

Instructor-oriented Authoring Tools for Educational Videogames

Javier Torrente¹, Pablo Moreno-Ger¹, Baltasar Fernández-Manjón¹, José Luis Sierra¹
¹*Universidad Complutense de Madrid. Facultad de Informática. C/ Profesor José García
Santesmases s/n. 28040 Madrid, Spain
{jtorrente,pablom,balta,jlsierra}@fdi.ucm.es*

Abstract

Digital games are becoming a rising trend in e-Learning due to their potential educational benefits. However, their application is hindered by issues such as their high production costs and the difficulty of getting instructors involved in the development process, a required aspect in order to achieve educational value. In this paper we propose to tackle those issues by developing new, instructor-oriented, authoring tools for educational videogames. These principles are carried out in the <e-Adventure> platform: a complete instructor-oriented authoring toolkit for educational point-and-click adventure videogames.

1. Introduction

Digital game-based learning has been pointed out as an attractive complement to traditional e-learning approaches (i.e. online HTML-based Learning Management Systems). Whilst typical web-based learning tools show high dropout rates due to frustration and lack of motivation [1], videogames manage to grip the attention of all kind of people for hours, as they are fun, motivational and challenging. In addition, they show very interesting educative traits as they are immersive, provide a very short feedback cycle as long as the player freely explores the domain of study and, finally, promote both collaboration and competition [2, 3].

However, the introduction of videogames in educational contexts is not free of barriers. In our opinion, the foremost ones are the astronomical development cost of commercial games along with the need of getting instructors actively involved in the development process.

In this paper, we propose to deal with these shortcomings by providing instructor-oriented authoring tools. These tools should not only be focused on the abstraction of the programming tasks, leaving the technical features of the game to programmers and media designers, but also on supporting those aspects

that are specific of the educational domain, usually not contemplated in toolkits for commercial videogames.

2. The need of specific authoring tools for educational game design

The use of videogames in online education has multiple benefits, as stated thus far. However, the cost for developing a custom educational videogame can be impressively high. High-profile commercial videogames often have budgets of several million dollars. As analyzed in [4], a high percent (40% approximately) of the educational game developments active or recently finished in 2005 were expected to cost more than 100.000\$. According to [2], the development costs of “next-generation simulations” are estimated on the 15-30 person-years range. Very few educational budgets could afford such expenditure.

On the other hand, there is no educational value in just entertaining students with a videogame. We consider that obtaining valuable educational content requires involving effectively instructors in the development process.

Taking into account these two premises, user-friendly authoring tools seem to be irreplaceable for educational videogames development, as they abstract the technical aspects and reduce considerably the costs. A study of the existing game authoring tools reveals that none of them were designed specifically for educational games. Although projects such as *The FPS Creator*¹, *Game Maker*² or *Mission Maker*³ provide intuitive utilities to create low-cost games, none of the alternatives in the market have been specifically developed for their use by instructors, as they do not support the inclusion of pedagogically relevant aspects. Thus educational game development should take a more instructor-oriented approach, moving away from commercial developments.

¹ <http://www.thegamecreators.com/>

² <http://www.gamemaker.nl>

³ <http://www.immersiveeducation.com/missionmaker/>

3. Instructor-oriented authoring tools

Most game authoring tools tend to be holistic, addressing aspects concerning many different people and fields (e.g. scripting languages used by programmers to configure game engines; media editors used by media designers to edit the art assets of the game, etc.). The results are high-tech products that can overwhelm a typical instructor, who is not expected to have any particular knowledge on computer science beyond the use of basic office tools.

Educational videogames require authoring tools which only incorporate what the instructor needs to control and customize in the game. In this way, aspects such as programming AI algorithms, or the creation of art assets such as graphics, sound effects or video clips should be designed elsewhere by the corresponding specialists and with an adequate tool support.

In addition to game-specific aspects, these authoring tools should address features specific to the educational domain. Among them, we highlight the need to track and evaluate the activity of the student (assessment), and how to adapt the behaviour of the game in order to fit different ranges of students (adaptation).

Regarding assessment, when games are integrated in educational environments instructors should be able to evaluate the work done by the student and, above all, check if the learning objectives are being attained (i.e. the learning experience is being effective) [5]. For this purpose, the information gathered during the interaction with the game (e.g. time consumed to get each task done) can be exploited for assessment purposes, moving towards assessment of the learning process as opposed to assessment of the knowledge memorized (as it is usual in traditional approaches).

Regarding adaptation to different aspects, such as students' preferences, learning styles or prior-knowledge, it is a desirable behaviour in order to maximize the effectiveness of the learning experience. Under this premise, educational game authoring tools should provide a mechanism to define alternative game behaviours according to the student's profile.

4. Instructor-oriented Authoring Support in <e-Adventure>

<e-Adventure> is an instructor-oriented environment for the development of educational *point-and-click* adventure videogames [6], a genre that has been highlighted because of their suitability for educational applications [7, 8]. An <e-Adventure> game follows the pattern of common adventure games such as the *Monkey Island*TM or *King's Quest*TM sagas, where the

player's avatar interacts with the different characters and objects in a narrative storyline.

4.1. Instructor-oriented abstractions

The <e-Adventure> platform provides a user-friendly editor that allows instructors to define those aspects of the games which are educationally relevant.

One of these aspects is the scene configuration, since it is related to the tasks the learner is expected to complete. For this purpose, a navigation panel is supplied where the elements of the game (e.g. objects, characters, books, scenes, etc.), are edited, accessed and modified, referencing art assets (e.g., the background image of a scene, sound effects, etc.), created by the media designers when needed. Then those characters and objects can be located in a scene by simply clicking on the scene, as Figure 1 depicts.

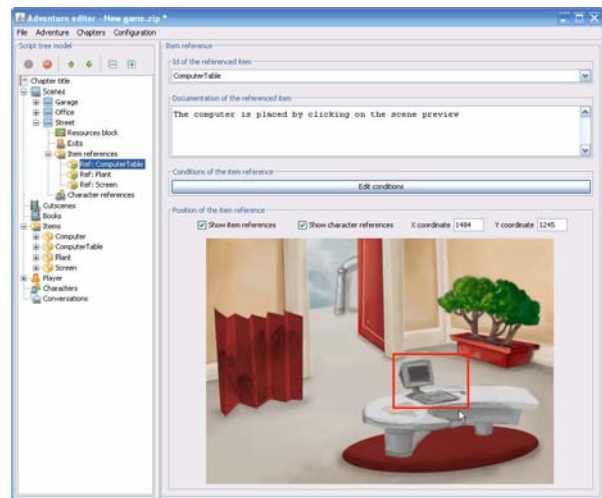


Figure 1. The <e-Adventure> visual editor

The editor also allows including sources of information to be accessible during the game. This is a key point, as educational games must provide knowledge concerning the subject of study. For this purpose, there are two mechanisms at instructors' disposal: conversations with characters and in-game books, useful when large portions of knowledge are to be transmitted, and which can include plain text, images, or even HTML content.

4.2. Education-specific features

The <e-Adventure> platform supports the educational features described in the previous section through its built-in assessment and adaptation mechanisms. Both mechanisms are based on the state-oriented implementation of the platform [9].

In <e-Adventure>, the games include a notion of *state*, represented by a set of flags that can be activated or deactivated during the game. The editor allows authors to specify *assessment rules* that can be associated to different states. The key idea is to identify relevant patterns of behaviour to be matched on the game's state space, and then to associate suitable assessment actions to these patterns. These rules can be created with a user-friendly graph-metaphor, as Figure 2 depicts.

For simplicity, thus far only three types of assessment actions are allowed: (i) association of final grades to end states of the game; (ii) association of partial increments of the total mark when a state is reached; and, (iii) association of time counters between states (i.e. measure the time needed to accomplish a set of tasks). When the game session ends, the a report of the outcome of these rules is automatically generated.

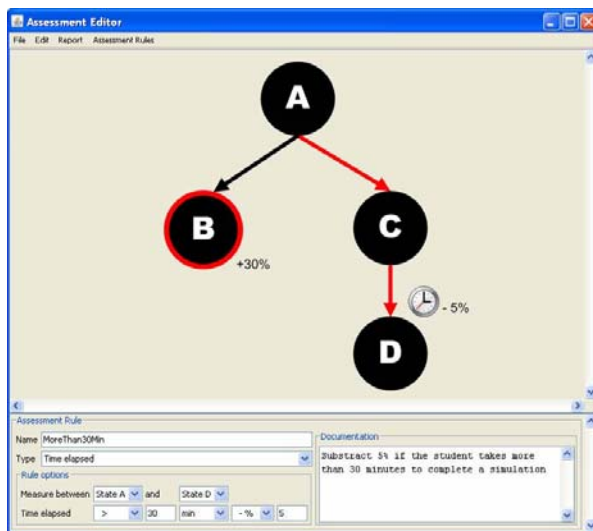


Figure 2. Edition of assessment rules

Similarly to the edition of assessment rules, <e-Adventure> also supports *adaptation rules*, which contain alternative initial states for the flags. The same game can exhibit completely different behaviours depending on specific state transitions, and these transitions are also defined using a graphical editor that allows the instructor to set up the relevant initial states for different student profiles.

5. Conclusions

The benefits of applying videogames in educational contexts have been widely discussed in the literature. However, some barriers must be overcome to generalize their use in online learning: reduce the high development cost and actively involve educators, who usually do not have the proper technological

background. To tackle these inconveniences, we need easy-to-use instructor-oriented tools. These tools will differ from conventional ones in incorporating instructor-oriented abstractions, and in addressing education-specific features.

<e-Adventure> was designed following these premises. The visual editor, along with the tools provided for the edition of assessment and adaptation rules constitute a platform that allows instructors to create games following the principles described in this article.

Acknowledgements

This work has been partially supported by the following project grants: TIN2005-08788-C04-01, FIT-350100-2007-163, TIN2007-68125-C02-01, CAM 4155/2005, UCM 921340, and Santander/UCM PR34/07-15865.

References

- [1] A. Parker, "Identifying Predictors of Academic Persistence in Distance Education," *Journal of the United States Distance Learning Association*, vol. 17, pp. 55-62, 2003.
- [2] C. Aldrich, *Learning by Doing: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences*. San Francisco, CA: Pfeiffer, 2005.
- [3] A. Mitchell and Savill-Smith, C., *The Use of Computer and Videogames for Learning: A Review of the Literature*. Trowbridge, Wiltshire: Learning and Skills Development Agency, 2004.
- [4] D. Michael and Chen, S., *Serious Games: Games that Educate, Train, and Inform*. Boston, MA: Thomson, 2006.
- [5] I. Martínez-Ortiz, Moreno-Ger, P., Sierra, J. L., and Fernández-Manjón, B., "Production and Deployment of Educational Videogames as Assessable Learning Objects," in *First European Conference on Technology Enhanced Learning (ECTEL 2006)*, Crete, Greece, 2006, pp. 316-330.
- [6] P. Moreno-Ger, Martínez-Ortiz, I., Sierra, J. L., and Fernández-Manjón, B., "A Content-Centric Development Process Model," *IEEE Computer*, vol. 41, pp. 24-30, 2008.
- [7] A. Amory, "Building an Educational Adventure Game: Theory, Design and Lessons," *Journal of Interactive Learning Research*, vol. 12, pp. 249-263, 2001.
- [8] R. Van Eck, "Building Artificially Intelligent Learning Games," in *Games and Simulations in Online Learning: Research and Development Frameworks*, D. Gibson, Aldrich, C., and Prensky, M., Eds. Hershey, PA: Information Science Publishing, 2007.
- [9] P. Moreno-Ger, Sierra, J. L., Martínez-Ortiz, I., and Fernández-Manjón, B., "A Documental Approach to Adventure Game Development," *Science of Computer Programming*, vol. 67, pp. 3-31, 2007.