Making Quests Playable: Choices, CRPGs, and the Grail Framework

Anne Sullivan, Michael Mateas, Noah Wardrip-Fruin

Expressive Intelligence Studio
University of California, Santa Cruz
{anne, michaelm, nwf} @ soe.ucsc.edu

Abstract

In table-top role-playing games, quests are goals that can be pursued in a variety of ways, providing structure to the game's ongoing fiction and providing opportunities for role-play. In computer role-playing games, on the other hand, quests are generally structured as lists of tasks or milestones, with variability present only in combat or as a binary choice between completions associated with one or another faction/morality. Given this, the computer role-playing game quest is rarely *playable*—instead existing as a motivation-supplying wrapper around game systems that can be played. While such games can be compelling, they do not manage to live up to the name or potential of the *role-playing* game. For computer games to reach this potential, however, will require a new generation of tools: ones that reify assumptions appropriate to the playable, goal-oriented quest, rather than the task checklist quest. The Grail Framework is a multi-element research project aimed at presenting a first working example of such a set of tools. This paper briefly describes one of its authoring elements (the QuestBrowser brainstorming tool) and one of its runtime elements (the GrailGM game master). While new tools cannot guarantee players a new experience, they can open up new potential spaces for designers and authors, creating conditions in which computer role-playing game fictions can become both authorially structured and meaningful opportunities for the deep experience of role-play.

1. Introduction

"A game is a series of interesting choices," influential designer Sid Meier is often quoted as saying. While there are clearly games for which this is not the case—we don't employ this as a definition—it points to one of the central elements that make games an effective form of interactive media. It also points to one of the central sites of audience attention for games: the systems that present interesting choices are an ongoing focus during play, while those elements that don't we may increasingly "look past" as our familiarity with a game grows.

But what makes an interesting choice? Rollings and Morris observe, "In an interesting choice, no single option is clearly better than the other options, the options are not equally attractive, and the player must be able to make an informed choice." [1] This outlines the moment of choice, but Salen and Zimmerman point out that such moments must also be meaningful within our larger experience of play. A meaningful choice is one in which the outcome of the choice is both "discernable and integrated." [2] Not only should the player be able to have interesting choices, but choices should have a noticeable (discernable) and significant (integrated) impact on the game world.

When looking at the Computer Role-Playing Game (CRPG) genre, we find that while players have some interesting and meaningful choices, these choices are often confined to combat. In contrast, it is typical for a CRPG to be restricted to a pre-set narrative; within which the player moves through the experience, fulfilling checkpoints to advance the story. These story checkpoints are delivered to the player in the form of quests, and these quests often lack interesting or meaningful choices, but are instead delivered as required actions for the player to complete.

Because of this lack of choice—combined with a lack of other possible types of ludic engagement—we argue that quests are not currently playable in many CRPGs. This is part of the reason that players, particularly in Massively-Multiplayer Online (MMO) CRPGs such as *World of Warcraft,* look past the fictional specifics of quests (as "flavor text") toward the tasks they specify for the game's more playable systems. However, when we look at table-top role-playing games, the precursor to CRPGs, we find quests that exist in a playable form that focuses on interesting, meaningful choices.

We believe the CRPG will never be able to reach its potential until its quests become playable. Only then will its players be able to engage in the activity from which it gets its name—role-playing—in a manner that approaches the power of true role-play, as seen in forms ranging from tabletop RPGs to Boalian "forum theater." [3] The playable quest works with the proven structuring power of the quest in the current CRPG form while simultaneously opening the door to experiences in which interesting, meaningful choice is central to experience of the fictional world.

In this paper, we will describe our proposal for another generation of borrowing from table-top role-playing games to CRPGs, one aimed at taking a first step toward playable quests. We begin by discussing the evolution from table-top to computer RPGs with a focus on when and why quests became non-playable. We then offer an overview of the Grail Framework, an ambitious project which aims to address the lack of interesting and meaningful choices within CRPG quests. We conclude with a few forward-looking thoughts.

2. Table-Top Role-Playing Games

To fully understand Computer Role-Playing Games (CRPGs) we must first look at their predecessors, table-top RPGs. Table-top role-playing games grew from war games, which were historical battle re-enactments played with miniatures. Over time, a contingent of war gamers moved away from the strictly historical battles and began to create their own battle campaigns. Some players began to experiment with changing the rule-sets, and some players began to focus on single characters (such as the *Chainmail* "Fantasy Supplement") as opposed to an entire unit, and campaigns moved towards a setting with less focus on historical accuracy. [4]

In 1974 Gary Gygax and Dave Arneson created what would become one of the first and most well-known rule-sets for role-playing games, *Dungeons & Dragons*. *Dungeons & Dragons* (D&D) is set in a fantasy world and has remained the most popular role-playing system to this day. [5]

Table-top role-playing games are played with a group of people, where one person takes the role of the Dungeon Master (DM) who creates the world and story, while the other players create characters that give meaning to the scenario through their actions. As the players move through the world, the DM adapts the story to incorporate player actions. This type of collaborative play involves a constant negotiation between DM and other players to create a story and good game experience with both interesting and meaningful choices.

2.1 Dungeon Master

In table-top RPGs, the Dungeon Master (DM) is the "God" of the game. [6] The player's available actions are ultimately defined by what the DM will allow in a game. Because of the critical role the DM has in the experience, each game is heavily influenced by the DM's abilities. An unsatisfactory DM is incapable of properly negotiating their overall vision of the story with their players' actions. There are two styles of bad DMing, falling at opposite ends of a DM story control spectrum.

On one side of the spectrum is the "railroading" DM. This style of DM has a story in mind and does not allow the players to deviate from it. Players are given no meaningful choices within the game, as their actions have no effect on the progression of the game. When a player offers an action that does not fit the DM's ideal storyline, it is not allowed. This can lead to the player feeling as if they do not have true control over their own character. A more skilled DM will allow an action, but adapt the story such that the action ultimately does not have a negative effect on the story.

For instance, in "DM of the Rings," a satire comic based on a railroading DM, one of the players attempts to dismount from a warg. The DM has already decided that the player and warg will fall off a cliff, so he disallows the player to dismount, even given the player's dice roll. [7] A more skilled DM would have allowed the player to dismount, even if the DM was guiding the players through a story in which falling off the cliff was absolutely necessary. The DM, for example, could have adapted the story through environmental changes (e.g. the cliff below the player's feet collapses) or NPC actions (e.g. another warg runs into the player pushing them off the cliff) to lead to the same outcome. This allows the player a feeling of control over their character, even if the world works against them.

On the opposite end of the continuous spectrum is a DM that does not exert any control over the game's shape. With a large world and many options, players often feel lost and unsure of where to go. Without any guidance from the DM, the players will get easily side-tracked or bored with the gaming experience. Experienced players may use their previous play sessions to guide them through the game, but without any constraints, the game lacks focus and interesting choices, and can become an exercise in frustration. [8]

Bad DMs are of particular interest to us because they are analogous to styles of game play within computer role-playing games. We discuss this in more detail in the next section.

3. Computer Role-Playing Games

Throughout the late '70s and early '80s, table-top role-playing games found their way into the computer domain. The well-defined combat rules from *Dungeons & Dragons* easily translated to the systematic nature of computers. Computers had an advantage over the table-top counterparts in that the computer could quickly and easily store all the rules for the game and effortlessly calculate turns within the combat. Instead of a single encounter taking (possibly) hours; it could be completed within minutes or even seconds, allowing for more progress in a shorter time.

However, the playable storytelling aspects of table-top RPGs were more difficult to represent computationally, and as such have long been less playable in computer role-playing games. CRPG gameplay instead focuses on battle, allowing very few player choices outside of those related to combat.

Due to the lack of storytelling support, CRPGs tend towards the two extremes of bad DMing mentioned in the previous section. Many classic CRPGs, such as the *Final Fantasy* series [9] have finely crafted stories which the player is railroaded into playing. While the stories may be grandiose and well-constructed, the player lacks meaningful choices, and is thus merely a character in someone else's pre-arranged story, given no options or chances to influence the narrative. In fact, major elements of such CRPG stories are often presented in non-interactive "cutscenes" (linear animations) driving home the fact that they are a part of the game that cannot be played, only consumed in set form.

At the other extreme are games such as *Oblivion*, [10] open worlds with a multitude of options for the player, but the overall player experience lacks cohesion and interesting choices for the player. *Oblivion's* designer, Ken Rolston went so far as to say that when designing the game, he kept in mind, "In games, stories suck, so focus on the other elements of narrative: setting and theme." [11]These shortcomings lead Chris Crawford to point out, "no games have approached what a good DM could do with players around a table." [12] How can this be addressed? One option would be to completely eschew CRPG conventions and develop a new game genre. We, however, are interested in extending the CRPG genre. We are encouraged by the fact that stand-out games within the genre such as *Planescape:Torment*, [13] *Star Wars: Knights of the Old* Republic, [14] and *Baldur's Gate II* [15] all have stories that are more playable than games at the polar extremes of the DMing spectrum. They accomplish this through embedding interesting and meaningful choices within their systems of quests.

4. Quests

Jeff Howard describes a quest as "a goal-oriented search for something of value." [16] The quest for the Holy Grail in Arthurian stories is a familiar example of a legendary quest. This style of quest is often used within table-top RPGs by DMs to give a general direction for player actions.

In a table-top RPG, the quest does not need to be followed, or the players can choose to solve the quest in a multitude of ways. For instance, a quest to overthrow a tyrannical king can be solved by combat (kill the king), subterfuge (convince the king's followers to dethrone him), diplomacy (work with nearby kingdoms to remove the king from power), or any other way the players can imagine – and convince the DM to allow.

Taking the lead from table-top RPGs, computer role-playing games also use quests as a staple in gameplay. [17] Quests are often used to direct the player through the game's story, or to give meaning to the player's actions. On the surface, this is very similar to the quest structure in table-top role-playing games, where the DM often provides at least a main quest that the players use to direct their actions. However, in a CRPG, there is often only one way to fulfill a given quest, with a combat-based solution being the most prevalent.

There are games that are exceptions to this, especially the stand-out games mentioned above such as *Planescape: Torment* and *Star Wars: Knights of the Old Republic*. Both *Planescape* and *Star Wars*, for example, integrated non-combat solutions for a selection of quests, allowing players to choose whether to take a traditional combat role or to fulfill one of the other supported solutions to complete the quest. But, given current tools, such multiple-solution quests are burdensome to implement and highly bug-prone. In our interviews with quest designers at a major MMORPG developer, we were told that—using their in-house tools—the effort required to implement quests with multiple solutions was not additive, but exponential. Similarly, Wardrip-Fruin has analyzed BioWare's Aurora editor, [18] showing how it reifies quests as a series of milestones, requiring those with other aims to work against the system's organization, leading ambitious RPGs toward narrative breakdown. [19] Wardrip-Fruin's example is *Star Wars: Knights of the Old Republic,* but we see it in many other CRPGs as well. For example, in Morrowind, [20] quests exist in which the player has the choice to kill a specific game character (referred to as a non-player character or NPC) or find another non-violent solution. Regardless of the choice made by the player, they can later receive a quest which requires them to talk to the NPC which they may or may not have killed. This not only demonstrates the bug-prone nature of this approach, but also shows that the choice the player makes does not affect the underlying storyline.

Partly for these reasons, an absence of player choice in quest actions is quite prevalent, particularly in Massively Multiplayer Online Role-Playing Games (MMORPGs). In an MMORPG, the game world is persistent, which means that the game continues to run even when a particular player is not playing. Up to thousands of players are simultaneously playing the game on a central server; therefore the game must support each of these playing experiences. Due to the persistence of the game world and the large number of simultaneous users, there is rarely a central story arc (a notable exception to this is *A Tale in the Desert,* [21] which resets the game world to the beginning state once a year, thereby having an "end" to the story). Quests are used mainly to give thematic meaning to the supported player actions – typically fighting with enemies scattered throughout the world – and to move players towards areas suitable for their level. Quests are popular in both single-player and MMO RPGs; however the online worlds generally have thousands of quests to examine, which is significantly more than most single-player RPGs. While we are going to briefly focus on quests in the extremely popular MMORPG *World of Warcraft* (*WoW*), [22] the issues we discuss common are across single-player and multi-player CRPGs.

There are a number of quest taxonomies suggested for *WoW*; we combined and adapted the systems available to make the following taxonomy [23, 24, 25]

- Kill X number of enemies (where X may be 1, and the enemy unique)
- Kill enemies until X number of a specific item drops (where X may be 1+)
- Collect X number of specific items from the environment (where X may be 1+)
- Deliver an item to a specific NPC.
- Talk to someone specific.
- Escort someone.
- Use a special ability.

As this taxonomy begins to highlight, in *WoW* the majority of the quests and experience points received are related to activities that revolve around killing. To illustrate this further, we examine the second expansion of *World of Warcraft: Wrath of the Lich* King, [26] which was released in late November 2008. One of the first regions (called zones) that a player can choose to explore is Howling Fjord. In this zone, there are 133 quests available to an Alliance player. Breaking down these quests into the above taxonomy and then weighting the categories based on the experience received for doing each quest illustrates the continued combat-centric game design.

Of the 133 available quests, 55% of the quest experience comes from straight kill-based tasks of the type “Kill X number of enemies,” “Kill a specific named enemy,” or “Get X number of items from killing enemies (drops).” Furthermore, 22% of the experience is received from collection-style quests, most of which require collecting items from areas that are infested with enemies.

Additionally, the only ways to gain experience in *World of Warcraft* are to complete quests, kill enemies, or discover new areas. The experience received from discovering new areas only accounts for a minute portion of the overall experience received by a player.

5. Task-Based vs. Goal-Based Quests

A striking feature of the style of quest described above is how different they are from the definition provided by Howard. In his description, the key concept is that of a “goal-oriented search.” In contrast, most contemporary CRPG quests present the player with a checklist of actions to take to complete the quest. The goal is to complete the list; there is no player choice involved other than whether they choose to complete the quest or not.

In such quests, which we refer to as *task-based* quests, the list of tasks provides the player with the collection of non-optional actions that must each be completed in order to complete the quest. In contrast, a *goal-based* quest presents an end point, or goal, for the player to achieve. The player chooses, within the constraints of the game world and mechanics, how to reach the given goal.

An example of a task-based quest that could be found within many RPGs is the quest to save a farm from wolves. A farmer will give a player a task of killing the wolves to save the farm. The player must kill the wolves in order to receive the reward for completing the quest, regardless of class or role-playing preferences. In contrast, a goal-based quest would simply explain to the player that there are wolves killing the local livestock. The player would then be allowed options for completing the quest. They could still kill the wolves if they choose, but other options would be available such as creating better fencing or helping restore the local deer population so the wolves no longer have to hunt livestock. In this way, the player is able to make an interesting choice (no choice is clearly better than the other) perhaps based on game-defined class or race, or on personal role-playing preference. With the inclusion of these choices having a meaningful effect on the game (e.g. different quest rewards, game world evolution, or even affecting future quest and solution availability) quests become playable.

The existence of player choice in how quests are completed is the key difference between goal-based and task-based quests. Early adventure games such as *The Secret of Monkey Island* [27]and *King’s Quest* [28] games often presented the puzzles within the game as a quest for the player to fulfill: “Become a pirate,” “Save the mayor,” “Save the land.” While on the surface these seem like goal-based quests, there was in fact no player choice involved in quest completion. This could easily lead to frustration where the player would be searching for the solution the designers had chosen for each quest, even though there were other options that made sense to the player but were not supported.

For example, in *Monkey Island* there exists a deceptively simple goal of getting past some deadly piranha poodles. At this point in the game, the player has become a sword master, but they are not allowed to fight the poodles, they must go find the exact item required to pass the dogs. The player has a chunk of meat, but this alone is not exactly what is required. Using the meat with grog (potent alcohol) does not work, but instead the player must eventually realize that they need to use the meat with a small flower they (hopefully) found while walking through the woods to drug the meat. Until the player stumbles upon this solution, they cannot progress further in the game.

These frustrations are due to these quests being presented as goal-based quests, when, in fact, they are task-based quests with opaque specifications. A true goal-based quest allows for interesting player choice, where there are multiple ways to fulfill the quest, and one solution is not obviously better than the others.

5.1 Agency

Giving the player interesting choices with which to solve their quests relates strongly to the theory of agency within game design literature. In particular, Wardrip-Fruin, et. al. describe agency as “a phenomenon involving both player and game, one that occurs when the actions players desire are among those they can take (and vice versa) as supported by an underlying computational model.” [29]

Achieving agency, in this account, does not require enabling players to "do anything." In fact, it is in many ways the opposite. It requires crafting the dramatic probabilities of the fictional world and developing player understanding of the underlying computational system so that the two are in concert with each other.

We see a version of this in traditional tabletop role-playing games. A set of dramatic probabilities are established for characters that connect to the underlying game system. For example, the expectations for classes like Wizards and Clerics – and races like Dwarves and Elves – are in part derived from literary sources (e.g., Tolkien) and in part formed by player knowledge of the game system (e.g., Clerics have a wide range of healing spells, Elves are generally better at agility-heavy tasks than Dwarves). These then come into action during gameplay, as players use their characters to attempt actions that are appropriate both for the dramatic probabilities of the situation and the specifics of their character – one doesn't role-play Aragorn and Gandalf the same way, even if they find themselves in the same situation.

The DM facilitates play that is appropriate to the different dramatic probabilities of each character, using the rule system, enabling an experience of agency.

This connects directly to our critique of quests in current computer role-playing games. Because quests are fixed lists of tasks, they assume that the dramatic probabilities are the same for a fighting character, a healing character, and a stealthy character. Furthermore, they assume the probabilities are the same for a character no matter what quests they have completed in the past, what special abilities they have, and so on. This, in turn, requires that combat be the center of almost every quest – because this is one of the few things that every character can do, even if it doesn't always make dramatic sense if we take particular characters seriously.

Our research is motivated in part by this mismatch. We want to enable quest authors to create quests, with a tractable amount of effort, that provide dramatically appropriate paths for different sorts of characters both in their fictional and game system representations. Beyond our near-term research, we are also interested in directions that will make it possible to dynamically alter and generate portions of quest structures to support dramatic probability for a wide range of characters with no more authoring effort than is required with today's relatively clumsy quest authoring tools.

6. The Grail Framework

The Grail Framework is our current research project, designed as a first step in addressing the issues discussed above. The system encompasses both authoring tools (including the QuestBrowser brainstorming tool) and an in-game system called the GrailGM (Grail Game Manager). The Grail Framework in its entirety is designed to create a framework in which the designer is able to author high-level rules together with relatively-atomic pieces of traditional content, both leveraging appropriate knowledge representations in the RPG quest domain. By not requiring the author to use traditional scripting methods to create quests, they are able to focus on the tasks of designing and writing, as opposed to programming.

The Grail Framework also allows for the development of new kinds of CRPG quests. As the designer creates content and rules, the Grail Framework is able to dynamically combine these in new and interesting ways. The designer is able to maintain authorial control over the world via designer goals—similar to a Game Master presiding over a table-top role-playing game. Quests and solutions available to the player are shaped by the player's own history—details such as where the player has traveled, who they have talked to and the types of relationships they have developed with these non-player characters (NPCs), as well as how they have solved previous quests.

To create enough content for the player to be able to make these choices requires much more work from the designer. Not only is there an issue of content generation, but it is also difficult for the designer to keep track of the inter-dependencies between quest lines and quest solutions. Because of this, it is necessary for the Grail Framework to also contain author support tools such that designers are realistically able to create the content required for such a system. Below we briefly describe one element of this: the QuestBrowser brainstorming tool.

We also discuss GrailGM, a system designed as a stand-in Game Master for the designer while the game is running. In some ways, the GrailGM is similar to a Drama Management system such as DODM; [30] instead of manipulating plot points in the game story, the GrailGM uses the player's history and the designer's goals to select the quests available to the player. As the player learns more about the world, GrailGM updates what quest goals and actions are available. In this way, the world is reacting dynamically to the player's movement throughout the world, giving the player a higher sense of agency.

7. QuestBrowser

To truly achieve playable quests, the designer will need to be able to create a large number of possible solutions for each quest. We have created the QuestBrowser brainstorming tool as part of the Grail Framework to help designers with this challenge as well as help alleviate the difficulties in thinking up multiple interesting solutions for each quest. QuestBrowser is a GUI interface that leverages the common-sense database ConceptNet3 [31] to find links between quest-related ideas. One of the benefits of using ConceptNet is that the database is able to easily supply relationships between objects that are unusual or surprising in some way, as it does not have a personal bias or preference for a specific outcome.

ConceptNet3 structures its data by storing concepts as nodes which are connected to other nodes based on their relationships. There are currently twenty discrete relationship types, such as *PartOf*, *ConceptuallyRelatedTo*, *UsedFor*, *CapableOf*, and *LocationOf*. In QuestBrowser we currently eliminate the most abstract link types (e.g., *ConceptuallyRelatedTo*) and limit the length of the paths returned by ConceptNet to 5 nodes, in order to focus results on those most likely to be useful for authors.

To interact with the system, a designer supplies the GUI tool with a quest concept, and, optionally, a possible quest objective. The system returns a list of linked paths through the knowledge space that connect the starting concept and quest objective specified. This gives the designer possible ways to reason about the relationships between these objects.

If a designer does not specify a goal or objective, QuestBrowser will show all nodes directly connected to the concept node chosen. This allows the author to explore the space of the concept they had in mind, possibly sparking ideas for quest goals based on that concept. To increase the authorial power of the tool, we have given the designer control over which relation types are used in the results giving the designer the power to narrow down the results returned. For instance, some designers may wish to know the temporal relation of subjects, and will want the results to include relation types such as *HasSubEvent*, *HasPrerequisite*, etc., while another designer may wish to exclude those types of relationships from their results.

These constraints are not always necessary, and often the default relationship links return usable quests. For instance, choosing church as the starting concept and heal as the quest objective generates the following non-obvious idea as one of its solutions:

Church –(*LocationOf*)→ **Music** –(*CapableOf*)→ **Heal**

A more detailed description of QuestBrowser is available in [32]

8. GrailGM

[Figure 1 near here]

The GrailGM is the run-time portion of the Grail Framework. Figure 1 provides a diagram of how the GrailGM interacts with the game. As the player takes action inside the game world (e.g. talking to non-player characters) the game updates the world and player states. The GrailGM contains recognizers, in the form of rules, which fire when the world/player is in a particular state. The particular states that are necessary are specified by the designer as part of the rules.

The Quest Manager uses rules to filter through the quest library, to find quests that are appropriate for the player given the state of the game. It also monitors the list of active quests to see if any quests have been completed. The Quest Manager uses this information to inform the player of new quests, enable quest-related actions, and to reward the player for completed quests. When a quest is accepted by the player, it is removed from the quest library and moved to the active quest pool. When an active quest is completed, it is removed from the list of active quests and stored in a library of completed quests, along with the actions taken for completion. This allows the Quest Manager access to this information which may be used for choosing future quests.

The GrailGM decomposes quests into different parts, storing each part as a separate entity within the quest library. The components of a quest that we store are:

- the quest goal: what the player needs to fulfill to complete the quest
- the available actions the player may take to complete the quest
- the reward for completing the quest
- non-player character (NPC) types who are able to give the quest
- dialog options available to the NPCs involved in the quest

We currently use Drools, [33] a forward-chaining inference based rules engine as the backbone of the GrailGM. Facts (relational data tuples) are asserted as working memory elements and these are matched against rule conditions. We have chosen to use a rule-based system as it naturally supports dividing quests into smaller component parts (NPCs, dialog, quest goals, quest solutions, etc.) and allows for authoring rules for dynamically recombining the pieces. This allows us to move away from scripting every possible recombination, which would quickly become intractable and bug-prone.

Player state—consisting of knowledge known by the player, as well as player history—and world states are asserted as facts. Each part of a quest is represented as a rule, with designer-specified preconditions stored as the patterns the GrailGM uses to match to asserted facts. The GrailGM filters possible quests and actions through the pre-conditions based on player history, appropriate NPCs, and current world state.

A more detailed description of the GrailGM is available in [34].

9. Conclusion

Currently, there are neither interesting nor meaningful choices for a player to make in most CRPG quests. Instead they are handed a to-do list to check off as they follow the instructions precisely or, at best, a binary choice between the completions associated with one faction/morality or another. This makes meaningful role-playing (which one might expect at the heart of a genre called the “role-playing game”) impossible—and a significant experience of agency unlikely except in combat-centric situations. Perhaps, then, it should be no surprise when even a famous CRPG designer such as Rolston quips, “I hate getting quests. I hate the toil of completing quests. I hate the formal and predictable resolution of quests. At best, I feel a Puritan sense of rectitude for laboring dutifully, of doing my duty to uncover the fog of narrative war.” [35]

One alternative—Rolston’s—is to turn away from the structuring mechanism of the quest in favor of environmental and systems-oriented exploration. But even with human-level intelligence, this approach often leads to a formless experience often associated with a certain kind of poor DM. Rather than this, we propose a third alternative for CRPGs, which we have here called the *playable* quest.

Ambitious computer game designers and authors have already attempted some exploration of the playable quest. But, given the available tools and the assumptions they reify, the attempts are partial, bug-prone, or both. In response, we are pursuing a research project called the Grail Framework, which seeks to give designers the tools necessary to create truly playable CRPG quests. If we are successful it will not guarantee players a new experience of quests—but we will be able to place a tool in the hands of CRPG creators that reifies a different set of assumptions. Specifically, our project constructs quests as goal-oriented pursuits that may move in a variety of directions, appropriate for role-playing different types of characters and influenced by the history of past role-play. These quests are supported by explicit knowledge representations of elements important to computer and table-top RPGs, including the evolving knowledge state of the player character. The construction of such quests is supported by an authoring approach that allows designers to build on our existing packages of rules or extend the system with new rules. It is supported by open-ended brainstorming tools that can help discover interesting connections between elements of the fictional world.

Together, we hope that these elements of the Grail Framework can help spark a new generation of computer role-playing games, one inspired by the non-combat play made powerful by successful DMs and GMs. Tosca states that "…the success of pen&paper games is precisely in the common creation of a story. […] Paradoxically, this is what cannot be reproduced by computer games." [36] We feel that while the success of table-top games might not be able to be reproduced exactly without the creation of an AI-complete DM, our system will move CRPGs one step closer towards the depth of story table-top games have enjoyed, while retaining the strengths that have made the CRPG a successful game genre.

End Notes

[1] Andrew Rollings and Dave Morris, *Game Architecture and Design* (Berkeley, CA: New Riders Publishing, 2003).

[2] Katie Salen and Eric Zimmerman, "Game Design and Meaningful Play," in *Handbook of Computer Game Studies*, eds. J. Raessens and J. Goldstein (Cambridge, MA: MIT Press, 2005).

[3] Augusto Boal, *Theater of the Oppressed*, (London: Pluto Press, 2000).

[4] Erik Mona, "From the Basement to the Basic Set: The Early Years of Dungeons & Dragons" In *Second Person*, eds. Pat Harrigan and Noah Wardrip-Fruin, 25-30 (Cambridge: The MIT Press, 2007).

[5] Greg Costikyan, "Games, Storytelling, and Breaking the String," in *Second Person: Role-Playing and Story in Games and Playable Media*, ed. P. Harrigan and N. Wardrip-Fruin (Cambridge: The MIT Press, 2007).

[6] Gary Alan Fine, *Shared Fantasy: Role-Playing Games as Social Worlds* (Chicago: The University of Chicago Press, 1983).

[7] Shamus Young's official Web Site, "The DM of the Rings," May 14, 2007, http://www.shamusyoung.com/twentysidedtale/?p=1017 (accessed July 12, 2009).

[8] Gary Alan Fine, *Shared Fantasy: Role-Playing Games as Social Worlds* (Chicago: The University of Chicago Press, 1983).

[9] Square Enix, "Final Fantasy," 1987-current.

[10] Bethesda Softworks, "Elder Scrolls IV: Oblivion." 2006.

[11] Ken Rolston, "My Story Never Ends," in *Third Person: Authoring and Exploring Vast Narratives*, eds., Pat Harrigan and Noah Wardrip-Fruin (Cambridge: The MIT Press, 2009).

[12] Chris Crawford, *Chris Crawford on Game Design* (Indianapolis: New Riders Publishing, 2003).

[13] Black Isle Studios, "Planescape: Torment," 1999.

[14] BioWare, "Star Wars: Knights of the Old Republic," 2003.

[15] BioWare, "Baldur's Gate II: Shadows of Amn," 2000.

[16], [17] Jeff Howard, *Quests: Design, Theory, and History in Games and Narratives* (Wellesley: A K Peters Ltd, 2008).

[18] BioWare, "Aurora Engine Toolset," 2002.

[19] Noah Wardrip-Fruin, *Expressive Processing: Digital Fictions, Computer Games, and Software Studies* (Boston: The MIT Press, 2009).

[20] Bethesda Softworks, Ubisoft. "Elder Scrolls III: Morrowind." 2002.

[21] eGenesis, "A Tale in the Desert," 2003.

[22] Blizzard Entertainment, "World of Warcraft," 2004.

[23] Faltin Karlsen, "Quests in Context: A Comparative Analysis of Discworld and World of Warcraft," in *Game Studies Journal* 8, no. 1 (September 2008).

[24] Gyuhwan Oh and JuYoung Kim, "Effective Quest Design in MMORPG Environment," (conference proceedings of Game Developers Conference, San Francisco, CA, 2005).

[25] Jill Walker, "A Network of Quests in World of Warcraft." in *Second Person*, eds., Pat Harrigan and Noah Wardrip-Fruin, 307-310 (Cambridge: The MIT Press, 2007).

[26] Blizzard Entertainment, "World of Warcraft: Wrath of the Lich King," 2008.

[27] Lucasfilm Games, "The Secret of Monkey Island," 1990.

[28] Sierra Entertainment, "King's Quest," 1984-1998.

[29] Noah Wardrip-Fruin, Michael Mateas, Stephen Dow and Serdar Sali. "Agency Reconsidered." (paper published in Breaking New Ground: Innovation in Games, Play, Practice and Theory: International Conference of Digital Games Research Association, London, 2009).

[30] Anne Sullivan, Chen Sherol and Michael Mateas., "From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience," (proceedings of the AAAI 2009 Spring Symposium on Intelligent Narrative Technologies II, Palo Alto, 2009).

[31] Hugo Liu and Singh Push, "ConceptNet: A Practical Commonsense Reasoning Toolkit," *BT Technology Journal* 22, no. 4 (October 2004): 211-226.

[32] Anne Sullivan, Michael Mateas and Noah Wardrip-Fruin, "QuestBrowser: Making Quests Playable with Computer-Assisted Quest Design," (conference proceedings of the 8th Digital Art and Culture Conference (DAC), Irvine, 2009).

[33] JBoss, "JBoss Rules," 2009.

[34] Anne Sullivan, Michael Mateas and Noah Wardrip-Fruin, "Rules of Engagement: Moving Beyond Combat-Based Quests." (proceedings of the 3rd Workshop on Intelligent Narrative Technologies, Monterey, 2010).

[35] Ken Rolston, "My Story Never Ends," in *Third Person: Authoring and Exploring Vast Narratives*, eds., Pat Harrigan and Noah Wardrip-Fruin (Cambridge: The MIT Press, 2009).

[36] Susana Tosca, "The Quest Problem in Computer Games," (proceedings of Technologies for Interactive Digital, Darmstadt, 2003).