Florian Mehm, Stefan Göbel, Ralf Steinmetz:
Authoring of Serious Adventure Games in StoryTec. In: Stefan Göbel, Wolfgang Müller, Bodo Urban, Josef Wiemeyer: *E-Learning and Games for Training, Education, Health and Sports*, no. LNCS 7516, p. 144-154, Springer, September 2012. ISBN 978-3-642-33465-8.

Authoring of Serious Adventure Games in StoryTec

Florian Mehm, Stefan Göbel, Ralf Steinmetz

Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
{florian.mehm, stefan.goebel, ralf.steinmetz}@kom.tu-darmstadt.de

Abstract. Adventure games, being characterized by a strong focus on narrative, interaction with virtual characters and solving puzzles, are a genre that can be used for Serious Games, especially those in the domain of educational games. However, the creation of a serious adventure game, similar to other game genres when being used for serious purposes, leads to a set of new problems. As new team members (such as domain experts and pedagogues) are added to teams, the common vision of the game can get lost, communication overhead is added and collaboration is harder to achieve. We propose that an authoring tool that integrates the tasks of the various groups found in serious adventure game development into one tool can help in mitigating these problems. We demonstrate this with the authoring Tool StoryTec that was used in re-authoring an existing commercial educational adventure game in StoryTec. Additionally, the integration of an open-source engine for third person adventure games in the authoring tool is shown.

In order to achieve this fully integrated authoring tool, we analyze the current game development processes of adventure games and the state of the art of adventure game authoring tools or editors. These processes are mirrored in the workflows that are captured in StoryTec, structuring the interaction and communication especially between game programmers and designers as well as domain experts. Based on a model for game content, the authoring tool StoryTec is described. The results of one usability and one focus group study show the applicability of the presented approach.

1 Introduction

Serious Adventure Games strive to combine the positive aspects of adventure games with a serious purpose such as education. The adventure game genre is defined by the focus on slow gameplay with comparably few action-laden or timed sequences as compared to other game genres. A strong narrative is used to embed puzzles [6]. Gros [7] established the adventure game as one of seven game genres for educational games. Using this genre for Serious Games has several beneficial aspects. For example, educational content (in the case of educational games) can be transported by means of the strong narrative. It can be embedded in the game world and puzzles by means of the game design and players can take their time in assimilating the presented knowledge due to the absence of time limits. On the other hand, gamers have lower expectations on the graphics or effects of adventure games as compared to other genres in which games have to push the limits of current hardware to be accepted, resulting in lower costs for assets and technology. Furthermore, adventure games follow a very common interaction structure and are similar to another concerning gameplay (while content can vary between realistic and fantastical and all literary genres). Therefore, tools for their production exist (see section 2) which aids in the production of the game.

The main field of application of serious adventure games has been in the educational sector, however, several adventure games have also been developed for other purposes, for example advertisement. The genre has received the attention of educators for building educational games both during their initial era of commercial success (see [5]) as well as in recent times [9].

The addition of a serious purpose to a game incurs higher production costs compared to non-educational/purely entertainment-focused games. This results from the additional team members with specialized backgrounds who augment a regular game development team involving game designers, artists and programmers. For an educational game, domain experts as well as pedagogues are added. This addition of team members and roles can then lead to communication problems involving different tools and processes, differing nomenclature or expectations.

We propose to offer a unified authoring tool, integrating the roles sketched here and identified in the following section in one authoring tool. This allows processes to be visible to all team members and collaboration to take place.

In the remainder of this paper we describe our approach to authoring serious adventure games. Sections 2 focuses on the current state of the art of (serious) adventure game production and section 3 on authoring tools for this genre. The concept and implementation of the authoring tool, StoryTec, are described in Section 4, followed by the results of two evaluations of the tool in Section 5.

2 Serious Adventure Games

First examples of serious adventure games stem from the early days of the genre, see for example [5]. In the following, we provide an analysis of the production process of adventure games. For this purpose, the authors interacted with a German developer of educational adventure games and carried out a literature survey, including [1], [8] and [13].

The production of an adventure game is initiated with a phase of game design. During this phase, the game's narrative as the means of binding the puzzles of the game together is written and fleshed out with the help of concept artists. Apart from the story, the main work at this stage is the definition of puzzles the player has to solve and their placement in the game. Since the adventure genre has strong standards concerning gameplay, game designers need not define completely new interaction methods but rather follow the existing conventions. For example, the way in which the player's inventory is managed is very similar in most adventure games, as is the navigation of characters (e.g. via pointing and clicking a mouse cursor).

During the design stage, domain experts for the domain of the Serious Game to be developed should be included, since the game's narrative and puzzles should conform with the serious purpose it is developed for. For example, the narrative could already embed domain knowledge, e.g. by making it historically accurate in the case of an educational game. Puzzles as the main form of interaction for the player should be linked to the purpose of the game, e.g. by reflecting real-world practices that a player should internalize while playing the game.

After this first phase of game development, the actual production of the game is carried out. During this stage, the game's engine is either created or re-used from another source (a previous project, a commercial or freely available engine). Using the engine, programmers implement the game mechanics as specified during the game design phase. Assets (such as virtual character models, images, GUI elements or sounds) are produced based on concept art and the game design and the integrated in the game. This step can involve programmers again, since content has to be integrated with the logic of the game. Adventure game engines commonly use scripting languages for programming at this level. This calls for a user trained in the tools necessary who is able to program in the scripting language of them.

Quality Assurance (QA) is carried out in parallel to the development of the game, with the goal of reducing technical and logical or content-related problems. For serious games, QA also involves assuring that the game is suitable for the serious purpose for the chosen target audience.

3 Authoring Tools for Adventure Games

Authoring tools are used for editing and composing content in various fields, including multimedia computing [4] and e-learning ([3]). They provide simple and customized interfaces that allow authors to work with content and publish it in various forms without author intervention (e.g. when publishing an e-Learning course as an interactive web site).

Several general-purpose authoring tools or game editors for adventure games are available. The free Adventure Game Studio (AGS)[1] has generated a large community of independent adventure game developers. The Visionaire Studio[2] was used in the development of the educational adventure game Winterfest [9].

Authoring tools for adventure games with a serious purpose that include functionalities specifically for this purpose are not found often in the literature. The e-Adventure authoring tool [15] provides the possibility to export a game created with the tool to a Learning Management System [2] and allows authors to create in-game books to transport knowledge textually.

[1] http://www.adventuregamestudio.co.uk
[2] http://www.visionaire-studio.net/cms/adventure-game-engine.html

4 Authoring Adventure Games using StoryTec

Two versions of StoryTec are described in this paper. One version was used to re-author an existing first-person commercial educational adventure game, using original assets and providing all interaction templates (including a puzzle, see figure 2) used in the original game. The second version recently integrated the Wintermute[3] adventure game engine, allowing the creation of 2D third person adventure games (see figure 4). Therefore, these two variants cover the most common variations of 2D adventure games.

4.1 Game Model

In this section, the model we use for capturing the content of serious adventure games is described. This model is based on the atomic unit of scenes, which are connected via links referred to as transitions. Objects are placed inside scenes to realize scenery, characters or logical objects such as variables, and parameters used to control properties of scenes and objects. Game logic is be configured by high-level commands, referred to as actions, which can be organized sequentially or branching and be conditionally executed. The execution of a sequence of actions is triggered by an event from the game environment, termed a stimulus.

The concept of a scene is chosen as the atomic unit of a game's structure in the model established here. In theater or movies, a scene comprises a small part of the overall narrative, delimited in space by the set or the theatrical scenery, with a set of fixed scene items (props) and a set of actors. Similarly, in this concept, a scene is intended to model a small part of the overall game, containing all objects and logic to capture the interaction of the player while the scene is experienced by the player.

"Object" is the umbrella term for all objects that are visible to the player in the game (images, 3D models, virtual characters, …), other media elements (e.g. sounds) or interactive elements (buttons, text fields, …). Apart from these objects the player interacts with directly, we have chosen to add also logical and control objects under the term "object". These objects include mainly variables (for keeping track of information about the game state) and other objects not directly perceivable by the player such as 3D cameras for a 3D game environment.

In order to model the connections between scenes, the concept of a transition is introduced. A transition is a directional relation between two scenes, indicating that the game can jump from the scene at which the transition starts to the scene at which it ends. Therefore, by adding transitions to the hierarchically organized scene structures, we arrive at a story model that is realized as a directed graph structure.

Game logic is handled by the concepts of stimuli (high-level events that are triggered in the game) and actions that the author specifies to occur when the event is triggered. Actions are aligned sequentially and can be made to be executed conditionally based on the state of the game world or variables.

---

[3] http://dead-code.org/home/

The intricacies of interaction with the player are handled by the concept of interaction templates (described in section 4.2)

For establishing a layer of abstraction above the actual target system (game engine), a “Story Engine” component is used. This engine parses files in which games are encoded according to the model presented here and sends high-level commands based on them to the actual game running on the game engine. Therefore, the technical development of the game is carried out in the game engine, whereas the content production for the game is carried out in the authoring tool by filling configurable game templates. In the case of integrating the Wintermute engine, the Story Engine was connected as a component to the engine, implementing details of actions in terms of Wintermute objects.

4.2 StoryTec

StoryTec integrates the work of the game development roles as described in section 2 into one unified authoring tool. Two possibilities for collaboration in game development teams [16] are acknowledged by this concept. It provides the possibility of a team-wide uniform vision of the game as all members of the team are able to see the whole structure of the game as it evolves and work directly on the game. Furthermore, it removes the bottleneck introduced by the need for programmers to be closely involved in most parts of the game development. The problem created by a plethora of tools being used is alleviated by using only one tool. Figure 1 shows the mapping of tasks from several groups into components of the authoring tool.

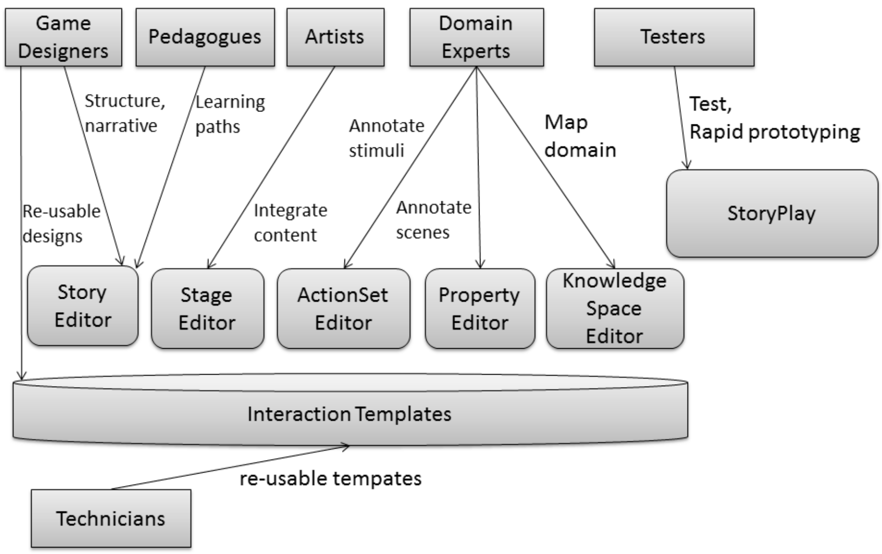

Fig. 1. Some of the possible mappings between user groups and the components of the described authoring tool. Note that not all combinations are shown, for example, game designers could also use StoryTec and StoryPlay in conjunction for storyboarding and rapid prototyping.

By mapping all users into the authoring tool and its components, the authors all share a common work environment and work with the same version of the game instead of everyone working in different tools (text editors, spreadsheets, programming IDEs etc.). The authoring tool then resembles a “virtual blackboard” on which all team members can see their activity along with that of other authors.
The modular authoring tool is composed of several interlinked editors. The first editor to be used commonly by authors is the Story Editor. This editor is used to configure the high-level structure of the game, by breaking down the whole game into a set of scene and defining the transitions between scenes. Figures 2 and 4 show two instances of the StoryTec GUI including the Story Editor.

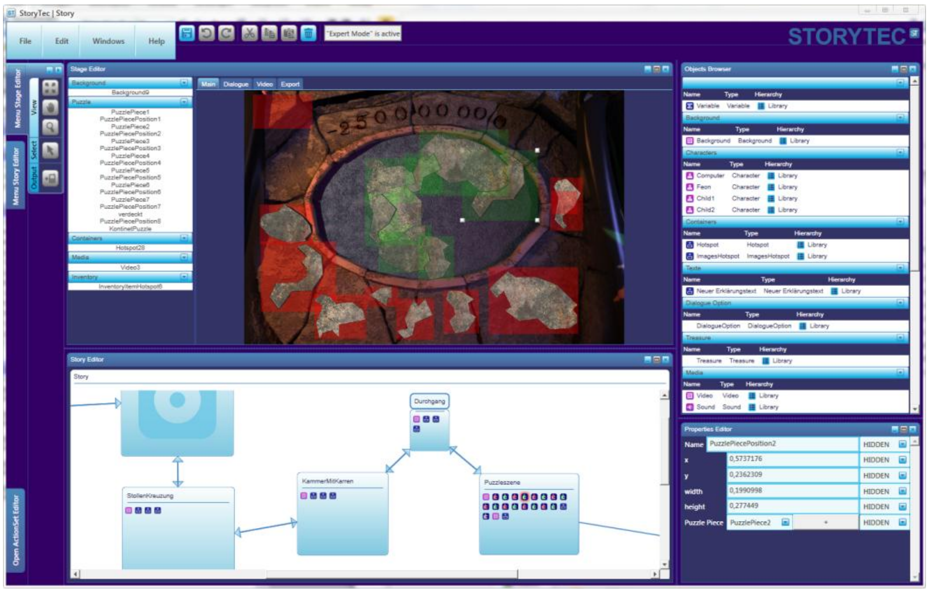

Fig. 2. StoryTec used for authoring a puzzle in an educational adventure game. The editors shown in this figure and figure 4: Stage Editor (upper left), Story Editor (lower left), Objects Browser (upper right) and Property Editor (lower right)

In the Stage Editor, authors work on one specific scene’s content in a WYSIWYG fashion. By dragging and dropping, objects from the Objects Browser are instantiated. The central concept behind the Stage Editor is that of interaction templates. Interaction templates are used to encapsulate the actual interaction of the player with the game, in essence the gameplay of the game. An example is a puzzle game, in which the interaction lies in the user dragging puzzle pieces to their correct location and the game signaling whenever the task is completed. An interaction template is programmed by game programmers and then provided in the authoring tool to be filled with arbitrary content by authors. A fitting counterpart from the world of analogue games are “frame games” [14].

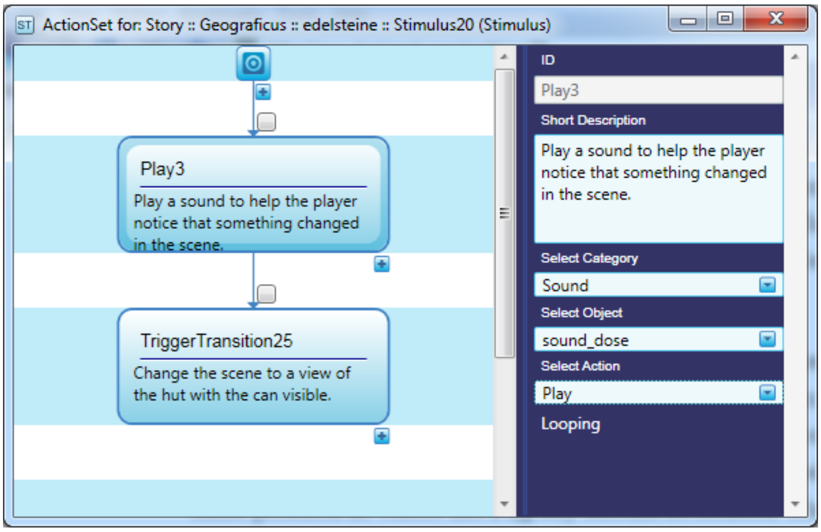

Fig. 3. The ActionSet Editor for configuration of control flow during the game

Actions as described in the previous section are configured in the ActionSet Editor (see Figure 3). Actions are aligned sequentially as well as branching based on runtime conditions. Each action has further parameters, such as the target in the case of a character movement action or the text in a dialogue action.

4.3 Rapid Prototyping, Testing

Rapid Prototyping is the practice of iterating over several versions of a software project such as a game quickly and incorporating the resulting feedback quickly. StoryTec is accompanied by the player application StoryPlay [10], which allows the simultaneous play of a StoryTec created game and the analysis of context information about the game flow and the internal data of algorithms (e.g. concerning adaptation) as well as the creation of log data, which can be evaluated later on. Authors can test game prototypes quickly and react to feedback. Musil et al. [11] call for such an agile approach in order to improve the production of games.

StoryPlay is split into a player aspect and an evaluation aspect. In the player aspect, the game's current state is shown in a playable version with which an evaluator or game tester can interact with. During play, log data and visualizations are created, which can be displayed in the evaluation aspect. Visualizations include the state of variables, the history of previous player choices and the state of internal variables such as the player model.

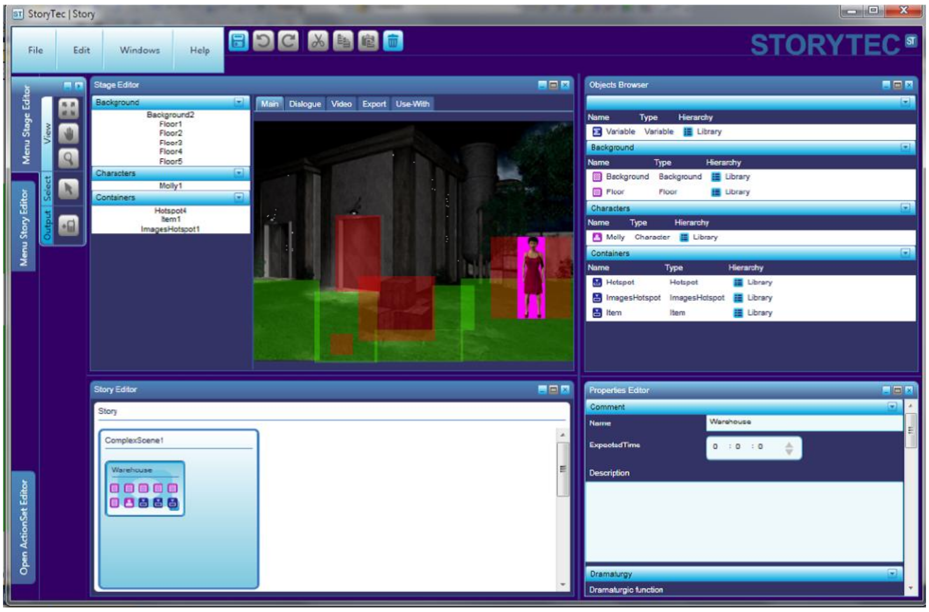

Fig. 4. StoryTec version targeting the Wintermute adventure engine. Red areas indicate interactive areas and the main character, green areas are walkeable by the character.

5 Evaluation

In the following, we describe the methodologies and results of two evaluation studies performed on StoryTec.

5.1 Usability Study

The first study was carried out in the form of a usability study involving students recruited from a lecture on Serious Games (N=26, one female). The mean age of the participants was 25.2 years with a standard deviation of 3.71 years. 13 participants stated to have previous experience with authoring tools. The tools that participants referred to included Unity 3D, Blender, 3ds Max, Audacity, Novelty, RPG Maker and Photoshop. The participants' rating of their knowledge of these tools varied strongly, on a scale from 1 to 7 a questionnaire resulted in a mean value of 3.69 (higher values indicate more perceived knowledge) with a high standard deviation of 1.70.

The test was carried out with groups of three participants. In order to motivate the participants to talk about their experience with the software in the style of the "Thinking Aloud"-method, they were given one of three roles, separating the tasks of reading and communicating the instructions ("reader"), executing them on the PC used ("performer") and observing the task and giving input ("observer"). The task consisted of restoring missing elements from a part of an educational adventure game modeled in

StoryTec. By design, the users could look up missing elements in the remaining parts of the game and understand concepts based on this.

A questionnaire based on the usability standard ISO 9241-10 was used to allow the participants to rate their experience with the tool. Table 1 shows the mean values and standard deviation of answers grouped in the seven basic principles of the standard.

Basic principle	**Mean value**	**Standard deviation**
Suitability to the task	4.74	0.88
Self-description	3.51	0.93
Controllability	5.48	0.77
Conformity with user expectations	4.55	1.06
Error tolerance	3.42	0.80
Suitability for individualization	4.42	0.72
Suitability for learning	5.14	0.78

Table 1. Results of the usability questionnaire (Values range from 1 to 7)

For the purpose of comparing StoryTec with other software tools, measurements from a test on 41 software tools with 1265 users [12]. Comparing the results of the study presented here and that by [12], StoryTec is rated average in all but two criteria (Self-description and Error tolerance).

Participants were asked to rate their perceived knowledge of StoryTec based on their experience in the test. This resulted in a mean value of 2.64 (SD = 1.44), lower than their knowledge of other authoring tools. However, this perception resulted from the very short initial exposure of less than one hour to StoryTec.

5.2 Focus Group Study

Subsequently, a focus group study of StoryTec with the goal of ascertaining whether StoryTec could be used in the context of a game development studio was carried out at a German development studio of educational adventure games. Three participants (aged 31, 37, 46, one female, three male) involved in the game development at the studio took part in this study. The participants were given similar tasks as during the first evaluation, adapted for only one person. Afterwards, the participants took part in individual guided interviews, including questions about the applicability of StoryTec in their fields of work (including game design, graphics and programming). The interview was based on a set of 15 questions, again drawn from and categorized into the criteria of ISO 9241-10. As a result, it can be noted that the experts were interested to hear about the development of StoryTec and, keeping the prototypical state of the system shown to them, they noted that a full version might be used for game development. In the state it was demonstrated to them, they regarded it as suitable for storyboarding and prototyping.

One question of the guided interview asked the participants to judge the applicability of StoryTec in the use case of a teacher using it for the creation of an educational game for class use and for game designers. All participants proposed that StoryTec in the state it was shown to them in was more suitable in the first use case that the second. One participant noted that the simple programming approach and the choice of the genre point and click-adventure were beneficial for this user group, another pointed out the necessity of a German translation for use in Germany. For game designers, the common judgment of the participants was that the present possibilities in StoryTec were yet too limiting on game designers. They noted that they perceived it as a good tool for creating game storyboards (requiring note and text capabilities, as one participant stated) and that it could be well suited in the presented state for hobbyist game designers.

A second question concerned the time that the participants estimated they would require to fully understand and use StoryTec. The answers to this question ("two hours" in two cases, "several days spent building more complex games") indicate that the participants' insight during their short initial exposure to StoryTec allowed them to understand the main concepts of the authoring tool.

Areas that required improvement as indicated by all test participants were the intended usage of the Story and ActionSet Editors, the details of which appeared unclear to the participants. Specific feedback on these editors has been incorporated in newer versions of StoryTec, for example offering more functionality in context menus for authors searching context-related commands there.

6 Conclusion

In this paper, we have described the genre of adventure games as a possible genre to be used for Serious Games. On the one hand, adventure games have very positive properties, especially when used in the educational sector. On the other hand, adventure games, just as other genres of games, lead to a set of typical game development problems.

The basic game model on which StoryTec is constructed was shown, followed by a description of the authoring tool itself. It maps the tasks of the typical members of adventure game development into one unified authoring tool, allowing collaboration and a project-wide uniform vision of the game by using the same interface for all involved users. The practical implementation of two versions of StoryTec has been shown, covering the two major strands of 2D adventure games (first and third person games). For the development of third person games, an existing engine (Wintermute) has been leveraged, thereby extending the reach of StoryTec on the one hand and demonstrating the viability of the software engineering approach of a Story Engine working on a structured game content model on the other hand.

Compared to existing tools (especially intended for serious adventure games as in the case of e-Adventure [15]), StoryTec allows a wider range of game tasks to be added due to the extensibility provided by the interaction template concept. Therefore, a workflow for adding tasks such as the puzzle shown in Figure 3 exists.

The results of two studies were shown, indicating the usability of StoryTec[4] in the domain of educational adventure games.

References

1. Amory, A.: Building an Educational Adventure Game: Theory, Design and Lessons. Journal Of Interactive Learning Research. 12, 2/3, 249-263 (2001).
2. Del Blanco, A. et al.: Easing Assessment of Game-based Learning with e-Adventure and LAMS. ACM Multimedia Workshop. 25-30 (2010).
3. Brusilovsky, P.: Developing adaptive educational hypermedia systems: From design models to authoring tools. Information Sciences. 377-409 (2003).
4. Bulterman, D.C.A., Hardman, L.: Structured multimedia authoring. ACM Transactions on Multimedia Computing Communications and Applications. 1, 1, 89-109 (2005).
5. Cavallari, B. et al.: Adventure games in education: A review. Australian Journal of Educational Technology. 8, 2, 172-184 (1992).
6. Dickey, M.D.: Game Design Narrative for Learning: Appropriating Adventure Game Design Narrative Devices and Techniques for the Design of Interactive Learning Environments. Educational Technology Research & Development. 54, 3, 245-263 (2006).
7. Gros, B.: Digital Games in Education: The Design of Games-Based Learning Environments. Journal of Research on Technology in Education. 1, 40, 23-38 (2007).
8. Hodgson, P. et al.: Managing the Development of Digital Educational Games. DIGITEL 2010 Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning. pp. 191-193 Ieee (2010).
9. Malo, S., Müsebeck, P.: Winterfest - An Adventure Game for Basic Education. Workshop-Proceedings Informatik 2 : 3rd Workshop on Inclusive E-Learning. pp. 61-65 Franzbecker, Hildesheim (2010).
10. Mehm, F. et al.: Bat Cave: A Testing and Evaluation Platform for Digital Educational Games. Proceedings of the 3rd European Conference on Games Based Learning. Academic Conferences International, Reading, UK (2010).
11. Musil, J. et al.: Improving Video Game Development: Facilitating Heterogeneous Team Collaboration Through Flexible Software Processes. In: Riel, A. et al. (eds.) Proc 17th European Systems and Software Process Improvement and Innovation Conference EuroSPI 2010. pp. 83-94 Springer (2010).
12. Prümper, J.: Der Benutzungsfragebogen ISONORM 9241/10: Ergebnisse zur Reabilität und Validität. Software-Ergonomie `97 – Usability Engineering: Integration von Mensch-Computer-Interatkion und Software-Entwicklung. pp. 253-262 Teubner, Stuttgart (1997).
13. Sommeregger, P., Kellner, G.: Brief Guidelines for Educational Adventure Games Creation (EAGC). In: Sugimoto, M. et al. (eds.) Fourth IEEE International Conference On Digital Game And Intelligent Toy Enhanced Learning. pp. 120-22 IEEE Computer Society (2012).
14. Stolovitch, H.D., Thiagarajan, S.: Frame Games. Educational Technology Publications, Englewood Cliffs, N.J . (1980).
15. Torrente, J. et al.: Introducing Accessibility Features in an Educational Game Authoring Tool: The <e-Adventure> Experience. 2011 IEEE 11th International Conference on Advanced Learning Technologies. pp. 341-343 IEEE (2011).

[4] Available for an open community at http://www.storytec.de

16. Tran, M.Q., Biddle, R.: Collaboration in serious game development. In: Kapralos, B. et al. (eds.) Proceedings of the 2008 Conference on Future Play Research Play Share Future Play 08. pp. 49-56 ACM Press, New York (2008).