Using Open Source Game Engines to Build Compelling Training Simulations

Perry McDowell, Rudolph Darken
MOVES Institute - Naval Postgraduate School
Monterey, California
mcdowell@nps.edu, darken@nps.edu

ABSTRACT

The arguments for increasing the amount of computer-based training are clear, convincing, and essentially over: almost every unit in the military is shifting its training strategy to include a larger percentage of computer-based training. The question now is how to create computerized training systems using the best and most cost-effective methods.

One solution is to increase the training done in simulators. Military simulator-based training has been used in substantial, expensive systems for over thirty years, first in flight trainers and later in vehicle simulators. Simulator training was limited to such large scale problems and solutions due to the cost of procuring, operating and maintaining these systems. However, with today's technology, simulators can be built for any area of the military, and operated on standard desktop computer systems or game consoles for a wider range of tasks.

In order to demonstrate such training systems' efficacy, we have built an application to train shipboard personnel in basic Damage Control, which is a task every shipboard Sailor must complete. The simulation is very similar to "First Person Shooter" games which are wildly popular with the military's primary demographic. We intend to conduct research using this system to determine its effectiveness in improving performance.

One of the most intriguing parts of this research is that this trainer was built with an open source game engine created in house. By using an open source engine, we have saved the licensing fees charged by large game companies, which normally run between $500,000 and $1,000,000. Additionally, this open source engine removes the legacy stovepipe that stunts the use of so many current training systems. This not only greatly reduces the initial cost, but significantly reduces the cost of follow-on because the application is not tied to a certain vender who is able to charge exorbitant rates.

ABOUT THE AUTHORS

Perry McDowell is a former Naval Nuclear Power officer. He has been on the faculty of the Naval Postgraduate School since 2000, where he teaches computer science and does research in virtual environments and training for the Modeling, Virtual Environments, and Simulation (MOVES) Institute. His research has been primarily focused on training in virtual environments with emphasis on creating simulators for Damage Control training. He is currently conducting research for his PhD. He graduated with a B.S. in Naval Architecture from the U.S. Naval Academy in 1988, and his M.S. (with honors) from the Naval Postgraduate School in 1995.

Rudolph Darken is an Associate Professor of Computer Science and a Technical Director of the Modeling, Virtual Environments, and Simulation (MOVES) Institute at the Naval Postgraduate School in Monterey, California. He is the Chair of the MOVES Curriculum Committee and directs the Laboratory for Simulation and Training. He also directs modeling and simulation efforts for the Center for Homeland Defense and Security. His research has been primarily focused on human factors and training using virtual environments and computer gaming media with emphasis on navigation and wayfinding in large-scale virtual worlds. He is a Senior Editor of PRESENCE Journal, the MIT Press journal of teleoperators and virtual environments. He received his B.S. in Computer Science

Engineering from the University of Illinois at Chicago in 1990 and his M.S. and D.Sc. degrees in Computer Science from The George Washington University in 1993 and 1995, respectively.

Using Open Source Game Engines to Build Compelling Training Simulations

Perry McDowell, Rudolph Darken
MOVES Institute - Naval Postgraduate School
Monterey, California
mcdowell@nps.edu, darken@nps.edu

BACKGROUND

Near the end of the Cold War, military training was significantly different than it is today. In the late eighties, almost all training was done in the field, at sea, or in the air. The presence of the Soviet Union as a potential foe drove large defense budgets which contained significant allocations for field exercises, steaming time, or flight time to train the military. Additionally, the presence of a major threat almost automatically overruled any other objections to training, such as environmental or noise concerns. Also, because there were only two major powers, it reduced the number of real world crises which required large deployments of troops, paradoxically giving them more time to train.

Since then, the Soviet Union has ceased to exist. This, among other factors, has led to military training budgets being slashed and many training areas being closed or having their use limited due to public outcry over the noise and/or environmental effects. Moreover, the U.S. military has been deployed on real world missions constantly since the beginning of Desert Shield in 1990, and the war on terror promises to increase the military's OPTEMPO in the future. While actual missions improve skills in those missions' areas, proficiency in other areas tends to lag since there is less time available to train. Obviously, these changes have forced the military to look to new methods of training to maintain combat readiness and effectiveness at high levels.

At the same time that military budgets were being cut, a remarkable rise in the capability of computer graphics occurred. Initially, this improvement was seen in the early 1990's mostly in large, multi-processor machines, such as Silicon Graphics' (SGI) Onyx or University of North Carolina's Pixel Planes. However, significant improvements soon spread to the home PC, driven largely by people's desire to play graphically advanced computer games, such as Castle Wolfenstein3D and Doom. The speed of graphics has improved incredibly in the intervening years. Tim Sweeny, chief executive of Epic Games, the maker of the UNREAL games, has estimated that graphics quality has increased 10,000 times since they started making games in 1991 (Takahashi, 2004). This new capability has pushed game developers to create bigger and better looking games, which has increased the sales of the video game industry above those of the motion picture industry, according to a study by Interactive Digital Software Association and Forrester Research Inc. (as quoted in Hong 2002), and Hong claims that the difference is widening.

These two facts, the changes in the military since the end of the Cold War and the explosion of the capabilities of the entertainment industry, have profoundly modified the computer graphics field. From when it started in the 1960's through the 1980's, most research and advancements were driven by government, primarily DoD, projects. The entertainment industry would look at the fruit of this research and try to determine what they could use for their purposes. In the 1990's, with the reduction in military spending and the rapid growth of the computer gaming industry, the roles were reversed. Most advances now come from the entertainment industry, and the defense industry looks to see what entertainment industry advances it can use for its own purposes (Capps, McDowell, and Zyda, 2001).

One other factor which occurred in the 1990's was the rise of the open source software (OSS) movement. Originally a small fringe movement, it has grown in recent years to become the software option of choice for many projects and applications. The best known example of OSS is the operating system Linux, which is being supported and marketed by some of the biggest computer companies, including IBM and Hewlett-Packard. Even agencies in the Federal Government are looking at implementing open source software; a recent report recommends that NASA move to open source software to save time and money and increase reliability (Moran, 2003).

We have combined these later two trends (advances in the entertainment industry and OSS) to build a prototype military training simulation to meet the DoD's new training needs.

SYNOPSIS

This paper will begin by describing why we feel that military training would be enhanced by using games to train personnel. It will then describe an open source game engine, Delta3D*, that we have developed at the MOVES Institute, and will conclude by describing a prototypical training game we have built to demonstrate the effectiveness of Delta3D and game based training.

GAMES FOR TRAINING

We believe that video (or computer) games are effective for the type of military training that will be required in the future for multiple reasons. Firstly, the military's roles have become more amorphous, requiring the ability to train quickly and effectively while deployed. Secondly, games are extremely well suited for training those coming of age now and in the foreseeable future.

Rapidly Shifting Roles

One reason why we see games as becoming a training medium is because military unit's roles have become much more ambiguous since the end of the 1980's. In the Cold War, every unit in the military had a well defined job, and it was rare for a unit to be required to perform outside its area of expertise. However, in the more amorphous war on terror, this is no longer the case. From the famous footage of special-forces soldiers going into battle on horseback with the Afghan Northern Alliance, to military police duties performed by many front line combat units after the fall of Saddam Hussein's regime, American forces had to quickly learn how to handle responsibilities and challenges far outside those for which they were trained (Miklaszewski 2004).

While these changes are happening to each of the military services, it is instructive to see how one service is reacting to them. Vice Admiral Albert Harms, responsible for all Navy training as the commander of the Naval Education and Training Command (NETC), has realized that the Navy must be much more adaptive. He is moving to modify Navy training to create the force needed to meet the challenge of rapidly changing combat roles. Because future Navy ships will have removable modules to allow reconfiguration for different missions and these ships will be manned by significantly smaller crews than current ships, each Sailor on these ships will have to be able to perform a significantly larger number of tasks than those on today's vessels. While in the past it was enough for a Sailor to be an expert in a single area of expertise, Harms predicts that tomorrow's Sailors will have to be experts in a wide variety of areas. Also, the knowledge that a Sailor needs a particular expertise may not be known until just before it is required. Harms refers to these Sailors as 'Hybrid Sailors,' and realizes that the Navy will have to use different methods to train them. Since the Navy can no longer make long term predictions on which tasks a specific Sailor will have to perform while deployed, training will have to become more embedded and deployable, able to train the Sailor closer to the time of the task. Harms also sees virtual environments and mission rehearsal playing a much more significant role in ensuring that 'Hybrid Sailors' are ready for their missions (Harms 2004).

Game-based trainers are ideally suited for this new type of training. They are easily deployable, being able to be deployed on systems as small as a laptop computer. Deployed troops will be able to design specific training scenarios for their particular task with a minimum of special training on the system. It will be easy to produce training games which can train a wide range of tasks, allowing the trainer to deliver "just in time" training to the person needing to learn a new task, such as VADM Harms' 'Hybrid Sailor.'

'Twitchspeed Generation'

The young people who are aged eighteen to twenty five the largest single demographic in the military today, have grown up in significantly different ways than previous generations. While it has long been stylish for older generations to point out the supposed differences between themselves and those currently coming of age, there is one factor which makes the difference between the current crop and their elders larger than any previous generation gap. This group grew up with computers and video games; they have never experienced a world where personal computers and game consoles were not in a vast number of homes. The video arcade was a favorite hangout and most have had computers in school for much, if not all, of their schooling. This group has experienced a completely different upbringing than any before them.

* Delta3D was code-named P-51 during its initial development.

Prensky calls them the 'Twitchspeed Generation' (Prensky 2000), and anyone who has seen the speed at which the thumbs and fingers of one manipulating a game pad certainly understands the term's origin. In the same work, Prensky points out how different their upbringing has been by assembling numbers produced by a number of studies. In his life, the average member of the 'Twitchspeed Generation' has:

- Spent 10,000 hours on video games;
- Exchanged 250,000 e-mails;
- Spent 10,000 hours on cell phones;
- Watched 20,000 hours of television;
- Seen 500,000 commercials; and
- Spent less than 5,000 hours reading.

Prensky also uses the term 'Digital Natives' to describe this group, since they have been raised since birth in the vernacular of computers and therefore have no "accent" the way those who learned computers after adulthood do. 'Digital Natives' have significantly different attributes than their predecessors as shown in Table 1 (Prensky 2004):

Table 1. Differences between Digital Natives & Others

Digital Immigrants	Digital Natives
Conventional speed	Twitch speed
Step-by-step	Random access
Linear processing	Parallel processing
Text first	Graphics first
Work oriented	Play oriented
Stand alone	Connected

These differences lead to significant variations between the members of this generation and those of previous generations. The military has already started to realize this; Brigadier General William Ard, Direction of Manpower and Organization for the United States Air Force put it this way: "This is the MTV generation, a generation used to seven second sound bites. How do we make the Air Force interesting to them?" (Ard 2004). VADM Harms has realized the need to capture the attention of the 'Twitchspeed Generation,' saying "Today's Sailors have different expectations than our generation. Are WE living up to their expectations?" (Harms 2004).

Their knowledge of video games is both a boon and impediment for implementing game based training systems. It is beneficial because these young people have played games extensively and have great experience in picking up a new game quickly. It is also helpful that they have a natural bent to play these games, and the games will attract them to spend even more time training – because they won't see themselves as "training." Instead, they will see themselves as "playing."

However, the difficulty in using video games for training is the level of sophistication of the 'Twitchspeed Generation. They have no patience for bad games, games which do not engage them or capture their interest. This makes it incumbent upon game-based trainer designers to ensure that the game aspect of their system is not overwhelmed by the training aspect of the system – they must remain in balance throughout.

An Example of An Effective Game

It would be useful to cite training effectiveness studies where several games were proven to deliver outstanding training benefit. Unfortunately, such studies do not exist. Most studies which have been done have relied upon anecdotal evidence ("I really think this would be effective" or "I learned a lot by playing the game") rather than empirical evidence. One such study (Beal & Christ, 2004) evaluated the Full Spectrum Command game at the Army's Infantry Captains Career Course at Fort Benning, Georgia. Despite there being no difference in performance between the students who played the game and those who did not, they decided that the game provided "tactical experiences with potential training value. It is also interesting to note that most of these officers are old enough to be on the cusp between 'digital immigrants' and 'digital natives.'

Since there are no studies which can point out the effectiveness of game-based military training, we will look to something similar. The United States Army has determined games were the preferred medium to get information on the Army out to a group of young people who had very little contact with anyone connected with the Army. Colonel E. Casey Wardynski, (Wardynski, 2004) the director and originator of the America's Army game, explains the rationale and the effects of using a game to inform the 'Twitchspeed Generation' about the Army:

> To counter (most young people's unfamiliarity with the Army), the game's originator reasoned that the Army would reduce search costs by framing information about Soldiering

> within the entertaining and immersive context of a game. This approach would also greatly reduce the assimilation costs of such information. Indeed, an official Army game with high production values would garner the interest of the large population of young gamers. For these individuals, having had no tangible contact with Soldiers, a game would provide virtual experiences and insights into the development, organization and employment of Soldiers in America's Army. In this way, such a game would place the Army in an accessible format and familiar environment for young Americans. In so doing, the Army would gain increased salience in their life-course decisions.… Thus, a well-executed game would put the Army within the immediate decision-making environment of young Americans. It would thereby increase the likelihood that these Americans would include Soldiering in their set of career alternatives….
>
> Due to its broad appeal, America's Army has found its way onto the computers of over 2.4 million registered users. As a result, a recent survey of the effectiveness of Army marketing and strategic communications found America's Army to be the Army's most effective medium for reaching young Americans. Indeed, the game engendered positive awareness of Soldiering among twenty-nine percent of young Americans age 16 to 24. *Importantly, the game achieved these results at an expenditure of about one-third of one percent of the Army's total marketing budget*. [italics added]

While the Army spends huge amounts of money to produce commercials, videos, printed material and web sites for recruiting, America's Army has been just as effective as *all of the rest* of the recruiting efforts in creating a positive image among the targeted demographic. This is a huge return on a relatively modest investment (approximately $2M/year).

Delta3D: AN OPEN SOURCE GAME ENGINE

Given the Navy's commitment to the use of games for training, it is obvious that future requirements will call for a myriad of applications for a wide variety of training domains. Our experiences with America's Army taught us an important lesson about the video game industry: It is structured around profit earned from selling games, not hardware or tools. This means that the business model of the game industry simply does not work for the Navy. The costs per application are way too high. Further, we anticipate the need for content reuse on a grand scale. Applications made by one developer will be built upon by another developer. Vendor "lock-in" will prove catastrophic. In order to achieve the Navy's vision of large-scale deployment of training games as a critical element of every warfighter's education and training, we must drive the costs of development (in dollars and time) down by an order of magnitude. We intend to achieve this lofty goal through the use of a modular open source game engine.

The development strategy of Delta3D is to maximize the use of existing source code wherever possible. We selected open source modules within the architecture that follow a "best of breed" approach. If at any time in the future a better component becomes available we can swap it in with minimal impact to the overall product. This allows us to be active contributors to the open source projects that we use while maintaining the development process for the unifying layer we produce. The open source modules we are currently using include:

- OpenSceneGraph (rendering)
- OpenAL (audio)
- Cal3D (character animation)
- Open Dynamics Engine (physics)
- Python (scripting language) and
- FLTK (GUI)
- XRTI (HLA interoperability)

among others. Delta3D consists of a thin unifying layer that standardizes the API and deconflicts the underlying modules where necessary (see Figure 1). The Python scripting language links directly to the C++ API. Delta3D also includes level editing capabilities and content development tools. The objective is to focus resources on content, not architecture, and to maximize reusability. This is the best business case for the Navy.

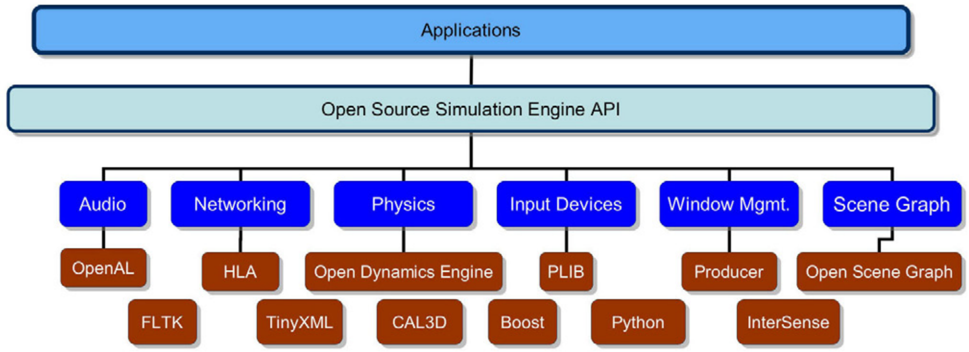

Figure 1. The Delta3D modular architecture

Delta3D is distributed under the GPL license and is freely available. Applications that are built in Delta3D are not expected to contain any proprietary source code so that they may be shared and extended as needed. The general game engine specification was written by NIST in cooperation with NETC to develop an unbiased view of what the Navy needs now and in the immediate future.

PROOF OF CONCEPT

NETC approached the MOVES Institute in October of 2003 and asked us to build a prototypical training game for demonstration in their booth at I/ITSEC 2003. While this time frame was very short (six weeks), we decided to accept the challenge and see if we could build a prototype trainer to demonstrate what a Navy training game might look like.

We had two concepts we hoped to prove by building this trainer. The first was to demonstrate that the Delta3D game engine, even at an early stage of development, was suitable to build a robust game. The second was to show the efficacy of a game in training military personnel in an important task. In this section, we describe the manner we attempted to prove both of these concepts.

In order to demonstrate the efficacy of the Delta3D engine in building games for military training, we constructed a prototype for training shipboard personnel in firefighting. This application was chosen for a number of reasons. Ships are highly complex visual settings, requiring a large number of polygons, textures, and advanced rendering techniques in order to produce realistic looking environments in real time. Testing the Delta3D engine in such an environment demonstrated its effectiveness for almost any other area that might be required. Also, a firefighting trainer required a large and varied group of features, such as smoke, fire, and water, which would test the readiness of the engine for a wide range of applications. Additionally, a firefighting trainer required multiple interactions between the trainee and the model which would test Delta3D's user interface.

A firefighting trainer also lent itself well to demonstrating the second concept of games as training media. One reason is that firefighting is a highly relevant activity: every Sailor stationed on a ship needs to qualify in damage control, and firefighting is a major part of that. A second reason is that firefighting consists of very definite tasks (which are described in the "Firefighting Tasks" section below). This made evaluation of the trainee significantly easier, since each task was either completed or not. While there are often subtle features to firefighting, the short time frame precluded trying to model and evaluate them, so we went with a simple "go/no go" criterion.

Another reason firefighting was chosen as the subject of the game is that the MOVES Institute has experience in building firefighting simulators (King & McDowell, 1995). It also currently includes personnel who can act as subject matter experts in the field. In addition, MOVES Institute personnel are working on other projects which require close liaison with the Afloat Training Group – Pacific (Damage Control) and Fleet Training Service Center – Pacific (Firefighting Training), allowing evaluation by those most likely to use such a trainer at an early stage of development.

Firefighting Tasks

We chose to create a game which simulates a class 'B' (flammable liquid) fire in number three generator room (3-370-0-E) on a DDG-51 class destroyer which the trainee has to extinguish. We chose a DDG because it is one of the newest and most common classes of the Navy's current ships. A class 'B' fire in an engineering space was chosen because it is the biggest and most complicated firefighting threat, in addition to being the one most commonly evaluated by inspection teams. Number three generator room was chosen because it was a very complex space which had a high likelihood of having a class 'B' fire, but was simpler to model than the significantly larger main engine rooms.

Shipboard firefighting requires a multitude of tasks to be completed, and certain aspects of firefighting should be approached in a "checklist" type manner. Normally, firefighting is a team endeavor, carried out by teams of approximately ten to fifteen individuals, under the control of a team leader who falls under the guidance of a repair locker. However, because of the short time frame between idea and demonstration, we greatly simplified the actual tasks required to fight a class 'B' fire down to seven, which would be completed by one individual. In this training game, the seven tasks to be completed by the trainee to be considered to have successfully combated the fire are:

1. Don a firefighting ensemble.
2. Don a self contained breathing apparatus (SBCA).
3. Trip the gas turbine generator's fuel valves.
4. Activate primary halon.
5. If primary halon is ineffective, activate secondary halon.
6. If secondary halon is ineffective, use the reentry hose reel to enter the space.
7. Extinguish the fire.

This game is designed to either teach the trainee these tasks or to evaluate the trainee's performance of the tasks. To accomplish this, the game has a menu tasklist which can be displayed with a keypress in the teaching mode (see Figure 2a-d). Once the trainee has completed the task, the task is "grayed out", indicating it has been completed and no longer needs to be accomplished. This feature is disabled in the evaluation mode.

The game begins with the standard game introduction as the camera pans around the outside of a DDG at sea, where the user's position (nozzleman on hose number one of the At-Sea Fire Party (ASFP)) and duties are described. After this intro, the camera flies into the ship outside repair locker three and gives the trainee control of the game. The ship's announcing system calls away the class 'B' fire in number three generator room and tasks the ASFP with responding to the casualty. At this point the user should move to the repair locker and carry out the seven tasks to complete the casualty.

Evaluation and Debrief

After all tasks are complete or the trainee exits the scenario, he is apprised of how many of the tasks were completed and which tasks were not correctly completed. One advantage of the game environment over traditional fire shipboard drills is that the Delta3D game engine records all actions in an XML file so that the trainee can see every one of the scenario's events. Although some first person shooter games allow replay of a short portion of a game, Delta3D was designed with military training in mind and allows the trainee to see exactly what actions he performed during the entire scenario.

This evaluation method is exceedingly simple, which is again a reflection of the fact that the time frame of this project was very short and evaluation was not a priority in building this application. However, computer based training, and games based training especially, will never succeed without a good, automated evaluation system. Eventually, we see this training device using an evaluation and tutoring system comparable to that being built at Stanford University for a similar domain. A group lead by Stanley Peters has built an evaluator/tutor for the performance of shipboard Damage Control Assistants, whose duties are comprised of many tasks. Their tutor is designed to perform an in-depth after action report with the trainee using natural language. (Pon-Berry, Clark, Schultz, Bratt, & Peters, 2004).

FUTURE WORK

Since we completed this project, we have continued work on Delta3D, and Version 1.0 is being released here at I/ITSEC 2004. We anticipate continuing development of the Delta3D game engine, using improvements made by the open source community to enhance the engine at low cost to the military.

We also have a large list of improvements to the firefighting trainer to make that system ready for deployment as a trainer. These include:

- Networking. (The capability to network exists in Delta3D, but was not implemented in this trainer.)
- Animations to represent other Sailors and equipment.
- Additional casualties, spaces and ship classes.

- An improved evaluation system.

Figure 2a. Outside the generator room.

Figure 2b. Immediately prior to entering the room.

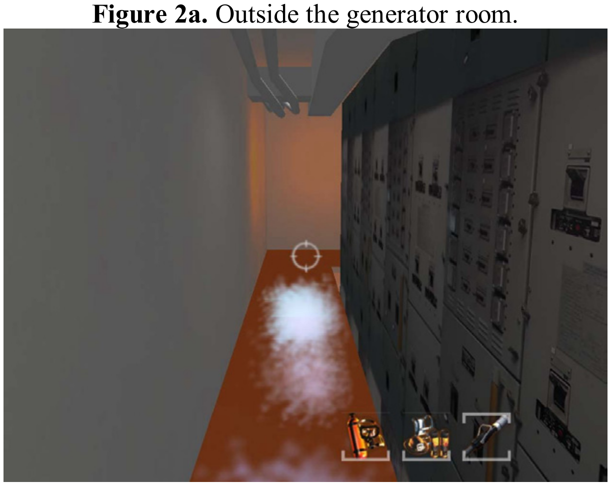

Figure 2c. Beginning to fight the fire.

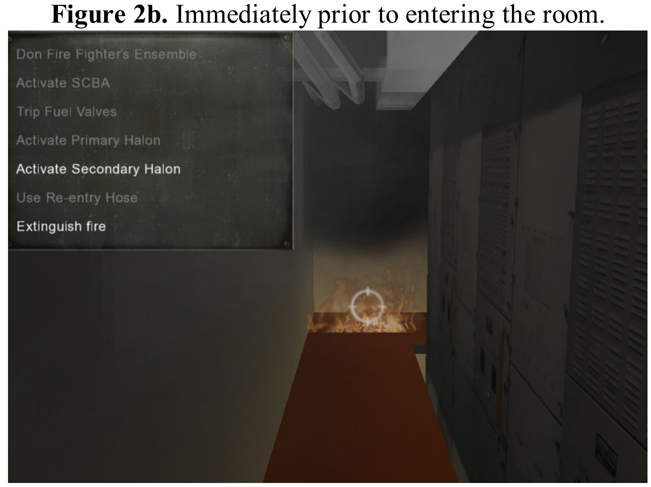

Figure 2d. Checking the task list (upper left)

ACKNOWLEDGEMENTS

We would like to acknowledge the sponsors whose funding supported this work, the Naval Modeling and Simulation Office (NMSO) and the Naval Education and Training Command (NETC).

REFERENCES

Ard, W. P. (2004). Remarks to the Navy Workforce Research and Analysis Conference, March 29, 2004, Arlington Va.

Beal, S.A, & Christ, R.E., (2004). Training effectiveness evaluation of the Full Spectrum Command game, technical report for the US Army Research Institute for Behavioral and Social Sciences, Ft. Benning, Ga.

Capps, M., McDowell, P., & Zyda, M. (2001). A future for entertainment-defense research collaboration, IEEE Computer Graphics and Applications, Jan-Feb, 2001.

Harms Jr., A. G., (2004). Investing in people, presentation given April 6, 2004, Pensacola Fl.

Hong, J. (2002**).** BCD Forum: The state of the online game industry, presentation delivered September 18, 2002, Halifax, Nova Scotia.

King, T. E., & McDowell, P. L., (1995). A networked virtual environment for shipboard training. Masters thesis, Naval Postgraduate School, March 1995, Monterey, Ca.

Miklaszewski, J (2004). War boosted Rumsfeld's mission, Retrieved June 20, 2004 from http://www.msnbc.msn.com/4400657.

Moran, P. (2003). Developing an open source option for NASA software, NASA Technical Report 003-009, April 21, 2003, NASA Ames Research Center, Moffett Field, California.

Pon-Berry, H., Clark, B., Schultz, K., Bratt, E. O., & Peters, S. (2004). Advantages of spoken language interaction in dialog based tutoring systems, Proceedings of ITS 2004, 7th International Conference on Intelligent Tutoring Systems, August 2004.

Prensky, M. (2004). Digital Game Based Learning, New York, McGraw-Hill Trade.

Prensky, M. (2004). Game design – A new language for communicating ideas, Presented at NASAGA 2004, March 11, 2004.

Takahashi, D. (2204). The new face of gaming, SiliconValley.com, retrieved June 17, 2004 from http://www.siliconvalley.com/mld/siliconvalley/news/local/8943813.htm.

Wardynski, E. C., (2004). Informing popular culture: The America's Army game concept. In M. Davis (Ed.), America's Army PC game: Vision and realization, (pp. 6-8). Monterey: U.S. Army and MOVES Institute.

http://owl.english.purdue.edu/handouts/research/r_apa.html#Examples