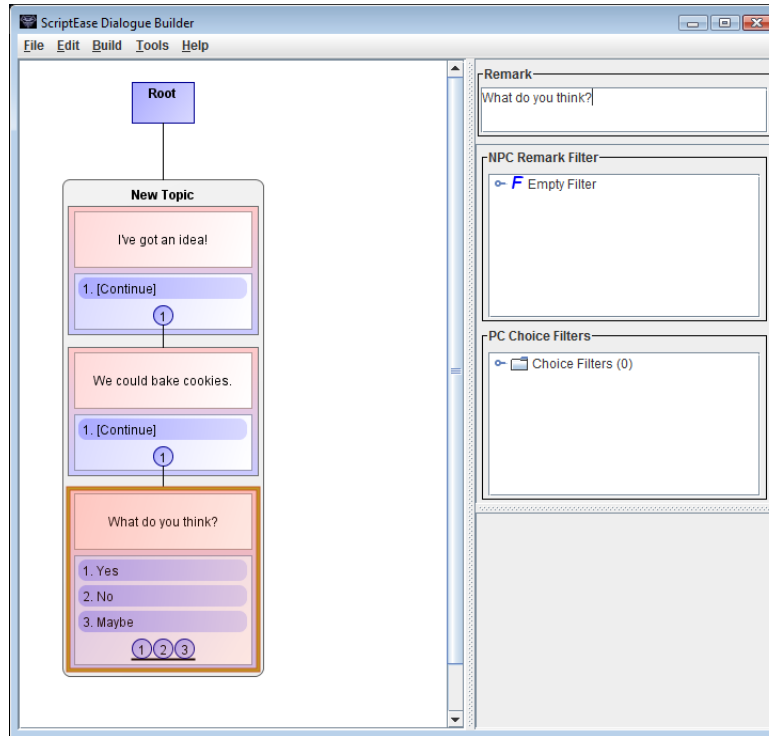


Figure 2.11: An example of a topic with multiple exchanges in the ScriptEase dialogue tool.

In order to facilitate the use of the structural dialogue patterns, ScriptEase provides its own dialogue editing tool. The tool allows designers to create exchanges and topics, and to link them together to form a dialogue graph. While the graph generated by the tool is presented as a tree, there are several improvements over the tree view found in other editors, such as the one in the Aurora toolset. In Aurora, as pictured in figure 2.12, the children of a given node are spread increasingly far apart as the depth of the tree increases. By presenting the dialogue as exchanges, the ScriptEase dialogue tool avoids this problem, as PC responses will always be grouped with the NPC remark that they are replying to. Additionally, when the ScriptEase tool uses symbolic links, it allows the user to re-root the original node at the position of the link for greater clarity. This compromise, between Aurora style symbolic links and using entirely direct links, allows ScriptEase users to avoid much of the confusion that symbolic links can cause while also avoiding the confusion that can be caused by excessive direct links cluttering the dialogue graph.

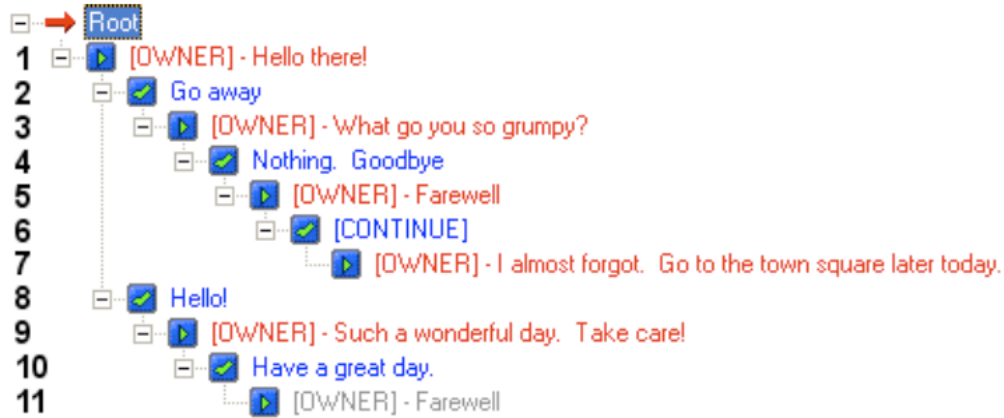


Figure 2.12: A dialogue as displayed in the Aurora toolset. Note the distance between sibling nodes near the root of the dialogue. (from Siegel [26])

In addition to the structural patterns, there are also *decision* and *choice* patterns. A decision pattern is used to select between different NPC remarks or whole topics. A choice pattern is used to control which elements of a set of PC remarks should be displayed. Both types of patterns are constructed using *filters*, which are conditionals that are constructed by combining multiple clauses that evaluate to true or false based on game state. Decision patterns differ from choice patterns in that decision patterns will select only the first clause that evaluates to true, while choice patterns will return all clauses that evaluate to true. The distinction between decision and choice patterns is necessary due to the nature of tree-based game dialogues. In a tree-based dialogue, scripts must decide on the exact line that the NPC should use. Players, however, are presented with a menu of dialogue options, and so the pattern must return all appropriate dialogue options instead of merely the first appropriate option. The combination of decision, choice and filter patterns allows designers to control the flow of a dialogue without needing to explicitly write the scripts to do so.

In order to construct a dialogue in the ScriptEase tool, the designer begins by adding either a topic or a decision to the root of the tree. To illustrate the process, consider the creation of the simple dialogue found in figure 2.13. To begin, the author must right click on the root of the dialogue and add a

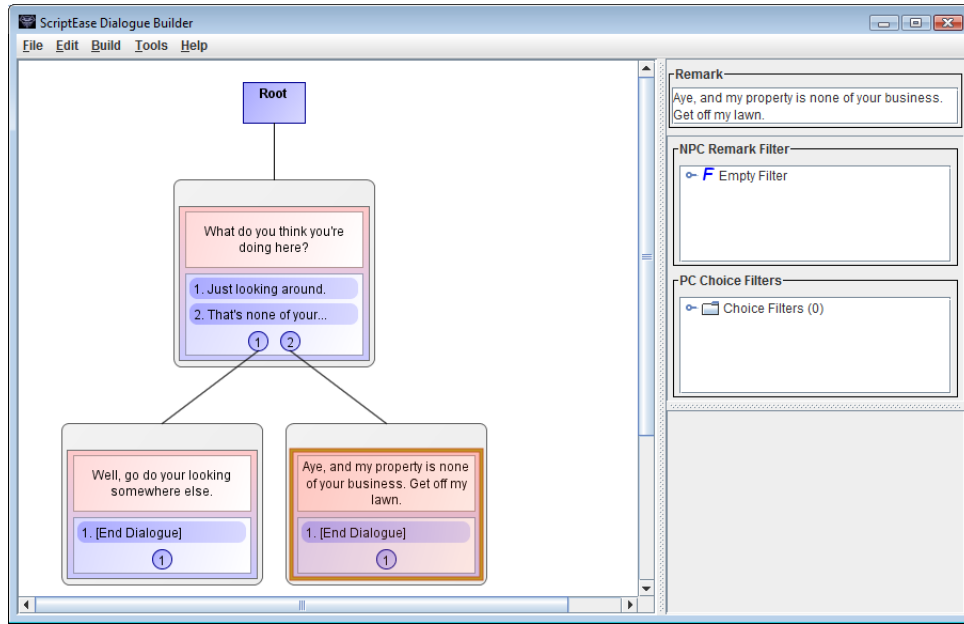


Figure 2.13: A simple dialogue built in ScriptEase.

new topic. Clicking on the exchange in this topic allows the author to enter the NPC's dialogue line. Any new topic will have a single PC dialogue line automatically. To set the PC's first dialogue line, the author need only select the line and enter the text in the text box in the upper-right corner of the editor. To add the PC's second dialogue line, the author must right click on the first PC line and select the option to append a new dialogue choice. By selecting the newly created PC line, the author can now set the text of this line, as well. At this point, the first exchange has been fully populated. To add the additional topics, the author right-clicks on the circular links at the bottom of the exchange and selects the option to add a new topic. This process is repeated to add the other topic, as well, and the dialogue lines of these new topics are set in the same manner as those in the original topic.

2.3 Classifying Dialogue Lines

It is clear, then, that while there is a need for better tools that reduce the amount of effort required by designers to write dialogues, the solution must be one that also leaves the designer in control of the final product. In chapter 1, I briefly outlined my *Intentional Dialogue* system. The base element

of intentional dialogues is the *Intentional Dialogue Line*, and an intentional dialogue line has, associated with it, a collection of actual dialogue lines. In order to ensure that appropriate actual dialogue lines are selected for a character's dialogue, there is the need to assign various labels to these lines. While it is possible to sort the lines by hand, such an approach is time consuming and therefore less than ideal. Instead, I have developed a machine learning classifier to automate the sorting process.

Previous work[1] has shown support vector machines (SVM)[30] to be successful at labelling the sentiment of a body of text. As labelling properties, such as the sophistication or disposition of a dialogue line, appears to be quite strongly related to such sentiment analysis, it seems likely that SVM would be an appropriate algorithm for the labelling required by my system, as well.

When using an SVM, input data is viewed as vectors in n-dimensional space (n being the number of features in the feature vectors). The SVM then attempts to construct a hyperplane that divides the data into two sets based upon their labels. The traditional SVM algorithm is only appropriate for use with input data that has two labels. However, there are extensions to the algorithm that allow for a greater number of labels by constructing a dividing hyperplane between each combination of labels and then testing a feature vector against each hyperplane in the set when performing classification. As my system uses properties with more than two values, I will be using such a multiclass SVM. There are a number of existing machine learning tools that are able to perform classification using SVMs[30][3]. I have chosen to use the libsvm library[3] for my classifier as it provides a large variety of SVM algorithms and can be easily integrated into the existing ScriptEase code base.

Chapter 3

Intentional Dialogues

3.1 Overview

There are a number of requirements that any new dialogue creation system must meet if it is to substantially improve the current state-of-the-art. Foremost, the dialogue system should be easy to use. Unlike the games industry of the past, modern game designers tend to be highly specialized in specific fields of game design. This specialization means that many of the writers who author a game’s dialogue do not have a technical background and so are not able to make use of tools that require programming knowledge. Secondly, the system should provide an efficiency gain over currently existing tools. The game studios are profit-driven, and in order for them to adopt a tool it must provide a measurable gain over what they are already using. Finally, the system must provide the user with sufficient control to ensure the quality of the dialogue that is created. For many game studios, especially those that make story-based games, the quality of their writing has a direct impact on their reputations, and reputation has an enormous impact on the success of a studio’s games.

To ensure the system’s ease-of-use, I am building upon the existing ScriptEase dialogue editor. As discussed previously, ScriptEase and its dialogue patterns allows for the creation and control of game dialogues by non-programmers through the use of generative design patterns. Furthermore, the ScriptEase editor provides a more intuitive and easy-to-follow view of dialogue than is provided by existing industry tools like the Aurora toolset.

In order to provide the necessary increase in efficiency, I am introducing the concepts of an *Intentional Dialogue* and an *Intentional Dialogue Line*. These constructs will allow game designers to quickly generate multiple dialogues by enabling the effective reuse of both new and existing dialogue content. While many current games are able to get away with mute NPCs or NPCs who share dialogues, they do not accomplish this without complaints from those playing the games. As the overall quality of games increases, so too will players' expectations increase. The ability to populate the dialogues of large numbers of NPCs without the dialogues being direct copies or being entirely hand written allows designers to satisfy the ever increasing expectations of players in a cost-effective manner.

Regardless of the gains achieved in usability and efficiency, however, no system will be adopted if it does not provide designers with the necessary control over the dialogue it creates. It is for this reason that I introduce a system that focuses on a combination of manual and semi-automatic methods. While the system does not exclude the possibility of including more fully-automated methods in the future, it ensures that the inclusion of such methods would be done in such a way as to leave the user in control of the final product.

3.2 Intentional Dialogues

An *Intentional Dialogue* is an abstracted generalization of a group of dialogues that share the same structure and intent. Examples of possible intentional dialogues include both the extortion and directory dialogues discussed in chapter 1. Considering the extortion example, the two dialogues were found to share the same structure, and while the exact text of the two dialogues differed, the underlying meaning of the dialogue lines, as well as the effect of the dialogue lines, was the same in both dialogues. If this shared structure and intent is abstracted away, the result is what is found in figure 3.1. An intentional dialogue would then seem to be a dialogue that is constructed of *Intentional Dialogue Lines*: abstract dialogue lines that represent a set of actual dialogue lines sharing a similar intent. These intentional dialogue lines are grouped

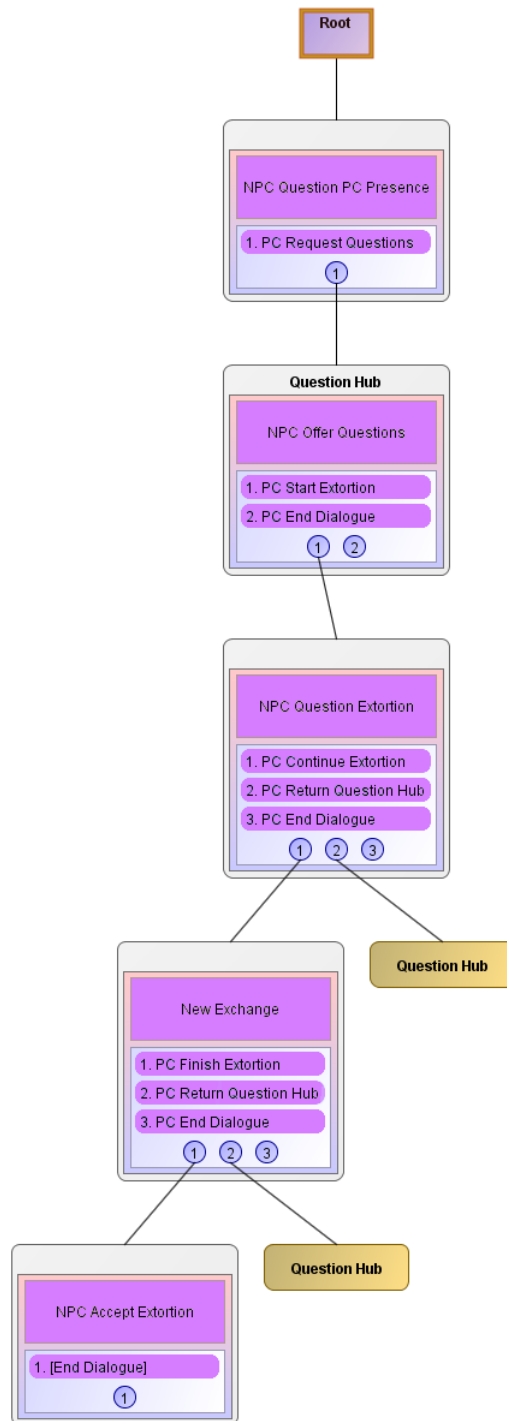


Figure 3.1: An Extortion intentional dialogue.

within exchanges, which are themselves found within topics, and these topics are then linked together to form the intentional dialogue.

An intentional dialogue can be simply a dialogue that is made entirely of intentional dialogue lines (with topics, exchanges, and links for structure), but is that sufficient to satisfy all of the uses a designer might have for intentional dialogues? What if, in the aforementioned extortion dialogue, the designer wanted the NPC to refuse to speak with the PC if the player follows through with the extortion attempt? In that case, a means of controlling the dialogue's flow is required. Fortunately, ScriptEase already has the required construct in its decision patterns.

In this case, the specific decision pattern that is required is a *Progress* decision pattern. A *Progress* decision is used to block access to part of a dialogue once a certain node has been reached. The pattern consists of two outcome nodes and a goal node. The first outcome node is the *Initial* outcome and it is the line of dialogue that will be displayed to the PC until the *Goal* node has been reached. The *Goal* node is reached either when it is displayed (for NPC lines) or when it is selected by the player (for PC lines). Once the *Goal* node has been reached, the decision will then redirect the conversation to the *Final* outcome node instead of the *Initial* outcome. For the extortion dialogue, the *Initial* outcome should be set to the line “NPC Question Player’s Presence,” the *Final* outcome should be set to the line “NPC Dismiss Player (Hostile),” and the *Goal* node should be set to the final NPC node of the extortion attempt in which the player receives the gold. The result of including the decision in the extortion dialogue can be seen in figure 3.2.

In addition to decision patterns, it would also be useful to include choice patterns to allow designers to guard PC dialogue lines. While not immediately applicable to the extortion dialogue, consider instead the dialogue of a bartender. The *Bartender* intentional dialogue depicted in figure 3.3 allows the PC to speak with the bartender NPC and buy a drink. There is a problem with the dialogue as it is shown, however. If the PC does not have enough gold to purchase the drink, they will still act as though they are able to. To solve this, two new choice patterns not in the existing catalogue must be created.

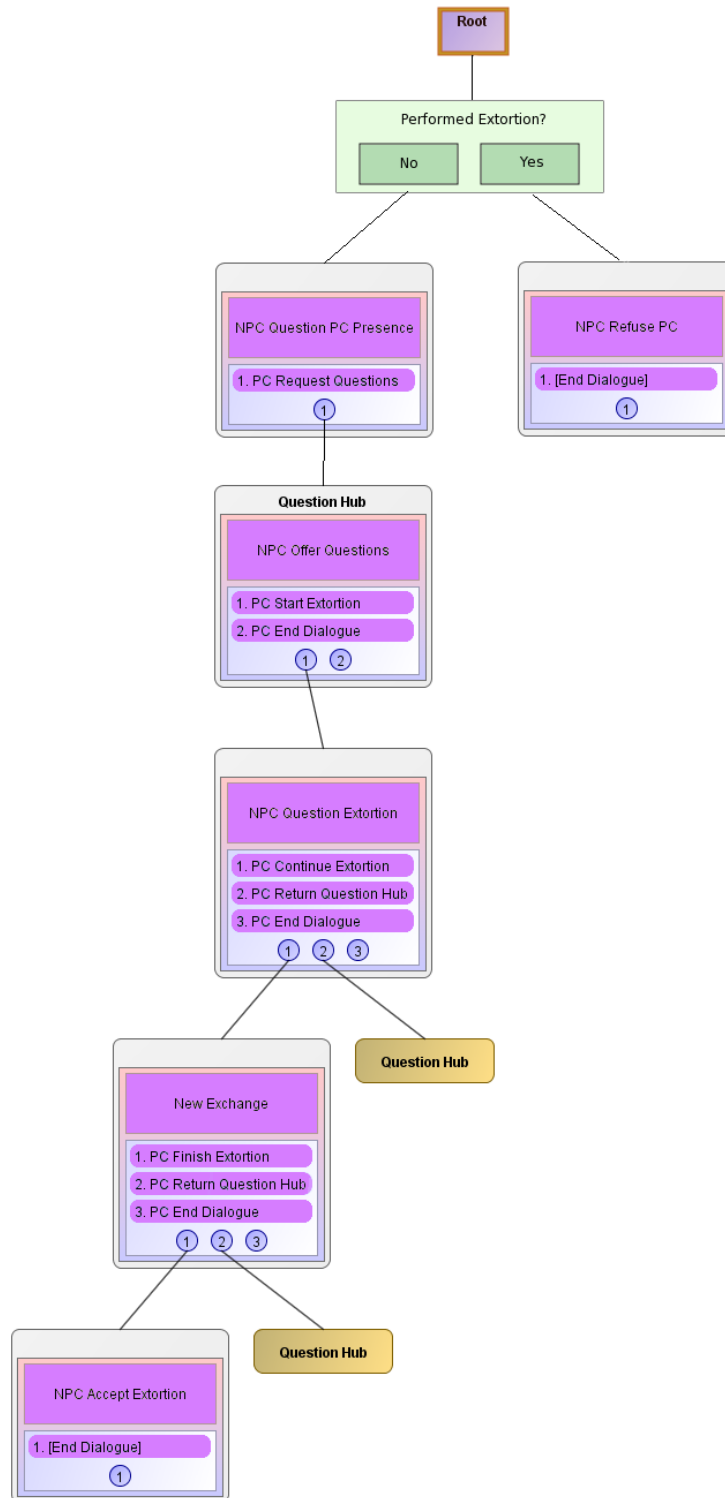


Figure 3.2: An Extortion intentional dialogue that includes a decision pattern.

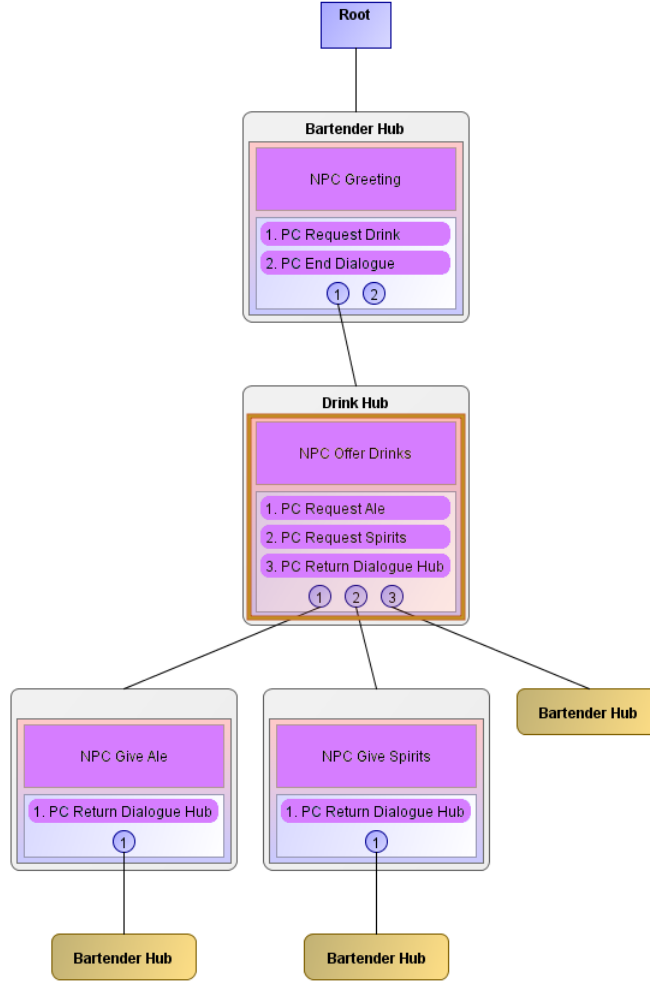


Figure 3.3: A Bartender intentional dialogue.

These choice patterns are the *PC has gold* and *PC doesn't have gold* choice patterns. As their names suggest, the *PC has gold* pattern will display the associated line if the PC has at least a certain amount of gold, and the *PC doesn't have gold* pattern will display its associated line if the PC doesn't have the specified amount of gold. By including some additional lines, and using these two choice patterns, a more robust *Bartender* dialogue can be created.

Are these all of the elements that should be included within an intentional dialogue? In order to take full advantage of intentional dialogues, it must be possible to adapt them in a similar manner to any other ScriptEase pattern. What if the designer wants to add some additional dialogue to the bartender NPC? This could be accomplished through the use of intentional dialogue lines,

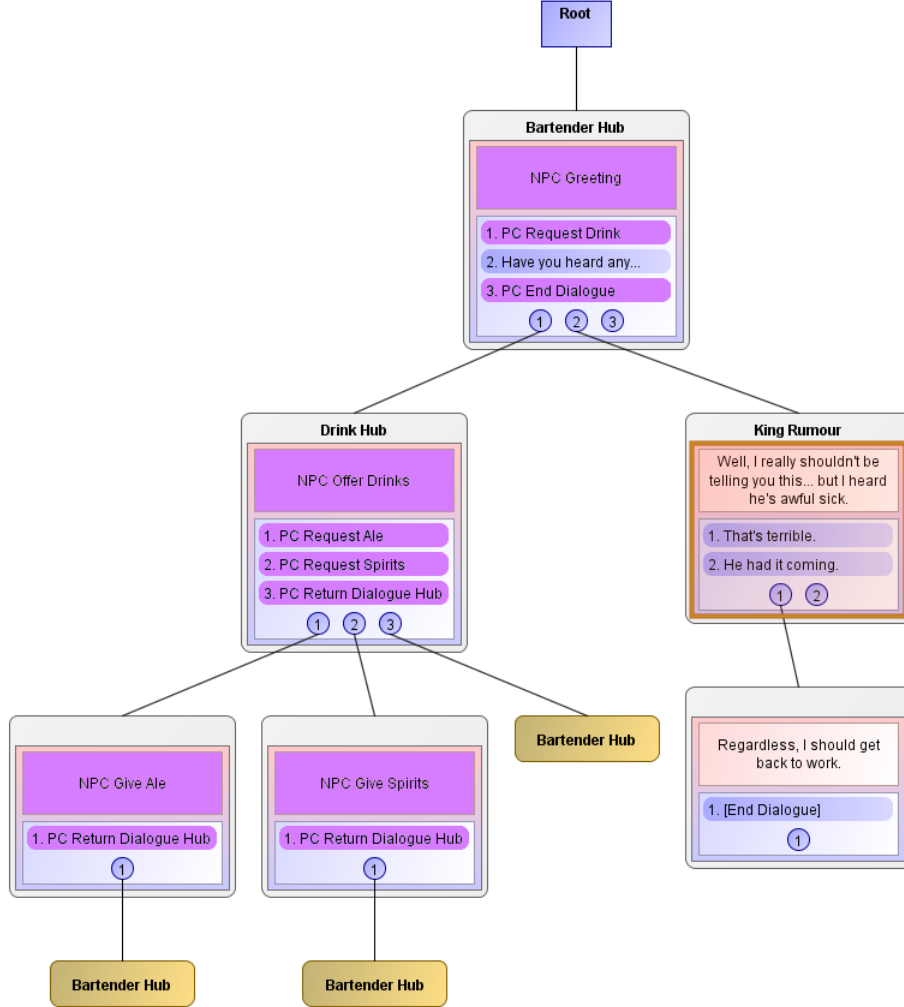


Figure 3.4: An adapted Bartender intentional dialogue that includes non-intentional dialogue lines.

but if the dialogue lines that are being added are only going to be used the one time on this one character, the use of an intentional dialogue line is somewhat excessive. Instead, if intentional dialogues were to allow the inclusion of non-intentional dialogue lines, as shown in figure 3.4, the designer would be able to perform the adaptation without the additional overhead of creating a new intentional dialogue line.

The topic of adaptation brings up yet more room for improvement in the formulation of intentional dialogues. While some intentional dialogues, like the extortion dialogue and the bartender dialogue, are intended to be used as the root of an NPC's dialogue, others like the *Directory* dialogue discussed

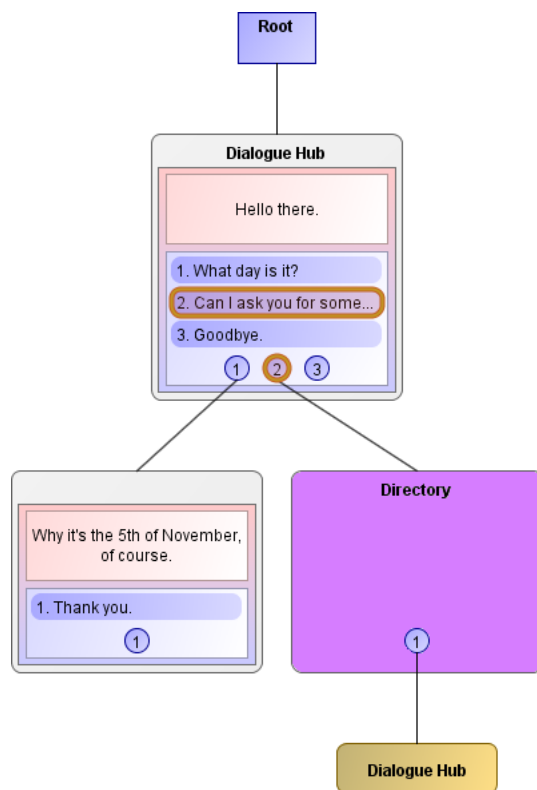


Figure 3.5: Prototype of an intentional dialogue collapsed into a single dialogue component.

in chapter 1 make more sense as a subsection of a larger dialogue. Therefore, I propose two more additions to intentional dialogues to facilitate their use within other dialogues. The first addition is a means of collapsing an intentional dialogue into a single element. This process, which is illustrated in figure 3.5, allows designers to treat an intentional dialogue as a single element similar to an exchange. This allows an entire intentional dialogue to be added to a larger conversation in the same manner that a single dialogue line would be.

The second addition is the inclusion of an optional *Hook* intentional dialogue line that is attached to the root of the intentional dialogue. Notice that in the previous example, the line that leads into the intentional dialogue (“Can I ask you for some directions?”) is not actually part of the intentional dialogue. It would seem beneficial to be able to include this line as part of the intentional dialogue. Unfortunately, ScriptEase dialogues, much like the NWN

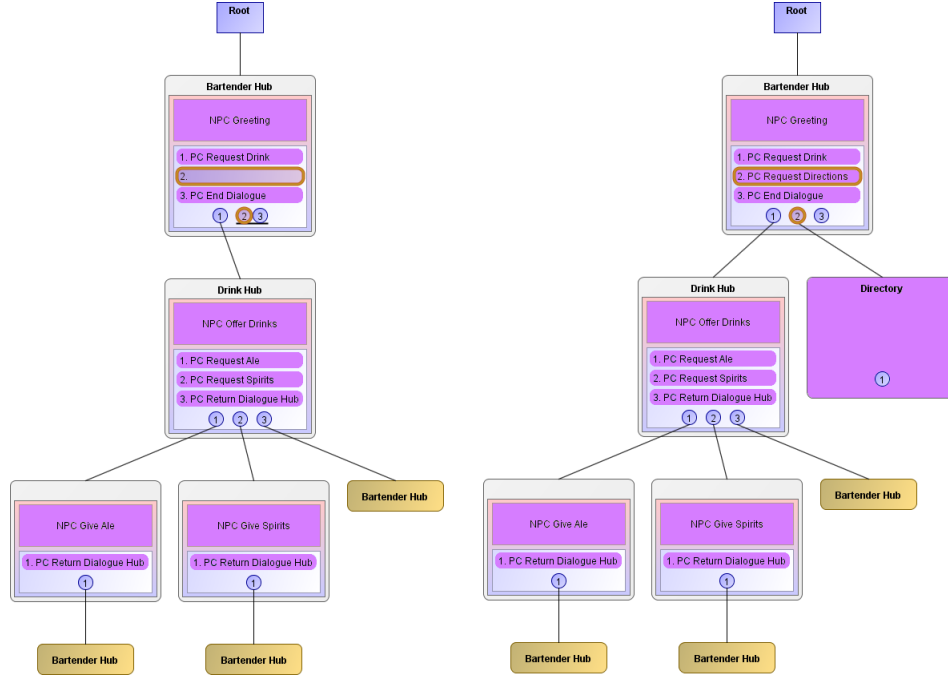


Figure 3.6: Prototype of a *Hook* line being propagated up when an intentional dialogue is included within another dialogue.

dialogues upon which they were designed, must begin with an NPC line, not a PC line. In order to address this problem, I propose that an optional *Hook* line is attached to the root of the intentional dialogue above the first dialogue line. If the intentional dialogue is attached to an empty PC remark, the hook line will be propagated up to replace the blank line. However, if the designer wishes to write the PC line manually or to use the intentional dialogue as the root of a conversation, the hook line will be ignored. This approach, as pictured in figure 3.6, is similar to the way that blank lines are replaced with “[Continue]” and “[End Dialogue]” in NWN’s Aurora toolset.

Is this, then, a complete formulation of an intentional dialogue? Not quite. What if the outer dialogue in the previous example was itself an intentional dialogue? What if a designer wanted to include the directory dialogue as part of the bartender dialogue? To do so would require that intentional dialogues could contain other intentional dialogues, as well. This ability proves useful as the composition of intentional dialogues within other intentional dialogues allows for the rapid construction of complex dialogues. This is similar to the

manner in which complex behaviours can be constructed within ScriptEase by including multiple simple behaviours within a role. There must, however, be one restriction placed on this ability. In order to prevent infinite recursion, an intentional dialogue cannot contain itself as a sub-component.

This, finally, is what an intentional dialogue is. For structure, an intentional dialogue contains topics, exchanges, and dialogue links. For content, an intentional dialogue contains intentional dialogue lines, as well as non-intentional dialogue lines if necessary or due to adaptation. For control of dialogue flow, intentional dialogues can contain both decision patterns and choice patterns. And for all of these needs, an intentional dialogue can contain other intentional dialogues. When included within another dialogue, an intentional dialogue has an optional hook line that can be used to connect it, and an intentional dialogue can be collapsed into a single element for ease-of-use. Most importantly, though, an intentional dialogue is a representative for a set of dialogues that all share a common structure and intent. By constructing an intentional dialogue once, a designer can then re-use the content across multiple character's dialogues with ease and without the static repetition found in so many of today's games. Furthermore, the use of an intentional dialogue need not be limited to the game for which it was initially created. By collecting dialogue content from existing games, a catalogue of intentional dialogues could be constructed much like was done for the other pattern types found in ScriptEase.

3.3 Intentional Dialogue Lines

The question that remains to be answered is how, exactly, does an intentional dialogue line work? As defined previously, an *Intentional Dialogue Line* is an abstracted generalization of a set of dialogue lines that share the same intent. The concept is an analogue of the intentional dialogue, but on the level of an individual dialogue line. Through the use of an intentional dialogue line, a designer is able to specify the desired meaning of a line while letting ScriptEase select an appropriate, actual dialogue line to be use in place of the intentional

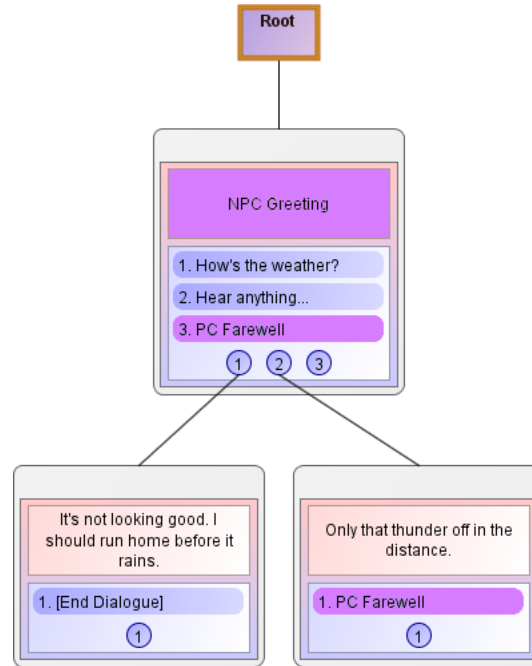


Figure 3.7: A non-intentional dialogue that makes use of intentional dialogue lines.

line at game time.

Intentional dialogue lines must, of course, be useable within intentional dialogues, but would it be useful to be capable of adding them to non-intentional dialogues, as well? If a designer is writing a one-off dialogue that will only be used once, it does not make sense to make the dialogue an intentional dialogue. However, certain intentional dialogue lines, such as *Greeting* or *Farewell* lines, may still be useful. Figure 3.7 shows such a situation. The use of intentional dialogue lines within a non-intentional dialogue allows designers to make use of the benefits of intentional dialogue lines without the added overhead of creating an intentional dialogue.

The potential for reuse of intentional dialogue lines between games is likely even greater than the potential for reuse of whole intentional dialogues. While certain character archetypes, and the intentional dialogues that were created for them, may not be applicable in all genres of story-based games, many of individual lines used to construct the larger dialogues may still be useful. By constructing a library of intentional dialogue lines from content present in existing games, the number of new intentional dialogue lines that must be

created when designing a new game can be greatly reduced.

3.3.1 The Grid

Knowing what an intentional dialogue line is does not explain how it works. If an intentional dialogue line is to act as a representative for a set of dialogue lines, it must have some way of associating the related actual dialogue lines to itself. While a generative process could, in theory, be used to create the actual dialogue lines from the intentional line, the deficiencies of such approaches, combined with the required degree of control over the finished product, makes such an approach less feasible in practice. In order to ensure that designers remain in control of the dialogues that appear in their games, it is necessary that an intentional dialogue line contains the full set of its associated actual dialogue lines at design time. This does not limit the techniques that a designer can use to create the actual dialogue lines; it simply requires that the lines are created before being associated with the intentional line. Therefore, an intentional dialogue line is, in many ways, a container for actual dialogue lines.

In addition to associating the actual dialogue lines with the intentional line, there must also be a means of selecting which actual line should replace the intentional line at game time. This could be accomplished through random selection, but the variation between actual dialogue lines with the same intent can still be quite large. It would be preferable if the intentional line could instead be replaced with the actual dialogue line that is most appropriate for the character speaking the line. This would ensure that the content of an intentional dialogue line is used in the most effective manner.

In order to select an appropriate actual dialogue line, both a means of labelling the actual dialogue lines and a method of mapping between game state and these labels is needed. To provide the labelling, I propose a partitioning system based upon the use of an arbitrary number of three-valued axes. Each axis represents a specific characteristic of the dialogue lines and is divided into three values: low, medium, and high. These axes are then combined to form a set of bins that partition the data based upon the characteristics of the axes.

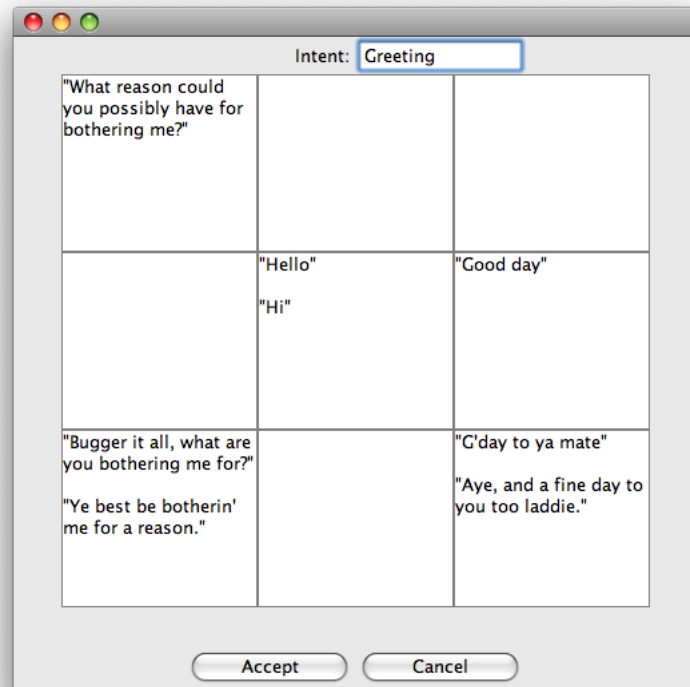
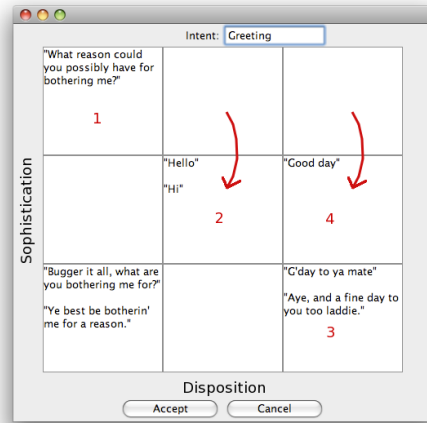


Figure 3.8: The partitioning of an intentional dialogue line by sophistication (vertical) and disposition (horizontal).

An example of a two-characteristic partitioning of an intentional *Greeting* line is shown in figure 3.8. Here, the partitioning is performed using sophistication and disposition of the lines as the two characteristics.

The partitioning system provides a means of differentiating between the actual dialogue lines associated with an actual dialogue line, but it does not provide a means of selecting which of the lines should be selected at game time. For this, I return to the *Decision* patterns provided by ScriptEase. A *Decision* pattern provides a way for designers to easily map elements of game state to a three-valued (i.e. high, medium, low) result. By attaching a decision pattern to each axis, a designer is able to specify which in-game conditions should equate to each bin of dialogue lines. Once the appropriate bin has been selected, an appropriate line can be chosen by randomly selecting from the lines in the bin. If the selected bin is empty, an axis, for which the mapping resulted in a value of high or low, will be selected and the mapping will be



- PC Charisma Low, NPC Intelligence High $\Rightarrow$ **1**
- PC Charisma Medium, NPC Intelligence High $\Rightarrow$ **2**
- PC Charisma High, NPC Intelligence Low $\Rightarrow$ **3**
- PC Charisma High, NPC Intelligence High $\Rightarrow$ **4**

Figure 3.9: Several mappings performed using a PC Charisma decision pattern for disposition and an NPC Intelligence decision pattern for sophistication.

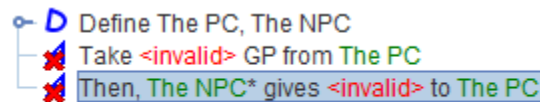


Figure 3.10: The action block of a *Purchase Item* dialogue line pattern with the parameters unset.

changed to a value of medium. This process will be repeated until a non-empty bin is found. To ensure that a bin is eventually found, the central bin that corresponds to all axes having a value of medium must contain a minimum of one dialogue line. Figure 3.9 shows a number of potential mappings and how they would be evaluated. Note that, while the redirected mappings in the example are both adjusted on the vertical axis, the axis that is shifted to medium is arbitrary and if a single shift is insufficient multiple shifts towards medium values can be combined to find a populated bin. Through the use of decisions, designers can quickly generate these mappings without any necessary programming knowledge, and, additionally, once a decision has been created, it can easily be reused in the future without any additional work being required.

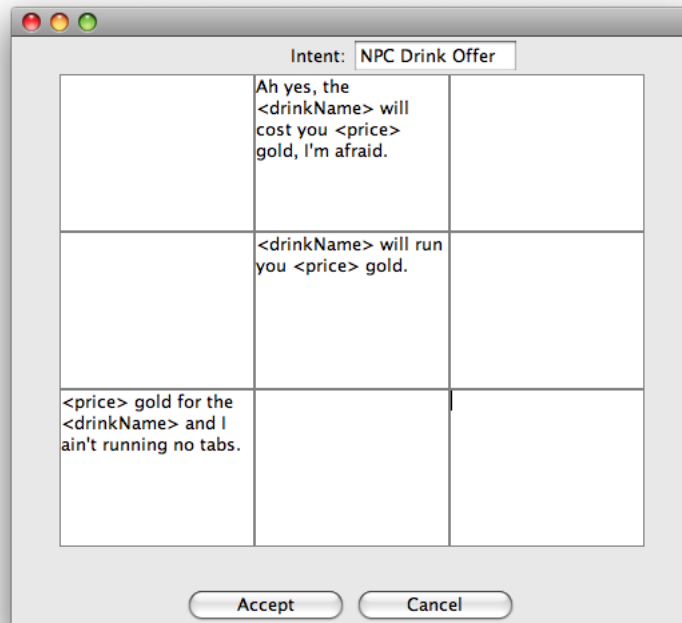


Figure 3.11: A parameterized intentional dialogue line.

3.3.2 Extending the Functionality of Remarks

Having arrived at a complete formulation of an intentional dialogue line, there remains some room for improvement. One shortcoming becomes apparent when considering the bartender intentional dialogue discussed previously. ScriptEase dialogue patterns currently do not provide any way to execute a set of actions when a certain dialogue node is reached. In order to have any actions executed upon reaching a specific dialogue line, a separate encounter pattern must be created. The inability to attach the necessary actions to a dialogue line from within the dialogue tool itself makes the creation of dialogues, like the bartender dialogue, unnecessarily difficult. I, therefore, propose that the existing *remark* structural pattern is replaced with a *dialogue line* pattern. A *dialogue line* can be either an intentional, or non-intentional, dialogue line and contains, in addition to its text, an action block similar to that found within a ScriptEase encounter pattern. Figure 3.10 displays the contents of an action block that would provide the necessary functionality for the *NPC Sell Drink* intentional dialogue line.

The bartender dialogue provides the motivation for one more improvement to intentional dialogue lines. As presented, the bartender dialogue requires a separate intentional dialogue line for each of the drink/price combinations that the bartender offers. This is far from ideal, as other than the drink name and price, the lines should be largely the same. Unfortunately, the ScriptEase dialogue editor does not provide support for parameterized dialogue lines. Prior to the inclusion of intentional dialogue lines, a single *remark* in a ScriptEase dialogue would always correspond to a single dialogue line, so there was no need for such parameterization. However, such parameterization becomes much more useful when dealing with sets of dialogue lines, instead.

In addition to their use within the text of an intentional dialogue line's actual dialogue lines, parameters could also prove useful within the action block of any *dialogue line* pattern. The parameterization of dialogue lines will be performed in a manner similar to the parameterization of other ScriptEase patterns. When designing a dialogue line, an author is given the ability to add option tabs. Each option tab corresponds to a single parameter, and each parameter will be assigned a name and variable type. These tabs, which will appear in the sidebar of the editor, allow users to assign a value to the parameters. Depending on the type of the parameter, the assignment may be performed via manual entry, a picker dialogue box, or a drop down list. When writing actual dialogue lines to be associated with the intentional dialogue line, an author can then make use of the parameter by including the parameter name surrounded by angled braces in the text. Figure 3.11 shows an updated version of the *NPC Drink Offer* intentional dialogue line that makes use of dialogue parameters.

3.4 Intentional Dialogues in Action

With intentional dialogues now formulated, it is important to demonstrate that they can provide real gains for designers in the games industry. While a complete case study of the dialogues in a number of games is beyond the scope of my research, even a cursory glance at a game like NWN shows a number of

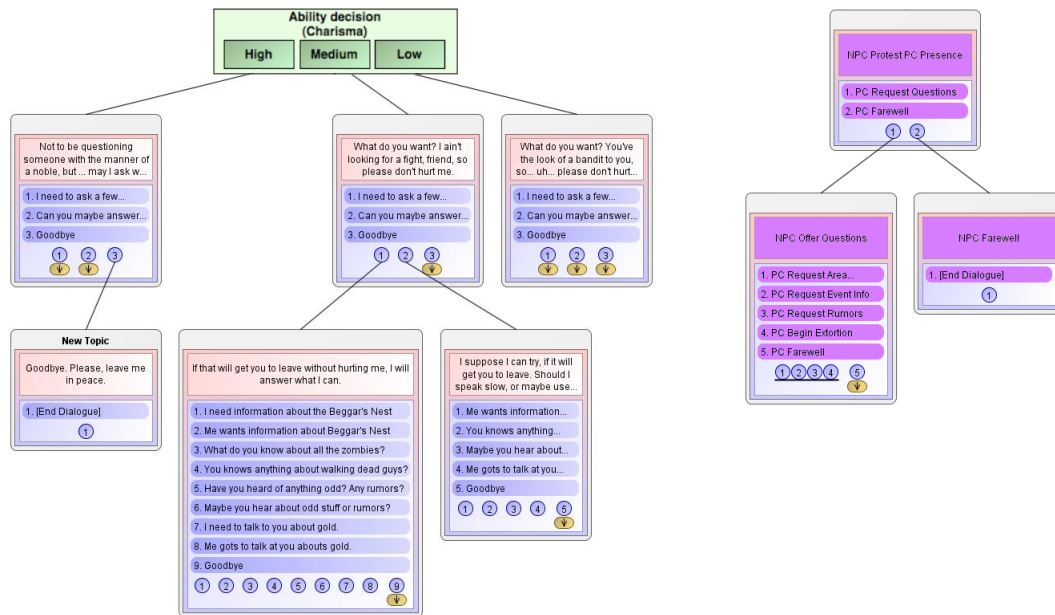


Figure 3.12: An excerpt from an extortion dialogue shown as a regular dialogue (left) and an intention dialogue (right).

situations in which intentional dialogues and intentional dialogue lines could be used effectively. The examples I will provide here are all from the first chapter of NWN and are by no means an exhaustive list of potential use cases within the chapter.

One of the first things that is likely to be noticed when examining the dialogues of NWN is that nearly all of the dialogues provide two copies of every PC line. There is the standard dialogue line that most players will see, and then there is a second copy of the line written for PCs with a low intelligence score. If these dialogue lines led to different sections of the dialogue, intentional dialogue lines might not be of use here, but while the immediate NPC responses do occasionally vary, both the regular and low intelligence branches follow this NPC response with the same set of PC lines. By taking advantage of the functionality of intentional dialogue lines, these pairs of lines could be combined into single dialogue lines. Many of these lines, such as the greeting and farewell lines, can be shared between dialogues, but the biggest gain in this situation does not necessarily come from the potential for reuse. Combining these pairs of lines into single intentional dialogue lines provides nearly a 50%

reduction in the number of lines in a dialogue. Even greater reduction can be obtained from the opening lines of many dialogues where the NPC will speak one of three dialogue lines depending on the player's charisma score. As shown in figure 3.12, the use of intentional dialogue lines in these dialogues provides a drastic simplification in overall structure by reducing the necessary number of nodes, branches, and symbolic links. This simplification in structure should reduce the cognitive load on the designer when working with the dialogues, and help to ensure that the dialogues are error free.

More dialogues that stand out are the doomsayer, extremist and ambush dialogues found throughout the chapter. The dialogues all consist of a number of NPC dialogue lines with no PC responses. Each of these conversations can be replaced with a a dialogue that contains only a single intentional dialogue line. While there is no immediate gain from doing so, that is due to the fact that, in essence, these conversations already are simply intentional dialogue lines without the benefits of organization and line selection that intentional dialogue lines provide.

A final example of the usefulness of intentional dialogue lines, and one that illustrates their use across multiple dialogue files, is the presence of lines like "Where is the prison?" and "What's going on here?" in numerous dialogues throughout the chapter. These dialogue lines could be combined into single *PC Request Location* and *PC Request Area Status* lines. These lines could then be combined with their NPC responses to form intentional dialogues that could then be placed in the dialogues of all of the NPCs that were using the original lines. In addition to the potential ability for reuse when creating new dialogues, these lines also allow designers to easily make updates to the information provided by all of the NPCs using the lines by simply editing the intentional dialogue.

In addition to the two intentional dialogues just identified, there are a number of other situations in which an intentional dialogue would prove useful. The most obvious example is the extortion dialogue, as it was the presence of two such dialogues in the first chapter of NWN that motivated the creation of intentional dialogues. However, other examples are found in the merchant

dialogues throughout the chapter. These merchant dialogues will generally begin with the merchant welcoming the PC to the store, and then the player can request to see the merchant's inventory. While many of the dialogues have additional lines as well, there is this shared base to the dialogues from which a merchant intentional dialogue could be formed. Similar to the merchant dialogues are the bartender dialogues. The bartender will welcome the PC and then the PC can request to buy a drink. Creating intentional dialogues out of these two dialogues has the added benefit of allowing the designer to include the scripting necessary for the operation of the NPCs store or the selling of drinks within the intentional dialogue. By doing so, other designers are able to make use of these base dialogues without needing to know how to create those scripts.

While the examples here do not detail all of the many uses a designer might have for intentional dialogues or intentional dialogue lines when creating a game like NWN, they do show that many such uses do exist. In addition to the potential for effective reuse that motivated the creation of intentional dialogues, intentional dialogues and intentional dialogue lines can also provide a great deal of simplification for large, complex dialogues.

Chapter 4

Supporting Flavours

4.1 Overview

Having arrived at a complete formulation of both an intentional dialogue and an intentional dialogue line in chapter 3, there remains the challenge of how to efficiently populate the intentional dialogue lines. Given both an intentional dialogue line and a set of actual dialogue lines that share the same intent, the actual dialogue lines must still be assigned labels for a number of properties that correspond to the axes of the intentional dialogue line’s partitioning grid. This labelling process can be performed manually by placing the actual dialogue lines into the correct bins, but this process is time-consuming and can lead to inconsistent labelling. As there is a large set of dialogue lines in existing games that could be used to populate intentional dialogue lines, what is needed is a means of sorting dialogue lines automatically.

Since the axes of an intentional dialogue line are not fixed, any technique for sorting actual dialogue lines must be both adaptable to different axis properties (e.g. disposition, sophistication, etc.) and scalable to larger numbers of axes. Developing a single classifier that performs sufficiently well under all of these conditions would be difficult. Therefore, I propose that classification be done individually for each axis, instead of on all axes at once. Once a dialogue line is classified on each of the axes of an intentional dialogue line, the combined labels can be used to identify the bin in which the line should be placed. This allows for greater scalability, as a classifier for any combination of axes with existing classifiers can be created by combining the individual classifiers.

As mentioned in chapter 2, previous work on the topic of sentiment analysis has shown that support vector machines (SVMs) are able to produce high-quality results[1]. Classifying dialogue lines based on properties such as sophistication or disposition seems to be a closely related problem. Therefore, I have chosen to focus my research on the development of an SVM-based machine learning classifier to label the sophistication of a dialogue line.

4.2 Collecting a Training Set

Before it is possible to train a classifier, it is first necessary to collect a set of data to train it on. Unfortunately, game dialogue tends to be stored within large game files that use proprietary file formats. As such, the automatic collection of game dialogue lines is not possible. While it would be possible to manually copy dialogue lines out of a game, doing so would be incredibly time-consuming. Instead, another source of dialogue lines is required, and for this I have turned to film scripts.

Much like games, many films feature a large quantity of dialogue. Unlike games, however, films are inherently linear, so the dialogue can be quite easily written in the form of a movie script. Movie scripts also have the benefit of being very rigidly structured. This rigid structure makes it possible to easily extract the dialogue by writing a simple computer program.

Even after extracting the dialogue lines of a film, there remain some challenges that must be overcome. In order to use this set for the training of a classifier, each dialogue line must be assigned a label based upon its sophistication (i.e. low, medium, or high sophistication). As a dialogue-heavy film could easily contain a few thousand lines of dialogue, manually assigning each of these would be too inefficient. Thankfully, movie scripts contain more information than just the text of the dialogue being spoken. In addition to the actual text, each line of dialogue is prefixed with the name of the character speaking the line. If labels are assigned by character, instead of by individual line, the amount of work necessary to label the dialogue lines is drastically reduced. Such an approach does introduce greater room for error, but by ex-

cluding characters who do not speak in a consistent manner, this error can be reduced.

Since the majority of the games that have been examined as a part of this dissertation are part of the fantasy genre, it seems that any film selected for the training set should also be a part of this genre. I have selected two such films from which to construct my training set: *The Princess Bride*[24] and *The Lord of the Rings: The Fellowship of the Ring*[13]. Combined, these two films provided 1410 lines of dialogue from which to build the training set¹.

While it seems plausible that a classifier trained on movie dialogue would translate well to game dialogue, it remains an untested assumption. This assumption, in combination with the assumption that characters' dialogue lines are consistent enough to allow labelling by character, makes it necessary to have a means of evaluating the classifier on individually-labelled, game dialogue lines. To accomplish this, I randomly selected a set of 225 dialogue lines from the first chapter of NWN. An attempt was made to ensure that the final set of lines contained at least 50 lines of each level of sophistication. These lines were then sent out to a group of seven graduate and undergraduate computing science students with game-playing experience. This group was to act as a jury to determine the sophistication of the dialogue lines by each individually assigning labels to the lines based upon their perception of what a high, medium, or low sophistication dialogue line was. The labeling by each member of the jury was performed independently and no specific instruction was provided as to what a low, medium, or high sophistication line should be in order to avoid biasing the results. Jury members were instructed that a minimum of 50 dialogue lines should be assigned to each label. The resulting labels were then combined, and any line for which three or more jurors disagreed with the consensus label was thrown out. This resulted in a final set of 181 manually-labelled, game dialogue lines with 55 labelled high, 76 labelled medium, and 50 labelled low sophistication². This set would then be the gold standard to evaluate the classifier with.

¹Data-set available upon request.

²Data-set included in Appendix A

$$(tf-idf)_{i,j} = tf_{i,j} \times idf_i$$

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

$n_{i,j}$: the number of t_i in line d_j

$|D|$: the total number of lines in the corpus

$\{d : t_i \in d\}$: the total number of lines containing t_i

Figure 4.1: The formula used to calculate *term frequency-inverse document frequency*.

4.3 Selecting the Features

When creating an SVM-based classifier, the most important step is the design of the feature vectors. The feature vectors are the only connection to the actual data that an SVM classifier has. Due to this, a poor selection of features can produce poor classification results regardless of the quality of the training data or the nature of the problem.

For the sophistication classifier I chose to use a bag-of-words-style feature vector [17]. In order to create a bag-of-words feature vector, the complete set of unique words found within the training corpus had to be identified. These words were then sorted alphabetically and used as the features of the feature vector. A feature in a feature vector was then assigned a value based upon whether or not that word appeared in the dialogue line that the feature vector represented. This approach discarded the information about grammar and word order found in the original dialogue lines, but in doing so was able to reduce the size of the feature space.

There are many techniques for deciding upon the exact value that a feature in a feature vector should be assigned. For the sophistication classifier I considered two approaches. The first approach was to use a binary-valued feature vector. In such a feature vector, a feature has a value of 1 if the feature word

”Why, hello there friend. What can I do for you today?”

$$avg\text{-}words = \frac{4 \text{ words} + 7 \text{ words}}{2 \text{ sentences}} = \frac{11 \text{ words}}{2 \text{ sentences}} = 5.5$$

$$|words : value| \Rightarrow |1 : 0|2 : 0|3 : 0|4 : 0|5 : 0|6 : 1|7 : 0|8 : 0|9 : 0|10 : 0|$$

Figure 4.2: An illustration of the average-words-per-sentence feature.

appears in the dialogue line and a 0 if it does not. The second approach was to use *term frequency-inverse document frequency* (TF-IDF)[25]. TF-IDF is a measure (formula shown in figure 4.1) of how frequently a word appears in a dialogue line in comparison to how frequently the word appears across the entire corpus. If a word occurred very infrequently throughout the corpus, its presence in a dialogue line was deemed to be relatively more important than if it had been present in the majority of dialogue lines. As words that appear in the majority of dialogue lines cannot be characteristic of a certain sophistication of line, due to their presence in so many lines of other sophistication levels, this seemed to be a valuable property.

In addition to the words in a dialogue line, both the length of the words and the length of the sentences may also provide some information as to a line’s sophistication. It seems plausible that a less sophisticated person would use shorter, simpler words than would a highly sophisticated person. The same seems likely of the length of the sentences used by characters of differing sophistication. In order to capture this information I also be evaluated an *average-syllables-per-word* feature and an *average-words-per-sentence* feature. In order to increase the effectiveness of these features, I used sets of binary-valued features in place of single floating-point-valued features. For each dialogue line, the desired average was calculated and then the result was rounded to the nearest integer value (any value greater than 10 was set to 10). Each feature in the set corresponded to an integer from 1 to 10, and the value of each feature was set to 1 if the calculated average was equal to that integer and 0 otherwise. Figure 4.2 illustrates how the sets of features were

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$F\text{-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figure 4.3: The formula's for Precision, Recall, and F-Measure.

constructed.

4.4 Evaluation

Two separate methods were used to evaluate the classifier. First, a classifier was trained on the automatically-labelled film data. This classifier was then evaluated by classifying the manually-labelled (juried) game dialogue lines and comparing the predicted labels to the juried labels. The second method was performed entirely on the juried data to provide a games-only result to compare the film-data-trained classifier against. The game data was divided into five sets, and then five separate classifiers were built by excluding a single set of lines for each classifier. The excluded sets were then used to evaluate their corresponding classifier, and the results were aggregated to obtain a final result. This approach is known as 5-fold cross validation. In order to reduce the variance in the results, 10 trials of the 5-fold cross validation were performed with new folds generated for each trial.

The best results are those in which the precision and recall of the predicted labels is high. The F-measure, the harmonic mean of the precision and recall, captures both of these in a single value. The F-measure is calculated individually for each label (i.e. low, medium, high). Due to the lack of a *no prediction* option for the classifier, if the precision and recall values for each of the labels are combined into aggregate precision and recall values, these aggregate values become equal. As can be seen in figure 4.3, if the precision and recall are

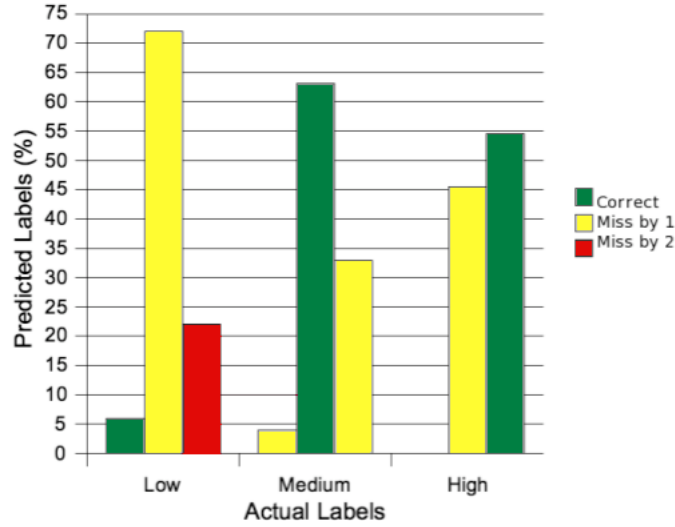


Figure 4.4: The results of the binary-valued classifier trained on film dialogue lines. (from Kerr [15])

equal, the F-measure must also be the same value. These values all become equal to the accuracy of the predicted labels. Therefore, the results presented here will consist of the error in the predictions for each label by a classifier and the overall accuracy of the predictions by a classifier.

4.4.1 Results with Automatically Labelled Film Data

The best results obtained when training on the film data set were with the TF-IDF valued bag-of-words classifier with the average-words-per-sentence feature included. These features provided an overall accuracy of 45%. Use of the binary valued classifier in place of the TF-IDF valued classifier caused a dropped of 11% in the accuracy. These results are displayed in figure 4.4. The bars represent the percentage of the lines with the given actual label to be predicted as low, medium, or high sophistication (as ordered left to right in each of the groups). Here, it can be seen that while the classifier performs relatively well on medium- and high-sophistication dialogue lines, the performance of the classifier on low-sophistication lines is quite poor. An inspection of some low-sophistication dialogue lines, from both games and movies, suggests that low sophistication movie dialogue is of higher sophistication than the low sophistication dialogue in games. Many of the low-sophistication game dialogue

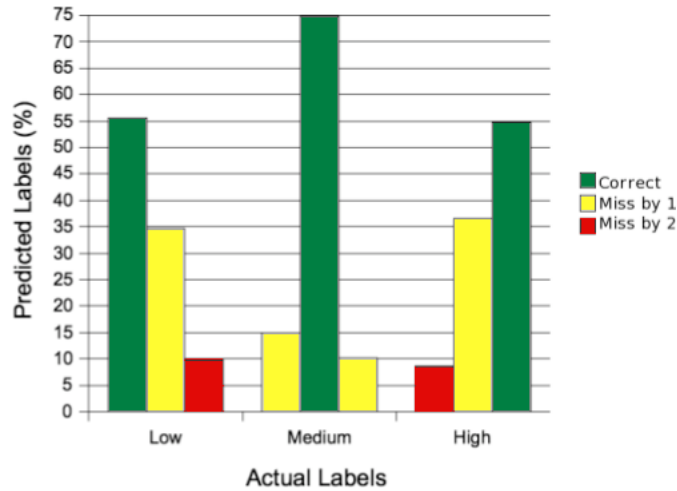


Figure 4.5: The results of the TF-IDF-valued classifier trained on game dialogue lines. (from Kerr [15])

lines are rather comically low, whereas the film lines are not so extreme.

When evaluated on their own, the film dialogue lines do prove to be more self-consistent than the previous results would indicate. Five-fold cross validation of the film data produced a mean accuracy of 52% with a standard deviation of 0.1% over the 10 trials. However, these values remain too low and further study is required for this method to be of practical use.

4.4.2 Results with Juried Game Data

For the classifier trained entirely on game data, the best results were obtained through the use of the binary valued bag-of-words classifier with the average-words-per-sentence feature included. With this classifier, I was able to obtain a mean accuracy of 63.37% with a standard deviation of 0.75%. The use of TF-IDF valued features in place of the binary values caused overall results to drop by approximately 14%. Figure 4.5 displays the results in the same format used for the film data classifier. The performance of this classifier, especially on low sophistication dialogue lines, is much improved. Furthermore, the majority of errors that occur involve misses by only a single category.

4.4.3 Additional Results

As noted previously, in addition to the average-words-per-sentence feature, an average-syllables-per-word feature was also tested. Unfortunately, the feature did not provide an increase in the quality of the predictions. This is likely due to the fact that information on word length is included implicitly through the words themselves. Increasing the number of syllable features to allow for greater precision by representing the average syllables as a decimal instead of integer value did provide a minor improvement in some cases, but it was not large enough to warrant the increased number of features necessary.

One of the concerns when deciding to adopt an automated method for the labelling of the film dialogue lines was that characters may not always speak at the same level of sophistication. In order to address this, attempts were made at culling the movie data to remove inaccurately labelled lines. The first approach involved performing 5-fold cross validation on the movie data and removing any dialogue lines that were predicted incorrectly. The remaining lines were then used to train a new, more self-consistent classifier. The second approach was similar except that instead of training a single classifier for each excluded fold, four classifiers were trained (by excluding one of the remaining folds for each classifier). The labels of the lines in the excluded fold were then predicted by the four classifiers and a voting scheme was used to decide whether to remove a line. Neither method provided a large enough increase in the quality of results to justify the possibility of over-training the classifier to the film data.

Another technique tested was the standard practice of excluding a *stop word* list: a list of commonly used words that are too general to provide much meaning. Several stop word lists of varying sizes were used, and while some did provide minimal gains to the accuracy of the movie-dialogue-trained classifier, all of the lists caused large decreases to the accuracy of the game-dialogue-trained classifier. Again, the minimal gains provided were not significant or consistent enough to justify the inclusion of this technique.

Finally, the use of word pairs as features (as well as more general n-grams)

was also investigated. This approach did show some promise when trained on the manually-labelled game data, but the size of the feature space was drastically increased by the inclusion of the features. Due to this increase, it was not possible to use the word-pair features with the movie-dialogue-trained classifier. It is likely that the witnessed increase in prediction quality was due to the implicit inclusion of additional grammatical information that was not included when using only the individual words. Unfortunately, due to the inefficiency of the feature, it was not included in the final classifiers.

Chapter 5

Future Work and Conclusion

5.1 Future Work

This dissertation described the design of an intentional dialogue system. However, while much of the ground work exists within the ScriptEase tool, the intentional dialogue system itself has not currently been implemented. In order to fully evaluate the efficacy of the intentional dialogue system, a fully functional tool must be developed. With an intentional dialogue tool, the development of a pattern catalogue of intentional dialogues and intentional dialogue lines could begin. As well, a number of case studies and user studies could be performed to determine the specific benefits provided to designers by intentional dialogues. Case studies in which the dialogue of a number of modern, story-based games is converted to intentional dialogues could be performed to determine how much the system is able to reduce the amount of dialogue lines that must be written by designers. Furthermore, a number of user studies would be useful to measure the exact impact that repetitive dialogue has on a player's game playing experience and to what degree the use of intentional dialogues improves upon the experience.

There is an upfront cost when creating an intentional dialogue or an intentional dialogue line. This cost, however, is offset by the efficiency gains that are provided when making use of them. Nearly all games are localized into a variety of different languages, and many modern games make use of voice overs in their dialogues. Both localization and the inclusion of voice over further increase the upfront cost of creating an intentional dialogue line. In order

to address this, further research into methods of reducing the upfront creation cost, as well as an analysis of the amount of usage required for an intentional dialogue or intentional dialogue line to offset its creation cost, is required.

In addition to the intentional dialogue system, this dissertation also detailed the development of a machine learning classifier that was designed to help designers more rapidly populate intentional dialogue lines. While promising results were achieved, the quality of the classifier’s predictions was not high enough for it to be of practical use. Further research on the selection of features for the classifier’s feature vectors is required. Potential avenues of research include the use of grammar-based features, word origin identification, and prefix/suffix identification.

Finally, the axis-based partitioning system used by intentional dialogue lines to group actual dialogue lines can become difficult to use with large numbers of axes. As the number of axes increases, the number of empty bins tends to increase dramatically. While rules were described to handle the substitution of neighbouring bins for empty bins, the rules are rather arbitrary and do not guarantee that the best substitution is made. Additionally, while many characteristics fit well into the three-valued-axis paradigm, others, such as a gender characteristic, do not. In order to address these problems, research into an alternative partitioning system is required. One potential system would be a feature-vector-based partitioning that assigns a feature vector to each actual dialogue line. Properties such as disposition and sophistication would then become features in the feature vector. Such a system would allow the use of non-three-valued features and would potentially enable future research on the use of machine learning classifiers to select the best match when an exact match is not found.

5.2 Conclusion

I began this dissertation with a discussion of the current state of dialogue in story-based games, and I detailed two major flaws present in the dialogues of these games. These flaws, which I identified as *single-character-repetition* and

multi-character-repetition, were due to the inability to efficiently produce large numbers of unique, interesting NPC dialogues. To explain why designers are unable to create unique dialogues for the secondary characters in a game, I then proceeded to examine a number of dialogue creation tools used throughout both the industry and academia.

After developing a clear picture of the current state of dialogue and dialogue tools, I then introduced the concepts of an *intentional dialogue* and an *intentional dialogue line*. Intentional dialogues allow designers to quickly generate large numbers of dialogues through the effective reuse of dialogue content. This allows designers to address both of the problems identified previously, and does so in a manner that leaves the designer in full control of the final product.

In order to reduce the effort required by a designer to populate intentional dialogue lines with actual dialogue lines, I then detailed the development of a prototype SVM-based machine learning classifier to label the sophistication of a dialogue line. While the results obtained leave room for improvement, the best classifier was still able to predict the correct label for nearly two-thirds of the dialogue lines it evaluated. Additional research is needed to improve these results, but in its current state, the classifier could be used to provide a rough sorting of the actual dialogue lines that a designer could then adjust.

Together, the intentional dialogue system and prototype sophistication classifier provide the design of a semi-automated dialogue creation system that could be used to drastically reduce the effort required to create large quantities of similar dialogue lines. By making use of this system, designers can ensure that even secondary NPCs have interesting dialogues. In doing so, designers can increase the realism of their game worlds and improve the players overall sense of immersion.

Bibliography

- [1] M. Annet and G. Kondrak. A Comparison of Sentiment Analysis Techniques: Polarizing Movie Blogs. In *Proceedings of the Tenty-First Canadian Conference on Artificial Intelligence*, pages 25–35, 2008.
- [2] G. Bowditch. Grand Theft Auto producer is Godfather of gaming. *The Sunday Times*, April 27, 2008.
- [3] C. Chang and C. Lin. LIBSVM: A Library for Support Vector Machines, 2001. Software available at <http://www.csie.ntu.tw/~cjlin/libsvm>.
- [4] BioWare Corp. Baldur’s Gate, 1998.
- [5] BioWare Corp. Neverwinter Nights, 2001.
- [6] BioWare Corp. Mass Effect, 2008.
- [7] CryTek. Crysis, 2007.
- [8] G. Ebersol. Nice skybox. <http://niceskybox.com/2008/01/25/mass-effect/> (accessed 2009).
- [9] Rockstar Games. Grand Theft Auto IV, 2008.
- [10] C. Hecker. Structure vs Style. Invited talk at Game Developers Conference, 2008. http://chrishecker.com/Structure_vs_Style (accessed 2009).
- [11] C. Hocking. I-fi: Immersive Fidelity in Game Design. invited talk at Game Developers Conference, 2008.
- [12] Epic Games Inc. Unreal engine 3, 2005. <http://www.unrealtechnology.com>.
- [13] Dir. P. Jackson. The Lord of the Rings: The Fellowship of the Ring, 2002. [DVD] New Line Home Entertainment, Inc.
- [14] G. Kacmarcik. Using Natural Language to Manage NPC Dialog. In *Artificial Intelligence and Interactive Digital Entertainment*, 2006.
- [15] C. Kerr and D. Szafron. Supporting dialogue generation for story-based games. In *Artificial Intelligence and Interactive Digital Entertainment*, 2009. Forthcoming.
- [16] M. Lebowitz. Story-Telling as Planning and Learning. *Poetics*, 14:483–502, 1985.

- [17] D. Lewis. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 4–15. Springer Verlag, 1998.
- [18] D. Lin and P. Pantel. Discover of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343–360, 2001.
- [19] M. MacNaughton, M. Cutumisu, D. Szafron, J. Schaeffer, J. Redford, and D. Parker. ScriptEase: Generative Design Patterns for Computer Role-Playing Games. In *19th IEEE International Conference on Automated Software Engineering (ASE)*, pages 88–99, Linz, Austria, September 2004.
- [20] M. Mateas and A. Stern. Façade: An Experiment in Building a Fully-Realized Interactive Drama. In *Game Developers Conference*, March 2003.
- [21] M. Mateas and A. Stern. Beyond State Machines: Managing Complex, Intermixing Behavior Heirarchies. In *Proceedings of the Game Developers Conference*, March 2004.
- [22] M. Mateas and A. Stern. Pocedural Authorship: A Case Study of the Interactive Drama Façade. In *Digital Arts and Culture*, Copenhagen, November 2005.
- [23] S. Rabin, B. Pfeifer, C. Hecker, S. Reynolds, and D. Isla. The Photoshop of AI: Debating the Structure vs Style Decomposition of Game AI. at 2009 Game Developers Conference - AI Summit. <http://www.cmpevents.com/GD09/a.asp?option=C&V=11&SessID=8279> (accessed 2009).
- [24] Dir. R. Reiner. The Princess Bride, 2001. [DVD] MGM Home Entertainment.
- [25] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–979, 1991.
- [26] J. Siegel and D. Szafron. Dialogue Patterns - A Visual Language For Dynamic Dialogue. *Journal of Visual Languages and Computing*, 2009. Forthcoming.
- [27] J. Skorupski, J. Lakshmi, S. Marquez, and M. Mateas. Wide Ruled: A Friendly Interface to Author-Goal Based Story Generation. In *Proceedings of the 4th International Conference on Virtual Storytelling (ICVS 2007)*, Saint Malo, France, December 2007.
- [28] Bethesda Softworks. The Elder Scrolls IV: Oblivion, 2006.
- [29] Bethesda Softworks. Fallout 3, 2008.
- [30] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2 edition, 2005.

Appendix A

Game Dialogue Lines

Dialogue Line	Low	Medium	High
Sorry, you must have me confused with someone else.	0	7	0
What do you want? I ain't got nothing to say to no strangers. Probably muggers and crap.	7	0	0
Who might you be, then? Speak up, speak up, I can't spend much time on pleasantries right now. Too much to do.	1	2	4
Me want ask you some questions.	7	0	0
I coulda gone to that sanctuary instead of here, but I prefer sensible talk over their pipe dreams.	2	4	1
Me want in.	7	0	0
Your precious lords and ladies mean nothing in these parts... Scram, you hear me?	1	3	3
I assume from your tone that you will want the opportunity for profit? I understand. Not everyone joins out of the goodness of their heart.	0	1	6
Well fancy that... I reckon you're with good man Desther, then, aren't you?	1	3	3
Where can I go to get some action?	2	5	0
Let's see what you have for sale.	0	6	1
Well fought! You have proven quite the force, defeating Callik like that. The lad was giving me trouble. No sense of style.	0	2	5
Good to see you. You must be one of the fine new recruits with the guards. No one else comes near the Beggar's Nest.	0	6	1
Just leave me be! I ain't going near that woman again! it just ain't worth it!	7	0	0
You better step back. Only those in the company of Mr. Hodge are allowed to enter his estate.	0	4	3
What are ya lookin' for?	7	0	0

Dialogue Line	Low	Medium	High
Relax, I will help, but I have some questions.	1	6	0
I am here to serve, though I can't offer much. The greatest tragedy is the interruption of my beer supply.	0	3	4
Why, it's a plantin' festival for folks like me. For the Helmites, though, they come to the keep and figger out how to protect us over the comin' year.	6	1	0
Nonsense. I merely issued a statement or two. How people choose to react to that is not my concern.	0	0	7
Why, this here's the road up to Helm's Hold, where all the priests live. There's no better lord than a Helmite, I say, 'cause they'll always protect you in hard times.	5	1	1
You are free. Wait until I make sure the way is safe.	2	5	0
Me want some information..	7	0	0
Dat's it! No more dumb jokes! Me flatten your head!	7	0	0
Even in times of plague you scheme. How petty.	0	2	5
I stumbled upon the remains of a fine little bird that Nasher and his lackeys carelessly lost. Imagine my surprise at seeing some dog in the street carrying it. I merely wished to put it to good use.	0	0	7
Please, you have to get me out of here.	0	7	0
Speak, but waste my time and you'll regret it.	0	2	5
Look at you. You are in no shape to fight. You need rest.	1	6	0
You better step back. Mr. Rumbottom does not permit strangers to enter his estate.	0	7	0
Of course, of course. I expected as much, though I was also hoping that Nasher would make an appearance.	0	4	3
I do what needs to be done. What is your offer?	1	6	0
I suppose from your perspective you are correct. Since you are the dominant force in the room, I bow to your wisdom.	0	0	7
What do you want? Come on now, I've got time if you're quick.	1	4	2
What in the Hells do you want? What reason would I have to speak with you?	0	5	2

Dialogue Line	Low	Medium	High
Hey, more people to talk to! Good, because me not want to go back to Aldo without wagon parts. Can't find none nowhere that will work. You see wagon parts?	7	0	0
Who be you?	7	0	0
Good morrow to you, citizen. Wait... are you a new recruit? Aye, they're replacing the ones we've lost, are they?	1	3	3
Whatever was getting the locals all worked up seems to be over, I guess. No trouble here at all besides the plague.	0	5	2
Anyone get let out in last little while?	4	3	0
I am sorry, m'lady, but my company is open to males alone. You might wish to see if Tanith is free.	0	2	5
Aye, an' keep yer pinching fingers clear of me crops, too.	6	1	0
I never thought I'd see it happen. I'm gonna miss this place.	0	7	0
I'm sure your quaint morality lets you sleep deep and long. Come back when you tire of being an errand boy for Tyr and Oleff.	0	1	6
Greetings, and welcome to this small house of Helm. Do you seek a blessing in this troubled time?	0	3	4
Now that he's back, mebbe things'll start gettin' back to normal 'round here.	6	1	0
I done told you already that I ain't taking any goods. I'm packing up and leaving this place. Now, get gone!	7	0	0
Hello there! I am pleased to see people of refinement can survive out there. My name is Bertrand.	0	1	6
The Wailing heralds the end of days!	0	3	4
You shouldn't be staying here. I could lead you out.	0	7	0
Me leave then. Bye.	7	0	0
Wait... haven't I already pleaded my case to you? Certainly you see the need to act in this instance?	0	0	7
You are not welcome in my sight! Go, and leave me to my work!	0	3	4
Farewell for now.	0	3	4
What are you doing here? State your intent to me.	0	2	5

Dialogue Line	Low	Medium	High
Time for me to get on to work. I'm heading off to Waterdeep, and I ain't got much time to show for the work to do.	5	2	0
I had almost given up hope. I can't believe it.	0	7	0
Greetings to you. Here, stand a bit closer. I look positively dashing next to you.	0	0	7
Welcome, ¡FullName!. I regret I did not get a chance to introduce myself properly to you while you were training at the Academy.	0	1	6
Get lost! Only guests of Mr. Rumbottom are allowed in. Please, I'm just doing my job.	1	6	0
I don't bargain with the devil's own. To arms!	0	4	3
Speak what ye will, friend. I will answer.	5	1	1
Need a drink? Times is tough, and a body could use a little lubrication to keep it movin'.	4	2	1
Sure. Give me blessing. Thanks much.	5	2	0
Why is this area so dangerous?	0	7	0
Greetings. Got a particular need this fine day?	0	4	3
You certainly bring a different flavor to the district, my lady. A treat for the eyes in this dark time.	0	2	5
Me help wherever me can.	7	0	0
I just want to go home, that's all.	1	6	0
You there, shall I bless you with this living water, once cupped within the Guardian's palm? It keeps the plague at bay, at least for a while.	0	2	5
You have something not yours!	6	1	0
Me got questions.	7	0	0
What you meaning? You please e'splain.	7	0	0
It is a small solace, I suppose, but I am more accepting of his passage now. My brother was a good person, you know?	0	0	7
Weapons should not be brandished on the main floor. You'll bring unwanted attention.	0	2	5
Move along, then, and keep that tail between your legs where it belongs.	1	5	1
Eh? Oh, you're that hero they be talkin' about. Whaddaya want?	7	0	0
Bah! I got no time for newcomers. You want respect, you step up to the Pit.	3	4	0
You're a holy man! Can you explain why the gods allow the privileged of Blacklake to be enslaved by fear of the plague? I thought not.	0	1	6

Dialogue Line	Low	Medium	High
Well met, friend. Fine night in the inn, eh. Right busy, it is.	4	2	1
By the Black Sun, how could she fell me?! The snake gods are pitiful! Beneath the notice of mighty Cyric!	0	2	5
Relax, I'm not here to hurt you.	0	7	0
Let me see your wares.	0	7	0
He's a dangerous-looking one, ain't he. Not mean, though - Just protective of that little girl of his. Anyone else?	2	5	0
Life must be an endless marvel for folks like you...	0	2	5
This is hardly the place to risk such behavior. Dress yourself - if not for decency, then at least for battle.	0	2	5
Still, I gather from your well-fortified self that I am in further trouble, am I not?	0	0	7
Greetings. Are you in need of a blessing? I would not travel the docks without it, lest you tempt the fates.	0	0	7
Your mind is feeble, my Lord. Never fear, I am used to dealing with patients whose minds have become enfeebled. I'll speak slow so you understand.	0	0	7
Kindly cover yourself. You'll bring unwanted attention to the tavern.	0	3	4
What do you want? And be quick about it - can't you see how busy I am here?	0	7	0
Woe to the unrepentant, for they shall be struck down!	0	0	7
Master Maugrim don't like meddlers... But Vardoc take care of dat.	7	0	0
Why can't I enter the estate?	0	7	0
Don't go drawing attention to me. Can't keep a coin in your purse for all the thieves out today.	1	5	1
Goodday, I wish you well.	0	5	2
Hemmel is such a boy. He'd have us stand here for ages thinking his precious heirloom will return. Hmph!	0	5	2
I wonder, by chance, if you've seen a man named Callik? He has an amulet of mine and he, ah, seems to have forgotten us.	0	4	3
I guess I should have tried to get to the City Core. There's certainly no help coming to us here.	0	6	1

Dialogue Line	Low	Medium	High
Degenerate! Show some modesty and dress yourself! This isn't Calimshan!	0	2	5
Farewell, then.	0	5	2
Rumor is that Luskan's sealed its gates to anyone comin' in. Some figger they came down wit' the Wailing but I thinks different.	6	1	0
Bad enough, though the man I thought I had lost made it back. Thanks to you if you had anything to do with it. He'll help us pick off the undead from here.	0	7	0
That's a blanketing statement. Why, may I ask, do you deem it necessary to kill me?	0	0	7
'At I do. I first met good man Desther the night of Shieldmeet. He must be an important fella, 'cause the whole place shut down when he left for Neverwinter.	7	0	0
Foul minion of darkness, you'll stay imprisoned here.	0	1	6
Yes? Is there something you require? These dark days have left my mind on other things so make it brief.	0	1	6
From what I've heard, the Docks district is almost settled down, even with the plague still a concern for most.	0	6	1
Yeah? What do you want? My folks ain't want me talking to no adventurers. You're an adventurer, right?	7	0	0
Yes? Is there something you need?	0	7	0
Who you be? Why you help dat man?	7	0	0
Thank you for saving me... if that is indeed what you have done. I do not know you, or your motives.	0	2	5
I just have a few questions.	0	7	0
Greetings. Harben told me to expect additional troops arriving. Let me tell you, you are much needed here.	0	5	2
I think I'll be going now.	1	6	0
Greetings in these dark days. Is there aught than I can offer one as esteemed as yourself?	0	0	7
Get some clothes on, fool. This ain't that kind of tavern!	5	2	0
I've no desire to speak with someone as rough-hewn as you, least of all in times such as these. Leave me be.	0	0	7

Dialogue Line	Low	Medium	High
Step right up! I've got just what you're looking for. My shop ain't what it used to be, but I'm still a shopkeeper, nonetheless.	1	6	0
Came here from the soup kitchen before it got overrun. Damned guards couldn't help. They didn't even know where it was.	0	7	0
At them! Yes, that's the spirit!	0	5	2
What? Has the bloody plague stricken ye dumb?	4	2	1
You will leave me be! I have no desire to speak with the likes of you! You disgust me!	0	1	6
More business! I am certainly busy this day, even with the streets so dangerous.	0	7	0
You're not lyin' to me again, are ya?	7	0	0
I will listen. What is it you wish to say?	0	6	1
Oh so polite. Bah! I have no use for manners. Can you steal and can you kill? That's what I need.	2	4	1
I am sure you will find what you require. I don't approve, but there are ways to capitalize on desperate times.	0	1	6
I am so grateful. I... I'm just stunned that someone would bother to help us... thank you.	0	5	2
I hold nothing against those that make their living as thieves, but I will not be a target. I need to make a living too.	0	5	2
I'm sorry for my stressed tone, but I'm growing tired of waiting for a wagon wheel in a zombie-infested street. You hear me, Aldo?!	0	5	2
Well then, what would a person of your manners be doing here? I've only a minute, so your introduction must be quick. Too much to do.	0	4	3
Can you tell me anything about my duties?	0	7	0
Then you don't get in. Goodbye.	3	4	0
The plague snaps at the heart of Neverwinter, but we have a history. We shall fight with the newly detailed fury of Halueth Never.	0	0	7
The glaring eyes of my wife wish to remind me that she did say the man was not to be trusted, but perhaps he is only delayed.	0	0	7
Your actions caused no end of chaos!	0	0	7
What do you have?	2	5	0
The end is nigh! Like the arrogant of Netheril, we shall fall to dust!	0	1	6
What would you care to know?	0	7	0

Dialogue Line	Low	Medium	High
I hope you're here as a friend. You look like decent folk, and I'd hate for you to have sunk to looting.	0	4	3
I take it you are the new owner. Luck be with you in these hard times.	0	7	0
Keep back, fool, or risk catching the Wailing Death yourself.	0	7	0
You and yer gnomes. I don't wanna hear another word of it, all right? Now get about yer business.	7	0	0
Me knew it could be done, but me not smart enough to do it. BAH! Me go back to real battles!	7	0	0
Thank you, that will be all.	0	7	0
Sure is good to see a friendly face. What brings you here in the midst of all this chaos? You wouldn't be a new recruit, would you?	0	5	2
Aw - what do you want? I'm telling you, this ain't the best of days so far. Captain Mung won't stop arguing with me.	5	2	0
Hey yous! Why yous gotta bother me? Sure, yous look like a noble-type, but only friends of Mr. Androd is supposed to come here.	7	0	0
What is it? I'm getting tired of explaining myself. My men are not risking themselves down there and that's final.	0	6	1
Heh, nothing. Just looking at the architecture.	1	6	0
What have you heard about the auction, friend?	0	7	0
You've got some nerve! I've never backed down from a challenge!	0	7	0
We ain't hiring new hands. Be off with ye.	7	0	0
Who are you to come here?! Do your worst! You will see how futile it is!	0	5	2
They said you was plain-looking all right, but you're the hero of Neverwinter, that much is sure.	6	1	0
I'm on the other side of a stout door, nitwit.	1	2	4
Keep movin', you're blocking my view.	4	3	0
Hmm? Leave me be, I got nothing to say. Can't be seen with the likes of you.	1	4	2
Perhaps I could arrange a meeting with the owner?	0	6	1
Even your uncouth kind are a welcome sight to me. Only zombies have populated the vigil for my brother.	0	0	7
We be simple folk here. Take your ego elsewhere.	6	1	0
I ain't saying anything to you. If I'm seen talking to potential looters or thugs, I'll be out on my ear.	4	3	0

Dialogue Line	Low	Medium	High
Jiminy, you can't even trust your neighbors these days. That's it, I'm not taking a chance on nobody no more. Do what you want. I'm leaving.	5	2	0
Greetings. Nice to see someone still alive out here. I'd prefer that we were talking somewhere safe, but it's still nice.	0	5	2
Greetings, my Lord. How may I be of service?	0	4	3
Of course I will. Oh, don't look so surprised. What am I going to do with some mangy bird? Besides, I know full well how important the feathers of this beast are.	0	4	3
You aren't royalty, are you? Pretty dangerous for you in the docks if you are. Pretty tempting target.	1	6	0
Go on with you, I got no reason to talk and you've got no reason to ask.	1	6	0
Greetings to you. May I ask what you expect to find here? I don't want to be rude, but I would like to be alone with my sorrow.	0	1	6
You can't run around town without anything on! Are you mad?	0	7	0
Aye, I remember wakin' up to hear shouts and the sound of fightin' comin' from the keep. Just thought the priests were in their cups after all the festivities.	4	2	1
I have been hired to make sure that nobody makes trouble here. If you won't tell me why you are here, then I won't let you pass.	0	5	2
Can I ask you some questions?	0	7	0
More meat for Pit? Good! Me tired of what here!	7	0	0
Don't bother me, all right? I did my part with the door and I don't care who goes through now.	0	7	0
Well I reckon a few questions won't harm an old fella like myself... Helm protect me if I be wrong.	4	3	0
My purpose here is not your business.	0	5	2
Hello to you. I hope the day finds you well. Looks like things are finally settling down in the Beggar's Nest. The undead are gone.	1	4	2
You madam, are an absolute disgrace to this fine city. Get out of my sight.	0	0	7
Could I trouble you with some questions?	0	3	4
I am thoroughly disgusted by the nature of people, and refuse to assist you. You may blame the actions of a small number for my mood.	0	0	7

Dialogue Line	Low	Medium	High
Goodbye. Perhaps we'll both live to talk another time...	0	6	1
Go on, why are you bothering me? I've no interest in you.	0	6	1
The remains of the creature are in a crate. Send it to Nasher with my regards, though I doubt he'll have time for a scoundrel like you.	0	4	3
He ain't here, is he... That's strange. Ever since the incident, he's been scared to set foot outside the place. Anyone else?	2	5	0
Blessings upon you. Is there aught I can do for you?	0	1	6
A proper member of society does not wander about with their weapon drawn.	0	1	6
I really don't have the heart for this today. I don't need the criticism.	0	6	1
By Cyric, I would speak with you!	0	6	1
Straighten up, I am here at the behest of Aribeth and Lord Nasher.	0	1	6
I had a cousin slow like ye. He were a mercenary, too... but he drowned in a puddle.	7	0	0
Must have been some fight, by the look of ye.	6	1	0
Oh, by the gods! Must all the dregs of society come in here?	1	3	3
I think the plague is a result of the poor classes we allowed into the district. The barricades were long overdue.	0	4	3
Welcome to my humble store. Before we continue, I draw your attention to my guards, and my many warning devices.	0	3	4
Damned if I know what settled them down, but I ain't seen any trouble for a while now.	4	3	0
Any word on the cause of the unrest?	0	6	1
Beggin' your pardon, but why are you in my home? Not that nobility or the like can't go wherever they want. Uh...	3	2	2
Just a man like any other, maybe driven mad by the plague.	0	7	0
Stay your wrath... this battle is lost to me... I yield.	0	1	6
By Lord Nasher and Lady Aribeth, grant me entrance.	0	1	6
Ayup. I reckon it's just the weather we've been having but better safe than sorry.	5	2	0

Dialogue Line	Low	Medium	High
Have you heard of anything Odd? Any rumors?	0	7	0
Me go now.	7	0	0
I urge you to keep away from these bodies, my lord. There is a risk of contagion, I understand.	0	1	6
Scram? Ooh, dat's it! Me flatten your head!	7	0	0
Me no see no Marcus anywheres.	7	0	0
May this child of the mortal realm be blessed, touched by the comforting waters that Desther has blessed in the name of Helm. Let the eye watch over your fate.	0	0	7
Me think you not know me. You thinking of some other one.	7	0	0
Sorry to hear that.	0	7	0
Go away! I want all scary people to go away!	5	2	0
On Lord Nasher's behalf I welcome you to the gates of Castle Never. Unfortunately, however, I cannot allow you entry.	0	1	6
Hello, miss. Forgive my forward manner, but you are the very model of feminine beauty.	0	0	7
I'm still makin' deliveries there, though they're not going through as much as they used to. They ain't coming to the gate no more, either.	6	1	0
I hate this. Someone should lay down the law.	0	7	0
More have come seeking shelter, I see. Do you wish to receive the blessing I offer?	0	4	3
You're one of the new recruits I was expecting, right? Well, travel in the Docks district is not recommended.	0	7	0
Missus Aribeth made me militia for li'l while.	7	0	0
We'll be all right. Bad things don't happen to our class of people.	0	6	1
Ruffians back again, hmm? Something you need?	2	3	2
No want. Take a hike.	7	0	0
What would you make of my time? Speak quickly, for I have much to do.	0	2	5
You may enter. Remember that you are a guest and must abide by the rules of common courtesy. The guards do not listen to excuses.	0	1	6
Tormentors! More of you walk free as I languish! Cyric, free me! The scaled worm Gulnan will not have my temple! She will not!	0	0	7