From Videos to Simulations: New Technologies in Training

Paper to be presented at Online Educa 2004
Berlin, Germany
1-3 December 2004

Frank Boosman
Chief Marketing Officer
3Dsolve Inc.

3Dsolve Inc.
1140 Kildaire Farm Road, Suite 200
Cary, NC 27511
919 469-9950 (office)
919 469-9954 (fax)
info@3dsolve.com

Introduction

Historically, video-based software training tools have often been used in training where interaction with other people comprises a large part of the subject matter. Examples of such subject matter include emergency response, customs inspection, facility security, and the like.

We are currently engaged in a project in which we are converting an existing video-based software trainer to a game engine-derived, simulation-based training tool. This paper describes our analysis of the development challenges of this conversion and our experience to date.

Goals and Challenges

The goals of this project are:

- **Reduce the long-term cost** of training materials development.
- **Create a more variable and realistic training environment** for students.
- **Increase the replay value** of training materials, encouraging ongoing usage.
- **Bring consumer-oriented interactive game design techniques** to training.
- **Provide trainers with a platform** for future simulation-based training development.

Two central challenges exist in this project:

- A **balance must be struck** between flexible interactivity on the one hand and ease of programming and validation on the other.
- The final software product must exhibit **sufficient rendering performance**—both in speed and in visual fidelity—on the target platform.

Interactivity in Training Today

To understand our approach to the challenge of interactivity, it is useful to briefly review the spectrum of interactivity in game engine-based training tools as it exists today.

At one end of the spectrum are highly interactive products such as first-person shooters (Unreal, Ghost Recon, Quake, etc.), flight simulators (both entertainment-focused and commercial- and military-grade), and other such simulations. In such products, players are given the maximum level of flexibility, including the finest level of control possible, over their ‘characters’ in the virtual worlds which they inhabit. For example, in games such as Ghost Recon, a player can move anywhere he or she likes in a world, with the ability to move as little as a few centimeters at a time—and moving just a few centimeters can mean the difference between life or death. Makers of such games are constantly striving to improve the fidelity of their simulations of the world, giving players the ability to perform more real-world tasks, and to perform them in as realistic a fashion as possible.

At the other end of the spectrum are branching-style products. These are not simulations, but rather branching stories. For each scenario, there is typically an opening scene and a small number of possible choices. Each choice may lead either to additional choices or to a final outcome for that scenario, with one correct outcome (and resulting video) per scenario.

Simulation-Based Training Concerns

For the purposes of this discussion, highly interactive simulations exhibit two primary problems:

- **Development costs for simulations are high** because the player (or student) must be able to interact with as much of the virtual world as possible. This means complex graphics, compiled code, and scripts; and it means pre-programming a variety of interactions between different objects in the virtual world—e.g., if a player throws a brick at a window, and the window 'breaks', it is because the developer anticipated and programmed that specific interaction. This very quickly can become prohibitively expensive.
- **Validation is difficult for highly interactive simulations** because there are typically an infinite number of states in which the simulation can exist. For training purposes, it is important to know that the student cannot get into a situation that would reward incorrect behavior. It is difficult (though not impossible) to ensure this is true in a highly flexible simulation.

Branching Video-Based Training Concerns

As for branching-style products, especially video-based products, they come with their own problems:

- **Lack of flexibility**. This comes from the script- and video-based nature of such products. Once a scenario has been filmed, it cannot be modified. For example, if a scenario is filmed during the day, it will always take place during the day—to give the student the option of participating in the scenario at night would require new video to be filmed.
- **Lack of extensibility**. If the creator of a branching-style product wishes to extend it with new scenarios, new filming is required. This may be cost-prohibitive, or even impossible if the original facility and/or actors are unavailable.
- **Lack of variability**. This results in training to unintended clues, and is an inevitable side effect of training tools that are based upon fixed scripts, especially when such tools use still images or video presentations. For example, if a blonde woman in a blue shirt has been infected with a biological agent, and must be quickly diagnosed and treated as such, and if a given student ever repeats that scenario, they can simply remember what to do for the blonde woman in the blue shirt, rather that properly diagnosing her. This can be ameliorated with additional

scenarios, but the nature of production—especially video production—makes this cost-prohibitive.

- **Lack of the routine**. Many jobs requiring training are, as one person once put it, "90 percent boredom and 10 percent sheer terror"[1]. The cost of developing traditional training tools, especially video-based tools, necessarily forces the focus on the sheer terror while leaving aside the boredom. This makes for an unrealistic training environment. For example, if one of the challenges of guard duty is in distinguishing the 1 visitor in 100 who should not be granted access, then an accurate training tool should provide 99 visitors who should be allowed to pass through.

Design Strategy

Our strategy for addressing the issues described above—the limitations of both highly interactive and branching-style simulations—is to build a fundamentally new type of game engine-derived, simulation-based trainer, one that is relatively inexpensive to develop and straightforward to validate, while at the same time flexible, easy to extend, variable, and incorporating routine interactions. Our approach is to create a larger number of variable scenarios that exist within a single, unified learning experience.

In our product, we are creating a fairly large number of scenarios, many of which are routine in nature, and only a small percentage of which are critical scenarios requiring the student to call upon all their knowledge. These scenarios can be thought of as "scriptlets," or miniature scripts. Students find themselves within the target environment. One at a time and in a semi-random sequence, the training tool plays back scriptlets within the environment. Students respond to the events in the scriptlets and receive feedback as appropriate.

Highly interactive simulations provide complex yet flexible interfaces, in which users can walk, run, crouch, shoot, hit, pick up, and perform a variety of other actions. As noted above, such complex user interfaces and the underlying simulations on which they are based pose problems with regard to development costs and validation. Branching-style trainers provide simple yet restricted interfaces, in which users choose one of a set of two or three options and then receive feedback on that option.

Our approach to the user interface is to provide a small number of actions that the student can take at any given time, and to provide these actions in a point-and-click manner. Students cannot go anywhere or do anything, but instead have access to a limited number of actions that might seem reasonable given their location and situation.

Conclusions

We cannot draw final conclusions about the ultimate success of our approach until the training tool is fully deployed and our customer has significant experience with it in the

[1] http://www.bartleby.com/63/92/1892.html.

field. That said, we are able to draw some preliminary conclusions based on development team experience and customer feedback to date:

- Game-engine derived training tools seem to provide a **better sense of place** than do their video-based equivalents. Given the obvious inherent quality of video footage, this seems counterintuitive until one realizes that the camera within video footage is fixed, whereas in a simulation-based training tool, if designed properly, students can have much more freedom to move and look within a space. This seems to dramatically increase their knowledge of the space even before entering it in the real world.
- Game-engine derived training tools are **far more extensible** than their video-based equivalents. Much of the investment in creating a simulation-based training tool—environmental artwork, character artwork, dialogue recording, animation sequencing, and so on—can be effectively reused, whether to update existing products (based on changing training requirements) or to create new extensions to existing products (based on expanding training requirements).
- Flexibility in development **can be a double-edged sword**. In the production of video-based training tools, the conclusion of filming is seen as a 'hard stop' by customers—the point after which the content cannot be changed without significant expense. In simulation-based training tools, a comparable milestone (in terms of the customer's perception) does not exist, and so this increases the requirement for the developer to finalize source content as quickly as possible and explain to the customer the implications of changes to this content.

About the Author

Frank Boosman is Chief Marketing Officer and a co-founder of 3Dsolve Inc. of Cary, NC. He served as the original product manager for *Adobe Acrobat*; designed *Tom Clancy SSN*, the first 3D submarine simulation game; co-founded pioneering game developer Red Storm Entertainment; and co-created *Tom Clancy's Rainbow Six*, the first realistic first-person tactical combat game.

About 3Dsolve

3Dsolve, The Simulation Learning Company, creates collaborative simulation learning solutions for government, military, and corporate applications, a market estimated to reach $6.1 billion by 2006. 3Dsolve's simulation learning products use realistic, interactive 3D graphics, based upon industry standards, enabling users to learn by doing. 3Dsolve has been named as one of *Military Training Technology* magazine's Top 100, the "companies that have made a significant impact in the military training industry."