Embedding 2D Standalone Educational Simulation Games in 3D Multi-User Environments: The Case of C-VIBE

Thierry Nabeth, Albert A. Angehrn
INSEAD CALT (Centre for Advanced Learning Technologies)
{thierry.nabeth, albert.angehrn}.insead.edu

Abstract

This paper presents insights gained in the design and evaluation of C-VIBE, a 3D multi-user simulation game developed within VERDI (Virtual Environment for Real time Distributed applications over the Internet), a project exploring advanced applications running over a satellite network infrastructure, and in particular innovative solutions for management education. The paper concentrates on factors to be taken into account when developing 3D multi-user versions of existing, standalone 2D simulation games.

1. Introduction

Education-oriented, multi-user virtual reality applications can be considered very attractive both from a pedagogical perspective (they provide support for engaging experiential learning) and from an advanced learning technology perspective (they provide the opportunity to integrate several technologies supporting advanced user interface and multimedia content design, as well as distributed collaboration). Nevertheless, such applications are still considerably limited by complexity and cost factors associated with their development.

Designers of e-learning simulation games are therefore reluctant to engage in the design of such complex systems, preferring to design simpler versions (typically 2D and standalone) that are already challenging enough to elaborate. An additional factor discouraging designers is the lack of evidence that 3D multi-user versions are effectively ‘better’ (compared to more basic standalone 2D ones), increasing the learning value significantly enough to make it worth undertaking the necessary additional efforts.

In this paper, we address these questions by presenting initial insights gained in designing and evaluating a 3D multi-user version [1] of a successful computer-based simulation (the “EIS simulation” [2](http://www.calt.insead.edu/eis)) used to help business students and executives develop and improve their change management competencies.

First, we describe the 3D multi-user version of the simulation from a technical, usage and pedagogical perspective, illustrating its main differences from the original standalone 2D version. Second, we discuss how design complexity can be addressed by defining a highly distributed architecture enabling the simulation engine to be re-used flexibly to support eLearning solutions based on 2D, standalone as well as different multi-user 3D technologies.

2. VERDI and C-VIBE: An Overview

2.1. The Objective of the VERDI project

VERDI (Virtual Environment for Real time Distributed applications over the Internet) is a research project co-funded by the European Space Agency (ESA) whose broad objective was the demonstration that advanced satellite broadband multimedia platforms can enable developers of Internet applications and publishers of rich-media contents to address the broadband Internet services market. The VERDI project was articulated on three key technologies: (1) a multicast satellite-based network infrastructure (OpenSky), (2) a multi-user virtual reality server technology using multicast to optimize the sharing of information among distributed users (VRSAT), and (3) a VRML and Java-based technology (Cortona) allowing end users to navigate and to interact in shared 3D virtual worlds.

2.2. The C-VIBE Demonstrator applications: Change management in 3D

One of the main demonstration applications of VERDI is C-VIBE [1]: a virtual reality role-playing business simulation game in which learners, operating in
distributed teams, are challenged to implement a major innovation in a fictitious company in which managers (represented by virtual characters) display dynamically different forms of resistance to change.

C-VIBE relies on an experiential learning [3] conceptual framework called “business navigator” [4], organizing the knowledge as a set of interconnected levels (informational, organizational and physical). It is also based on an existing 2D standalone change management simulation game, the “EIS simulation” [2] (http://www.calt.insead.edu/eis) that has been developed over years at INSEAD CALT (Centre for Advanced Learning Technologies), and which is used extensively in top business schools and universities (MIT, Wharton, Stanford, etc.) and companies to teach interactively how to conduct change in organizations. This EIS simulation is a standalone application, run by teams of 3-6 co-located players spending 1.30h to 2.30h to accomplish the mission, supported by the simulation software but also by handouts, whiteboards and other traditional meeting support tools.

2.3. The user experience

The C-VIBE simulation starts when a user connects to the C-VIBE login window, joins a player team and selects an avatar. After the login procedure, users come to the "Main Window", which is a web page containing a 3D panel displaying the virtual world, a Chat panel, and an Information & work panel [5]. Figure 1 gives a snapshot of this main window.

Figure 1: The main user interface

In C-VIBE, the 3D world consists principally in a Virtual Board Room, in which the team members, represented by avatars, access information and execute a set of management decisions. This Virtual Board Room is organized as a set of walls supporting a set of functions such as: (1) getting access to information about the state of the “simulated” managers of the organization that the player’s objective is to convince; (2) getting information related to the different actions; (3) taking some decisions (some white boards are used to specify the actions to be taken and to specify the different parameters); (4) accessing overall information and results related to the whole organization (displaying the social networks of the organization, getting some overall statistics of the current adoption level, etc.). To add to the realism and to stimulate the engagement of the learners, the C-VIBE application pilots a special avatar, the ‘Assistant’ bot. The role of this bot is to introduce the game to the players at the beginning of the session, and to participate in the decision making: when a decision has been selected, the bot activates a small animation, and reports on the outcome of the initiative.

Finally, the C-VIBE 3D multi-user version of the simulation provides the support of distributed teams through a provision of mechanisms helping the coordination of the members of the team such as communication tools and voting tools.

3. The Design of the C-VIBE

3.1. C-VIBE computer application

Technically, C-VIBE is a client-server application consisting of a 3D virtual reality server and a set of client applications communicating dynamically with this server via the Internet. The client applications include two components: (1) an Internet browser application providing the main end user interface -this application integrates a VRML visualization panel with a set of Java applications used for providing information or real time communication; (2) the C-VIBE engine, which connects to the 3D server and implements the logic of the change management simulation game. The C-VIBE engine is able to capture the actions happening in the 3D world and in the 2D panels (used to display information and to communicate), and to act upon them by executing simulation-specific actions and displaying the results.

3.2. The Architecture

The architecture of the C-VIBE application consists of a set of distributed components articulated around the VRSat Virtual Reality server (figure 2).

This server uses multicasting to update the state of the world and distributes it efficiently to the different client applications (the end user browsing application and the C-VIBE application). In order to clearly
separate the game simulation logic from its visualization part, the EIS simulation software has been extended to operate in server mode via HTTP protocol, serving hence as a distributed game simulation engine.

At the beginning of each session, the C-VIBE application retrieves from the simulation engine all the necessary information related to the setting-up of the game. Further connections to the game engine take place every time the users trigger relevant actions in the Virtual Board Room.

Figure 2: The C-VIBE architecture

3.3. The Design

The design of the C-VIBE application has represented a real technical challenge given that some of the key technologies (OpenSky and VRSat) were also developed during the project, or were in a stage that can definitively not be considered as mature (VRML 2, Cortona, Microsoft Java virtual machine). Second, although the choice of reusing the existing game simulation engine helped to reduce the development effort, it also introduced some difficulties. For instance this engine was only available via the 2D version, and we didn't want to create a special version of this engine because this could have meant maintenance problems later. Besides, separating clearly the visualization layer from a pure logic layer is complicated, even if this represents an important research direction in Computer Sciences (in particular with the extensive work conducted around the MVC – Model View Controller – Framework). This is even more true in the case of a 3D visualization layer for which less experience is available.

4. Results and Conclusions

This approach has proved to be successful since we managed to create a system able to reuse the existing simulation kernel (and therefore reused the previous game engine component and maintained a unique version), and produced an architecture for which the different components could be flexibly distributed on the Net. This undertaking has appeared, however, to be more difficult than originally foreseen (immaturity of the technologies) but has proved to be a valuable experience of better understanding the technical, conceptual implications, and future possible evolutions. In particular, this project has demonstrated that it is possible to define architecture in the development of multi-user 3D applications on top of existing 2D ones without compromising the functionality provided to the users and retaining a maximum level of flexibility. This was done at the cost of an increased level of complexity of the overall architecture, but with the benefit of a much better modularity, flexibility in the deployment, and maintenance (only one kernel is used for the multiple versions). This design has also opened the possibility to design a system supporting multiple virtual reality platforms (a prototype based on the Active World virtual reality environment is currently under development), with a limited amount of effort.

5. References
