e-Training DS: An Authoring Tool for Integrating Portable Computer Science Games in e-Learning

Roberto Tornero, Javier Torrente, Pablo Moreno-Ger, Baltasar Fernández Manjón,

Universidad Complutense de Madrid. Dpto. de Ingeniería del Software e Inteligencia Artificial. Facultad de Informática.
C/ Profesor José García Santesmases s/n. 28040 Madrid, Spain
{rtornero, jtorrente}@e-ucm.es, {pablom, balta}@fdi.ucm.es

Abstract. This paper presents *e-Training DS*, an instructor-oriented toolkit for the integration of portable game consoles in modern e-learning environments (e.g. *Moodle*™). The tool is based on a general model in which the instructor maintains a library of mini-games that are easily produced by customizing generic game patterns (e.g. a simple jigsaw puzzle). The authors can then compile a working set of mini-games and distribute it to the students so that they can play using their own portable game devices. Moreover, the game packs produced with *e-Training DS* can track the students' activity and send the results to the back-end e-learning environment. In this manner instructors can evaluate the students' performance using the facilities provided by the e-learning system. While *e-Training DS* is focused on the Computer Science field and the *Nintendo DS*™ game console, the approach proposed can be repurposed for other knowledge areas and devices.

Keywords: integration, e-learning, portable game devices, authoring tool, learning management systems, instructors.

1 Introduction

Although the discussion about the effectiveness of educational games is still ongoing [1], the academic community is increasingly recognizing digital Game-Based Learning (GBL) as an engaging and self-directed educational paradigm [2]. Additionally during the last years the increasing interest in GBL has expanded, reaching also the mobile gaming field as mobile devices can foster pervasive and ubiquitous learning just at the moment and the place required [3].

However it is not clear how these games can be integrated into broader learning scenarios. This is especially troublesome in traditional brick and mortar education, where the introduction of new technologies is always difficult [4].

A possible approach is to focus on introducing these digital games into e-learning environments. First, these environments are strongly supported by technology, which facilitates the introduction of new digital media in a natural way. Second, modern e-Learning environments (or Learning Management Systems – LMS) like *Moodle*™ [5] or *Blackboard*™ [6] are web-based systems that can deliver highly interactive and complex content and include advanced tracking and assessment features for the instructors.

The integration of mobile educational games in e-learning environments would result in more interactive and rich learning scenarios that combine traditional web-based contents and game-based modules [7]. However, this pathway is not free of barriers. On the one hand, as opposite to web-based content (e.g. HTML or PDF documents) instructors cannot usually create their own educational games as this task is time-consuming, requires a multidisciplinary development team and demands some technical background from the instructor. On the other hand, tracking the students' performance involves an extra cost as games usually behave as black-boxes (i.e. closed products where information cannot be extracted from), forcing teachers to conduct post-tests or debriefing sessions [8]. Finally, the emergent importance of portable game platforms does not necessarily fit the web-oriented nature of LMS.

In this paper we propose a general model for integrating portable games in modern LMS that is based on two main ideas: game patterns that could be easily configured with a game editor and built-in tracking and assessment features that could be configured to deliver the data to the assessment system of a back-end LMS for latter processing. Besides we describe how these ideas have been implemented in the *e-Training DS* game platform (formerly *CS-Training*) [9],

The paper is structured as follows: In section 2, we present some related work regarding the integration of games in e-learning. In section 3 we present *e-Training DS,* describing the general model and the current implementation for an undergraduate Computer Science course. In section 4 we describe the process of compiling a customized set of mini-games and the execution of an online course that uses *e-Training DS*. In section 5 we describe a case study and finally in section 6 we present some conclusions and outline future lines of work.

2 Mobile Games and E-learning

Several good examples of applying portable games in education can be found in the recent literature. For example, in [3] the authors describe an experience with mobile games in rural India, which depicts how portable devices can be used to enhance learning processes where computers are not available. There are other interesting examples in the literature that report on the potential of portable devices to provide blended learning experiences where the game is integrated into the real environment (a.k.a. augmented reality) [10]. This has been used, for instance, to enhance tours in museums and botanic gardens [11, 12].

An additional advantage of mobile games is that they are cheaper to produce than desktop or console games. This is a very important issue as one of the main arguments against game-based learning is the high budget they demand, which is not always available in educational settings [13]. One of the better examples is the *Touch! Generations*™ (e.g. *BrainTraining*™) brand promoted by Nintendo[1], which targets a wide audience of different age, gender and gaming experience. These games have

[1] http://www.touchgenerations.com/

been successful in gathering new audiences as the interaction is more natural and easy to understand with a very low investment. This is an important design issue as the game atmosphere and fostering of competition seems to be more important in engaging students that a complex and costly 3D immersive world [14]. Although these games still have detractors [15] that argue that there is no evidence of the effectiveness of mobile games beyond the *drill & practice* paradigm, it is clear that they are useful in engaging students and encouraging them to practice and learn.

There are also other approaches to reduce the cost of the games. In this regard the application of specific game authoring tools to develop the games can be useful. This is, for instance, the case of the <e-Adventure>[16] or GameMaker[17] authoring tools. However these tools can be still too complex for some instructors. Another approach is to apply wizard-like tools that instead of allowing instructors to create a whole video game (which might be too ambitious), they provide pre-made games that can be easily configured by the instructor. Some good examples of these tools are *DS Game Maker*®[2] or *Raptivity*©[3], one of the most popular learning tools for creating interactive learning exercises.

However, there are still open barriers for the introduction of mobile games in educational settings. One of the most important is the extra workload they add to the assessment process [8]. An approach to address this issue is to integrate the games into Learning Management Systems so instructors can access the results of the game experience using tracking and evaluation interfaces they are used to. While there are some initiatives in this line for desktop applications [14, 18-20] this research line is almost unexplored for mobile platforms.

3 e-Training DS

Our approach to introduce portable games in e-learning environments is based on a general game model that conceives an educational game composed by small mini-games running on the portable device that uses the back-end LMS as a data server.

In order to allow instructors with no technical background in game programming to create and customize different instances of these mini-games, *e-Training DS* includes a visual editor that permits the configuration of a library of mini-games. Based on a user friendly wizard-like, the *e-Training DS* editor allows the instructor to create custom instances of the game patterns in a few simple steps. To provide a friendly methodology, the editor contains several templates for each game pattern and allows the configuration of how the students' performance must be tracked and evaluated.

3.1 General Game model: Mini-games, training and evaluation

The central point of the general model is a set of mini-games, each one supported by a specific local evaluator (assessment module), encapsulated in a bigger game that launches these mini-games and manages the final evaluation taking into account the

[2] http://dsgamemaker.com/

[3] http://www.raptivity.com/

results produced by the local evaluators (Fig. 1). In order to manage this evaluation, the launcher receives the score of every included mini-game to produce a weighted score. Each mini-game is an interactive version of a practical problem studied in Computer Science, and adapted to the specific features of the portable game device (e.g. touch interaction, small canvas for rendering), etc.

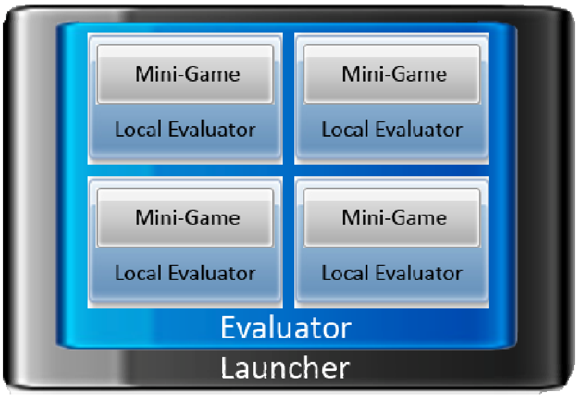

Mini-Game
Local Evaluator
Mini-Game
Local Evaluator
Mini-Game
Local Evaluator
Mini-Game
Local Evaluator
Evaluator
Launcher

Fig. 1. The structure of the general game model is based on a set of mini-games managed by a main loader.

Each of these short, simple exercises includes a different evaluation logic that determines the grade the player gets upon completion. By default, the system includes 10 different mini-games, focusing on different problems and featuring different types of interaction (e.g. point-and-click, drag & drop, etc.). (Fig. 2).

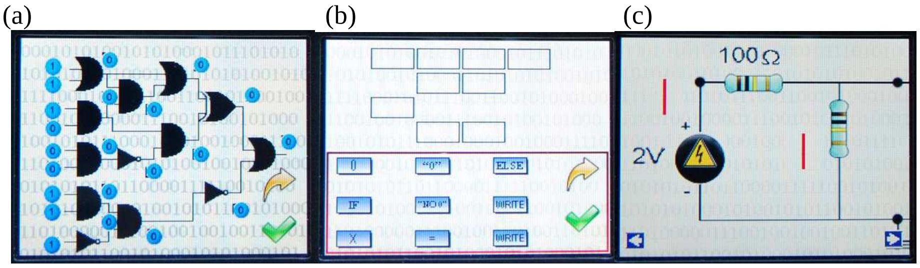


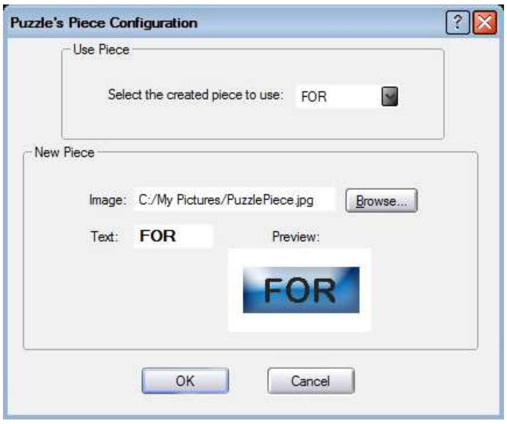
Fig. 2. Some mini-game samples. (a) Logic Gates: click on the input nodes to change their binary values in order to obtain the correct output values; (b) Coding Puzzle: drag and drop each puzzle piece to its correct position; (c) Electric Current: mark the right direction of the current through the circuit.

These mini-games are conceived to be used as training exercises but also as assessment tools. When the game is launched, students can choose whether they want to practice with the exercises or to perform an online evaluated test. The training mode allows students to practice freely with the interactive exercises as many times as desired. There are no grades or time limitations. This is an unsupervised mode that allows students to practice in their spare time, so it does not perform any interaction with the back-end LMS.

In turn, the evaluation mode requires students to complete a set of tests in which some of the mini-games are played sequentially with a time limitation and an automatic evaluation. The evaluations are executed in communication with the LMS, and the results are submitted automatically to the server for instructor supervision. The time limit, the number of exercises and their order are configured by the instructor using the editor.

3.2 Managing the Library of Games

Instructors develop their own mini-games by customizing pre-defined generic game patterns. Exercises of the same type share aspects such as the interaction type (drag & drop, point-and-click, etc.) and the evaluation logic, while the graphical elements, and in some cases, the theme of the exercise, will be fully configurable. The e-Training editor allows the instructor to create instances by selecting a basic pattern and then completing a multiple-step wizard to determine aspects such as interaction and contents (Fig. 3).



Select the created piece to use: FOR

Image: C:/My Pictures/PuzzlePiece.jpg

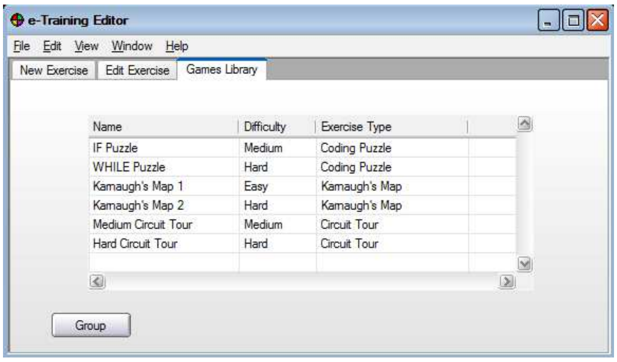
Text: FOR

Fig. 3. Creating a new mini-game of the Coding Puzzle (Fig. 2b) type allows the instructor to customize each of the pieces included in the puzzle.

Each new interactive exercise that the instructor creates is stored in a local library of games identified by a name, a difficulty and a type, according to the choices made during the customization process. The library is displayed as a list of games that can be selected to package an executable game as described in the next section (Fig. 4)

3.3 Packaging the final *e-Training* game

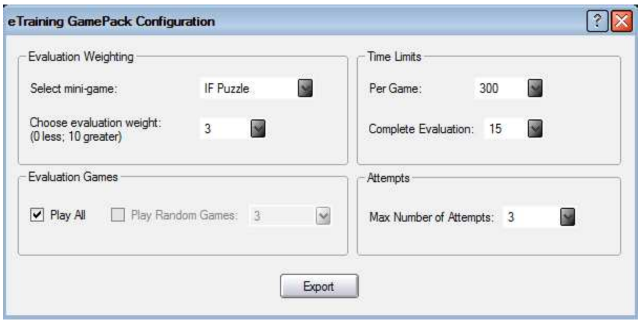
The instructor can select a group of mini-games from the game library and compile a functional *e-Training DS* game package with training and evaluation features. After choosing the games that the package will include, the instructor must set some features on a brief configuration form. These features include a numeric field to edit the weight each exercise will have on the final grade, a checkbox to select whether the exercises are executed randomly in each game run in evaluation mode, etc. (Fig. 4).



Name	Difficulty	Exercise Type
IF Puzzle	Medium	Coding Puzzle
WHILE Puzzle	Hard	Coding Puzzle
Kamaugh's Map 1	Easy	Kamaugh's Map
Kamaugh's Map 2	Hard	Kamaugh's Map
Medium Circuit Tour	Medium	Circuit Tour
Hard Circuit Tour	Hard	Circuit Tour

Fig. 4. Screenshot of the e-Training's authoring tool, showing the content of the games library tab.

During the exportation process the *e-Training* DS editor will build the low-level C files required and compile all the mini-games, evaluator modules and other wrappers into a single DS executable game. Besides, the exportation process also produces a configuration script for the LMS that configures the communication assessment features for that specific game (Fig. 5).



Evaluation Weighting

Select mini-game: IF Puzzle

Choose evaluation weight:
(0 less; 10 greater) 3

Time Limits

Per Game: 300

Complete Evaluation: 15

Evaluation Games

☑ Play All ☐ Play Random Games: 3

Attempts

Max Number of Attempts: 3

Fig. 5. Screenshot of the configuration form, in which the configuration script for the LMS is automatically created.

4 Running a Game-based/enabled Web Course

Once the instructor has created a complete package of exercises for the portable device, the game is ready to be introduced in a broader course. The instructor should now configure the LMS to support the course. The current version of *e-Training DS* includes built-in features to actively integrate the games in a *Moodle*™ environment. When preparing the different lessons that compound the course, the instructor should create an assessment tool that will serve as a placeholder for the integration of the portable game, and then run the script that was generated by the authoring tool. The instructor will then be able to use that assessment instance to retrieve the information generated by the game.

4.1 Evaluation life-cycle

Students will be able to train and solve the mini-games included in the packaged game at any point and at their own pace. However, when the student runs the game in evaluation mode on the portable device, the execution is performed in communication with the LMS, according to the following steps (Fig. 6).

1. Authentication: The portable game attempts to authenticate the student in the LMS using a login process (the student has to introduce these data manually in the device). If the login is valid, a new remote session is created.

2. Course Information: Once the student is authenticated, the game requests a complete list of courses in which the student is enrolled. These courses are checked seeking the presence of assessments created with the configuration script. Only compatible tests are shown on the device.

3. Final Evaluation: Finally, once the game produces the weighted score of test performance, it is forwarded to the LMS along with a brief performance report. After this step, the device will end the session in the LMS.

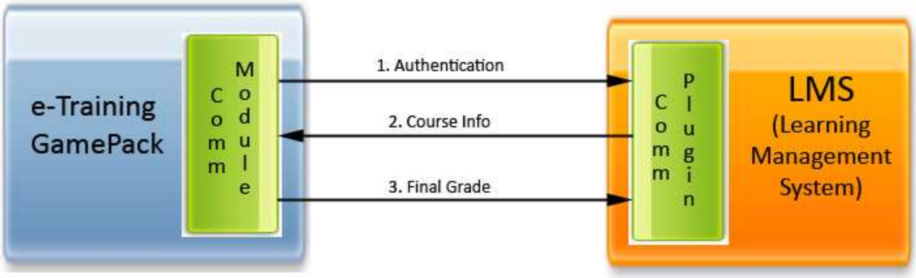
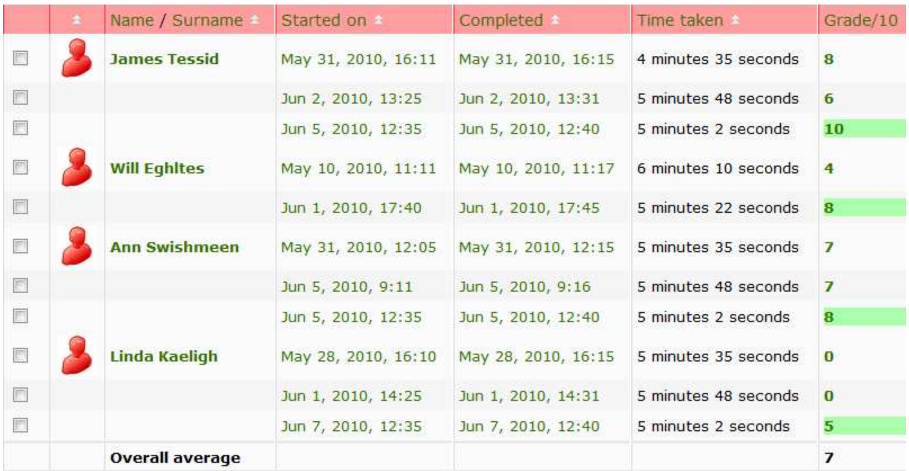


Fig. 6. Communication and interaction between both platforms is established in three steps.

4.2 Student Tracking and Assessment

Through the third step described in Figure 6, the LMS receives and stores a report detailing the game experience: data like the time consumed and the grade obtained in the resolution of each interactive exercise are included in the report. Using this tracking feature the instructor can check the performance of each student.

Students have in their profile a history of attempts (Fig. 7) that have been made for each test, sorted by descending score. The LMS generates a ranking of the students that have obtained better results. The instructor can decide to make this ranking public to encourage competitiveness and the spirit of excellence to achieve the best positions in the ranking.



		Name / Surname	Started on	Completed	Time taken	Grade/10
☐	👤	James Tessid	May 31, 2010, 16:11	May 31, 2010, 16:15	4 minutes 35 seconds	8
☐			Jun 2, 2010, 13:25	Jun 2, 2010, 13:31	5 minutes 48 seconds	6
☐			Jun 5, 2010, 12:35	Jun 5, 2010, 12:40	5 minutes 2 seconds	10
☐	👤	Will Eghltes	May 10, 2010, 11:11	May 10, 2010, 11:17	6 minutes 10 seconds	4
☐			Jun 1, 2010, 17:40	Jun 1, 2010, 17:45	5 minutes 22 seconds	8
☐	👤	Ann Swishmeen	May 31, 2010, 12:05	May 31, 2010, 12:15	5 minutes 35 seconds	7
☐			Jun 5, 2010, 9:11	Jun 5, 2010, 9:16	5 minutes 48 seconds	7
☐			Jun 5, 2010, 12:35	Jun 5, 2010, 12:40	5 minutes 2 seconds	8
☐	👤	Linda Kaeligh	May 28, 2010, 16:10	May 28, 2010, 16:15	5 minutes 35 seconds	0
☐			Jun 1, 2010, 14:25	Jun 1, 2010, 14:31	5 minutes 48 seconds	0
☐			Jun 7, 2010, 12:35	Jun 7, 2010, 12:40	5 minutes 2 seconds	5
		Overall average				7

Fig. 7. Students' attempts are stored in the LMS, highlighting the best grades for each one.

5 Case Study: Creating a Computer Science training module

In order to evaluate the model described in this article we have developed a simple case study in the context of Computer Science education. In this section we describe the process of creating an *e-Training DS* module.

First we create ten different mini-games based on the different types that are included in the editor, such as a coding puzzle, iteration over binary trees, logic devices, Karnaugh maps, etc. Thanks to the wizard-like orientation of the *e-Training DS* editor, the process of creating each mini-game has a reasonable cost in time, as any specific design is not required. Both theme and game logic are predefined, so the creation process just requires setting the properties through the editor and choosing the graphical elements (images, backgrounds, etc).

Secondly, once the library of games is ready, we select the characteristics of the evaluation through the configuration form and choose between different options, such as selecting the assessment weight for each game included. Then we export our selection of mini-games to an *e-Training DS* game package, which is an immediate process. The exportation process also generates a configuration script that should be uploaded to the LMS to configure the communication between the game device and the LMS. Finally we configure the features of our own LMS course, creating the lessons and the appropriate tests, and registering the students. As Table 1 depicts the whole process is affordable for a single instructor.

Table 1. The time spent for each different task in the case study is the key in measuring the cost of using the whole e-Training's module.

Task	Time Spent
Creating ten different mini-game instances	7 hours
Customizing the evaluation process	1 hour
Extracting the *e-Training DS* Game Pack	Immediate
Course configuration (in Moodle)	2 hours
Deploying and running the configuration script	1 hour

6 Conclusions & Future Work

In this article, we introduce *e-Training DS* as a platform to facilitate the integration of portable game devices in the learning flow. We consider that casual games played in a portable gaming device can be an accessible pathway to promote game-based learning experiences. In addition, these portable devices are now pervasively connected to the Internet, which allows them to communicate with a web-server. However, the technical challenges are significant. In many cases, instructors already find themselves struggling to use simple games as an educational medium, due to the lack of resources and technical knowledge.

The tool offers an assisted process for the instructors to create new mini-game instances, customize their evaluation and configure the several tests that the students will have to solve. In addition to creating their own learning contents and managing the courses, instructors are able to track students' progress even considering the implicit mobility of portable devices, thanks to the wireless communication with the LMS. These kinds of devices are also especially attractive for the students, which results in an overall improvement of the learning experience.

e-Training DS was also conceived as an extensible framework for different fields and technologies, even if the tool described in this paper only focuses on Computer Science. We have also based our implementation on the *Nintendo DS*™ game console and *Moodle*™, but the characteristics of this model make it suitable to support other kind of portable devices (such as cell phones or tablet PCs) and any other LMS platform (such as Web-CT or Sakai).

We intend to explore this flexibility further, with the extension of the model to support other mobile platforms (with special focus on the *Android*™ operating system). The objective would be to use a single library of games that may be grouped and compiled for different mobile platforms (e.g. the *Nintendo DS*™ or an *Android*™ device). This would protect the investment in the mobile platforms field, which moves at a great speed with new technologies emerging every year.

Acknowledgments. The Spanish Committee of Science and Technology (TIN2007-68125-C02-01) and the Ministry of Industry (grants TSI-020110-2009-170 and TSI-020312-2009-27) have partially supported this work, as well as the Complutense University of Madrid and the regional government of Madrid (research group 921340 and project e-Madrid S2009/TIC-1650) and the EU GALA network of excellence and PROACTIVE EU project (505469-2009-LLP-ES-KA3-KA3MP).

References

1. Hays, R.T.: The effectiveness of instructional games: a literature review and discussion. Naval Air Warfare Center, Orlando, FL. (2005)
2. Van Eck, R.: Digital game-based learning: It’s not just the digital natives who are restless. EDUCAUSE Review **41** (2006) 16-30
3. Kam, M., Rudraraju, V., Tewari, A., Canny, J.: Mobile Gaming with Children in Rural India: Contextual Factors in the Use of Game Design Patterns. 3rd Digital Games Research Association International Conference, Tokyo, Japan (2007)
4. Reynolds, D., Treharne, D., Tripp, H.: ICT—the hopes and the reality. British Journal of Educational Technology **34** (2003) 151-167
5. Dougiamas, M., Taylor, P.: Moodle: Using Learning Communities to Create an Open Source Course Management System. World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003. Honolulu, Hawaii, USA (2003) 171-178
6. Blackboard Inc: Blackboard Building Blocks. Vol. 2006 (2006)
7. Lavín-Mera, P., Torrente, J., Moreno-Ger, P., Fernández-Manjón, B.: Mobile Game Development for Multiple Devices in Education. 4th International Conference on Interactive Mobile and Computer-Aided Learning, Amman, Jordan (2009)
8. Squire, K.: Changing the game: What happens when video games enter the classroom. Innovate, Journal of Online Education **1** (2005)
9. Tornero Santamarina, R., Moreno-Ger, P., Torrente, J., Fernández-Manjón, B.: CS Training: Introducing Mobile Educational Games in the Learning Flow. IEEE International Symposium on Industrial Electronics 2010 (ISIE 2010). IEEE, Bari, Italy (2010)
10. Squire, K., Klopfer, E.: Augmented Reality Simulations on Handheld Computers. Journal of the Learning Sciences **16** (2007) 371-413
11. Doswell, J., Harmeyer, K.: Extending the 'Serious Game' Boundary: Virtual Instructors in Mobile Mixed Reality Learning Games. Digital Games Research Association International Conference (DiGRA 2007), Tokio, Japan (2007)
12. Ardito, C., Buono, P., Costabile, M.F., Lanzilotti, R., Pederson, T.: Re-experiencing History in Archaeological Parks by Playing a Mobile Augmented Reality Game. OTM Workshop 2007 (2007)
13. Michael, D., Chen, S.: Serious Games: Games that Educate, Train, and Inform. Thomson, Boston, MA (2006)
14. Sancho, P., Fuentes, R., Gómez-Martín, P.P., Fernández-Manjón, B.: Applying multiplayer role based learning in engineering education: Three case studies to analyze the impact on students’ performance. International Journal of Engineering Education **25** (2009)
15. Pivec, P.: Game-based Learning or Game-based Teaching? Becta (2009)
16. Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., Sierra, J.L.: Instructor-oriented Authoring Tools for Educational Videogames. 8th International Conference on Advanced Learning Technologies (ICALT 2008). IEEE Computer Society, Santander, Spain (2008)
17. Overmars, M.: Teaching Computer Science through Game Design. IEEE Computer **37** (2004) 81-83
18. McDowell, P., Darken, R., Sullivan, J., Johnson, E.: Delta3D: A Complete Open Source Game and Simulation Engine for Building Military Training Systems. The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) (2005)
19. Del Blanco, A., Torrente, J., Moreno-Ger, P., Fernández-Manjón, B.: A General Architecture for the Integration of Educational Videogames in Standards-compliant Virtual Learning Environments. 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009). IEEE Computer Society, Riga, Latvia (2009)
20. Kemp, J., Livingstone, D., Bloomfield, P.: SLOODLE: Connecting VLE tools with Emergent Teaching Practice in Second Life. British Journal of Educational Technology **40** (2009) 551-555