A WEB BROWSER-BASED 3D SIMULATION ARCHITECTURE FOR EDUCATION AND TRAINING

Miguel Ángel Gamón, Miguel Ángel Expósito, Ángel Rodríguez-Cerro
Marta Pla-Castells and Rafael J. Martínez-Durá
Instituto de Robótica, Universidad de Valencia, PO-Box 2085. 46071, Spain

Keywords: Web-based Learning, Virtual simulation and training, e-Learning platforms, Games for education and training.

Abstract: This paper shows a web browser-based simulation architecture for education and training. This architecture integrates multimedia content with an interactive virtual environment. This is achieved combining the use of OSG4Web, a plugin for managing a scene graph inside a web browser, with the Flash platform. This architecture has been used to develop a training system focused on building and masonry, and enables students to practice their skills and acquired knowledge in a 3D immersive environment. It also includes an instructional design and an evaluation module by means of several didactic units and exams.

1 INTRODUCTION

Nowadays, academic institutions related to Information Technologies for communication in education fields are progressively incorporating the use of web browser-based applications (Hosmeier and Dicesare, 2000), including e-learning technologies for learning and training in different degrees and professional areas (Garrison and Anderson, 2003) (Michael and Chen, 2005).

These applications allow students to get access to educational materials related to a particular course or qualification, (e.g. professional modules of masonry), and also typically include on-line exams to validate the correct assimilation of the contents, leading to self-learning and self-evaluation.

The main goal in these applications is to increase the skills acquired by the user by means of an efficient learning method (Holzinger and Ebner, 2003). In fact, most of them are based on exercises implemented as interactive 2D animations. However, the limitations of 2D interactivity in most cases diminishes the effectiveness of this technique (Virvou et al., 2005).

Recently, e-learning solutions, especially those related to games for education (Squire, 2003)(Amory et al., 2002), tend to include features like simulation and virtual reality via 3D interactive environments (Elliott et al., 2002). This new concept allows the implementation of real working methods in a realistic way (e.g. placement of protective barriers, or the removal of building partitions).

Thus, the functionality of the simulators enables the user to freely navigate through a 3D scene, (e.g. a building under construction), the selection of objects like tools and construction materials, the use of working methods, and the completion of tasks or objectives required in an exercise (Kim, 2005). In addition, through an instructional design the simulator provides a better use of the teaching-learning process.

In this context, this paper presents a web browser-based 3D simulation architecture for education and training. It shows the techniques that allows to practice real life tasks, through a realistic environment based on virtual reality. This architecture has been used to develop a training system focused on building and masonry, wich improves the learning process enabling the students to practice their skills and acquired knowledge in the same web page.

The rest of this paper is structured as follows: section 2 describes the technical background of web browser-based simulators; section 3 presents the proposed architecture for education and training; section 4 exposes the instructional design of the developed application; section 5 shows the results and usage of the system; finally, section 6 draws some conclusions and further work.

2 TECHNOLOGY BACKGROUND

As technology evolves, e-learning systems are progressively moving towards web-based simulation,

Ángel Gamón M., Ángel Expósito M., Rodríguez-Cerro Á., Pla-Castells M. and J. Martínez-Durá R. (2010).
A WEB BROWSER-BASED 3D SIMULATION ARCHITECTURE FOR EDUCATION AND TRAINING.
In *Proceedings of the 2nd International Conference on Computer Supported Education*, pages 307-312
DOI: 10.5220/0002800103070312

more powerful and didactic. This enables web pages to display interactive 2D and 3D content as well as other advanced capabilities such as immersive audio and video, thus improving the user's experience. A brief description of the most commonly used technologies has been exposed in (Martnez-Dur, 2009).

For this purposes, Adobe Flash is an excellent option since it's a popular standard for rich internet content (*www.adobe.com*) powered by the ActionScript scripting language (*www.actionscript.org*). Flash files (.swf), also known as Flash movies, run on a virtual machine that displays animations and executes ActionScript code. This virtual machine, also called Flash Player, can be embedded into a web page as a browser plugin. Adobe Flex (*www.adobe.com/products/flex*) is a framework built on top of the Flash API consisting of an XML-based language and a set of libraries for building Flash-based RIAs very quickly. In summary, it's a really well-suited development tool for building rich GUIs and 2D content that runs on almost every browser.

Last versions of the Flash API allow raster operations on 2D polygons, which has been the starting point for emerging Flash-based 3D rendering engines, like PaperVision (*www.papervision3d.org*) or Away3D (*away3d.com*). However, Flash-based 3D rendering engines can't take full advantage of the hardware graphics pipeline due to the lack of support in the Flash API. That leads to the implementation of most "‘hardware"' tasks in ActionScript, which is a slow language for the extremely heavy load that these calculations represent.

In order to find a technology capable of supporting a real-time 3D interactive environment, the main requirements was the ability of handling a scene graph and OpenGL or DirectX graphics. This way, we have found several alternatives to take into consideration.

Firstly, Java allows to build web-based simulation applications. Hardware-accelerated 3D graphics in Java are possible through native bindings like jOGL (*jogl.dev.java.net*) or jMonkey (*www.jmonkeyengine.com*), that export operating system-dependent APIs such as the OpenGL implementation to the language.

Other alternative developed by Google is O3D (*code.google.com/apis/o3d*), an open-source JavaScript API for creating interactive 3D graphics applications that run within a browser window, such as games or virtual worlds. It uses OpenGL or DirectX through scene graph data structures and API.

Another interesting possibility is OSG4Web (Calori et al., 2009) (Pescarin et al., 2005), wich provides an open-source framework for in-browser OpenGL-based application wrapping. The framework allows the development of OpenGL and OpenSceneGraph (*www.openscenegraph.org*) based applications, that open windows within the browser, allowing bidirectional interaction with page elements via JavaScript. OSG4Web decouples the application code from the rendering engine, therefore the same plugin can run totally different applications, also called cores. Cores are dynamic OS libraries that are loaded and linked in run-time, making it very efficient but absolutely dependent on the operating system.

3 ARCHITECTURE

The architecture that is presented in this paper supports both 2D and 3D contents. 2D graphics are used to display interactive menus, interface controls and theoretical content, while 3D graphics provide quality practice exercises. Our requirements for the 2D part have been to make an usable, easy and intuitive interface by means of clear and simple buttons and organized layouts without sacrificing compatibility. The Adobe Flex IDE, based on Eclipse, provides an easy way of designing layout-based GUIs with a visual editor. It also incorporates all the capabilities and versatility of Flash and ActionScript. Due to this features, the choice has been to use this tool for all the 2D-related parts.

Regarding 3D content, the main goal has been to display fast and neat graphics that supported the formative goal of every e-Learning application. Unfortunately, as mentioned before, Flash can't provide quality 3D graphics at this moment. That implied two main aspects: it would be necessary to choose a separate platform for rendering only the 3D part, and it would be needed some kind of integration between Flash and the former. Thus, OSG4Web has been the choice for that task due to its high speed OpenSceneGraph (OSG) infrastructure. It allows quick porting of existing OSG applications. The fact that the plugin implements JavaScript bidirectional communication, as it does Flash, allows a JavaScript-based integration.

Thus, the architecture is composed of four basic blocks (Figure 1):

- **HTML Container.** Instantiates both browser plugins. Contains the JavaScript bridge and some other handy code to manage on-screen elements.
- **Flash Plugin.** Displays 2D contents and incorporates the application logic.
- **JavaScript Bridge.** Supports the communication between Flash and OSG4Web.
- **OSG4Web Plugin.** Displays 3D contents, and incorporates the 3D scene graph management code

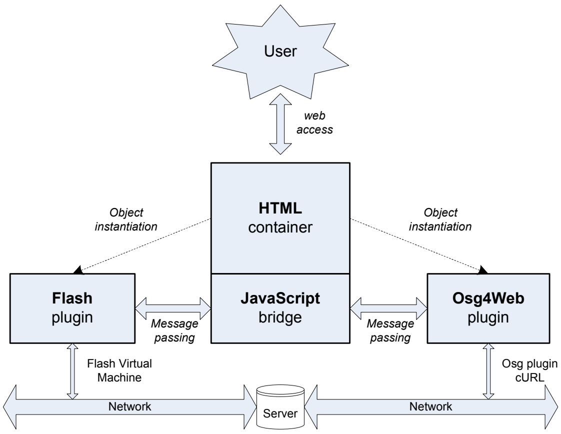


Figure 1: Web simulator architecture.

and the exercises logic.

A detailed explanation of the internals follows:

3.1 HTML Container

When the user's browser requests the HTML container web page by accessing its URL, it (as defined in its source code) instantiates the Flash Player into a DIV layer and also the OSG4Web plugin into another one. A JavaScript detection code triggers a pop-up message in the case the user don't have either the OSG4Web plugin or the Flash Player plugin installed on his machine, and the user is pointed to the right place where he can download and install it.

3.2 Flash Plugin

The Flash plugin is loaded with a movie when the page loads. The Flash movie implements the main application code and logic. On the one hand, it is responsible for the GUI, the navigation through menus, the 2D theoretical module management and display, the multimedia content (images, videos, sounds) loading and managing, and the self-checking tests. On the other hand, it calls JavaScript for showing/hiding the OSG4Web layer as necessary, sends commands to the 3D plugin and receives the events from it.

3.3 JavaScript Bridge

As mentioned before, both Flash and OSG4Web are able of calling functions written in a foreign language resident in web container. In this particular case, JavaScript is used because it is a well-known language that runs in many browsers. The communication between Flash and OSG consists of a JavaScript bridge which sends messages from one to another in both directions. We have implemented it with a pair of functions. The first one is called by Flash which calls a OSG4Web-JavaScript interface function, while the other is called by OSG4Web and calls a Flash-JavaScript interface function.

Flash can call any JavaScript function, with arbitrary name or arguments count and, even, get back its return value in a transparent way. In the case of OSG4Web this is more restricted. It only allows cores to call a function named "'eventCatcher"' which takes a single argument: a text string. For this reason, we had only the ability to interchange one text string in each call.

That led us to design and implement a text based protocol for communicating messages, consisting of sub-strings separated by one special separator character. There are two types of messages: **Commands** and **Events**. Commands are orders or requests that usually flow from Flash to the 3D core, and events are asynchronous situations that require a notification from one side to the another, mainly for synchronization or other purposes.

Typical tasks that the protocol implements are:

- Request from OSG4Web to load a 3D scene model.
- Request from Flash to play an MP3 sound.
- Notify to OSG4Web the completion of the playing of a sound.
- Notify to Flash that the exercise has finished and also its results.

3.4 OSG4Web Plugin

OSG4Web is based on two components (Figure 2): the first is a browser plugin, and the other consists of a communication interface to interact with the particular graphic application. In this way, as mentioned before, it is possible load different graphics application with the same plugin.

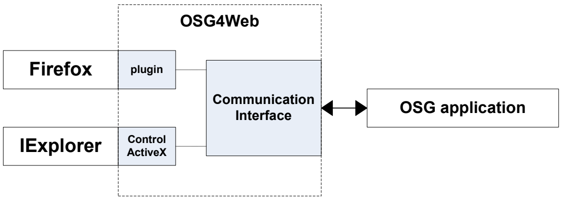


Figure 2: OSG4Web components.

OSG4Web works on Microsoft Internet Explorer and Mozilla Firefox under Windows. We have improved bidirectional communication between the OSG4Web plugin and JavaScript which only received messages on the ”window refresh” events. We have also implemented a queue for avoiding message leaks that emerged randomly.

OSG Application Core

The OSG4Web plugin is loaded with a custom-designed core. We have implemented a set of exercises that consist of a typical work scene, where the user has to pick some elements and interact with them in a point-and-click fashion. Because of that, there is a special event to notify Flash when the user picks an object, receiving the name of the scene graph node picked. This way, Flash can display it integrated in the 2D GUI.

When a new exercise starts, Flash provides OSG4Web with a command telling it to load a XML file that contains the name of the exercise scene model and also other information about the exercise. The application core downloads the OSG model file (.ive) hosted in the web server and attaches the scene model to the root node.

The simulator logic is splitted across the Flash movie and the OSG4Web application core. The most elegant solution would have been extending our protocol to allow scene graph management directly from Flash, but for the sake of simplicity, we adopted the solution of having the main simulator logic in ActionScript, and keep the exercise-related scene graph management code in our OSG4Web core, as a compromise between functionality and required effort.

As stated before, each exercise is described by using the XML language. We have also defined a DTD that describes a pseudo-scripting language, allowing to define several configurable aspects, like the models to load or the positions of the objects to show. It also defines the implementation of the exercise logic by means of select statements, jumps, loops or conditional branches.

By using this XML scheme, developers can create new exercises with very few changes and also debug them with ease.

4 INSTRUCTIONAL DESIGN

The objective of the developed training system is to teach the procedures, materials and safety measures needed to perform operations in the construction of fences and brick walls, while behaving in a safe way. For this purpose, an instructional design model has been specified. This model allows properly to exploit the teaching process and optimize the training time.

The instructional design for this training system refers to the exercises that the user have to follow in order to acquire the relevant skills. Their contents and the evaluation criteria have been structured in a convenient way, taking into account the age and formative level of the students.

In first place, the instructional design includes a set of skills related to masonry that the user must acquire through the use of the simulator:

- Prepare and maintain tools, equipment, materials and aids.
- Wall partitioning.
- Prepare cement mortars.
- Build walls and partitions.
- Position and fix frames to brick structures.

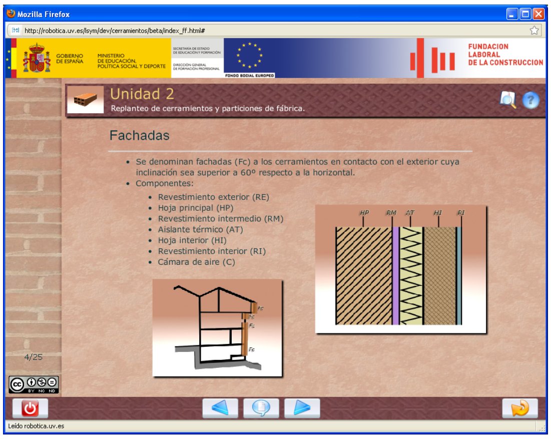
The educational content of the application is divided into five different units, so each one presents the techniques and methodologies required to achieve the skills described above. Each learning unit has been divided into three parts: a theoretical part, which explains the theoretical concepts related to each skill; a practical part in which students must perform a set of masonry tasks in a 3D virtual environment; and an evaluation part, where multiple choice questions extracted from the didactical contents are asked and then evaluated.

In addition, an evaluation system provides a measurement of the theoretical concepts that are being assimilited. Thus, it evaluates the right choice of appropriate security measures, the mistakes made, and the correctness in the decisions made to solve the tasks.

The proposed instructional design provides several advantages to the learning process. Particularly, the obtained results are:

- Provide a training tool, empowering the process of building workers training.
- Give knowledge, skills and abilites to workers, which are necessary to develop safe behaviors.
- Establish secure protocols for typical operations and raise workers awareness of the importance of security within the organization.
- Assess individual workers and evaluate the level of compliance of security protocols.

In this manner, the instructional system offers a number of defined characteristics that allows a better use of the teaching-learning process.


Mozilla Firefox
http://robotica.uv.es/lsym/dev/cerramientos/beta/index_ff.html#
GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN, POLÍTICA SOCIAL Y DEPORTE
SECRETARÍA DE ESTADO DE EDUCACIÓN Y FORMACIÓN
DIRECCIÓN GENERAL DE FORMACIÓN PROFESIONAL
FONDO SOCIAL EUROPEO
FUNDACION LABORAL DE LA CONSTRUCCION
HP
RM
AT
HI
RI
Fc
Fc
Fc
Fc
4/25
BY NC ND
Leído robotica.uv.es

Unidad 2

Replanteo de cerramientos y particiones de fábrica.

Fachadas

- Se denominan fachadas (Fc) a los cerramientos en contacto con el exterior cuya inclinación sea superior a 60º respecto a la horizontal.
- Componentes:
  - Revestimiento exterior (RE)
  - Hoja principal (HP)
  - Revestimiento intermedio (RM)
  - Aislante térmico (AT)
  - Hoja interior (HI)
  - Revestimiento interior (RI)
  - Cámara de aire (C)

Figure 3: Theoretical module: The slide contains multimedia elements related to learning content. It also contains navigation buttons to browse between slides or to access to the different sections and didactical units.

5 RESULTS

By means of using the proposed architecture defined before we have developed a training system composed of several units containing: a theoretical module with the typical features of an e-learning website, a practice or simulation 3D-based module, and a test module in the form of questionnaires.

The operation details of each of the modules are described below.

5.1 Theoretical Module

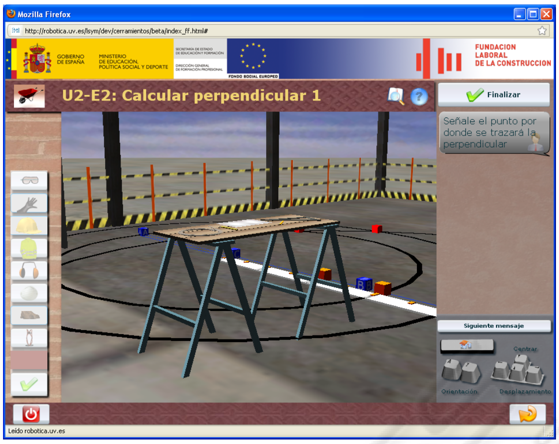
In the theorethical module, different multimedia content (like static pre-rendered images, movies and other 2D content) is shown using a presentation with Flash (Figure 3). Navigation buttons allow to browse between slides or access the different sections and didactical units, as well as consulting a glossary of terms related to construction.

The slides of each module show the above mentioned text, images and videos. All the text strings are defined and indexed in a localized XML file for multi-language support. The contents are also supported by a pre-recorded voice that reads them.

5.2 Practice Module

In the simulation or practice module, the user navigates in a virtual environment similar to a building site, where he must perform a set of particular actions that lead to achieve the aim of the exercise (Figure 4).

At the beginning of the exercise, a voice, along with some screen messages, explain the objectives to accomplish and indicates the tasks to do in every moment.


Mozilla Firefox
http://robotica.uv.es/lsym/dev/cerramientos/beta/index_ff.html#
GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN, POLÍTICA SOCIAL Y DEPORTE
SECRETARÍA DE ESTADO DE EDUCACIÓN Y FORMACIÓN
DIRECCIÓN GENERAL DE FORMACIÓN PROFESIONAL
FONDO SOCIAL EUROPEO
FUNDACION LABORAL DE LA CONSTRUCCION
Finalizar
A
B
C
Siguiente mensaje
Centrar
Orientación
Desplazamiento
Leído robotica.uv.es

U2-E2: Calcular perpendicular 1

Señale el punto por donde se trazará la perpendicular

Figure 4: Practice module: The virtual scene shows a building site. The user can navigate through the environment using the keyboard and the mouse. Also, the user can select and use different tools for complete the tasks.

The system provides a mechanism for navigating through the environment and selecting objects, using the keyboard and the mouse. The possible movements are the classic ”first person shooter” ones: change head/pitch with the mouse, and walk with the arrow keys.

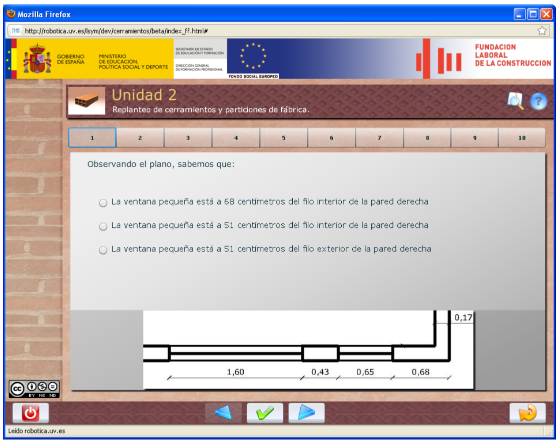
The experience with users has shown that it’s important to establish properly the camera view when the exercise starts, focusing on the tools and the workplace. In addition, a ”home key” has been implemented to allow the user to return to the starting position when has moved to an undesired area. Another suggestion has been to identify the selected objects in the exercise with labels on the screen.

The decisions made by the student may not be correct, but the implementation of the exercises allows for situations that could become erroneous. This helps the student to learn from his mistakes by means of choosing another working methodology.

5.3 Test Module

The test module shows a series of questions spoken by a voice and defined in an XML file, where the system prompts the user for selecting one of up to six possible answers (Figure 5).

When the student considers that the test is completed, a button allows the user to submit it. Then, a summary window is shown, reporting the total taken time, the number of hits and misses represented by a pie chart and the qualification (test passed or test failed).


Mozilla Firefox
http://robotica.uv.es/lsym/dev/cerramientos/beta/index_ff.html#
GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN, POLÍTICA SOCIAL Y DEPORTE
SECRETARÍA DE ESTADO DE EDUCACIÓN Y FORMACIÓN
DIRECCIÓN GENERAL DE FORMACIÓN PROFESIONAL
FONDO SOCIAL EUROPEO
FUNDACION LABORAL DE LA CONSTRUCCION
1
2
3
4
5
6
7
8
9
10
0,17
1,60
0,43
0,65
0,68
BY NC ND
Leído robotica.uv.es

Unidad 2

Replanteo de cerramientos y particiones de fábrica.

Observando el plano, sabemos que:

○ La ventana pequeña está a 68 centímetros del filo interior de la pared derecha

○ La ventana pequeña está a 51 centímetros del filo interior de la pared derecha

○ La ventana pequeña está a 51 centímetros del filo exterior de la pared derecha

Figure 5: Test module: In this module, the user answers several questions about the theoretical content. The user can freely navigate through the different questions using the upper bar and change his answer at any time.

6 CONCLUSIONS AND FURTHER WORK

In this paper, a web browser-based simulator architecture for learning and training has been presented. As a result, a simulator to use in the educational field has been implemented.

The proposed architecture combines the advantages of Flash, which provides rich multimedia presentations with bright and usable interfaces, with the advantages of OSG4Web, capable of loading and executing an OSG application for the displaying of fast and high quality 3D graphics.

The developed application allow the students, through a web page, to learn new theoretical concepts, to practice their skills and acquired knowlege in a 3D immersive environment, and finally to be evaluated. The specification of an instructional design has allowed to achieve a better use of the learning process.

As future work, we are planning to improve our web-based simulator's look and feel by using shaders and physical simulation. The enhancement of the web integration, extending its compatibility across different browsers and platforms is also planned.

ACKNOWLEDGEMENTS

This work has been supported by Fundacin Laboral de la Construccin (FLC), Ministerio de Educacin — Centro Nacional de Investigacin y Comunicacin Educativa (CNICE) and European Commision — European Social Fund (ESF).

REFERENCES

Amory, A., Naicker, K., Vincent, J., and Adams, C. (2002). The use of computer games as an educational tool: identification of appropriate game types and game elements. *British Journal of Educational Technology*, 30(4):311–321.

Calori, L., Camporesi, C., and Pescarin, S. (2009). Virtual rome: a foss approach to web3d. In Spencer, S. N., editor, *Proceedings of the 14th International Conference on 3D Web Technology*, pages 177–180. ACM.

Elliott, J., Adams, L., and Bruckman, A. (2002). No magic bullet: 3d video games in education. *Proceedings of ICLS*.

Garrison, D. and Anderson, T. (2003). *E-learning in the 21st Century: A Framework for Research and Practice*. Routledge/Falmer, New York.

Holzinger, A. and Ebner, M. (2003). Interaction and usability of simulations & animations: A case study of the flash technology. In *Human-Computer Interaction - INTERACT'03*, pages 777–780. IOS Press.

Hosmeier, J. A. and Dicesare, C. (2000). System response time and user satisfaction: An experimental study of browser-based applications. *AMCIS 2000 Proceedings.*, page 347.

Kim, G. (2005). *Designing Virtual Reality Systems: The Structured Approach*. Springer.

Martnez-Dur, R. J. (2009). Real-time simulators developments: Current and new trends. *International Conference on Harbor Maritime & Multimodal logistics modeling and simulation*, pages 1–9.

Michael, D. and Chen, S. (2005). *Serious Games: Games That Educate, Train, And Inform*. Muska & Lipman/Premier-Trade.

Pescarin, S., Calori, L., Camposeri, C., and Forte, M. (2005). Interactive landscapes reconstruction: a web 2d and 3d opensource solution. *The 6th International Symposium on Virtual Reality, Archeology and Cultural Heritage VAST*, pages 33–38.

Squire, K. (2003). Video games in education. *International Journal of Intelligent Simulations and Gaming*, 2:49–62.

Virvou, M., Katsionis, G., and Manos, K. (2005). Combining software games with education: Evaluation of its educational effectiveness. *International Forum of Educational Technology & Society (IFETS)*, 8(2):54–65.