REAL-TIME WARFARE SIMULATION GOES WEB 2.0

Gustavo Henrique Soares de Oliveira Lyrio
Roberto de Beauclair Seixas

Institute of Pure and Applied Mathematics – IMPA
Estrada Dona Castorina 110, Rio de Janeiro, RJ, Brasil 22460-320
e-mail: glyrio,rbs@impa.br

Computer Graphics Technology Group – TECGRAF
Pontifcia Universidade Catlica do Rio de Janeiro – PUC-Rio
Rua Marqus de So Vicente 255, Rio de Janeiro, RJ, Brasil 22453-900
e-mail: glyrio,rbs@tecgraf.puc-rio.br

KEYWORDS
War gaming, serious gaming, real time tactics, WEB 2.0

ABSTRACT

This work presents the authors' experience in the process of developing a massive real time multi-player for military tactical online training simulations, and how some web 2.0 applications provide helpful tools to overtake common problems found during this process.

INTRODUCTION

As long as organized conflicts are known in the history of humanity, there are also war gaming. These games were used for training or entertainment.

Although as popular as table-top games, it has taken to tactical war games a little longer to be available for computers. Its large number of units and the advanced set of rules demanded processing power beyond the capabilities of hardware available at the moment. Also, the most established set of rules were made for turn-based war games so, it took a while to translate these rules in real time and also to release the first real-time tactics (RTT) (Adams 2007) computer game.

Real-time tactics is a sub-genre of tactical war games. It is differentiated from real-time strategy (the most popular) due to the lack of resource micro-management and base building, the greater importance of individual units (Adams 2007) and the focus on battlefield tactics.

With the evolution of the hardware resources and the popularization of the Internet, war games became available in online paradigm, where a player create a server game with its own world and other few players connect to this server and play together. It didn't take too much time to the players to demand a unique persistent world, where all the players could play together. The Massive Multi-player Online games (MMO) were born.

About ten years ago, a new way to use the internet was arising. Called "Web 2.0", this second generation of web applications facilitates communication, secure information sharing, interoperability and collaboration on the World Wide Web.

In the last twelve years, we have been working with the Brazilian Marine Corps, in the development of their training system. In the beginning, the goal was to simulate an Amphibian Assault Operation. With the expansion of the system to cover other Brazilian Marines' operations, like riverine operations, amphibian incursions and peace operations (military operations other then war), came the demand for new technologies to provide more details.

Then, it came the idea of joining the concepts and technology used among Web 2.0, MMO and tactical war games to develop a Massive Multi-player Online Real Time Tactics (MMORTT) training tool adding the necessary portion of reality to reach the training goals.

The goal of this paper is to present the adversities found in development of a MMORTT military training system and to show how some Web 2.0 applications could take part in this system providing useful solutions.

COMMON ADVERSITIES FOUND IN DEVELOPMENT PROCESS

During the development process three major adversities where identified: real terrain and 3D units modeling, development of a specific communication system and keep track of units actions.

3D Real world and units modeling

Differently from an entertainment game, the user will be expecting a real world. The training should be done in some place that really exists in a known region of our world. Otherwise the system will lack of reality. That is a huge problem to developers because the system will demand 3D modeling of all the training regions with its characteristics (mountains, rivers, lagoons, swamps and vegetation, for instance).

Real terrain demands real edification, bridges, and roads, which will be another problem. Again the designers will have a lot of work to do. Also you will need units that user can identify as real military units.

- **What we have learned**: Real world means real terrain, real scructures and real units 3D modeling;

Communication system

An additional problem arises from the fact that you are training a group of persons together, not entertaining isolated players. Since the results will be evaluated at a team level those persons, of course, need to interact with it each other. In the real world it would be done by radio communication, with a lot of features like static, interference and electronic warfare, so we need to simulate a radio communication system.

- **What we have learned**: Real world also means real communications with all real features and real problems;

Units line of actions tracking

The last problem, which we have discovered with some field experience, is that after the training is done, a debriefing is made and the users has to discuss about the actions each one took during the game. If it was a long game (more that one day, for instance), the large amount of information and the high level of interaction between users, makes it hard to determine why a specific action has been taken in a specific moment. It would be easier if each users had something like a journal to write down, and explain their line of actions. Such procedure is know on ships as the captain's diary.

- **What we have learned**: Discuss the game result is more important then the game itself;

WEB 2.0 SOLUTIONS

In the previous session we identified three major problems in the process of developing a MMORTT for military training. We next present three solutions provided by Web 2.0 applications that will make the developer's life easy when dealing with these problems.

Terrain and 3D unit 3D modeling with Google Earth API

The google earth API was chosen to handle this issue due to the fact that it alone, handles all the 3D rendering, dismissing all the design work.

In the middle of 2008 the Google company has made available for developers an API for including Google Earth system inside a web page. The benefits from this API are notorious. The developer uses javascript to gain control of the Google Earth plugin and its capability to display a 3D model of the whole world with no cost. Also it is possible to personalize the system with images, 3D models and XML/KML files.

Google has also provided Google SketchUp, a 3D software tool that enables to draw and place models in real world coordinates using Google Earth API. Users are encouraged to make their models available in Google Earth Warehouse, madding huge the probability to someone already has modeled that bridge in your terrain or even that military unit you need.

Below we will show some of the benefits that Google Earth API can bring to the development process of a MMORTT.

Aerial Reconnaissance
Aerial reconnaissance is a system where the user fly through a real terrain and try to detect the positioning of enemy forces. This task showed to be complex due to the necessity of builds a 3D model of every area of training and of each military unit. Terrain modeling can be a real headache when working with huge terrains. It demands a modeling team with designers and developers. Rendering algorithms with culling, LOD and other techniques. Would not be amazing if all that could come ready to use in a single API? Well, it became possible with Google Earth. The Figure 1 shows the reconnaissance system developed using a raster image of a UH helicopter cockpit (with transparent glass) above the Google Earth API. It took about 30 minutes to develop such application.

3D Visualization
The Brazilian Marines instructors complained about the difficulty to understand the line of sight and communication obstructions using a 2D map. They requested a way to really see the game instead of little marks on a map. In addition, a 3D view always makes a game more impressive. Again the same problems discussed above in the Aerial Reconnaissance session came back. Fortunately Google API has resolved it for us providing a tool called Google SketchUp.

Google SketchUp is a tool that allows its users to create

Figure 1: The aerial reconnaissance system using Google Earth API to display an raster image with transparent helicopter cockpit above the Google Earth terrain

3D models and made it available on the Internet. This models are fully supported by the Google Earth API (Figure 2).

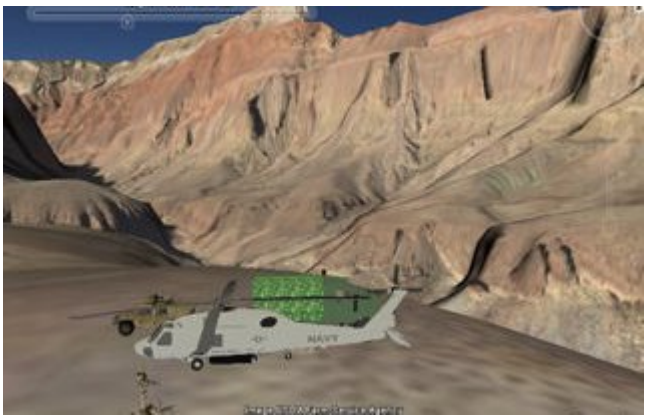

Figure 2: 3D visualization showing some Google SketchUp models (downloaded) placed over a real terrain (Grand Canyon). The terrain model is provided by Google Earth API

Communications using Team Speak SDK

Our simulation communications are done with three real radios installed on each simulation room: one for tactical issues, one for logistic requests and one for request fire support. Although the system considers electronic warfare in unit-unit and units-user communications, the communication between two users is done directly by those radios and doesn't pass through the system. The idea is to replace those radios by software using TeamSpeak server and developing an interference module bringing the effects of electronic warfare to the user-user communications also.

TeamSpeak is a software that enables people to speak with each other over the Internet. It consists of both client and server versions. The server acts as a host to multiple client connections, capable of handling literally thousands of simultaneous users. A relevant feature the TeamSpeak present is the ability to create channels. A channel works exactly as a radio channel does. One user creates a channel and may turn it public or protect it with a password. Other users connected on the server may then connect to this channel (providing the password if it has one). TeamSpeak became popular with the Counter-Strike game as a "radio" system used by the players to manage their teams. Another possible solution is to use Skype.

Unit journals through Twitter or Blog

Tracking a combat unit progression over the simulation is important to allow the instructor to make a debriefing and points out to the players which actions went wrong, what should be improved or even to show a great maneuver. However, it has been a hard task. The system needs to be able to store each action taken by each unit. This kind of storage demands a lot of work to detach what was the action line of a particular unit.

Twitter is a free social networking and micro-blogging service that enables its users to send and read other users' text-based posts of up to 140 characters, which makes it fast to read. The posts are known as tweets and are displayed on the user's profile page and delivered to other users who have subscribed to them (known as **followers**). Users can send and receive tweets via the Twitter website, Short Message Service (SMS) or external applications.

A proposal to solve our problem is to generate a twitter account for each unit, where the unit by itself, through the twitter API, informs which actions were taken in a scenario when looking for a goal.

For a debriefing process instructors can subscribe as followers of any unit and follow the unit progression with the 3D visualization tool, and consulting the unit tweets to determine why an unexpected action was taken. This process increases the capability of identifying wrong actions taken by the officers or even registering outstanding unexpected maneuvers that should be shared and discussed with other officers.

Recognized potential in Web 2.0 tools

The United States Army and Air Force saw potential in Twitter and other Web 2.0 tools. Recently, the news has reported that these tools have been used by the military to communicate with people that uses the Internet as alternative source of information to the press media in Afghanistan. One of the goals is to oppose the Taliban

advertising.

RESULTS

Using the Google Earth API we were able to develop a 3D visualization and a 3D aerial reconnaissance tool, solving problems that have persisted in the project for many years with no more than 150 lines of codes each. In addition, the visual including 3D models, water movement and sun light effects caused a good impression to the users. The integration with the whole system was also easy, since it was only necessary to call a new function that builds an html page with Google Earth Javascript inside. The whole development process took no more than one hour.

The process of developing radio communication by software using TeamSpeak was slightly more complicated due to the necessity of acquiring data for each type of radio used by the Brazilian Marine Corps and to implement interference and electronic warfare over the TeamSpeak SDK. The idea was also well received by the officers and is currently in a process of evaluation and homologation.

Using Twitter as a track tool for the units was praised by the instructors as an alternative to an email system that has been font of problems and missed units' information for a long time. The tool is also in evaluation. As occurs with the radios, replacing something that has been used by twelve years requires some time to settle down as the new way to do the job.

DISCUSSION

Unfortunately Google Earth API still presents some issues that prevents its use in a training application. The first one is the lack of a Google Earth plugin for Linux. The Brazilian Marine Corps is on a process of free software adoption and would be bad to demand lots of Windows or Mac licenses to support the system. Other issue that demands attention is the time that Google Earth spends to render. In a commercial game it would be not acceptable to wait the rendering process looking the world slowly appear in front of your eyes. Additionally, there are some flickering problems during the render process. An example is when the camera gets a huge height related to the ground.

However, no one can deny that Google Earth API has a promising future. Google Team is working hard to solve these issues and to implement new features to keep the API up to date with the Google Earth software version. The last huge improvement was the ability to go "underwater" available from version 5. This feature inspired us to start working in the 3D representations of submarines and combat divers.

After all, we must consider that Google Earth API is just a one year old API and the contributions that it has made are huge. We believe that in a few years Google Earth API will be "the direction" to follow when doing real terrain rendering.

REFERENCES

Adams D., 2007. *The State of the RTS.*

Graham P., 2006. *Web 2.0.*

Montenegro R.S.R.S.A., 2009. *Objects Visualization in Digital Terrains Using Adaptive View-Dependent Techniques.*

O'Reilly T., 2006. *What is Web 2.0.*

O'Reilly T., 2007. *Web 2.0 Compact Definition: Trying Again.*

Seixas G.L.R., 2004. *Riverine Operations: Modeling and Simulation.*

T. B., 2003. *Strategy Game Programming With Directx 9.0.* Wordware Publishing, Inc.

BIOGRAPHY

Roberto de Beauclair Seixas works with Research and Development at Institute of Pure and Applied Mathematics - IMPA, as member of the Vision and Computer Graphics Laboratory - Visgraf. He got his Ph.D. degree in Computer Science at Pontifical Catholic University of Rio de Janeiro - PUC-Rio, where he works with the Computer Graphics Technology Group - TecGraf. His research interests include Scientific Visualization, Computer Graphics, High Performance Computing, GIS, Simulation Systems and Warfare Training Games. Currently he is the advisor of the Warfare Games Center of the Brazilian Navy Marines Corps.

Gustavo Henrique Soares de Oliveria Lyrio works with the Computer Graphics Technology Group - TecGraf. He got his B.Sc. in Computer Engineering at Pontifical Catholic University of Rio de Janeiro - PUC-Rio. His interests include Computer Graphics and Warfare Training Games. Currently he is developer of Warfare Games Center of the Brazilian Navy Marines Corps.