

Quest-centric Authoring of Stories, Quests, and Dialogues for Computer Game Modifications

Robin Horst, Micha Lanvers and Ralf Dörner

RheinMain University of Applied Sciences, Unter den Eichen, Wiesbaden, Germany

Keywords: Computer Game Modification, Modding, Story Writing, Layperson Authoring of Real-time Interactive Systems.

Abstract: A computer game modification (mod) is an extension or variation of an already published computer game, usually developed by hobbyists. Creating computer games requires expertise from different disciplines. For role-playing games (RPGs), one of them is the development of coherent stories, for example, through quests and dialogues between characters. Different expert authoring tools exist supporting story writing specialists in their profession. However, based on the results of our related work analysis and expert interviews with two lead storywriters of successful mods (*Enderal* and *Legends of Ahssûn*), we identified a gap of research regarding story writing tools suitable for the modding community. In this paper, we introduce quest-centric concepts for supporting modders in designing and managing quests for RPG mods and illustrate how to use graph-based visualization techniques for modeling interactive dialogues and dependencies between quest components of a story. Based on statistically significant results of an expert user study utilizing a reference implementation of our concepts, we conclude that the quest-centric concepts supported our participants and that the proposed graph visualizations and automatic optimization suggestions were particularly helpful to them.

1 INTRODUCTION

In role-playing games (RPGs), such as *The Elder Scrolls: Skyrim* or *Gothic 2*, players experience a story mainly through completing quests instructed by non-player characters (NPCs) (Howard, 2008). A core task of creating RPG game software is the authoring of a story including quests and dialogues. A storywriter must also plan for potential decision possibilities of players within the story, what can make the authoring of stories, quests, and dialogues more complex. For example, the assessment of a story's coherence with its dependencies on the players' decisions is difficult to overview. To support professional game developers there already exist software tools, however, not everyone with a good idea for a story is a professional storywriter.

A modification (mod) of a computer game is an expansion or adjustment of an already released computer game and is generally developed by non-professionals. Successful examples are *Enderal* for *Skyrim* with an estimated 3,000,000 downloads and a playtime of 30-100 hours and *Legend of Ahssûn* (LoA) for *Gothic 2* with more than 60,000 downloads and an estimated playtime of 50-200 hours. With a

story-mod's main goal of providing additional story, resulting in extra playtime, the story writing is an essential part of the modding process. However, hobbyist storywriters face various challenges when developing a mod.

In this paper, we explore the little-researched RPG-modding domain and focus on challenges of authoring quests and dialogues. We make the following contributions:

- We point out requirements and investigate in the motivation of storywriters for mods within semi-structured expert interviews with the lead storywriters of two successful mods (*Enderal* and *LoA*).
- We introduce *quest-centric* authoring concepts for supporting modders in designing and managing quests as the most prominent element in RPG story mods. Furthermore, we propose using graph-based visualization techniques for modeling interactive dialogues and dependencies between quest components of a story.
- We utilize a reference implementation of our concepts within an expert user study and state lessons learned.

2 RELATED WORK

Authoring procedures of content for real-time interactive systems such as games, virtual realities, etc. can be highly challenging for layperson authors (Horst, 2021). In his work (Sotamaa, 2010; Sotamaa, 2003), Sotamaa examines motivations of hobbyists for participating in mod projects. He states that first of all, people contribute in mods for the fun of the original game itself, as artistic self-expression, being part of a community, and lastly to get a chance to enter the professional gaming industry. However, similar to work by Hurel (Hurel, 2016), Sotamaa (Sotamaa, 2010; Sotamaa, 2003) concludes that the modding domain deserves and needs more research.

Concerning the design of quests and stories, Howard (Howard, 2008) elaborates on the differences between quest games and quest narratives, and puts a strong focus on the description of entities associated with quests such as levels, characters, items, dialogues, journal/logbook entries, and events. He makes clear that their design has a significant impact on the quality of a quest's design. Howard (Howard, 2008) also points out different quest types in games with a focus on quests to tell the story. He distinguishes types such as kill quests and fetch quests and states practical tips on quest design.

Smith et al. (Smith et al., 2011) analyzed 20 quest-focused RPGs in their work in order to derive quest and level design patterns and to gain a better understanding of the interrelationships between these areas. This work indicates that the quest design in such games is highly dependent on the level design. This conclusion is also supported by work of Aarseth (Aarseth, 2005). From these works, we derive the recommendation that quest design and quest management software should provide the ability to link quests with information about locations, characters, and items in order to design them synergistically.

In recent work of Veloso and Prada (Veloso and Prada, 2021), a debugging application was developed for storywriters to automatically test designed stories to detect inconsistencies or predict the outcome of different player types' playthroughs. The authors conclude that such an application encourages authors to create more complex narratives and thus to develop artistically without feeling restricted. A user test confirms these assumptions and also indicates positive effects on the efficiency of the story writing process. A tree visualization was used to prepare the structure of the story. Overall, the work by Veloso and Prada (Veloso and Prada, 2021) indicates that storywriters felt supported by the visual overview of their stories and automated error checks.

Besides work in academia, several authoring tools have been introduced in the industry. Articy:Draft (Articy Software GmbH & Co. KG, 2021) is a commercial software for game writing, planning, and content management. The core of the application is the *Story Flow*, a graphical editor that can be used to design dialogues and story events again as a graph (flow graph). Another application is ChatMapper (LearnBrite, 2021), which also represents dialogue texts and events as a flow graph. DiaDepp (Sektenspinner, 2009) is a dialogue manager specifically for modifying the game Gothic 2. It can be used to create dialogues and events that can be stored in their own folder structure. These contain script code, where dialogue lines and frequently used functions like passing items can be written in a highly simplified syntax. The export function resolves the simplified syntax so that created dialogues and events can be used directly in the game. Other applications with similar approaches we identified include Twine (Interactive Fiction Technology Foundation, 2021), Arcweave (Arcweave OU, 2021), Dialogue Designer (radmatt, 2021), and TalkerMaker Deluxe (digiwombat, 2021).

Overall, we found a lack of research for studies and tools applied within the modding domain. Some mentioned works and authoring applications take a general approach to story writing. However, RPG mods typically structure their content into quests, which in turn are often grouped and presented according to importance and game stage. Furthermore, the ability to manage quest-associated entities varies across applications. Graphs were found to be suitable for representing interactive dialogues and can possibly be utilized for quest-specific approaches. For example, one feature we found missing existing work is the automatic generation of a visualization of dependencies between a quest's individual component, which was investigated in the work of Veloso and Prada (Veloso and Prada, 2021) and seems practical for a better overall view of the story but also for debugging purposes, that might be crucial for non-professional authors. At last, further assistance functions for inexperienced storywriters are only little researched.

3 EXPERT INTERVIEWS

To get more specific insights into story writing within the modding domain, we conducted two one-hour interviews with the lead storywriters of Enderal (participant 1, P1) and LoA (participant 2, P2). In this section, we briefly state the results. The interviews were conducted as remote semi-structured interview

divided into three main questions that asked them about their mod projects to gather postmortem information:

1. Which Tools Did You Use?

Our participants stated that they used particularly Articy:Draft (P1) and Word (P2) for planning and writing quests and dialogues for the story-mods. A key message of the interview with P2 was that the roles of *scripters* and *storywriters* were differentiated within their mod team and that, from P2's experience, 'scripters were more eager to develop features than to implement dialogues that have already been prescribed'. This was the main reason for utilizing DiaDepp since it facilitates to export prescribed dialogues into game-scripts. However, for the sake of the overall authoring workflow, P2 had to adopt the dialogue syntax technically determined by DiaDepp, which was perceived 'restricting' within the design workflow.

For the design of the interactive dialogues for Enderal, P1 utilized Articy:Draft. P1 used its export functionality to provide dialogue texts to voice actors in a presentable form. However, for the mod itself, the dialogues had to be integrated manually, by the storywriter. To plan characters and their backstories, P2 used common text-formatting tools such as Google Docs.

2. How Should a Story Writing Tool Be Designed to Support Modders in Creating Quests and Dialogues?

Both participants stated features they would add to their existing tools or wish for novel ones. P2 noted that quests are essential parts of the overall modding of game stories and that '[...] in DiaDepp, quests are structured and sorted by chapter of the story and the location a quest takes place'. Such sorting and structuring functionality was recommended for future tools to help novel authors getting a quick overview over the story and also for finding and adjusting certain quests more efficient.

Concerning the structures of quests, P1 further demands from future tools to provide story mod authors with support for the narrative design of quests. For example, providing typical quest structures (Howard, 2008) such as fetch quests or kill quests as templates so that the hobbyist authors could use them as a starting point. P1 did also mention a particular opportunity for more assistance functionality for a story writing tool. Further automated guidance throughout the authoring process would be appreciated, for example, regarding the game-play loops of a mod. Assistance should be an optional element, since P2 noted that in the initial phase of modding, people will often 'lack necessary expertise' in quest writing.

Both participants mentioned that, for modders

even more than for professional storywriter for games, technical aspects of a game should be abstracted as much as possible ('the less technical the better', (P1)), since writers within the modding might not want to deal with the technical aspects in their free-time ('I want only to write' (P1)). Export functionalities for quests and dialogues to game-scripts should be included similar to DiaDepp, however, the participants assumed for compatibility for specific mod kits rather than generic scripts.

Besides a 'neat graphical user interface', P1 stated that the visual representation of quests and specifically their sub-components such as dialogues, dependencies to other quests, or involved NPCs would help writers after the initial creation of a quest. A visualization and distinction between such elements could help 'especially for [updating and maintaining] non-linear dialogues' (P1).

3. What Would You Do Differently in Hindsight?

When asked what they would do differently in hindsight, P2 replied 'to put more focus on actual story writing'. In the initial phase, this was given a lower priority than the level design, so that the mod's game world was already finished before the design of the main story then had to be strongly oriented towards it. 'Quests were often developed for the sole reason that a game location would otherwise have had too few' (P2). As a second lesson learned from this, P2 stated that an option to display information about specific items and characters and linking them to particular dialogues was identified a useful feature for future tools. This insight is underlined by P1, who further demands from a suitable tool to attach pictures for characters, items, and locations.

Finally, the interviews underlined that story writing plays a major role in the modding domain and that existing applications used within successful mod projects did not entirely fit the modders' needs. A corresponding application should support storywriters in their creative process without restricting their artistic freedom and pursue a UI design that is as de-technicalized as possible. Furthermore, the idea of direct game-script export functionality was considered useful, as it can lighten the workload of scripters within a mod team and enables storywriters to create their own adjustments to the mod project without in-depth scripting knowledge. Also, both participants noted that the modding workflow requires writers to restructure and adjust quests and dialogues of a story-line multiple times as a story mod project 'actually always starts unstructured and with a rough idea and everything is thrown around 1000 times' (P2). Both writers regarded quests as most essential element when creating a story mod of a game.

4 THE QUESTMANAGER AUTHORIZING TOOL

In the following, we introduce six core concepts (*C1* to *C6*) designed based on the findings from our literature analysis and expert interviews. The concepts are then utilized within *Questmanager* – a story authoring tool for usage within the modding domain that focuses on quests as the central part of creating a story mod.

C1 - Quest-centric Structure:

Quests consist of conversations with NPCs (dialogues) and events that are executed under certain conditions (e.g., at a certain location). We define the following *quest components* regarding dialogues and events:

- Dialogue line: Text that the player speaks to the interlocutor, interlocutor speaks to the player, or player speaks to him/herself
- Event: Generally everything game-specific that can happen in a dialogue or event; e.g. action of the player/NPCs, starting a quest, generating a journal/logbook entry.
- Condition: Control element to execute dialogue lines/actions only conditionally; e.g., only after conducting a dialogue or obtaining a specific item.
- Choice: Possibility for the player to accept or reject a quest, for example.

Dialogues and events also have preconditions that define under which circumstances they can be conducted. Dialogues have additional parameters such as the dialogue partner and the information if the dialogue partner starts the dialogue by himself and if a dialogue can be repeated. Quests should also be structurable, for example, to distinguish between main and side quests or to assign quests to individual game sections.

C2 - Game-specific Configuration and Export:

A Mod is always based on exactly one game, which possesses own functionalities, characteristics and script syntax. As stated by Howard (Howard, 2008), the consideration of game-specific mechanics in the design process leads to more immersive quests. We suggest that actions and conditions from *C1* have their own configuration per supported game, in which necessary parameters, representation aspects in an authoring tool, as well as information for the script export can be defined. Through this, quests designed in an authoring tool can be imported directly into the game with little effort and scripting experience.

C3 - Dependency Visualization of Quest Components:

Quest components such as dialogues and events can be dependent on each other in different ways,

which can make it difficult to keep track of them, especially in complex quests. An example is illustrated in Fig. 1. Dialogue D3 should only be able to be conducted by a player if the previous D2 was already held. In this case D3 is directly dependent on D2. A quest also has a certain status such as *not started*, *started*, *done* or *failed*. If D2's quest status is set to *started* and D4 can only be held when Q2 has the status *started*, D4 is directly dependent on D2. Furthermore, variables can be used to represent more complex conditions in interactive dialogues. For example, if the variable Var is set in D4, and D5 checks for the value of Var, then D5 is directly dependent on D4.

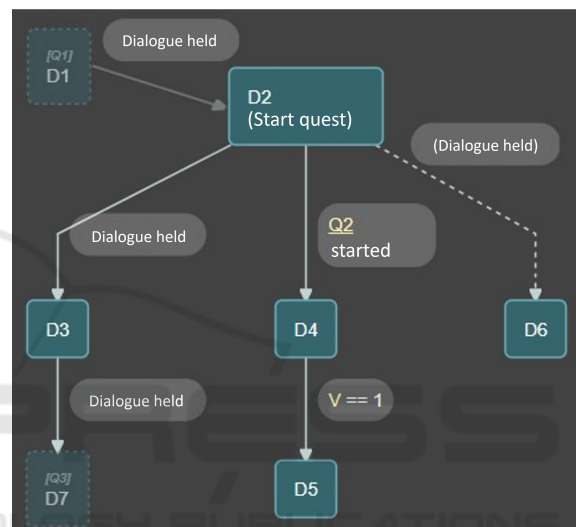


Figure 1: Dependencies of exemplary dialogues D1-D7.

We differentiate direct and indirect dependencies of dialogues. While a direct dependency (Fig.1 solid arrows) determines whether a dialogue or event can be conducted at all, indirect dependencies only query conditions within an ongoing dialogue (Fig.1 dashed arrows).

By visualizing the pre- and post-conditions of dialogues and events, authors can identify previous quests (e.g., in case of D2: dialogue D1 of quest Q1). By looking at the preconditions of all quest-unrelated dialogues and events, subsequent quests can be identified (e.g., dialogue D7 of Q3). The display of such non-quest dialogues and events should be optional for extensive quests in order to be able to control the clarity. In such visualizations, also dependency conflicts can be identified. This includes, for example, circular dependencies, where a dialogue depends directly on a second dialogue, and at the same time the second dialogue depends directly on the first, which leads to a deadlock situation. By identifying such conflicts early on, logic errors can be uncovered and later game errors can be avoided.

C4 - Quest Component Visualization:

As examined in Section 2 and apart from dependencies, flow graphs are also a suitable representation for interactive dialogues themselves. They provide authors with an advantageous overview compared to the usual text form, particularly when many conditions and choices are involved. We also illustrate the quest components proposed in C1 within a graph depicted in Fig. 2.

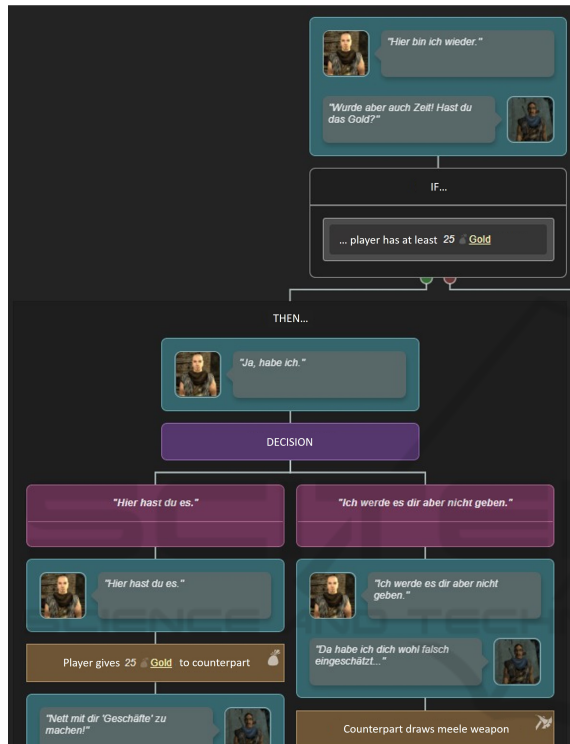


Figure 2: A dialogue excerpt of a German mod project.

C5 - Linkage of Quest Components to It's Entities:

Various studies (Howard, 2008; Smith et al., 2011; Aarseth, 2005) and applications (Arcweave OU, 2021; Articy Software GmbH & Co. KG, 2021; digiwombat, 2021; LearnBrite, 2021; Sektenspinner, 2009) illustrate that the design of a quest's entities such as characters, items, and locations is strongly related to the quest design itself. However, results of the expert interviews suggest that text editing programs are commonly used to create character profiles etc. As a result, information may be scattered across multiple files or software that have no links to each other, making it difficult to keep track. We propose linking the entities of a quest with their essential information to the quest components of a quest within an authoring application. If a character is necessary for an action/condition (e.g., a query if an NPC is dead) it can

be linked directly to its information. If a character or location name appears in the text of a dialogue line, a link is generated automatically.

C6 - Quality and Efficiency Enhancing Assistance Functions:

In addition to the visualization of dialogues as a flow graph, other views can help with designing quests. For example, a listing of all of a character's dialogue lines from different dialogues can help examining a uniform language style. The information that a certain character does not yet have a single dialogue line can also reveal gaps in the quest design. Another example is a view specifically for writing dialogue lines. Other quest components are neglected in this view and the author is provided with supplementary functions for writing, such as spelling checks, context-specific word suggestions, or checks for words or phrases of a character used too frequently. Furthermore, an indication that a dialogue line is too long to be displayed correctly in the game can save time for mod project members.

The possibility to add optional data to a quest, such as a quest type or a start/end location, enables providing additional assistance functions. For example, a visualization of the quest type distribution across all quests can help preventing designing too many similar quests, or the mapping of quests to locations can ensure an even quest distribution throughout the game world. Furthermore, quest types can be used to provide authors with templates for the given types to speed up their development process. However, instead of restricting authors to existing quest templates of a game, novel quest types can be designed to create a growing quest type framework for a mod. Authors may specify parameters of a type necessary for the template, from which a quest with several dialogues and events is then generated. This can also give new authors an idea of how quests can be structured.

5 EVALUATION

We implemented our concepts and evaluated Questmanager within a moderated remote expert user study that involved 11 unpaid and voluntary participants between 24 and 33 years with Ø 29.0 and SD 2.9. Their experience in story- and quest-writing as well as experience in RPG modding were asked using a ten-point scale. With Ø 7.3 and SD 2.3 (writing) and Ø 7.2 and SD 3.5 (modding) we classify them as expert users. Five participants stated to be part of a permanent mod team.

The user tests included seven tasks (T1-T7) to be accomplished by our participants using Questman-

ager. Finally, we asked our participants to fill out a questionnaire and evaluated four aspects with it:

- [A1] *Quality Enhancement*: Which functions can be used to enhance the quality of quest design?
- [A2] *Efficiency*: How is efficient creation of dialogues achieved?
- [A3] *Clarity*: How can the overview of a story and its components be ensured?
- [A4] *Dialogue Graph*: How well is the developed graph suited for representing and creating interactive dialogues in the modding domain?
- [A5] *Product Character*: The product character (Hassenzahl, 2018) is a measure incorporating both pragmatic and hedonic qualities.

A5 was measured using the standardized abbreviated AttrakDiff questionnaire (User Interface Design GmbH, 2021). A single user test was conducted within the time frame of 1 hour.

Figure 3 (left) shows the value distribution of the questions answered using the 7-point scale. The box plots show that all mean values lie above the neutral value of the scale (3) except for Q9 and Q17. With regard to the value distributions, Q9, Q17, Q19 and Q25 show the greatest deviations across the entire scale. Outliers are present at Q5, Q19, Q20, Q23, Q24, and Q25. We performed Wilcoxon signed-rank tests (Wilcoxon and Wilcox, 1964) on the questions to analyze how Questmanager was rated by our participants relative to a hypothetical neutral rating. With a threshold for statistical significance of 5%, all tests except for Q9 and Q17 show statistically significant differences.

Figure 3 (right) shows the distribution of values aggregated for A1-A4. Again, all mean values lie above 3. A2 shows the highest deviation ranging from 4 to 6. Outliers are present for A1, A2, and A3. We also performed Wilcoxon signed-rank tests against a neutral score for A1-A4. All tests confirm statistically significant differences.

The comments and observations were used to obtain qualitative data and were assigned to A1-A4. Regarding A1, it became clear during the tests that the optimizations suggested on the overview page were perceived as valuable. Two participants mentioned that the identification of unfinished quests, characters without a quest reference, and dialogues that cannot be conducted due to dependency conflicts would also be valuable assistance functions. Concerning A2, we observed that the function to create a quest based on a template could be used by all participants without any problems. One participant proposed offering ‘frequent rewards like gold’ as a favorite in the quest to

speed up the creation process. Another one wanted custom templates to be created, but at the same time expressed a caution about seeing ‘100 [generic] fetch quests in front of you’ with this feature. Regarding A3, we noticed that only six participants were intuitively able to create a quest event. Here, one participant suggested offering an interactive tutorial to explain Questmanager’s own concepts. During testing, it was noticeable that the concept of indirect dependencies was not correctly understood by nine participants and therefore had to be explained again verbally for the task. The dependency graph was highlighted as a particularly positive feature by three participants in Q26. To this end, we observed that one participant used it extensively for orientation during T1. Regarding A4, the necessity of integrating drag-and-drop for rearranging elements in the dialogue graph was frequently highlighted. As an additional feature idea, one participant mentioned a thumbnail view of the dependency graph right next to the dialogue graph to be able to switch between dialogues faster.

We analyzed the results of the AttrakDiff questionnaire to determine the product character (A5) of Questmanager. The aspects pragmatic quality (pq), hedonic quality (hq), and attractiveness (att) are illustrated in Fig. 4. It shows that Questmanager’s attractiveness was rated highest, and its pragmatic quality was rated lowest. The word pairs ‘bad/good’ and ‘impractical/practical’ (Fig. 4 right) were rated highest with values of 5.1 and 4.9, whereas the word pairs ‘complicated/simple’ and ‘confusing/clearly structured’ were rated lowest with values of 3.7 and 4.2.

Overall, the results of the evaluation show that storywriters from the modding field could successfully use Questmanager to design quests. The functionalities for improving the quality of quest design (A1) were rated higher than a neutrally rated quest management application, taking into account the mean values and statistical significance. None of the functions regarding A1 were criticized in Q27. However, suggestions were made for further automatic checks (Q22).

The approaches to increase the efficiency of creating interactive dialogues (A2) were rated lower in relation to the other aspects. However, the results show that the efficiency is still significantly higher than for a neutrally rated quest management program. While the creation using the graph with an initially automated positioning was perceived as easy (Q3), our approach of arranging elements exclusively automatically was perceived restricting (Q9-10). Accordingly, the option of individually structuring the graph elements of a dialogue should be available. The generation of a quest based on a template was found helpful (Q13) and can be optimized by the mentioned sug-

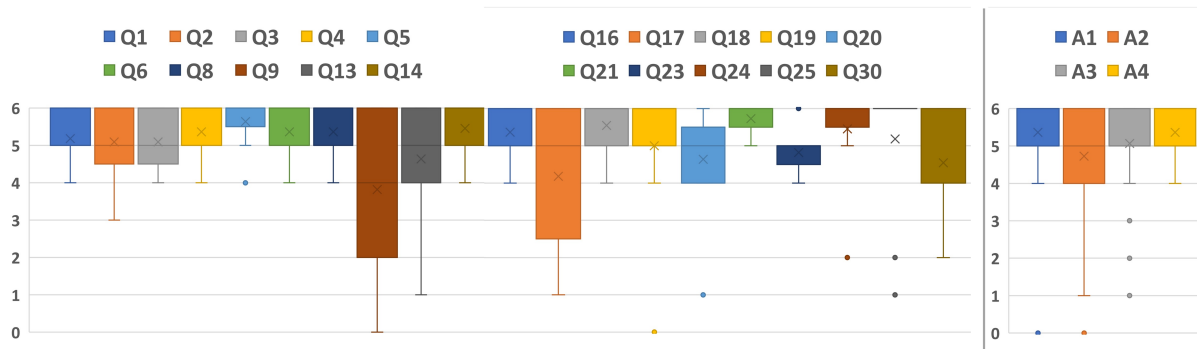


Figure 3: Box plots for the 7-point scale questions and the aggregated aspects 1-4.

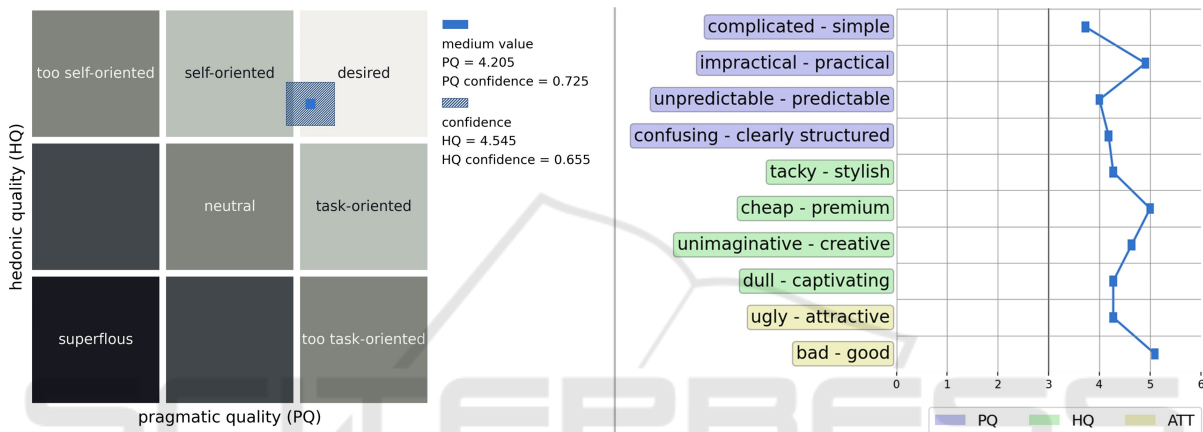


Figure 4: AttrakDiff evaluation: Portfolio representation (left) and description of the word pairs (right).

gestions. The recommended feature of using existing game-scripts as a basis for an import function to create a quest manager project can offer a benefit for mod projects already in development.

Concerning the clarity (A3), the quest structure we proposed was rated useful (Q2). Being able to manage characters, items, and locations was found an elementary part of our quest-centric approach, as Q16, Q26 and the desire for further capture options (questrelevant monsters, lore) indicate. Q17's results confirmed the usefulness of the manually creatable folder structure for quests and entities. This concept can contribute to a better visibility by an own sorting possibility instead of the currently used alphabetical sorting. Furthermore, the automatically generated links in dialogue lines were perceived helpful (Q20), but the current algorithm, which only considers character and location names, is potentially error-prone. This function should be optional and the concept should be extended by the possibility to link entities manually. Q24 shows a high willingness to specify optional information to improve clarity. This indicates that further thought should be given in this direction.

The usefulness of the graph-based representation for interactive dialogues is also supported by the positive evaluation of A4. The concept of the dependency graph, which is new to this area, was rated one of the most valuable aspects of Questmanager, as Q14, Q26 and the qualitative date suggests. At the same time, it was rated one of the most complex aspects of Questmanager, demonstrating the inability of most participants to understand the concept of indirect dependencies without a verbal explanation or to resolve circular dependencies correctly around it. Aspects such as existing dependency conflicts, the quest affiliation of dialogues unrelated to the quest, and completely independent dialogues more visually obvious should be made more visible to guide our mod authors. Overall, the proposed features to extend the functionality suggest that there is still more potential in this concept. Concerning the dialogue graph, Q7 provided only few suggestions for further element types, indicating it contains all major functions necessary for the quest design in the modding domain.

The AttrakDiff results indicate that the proposed UI for our quest-centric concepts was well received. Some mentioned usability optimizations, also re-

flected in the lower rated pragmatic quality. However, despite mentioned improvements, all participants could imagine using Questmanager for a future mod project (Q24).

Finally, we derive that our quest management application can support storywriters with our quest-centric approach, providing context-specific views and automatic optimization suggestions. Questmanager provided our participants with a clear entity management and the dependency graph proved to be valuable after initial understanding. Furthermore, the graph-based display of dialogues was found to be clearer than continuous text representations.

6 CONCLUSION AND FUTURE WORK

In this work, we investigated in the little-researched RPG-modding domain. We pointed out requirements and thereupon introduced quest-centric authoring concepts that support in designing and managing quests. We also showed how to utilize graph-based visualization techniques for modeling interactive dialogues and particularly dependencies between quest components of a story. We implemented our concepts within Questmanager – a lightweight authoring tool. With the results of our expert user study, we conclude that our concepts can support storywriters in the modding domain and also pointed out aspects that should be included in future applications.

For future work, we see potential to apply our concepts and Questmanager itself outside of the modding domain, for example, for professional story writing of interactive media such as computer games. However, future work should also include novel ideas such as the notion of game-specific modding aspects. For example, we included a JSON configuration file in a Domain Specific Language (Fowler, 2010) to define actions and conditions that the base game actually does support. However, how can generic dialogues be designed and exported to fit into the format of specific games? How can the need for external documents be eliminated by describing domain-specific entities directly in an authoring application for mods? How can existing lore of a game be included in an authoring tool?

REFERENCES

- Aarseth, E. (2005). From hunt the wumpus to everquest: Introduction to quest theory. In *International Conference on Entertainment Computing*, pages 496–506. Springer.
- Arcweave OU (2021). Arcweave. <https://arcweave.com/>. [Accessed: December 23, 2021].
- Articy Software GmbH & Co. KG (2021). Articy draft. <https://www.articy.com/en/>. [Accessed: December 23, 2021].
- digiwombat (2021). Talk maker deluxe. <https://github.com/digiwombat/TalkerMakerDeluxe>. [Accessed: December 23, 2021].
- Fowler, M. (2010). *Domain-specific languages*. Pearson Education.
- Hassenzahl, M. (2018). The thing and i: understanding the relationship between user and product. In *Funology 2*, pages 301–313. Springer.
- Horst, R. (2021). *Virtual Reality Nuggets - Authoring and Usage of Concise and Pattern-Based Educational Virtual Reality Experiences*. PhD thesis, RheinMain University of Applied Sciences.
- Howard, J. (2008). *Quests: Design, theory, and history in games and narratives*. CRC Press.
- Hurel, P.-Y. (2016). Playing rpg maker? amateur game design and video gaming. In *DiGRA/FDG '16 - Proceedings of the First International Joint Conference of DiGRA and FDG*.
- Interactive Fiction Technology Foundation (2021). Twine. <https://twinery.org/>. [Accessed: December 23, 2021].
- LearnBrite (2021). Chat mapper. <https://www.chatmapper.com/>. [Accessed: December 23, 2021].
- radmatt (2021). Dialogue designer. https://store.steampowered.com/app/1273620/Dialogue_Designer. [Accessed: December 23, 2021].
- Sektenspinner (2009). Diadepp. <https://forum.worldofplayers.de/forum/threads/623230-Dialog-Tool-DiaDepp>. [Accessed: December 23, 2021].
- Smith, G., Anderson, R., Kopleck, B., Lindblad, Z., Scott, L., Wardell, A., Whitehead, J., and Mateas, M. (2011). Situating quests: Design patterns for quest and level design in role-playing games. In *International Conference on Interactive Digital Storytelling*, pages 326–329. Springer.
- Sotamaa, O. (2003). Computer game modding, intermediality and participatory culture. *New Media*, pages 1–5.
- Sotamaa, O. (2010). When the game is not enough: Motivations and practices among computer game modding culture. *Games and Culture*, 5(3):239–255.
- User Interface Design GmbH (2021). Attrakdiff questionnaire. <http://attrakdiff.de/index-en.html>. [Accessed: December 23, 2021].
- Veloso, C. and Prada, R. (2021). Validating the plot of interactive narrative games. In *2021 IEEE Conference on Games (CoG)*. IEEE.
- Wilcoxon, F. and Wilcox, R. A. (1964). *Some rapid approximate statistical procedures*. Lederle Laboratories.