A VIRTUAL REALITY SCENE GENERATOR THROUGH OPEN COMMERCIAL STANDARDS

W. Garth Smith
MetaVR, Inc.

http://www.metavr.com

KEYWORDS
Microsoft: DirectX, Direct3D, Direct Sound, Visualization, PC, Multiplayer, DIS, 3Dfx

ABSTRACT

Commercially available packages for Virtual Reality applications typically attempt to serve the widest possible audience. Consequently such packages are large, unwieldy, and unsuited to real-time applications such as high-fidelity, multi-player, low-cost, networked worlds. Moreover, many such packages are written using proprietary software libraries. A small, compact software package based on open commercial standards is needed for applications that require high speed and a high degree of reconfigurablity among low cost personal computer (PC) platforms so as to enhance simulation interoperability (Ref. 1). The industry trend of real-time graphics is the confluence of commercial gaming and visual simulation needs into a single set of requirements. There are many technical as well as market driven reasons for choosing which approach will allow the greatest amount of capitalization upon existing commercial work so as to lower the cost of visualization applications.

This paper discusses an implementation of a low cost Virtual Reality Scene Generator (VRSG™) application providing real-time textured visualization on a PC. Specifically, the VRSG system provides visualization of entity interaction and weather phenomenon on a Windows/Intel platform at a 30 Hertz update rate. The application uses free and open commercial standards to visualize up to 120 simultaneous ground and air vehicles on a high-fidelity terrain database fully inter-operable with the Modular Semi-Automated Forces (ModSAF). Design considerations for migration to the High Level Architecture (HLA) will be discussed in terms of maintaining real-time performance as this application currently uses the Distributed Interactive Simulation (DIS) 2.0.4 protocol. The advantages of such open standards lead to application capabilities such as near real-time terrain database generation from distributed government data sources, entity threat visualization, and weather visualization from environment servers.

1.0 ENABLING CRITERIA FOR PC-BASED REAL-TIME VISUALIZATION

A series of commercial strategies and technologies are required to achieve visualization on a personal computer (PC) that is sufficiently sophisticated to meet the needs of the simulation community (Ref. 2):

- Commercial software development kit (SDK) that includes all source code and documentation without licensing schemes
- A ubiquitous simulation host architecture and operating system to take advantage of economies of scale
- Low cost three dimension graphics accelerator cards that support at least: 800 by 600 pixels resolution, one million triangles/second of textured polygons, and 45 megapixels/second sustained fill rate
- Software development tools that support C++
- Host hardware platforms that are open to the extent that they can be upgraded via Web or mail order houses independent of a particular vendor
- Plug-compatible simulation hardware host components such that vendors components are readily interchangeable
- Simulation hardware host upgrade paths that support upgrading isolated components for significant performance increases
- Software API that supports extensive networking capabilities for multi-player interaction
- Software API for adding user controller hardware devices

The Winter of 1997 will be remembered as a watershed moment in the PC based simulation market as all of the aforementioned capabilities have come to market. This paper will discuss an implementation that attempts to maximize PC based multi-player visualization capabilities in both the technological and commercial sense. Prior to development of this implementation, particular emphasis was placed on building a software infrastructure that would afford the resulting application the greatest possible longevity to the community.

2.0 API FOR RENDERING REAL-TIME THREE DIMENSIONAL GRAPHICS

There exists a multitude of software APIs that support three dimensional (3D) real-time rendering, each with its own set of advantages and disadvantages. It is not enough to focus development on a given API that, at the moment, is considered technically superior. It is equally vital to choose an API that will evolve and has broad acceptance in the developers community. The following set of criteria was established for choosing a 3D API.

- Available via the World Wide Web as a free resource with no run-time or developers licenses
- Extensive documentation
- Broad industry and developer support
- Significant enhancements for each release so as to take advantage of the rapid cycles of hardware improvements
- Sufficient performance capabilities when compared to proprietary applications

Microsoft® Direct3D™ (D3D) is believed to be the only API to meet all of these criteria. D3D (Ref. 3) is a subcomponent of the of Microsoft DirectX™ interactive media technologies, which include the DirectDraw™, DirectSound™, DirectInput™ and, DirectPlay APIs, offering software developers a comprehensive set of interactive media services. Choosing the D3D API was expected to yield the additional benefit of significant inter-operability with other DirectX components. For instance, the sound generation environment (DirectSound) and networking API (DirectPlay) are meant to be highly inter-operable with D3D thus reducing development time for additional functionality beyond the graphics requirements.

3.0 SIMULATION HOST ARCHITECTURE AND OPERATING SYSTEM

The WinTel environment was then identified as the simulation host and operating systems of choice given the availability of the Direct3D API on such an extensive installed base of Windows 95 and NT based Intel hardware. A significant consequence of this choice is the enormous capacity for reuse that the WinTel platform provides for tasks other than simulation. These platforms are typically used for presentation creation and analysis reports related to simulation exercises. The advantage of having such ready access to the simulation environment is a benefit to the target user audience which consists primarily of simulation professionals.

4.0 3D GRAPHICS ACCELERATOR CARDS

As the spector of a high volume consumer market for gaming and visualization becomes tangible, numerous companies have become focused on producing 3D graphics accelerator cards (Fig. 1).

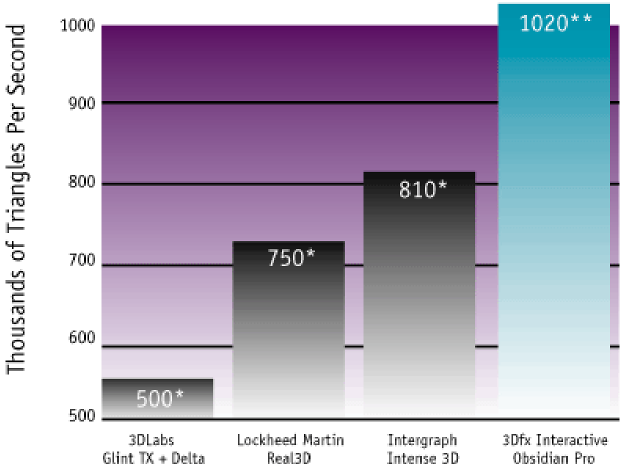


Figure 1. Performance comparison between a number of graphics accelerator manufacturers.

The industry for PC-based visualization is undergoing a remarkable convergence of requirements for what were formerly the two separate realms of the gaming and simulation industries. Traditional high end military simulations are the ideal candidate for the ravenous, fidelity-driven gaming community. Until now, the cost of such military-style, high-fidelity simulations has been unacceptable to a commercial community accustomed to shrink-wrapped $59.00 U.S. games. In the very near future it is believed that the distinction between game and simulation visualization requirements will disappear.

The simulation community has formally benefited from a series of graphics accelerator offerings that begin to meet the lower end of visual simulation requirements. The heuristics threshold requirements established here for creating a virtual reality scene generator (VRSG) based on the DIS requirements were largely generated by a series of failed attempts at developing real-time visualization with one new graphics card after another received by overnight shipping services. It can be difficult to ascertain from a specification sheet the actual performance of a graphics card as capabilities are interpreted to the advantage of each manufacturer; it was for this reason that an anecdotal evaluation process was chosen rather than relying solely on manufacturers claims.

Graphics accelerator cards based on the 3Dfx Interactive's Voodoo Graphics™ chip set were chosen after bench-marking a number of manufacturers' devices against the VRSG software. The Obsidian 4220 enables applications to dynamically manipulate fully shaded, texture-mapped, LOD MIP-mapped, filtered, Z-buffered models and animation effects (Fig. 2).

Obsidian 50:4220	
Number of pixelfx chips	1
Total frame buffer memory	4 Megabyte
Maximum resolution: double buffered with Z and 24-bit effective color	800 X 600 pixels
No. of texelfx chips	2
Texture memory per texelfx	2 Megabyte
Total effective texture memory	4 Megabyte
Sustained bi-linear fill rate	45 Megapixels/sec
Sustained tri-linear fill rate	45 Megapixels /sec
Maximum sustainable textured triangle	1.25 Megatriangles/sec

Figure 2. Specifications for the 3Dfx Obsidian 50:4220 3D graphics accelerator card.

5.0 THE SIMULATION HOST HARDWARE AND DEVELOPMENT ENVIRONMENT

The Obsidian graphics accelerator board is PCI bus 2.1 compliant and supports DOS, Windows® 95 and Windows NT environments on either the Pentium™ (P5) or Pentium Pro™ (P6) Intel processors. The graphics mode operates transparently with VGA and Windows accelerators in dual or single monitor configurations via analog pass-through. 32 Megabytes of RAM are required on the simulation host for sufficient operation of graphics processing. This configuration of the simulation host in terms of first quarter 1997 prices places a low end hardware configuration in the price point range of low cost home systems of approximately $3,000.00 U.S.

The development environment was chosen on the basis of the most prolific resources for an object oriented language. Hence, the Microsoft Enterprise Edition of Visual C++ version 4.2 was chosen as the software development environment. This version is bundled with release applications such as InstallShield and the configuration management tool Source Safe that further adds value to the development process.

6.0 THE VIRTUAL REALITY SCENE GENERATOR (VRSG)

The virtual reality scene generator was designed to combine all of the aforementioned capabilities and strategic development direction into a low-cost, multi-player application based on the DIS protocols and entity interaction model (Fig. 3).

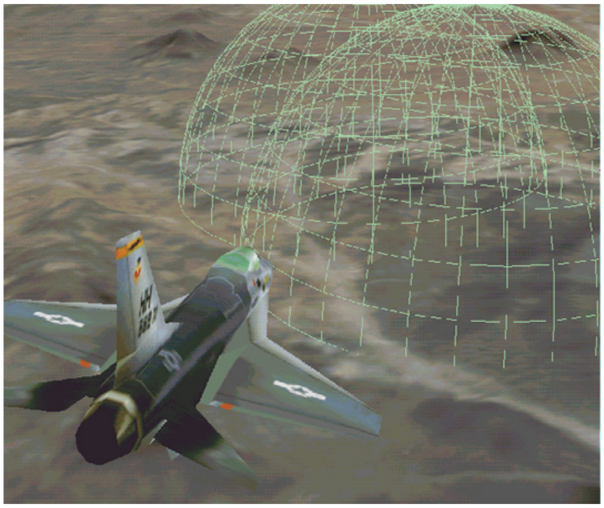


Figure 3. The Virtual Reality Scene Generator (VRSG) displaying a threat sphere to an approaching F-18 for mission planning on a P6/200 at 30 Hertz.

The meta-design philosophy, however, was to establish a software infrastructure that would provide arbitrary viewpoint management in a virtual world. A DIS specific application known in the vernacular of the industry as a "Stealth" and, in this case, a "PC Stealth," was developed for the initial implementation.

The general functionality of the VRSG was determined on the basis of interviews with members of the DIS testing and exercise evaluation community. A series of requirements was then generated based on feedback from this community which relies heavily on such a system. These requirements were then built into the VRSG system as a fully operational, PC-based DIS Stealth:

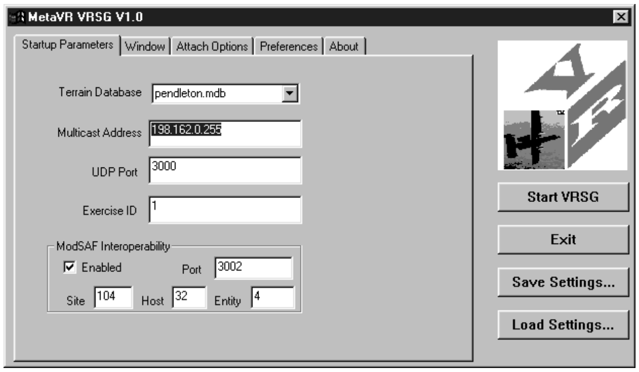
- DIS 2.0.4 compliant with migration path to HLA
- High-sustainable frame rates on conventional PCs
- Databases (MDB) are generated from ModSAF CTDB resulting in absolute terrain correlation
- Support any 3D graphics accelerator that uses Microsoft Direct3D
- ModSAF PVD inter-operability via Stealth control PDU
- Dynamic environmental effects
- Screen snapshots to Windows 95 clipboard

- Save and recall named viewpoints
- Choice of coordinate system (UTM Grid, LL, or Terrain)
- GUI-based control over database scenarios
- Support UDP Multicast, Broadcast, or Point-to-point
- Teleportation to an arbitrary location via specifying location using coordinate system of choice
- Save configuration settings to a file - system always recalls and starts with settings used when system last ran
- From GUI, specify FOV, Horizon, Fog, Cloud height
- Multiple attachment modes
- Store more interesting entities in a Preferred List
- Selectable resolution such that a user can optimize between image quality and performance
- Personalize controller gain
- User-friendly interface with onscreen help
- Keyboard control which may be used concurrently with other input device using the Microsoft Gameport API

While the list of functionality is familiar to the general simulation community, the resulting decision to produce a new database format was not an explicit requirement but rather a consequence of the stated requirements. To be sure, a significant amount of effort has gone into the development of 3D analogs to the 2D environment of the ModSAF artificial intelligence-based entity generator. The most common format for entity models and terrain database in 3D applications is the Multigen Flight™ format. However, it was determined that the overriding requirement for the community was to provide absolute terrain correlation with the ModSAF Compact Terrain Database (CTDB) terrain format.

A new terrain database format was chosen since no conversion capability exists within the Multigen tool suite to produce a CTDB correlated Flight format database. The option to write a Multigen to CTDB converter is not possible due to the proprietary nature of MultiGen, which reserves control of applications writing its files. Creating a new format based on lessons learned from the simulation community had other advantages as well. It allowed for a design that specifically lends itself to real-time performance and support for dynamic terrain. The resulting Meta-Database (MDB) format can currently be generated from CTDB or Digital Terrain Elevation Data (DTED).

The VRSG system uses a Windows-based GUI for user control of the virtual world (Fig. 4).



Terrain Database: pendleton.mdb

Multicast Address: 198.162.0.255

UDP Port: 3000

Exercise ID: 1

ModSAF Interoperability

☑ Enabled Port: 3002

Site: 104 Host: 32 Entity: 4

Figure 4. The VRSG graphical user interface.

Further virtual world navigation is enhanced by a 2D overlay of entity and help information within the virtual world. The primary user control device is the SpaceOrb 360 controller whose genesis is from the Spaceball control devices.

7.0 REAL-TIME CONSIDERATIONS FOR VISUALIZATION PERFORMANCE WITHIN HIGH LEVEL ARCHITECTURE (HLA)

The primary considerations for performance effects on visualization environments by the High Level Architecture (HLA) involve network interaction prior to rendering the scene graph. It is assumed that any implementation of the HLA Run Time Infrastructure (RTI) will need to support heterogeneous reliability of network interactions between simulation hosts.

Heterogeneous reliability is the segmenting into hierarchies of reliability requirements the protocol data unit (PDU) types based upon their Quality of Service (QoS) requirements. The hierarchies are based on the reliability requirements of individual entities and hosts and are used to dynamically select between reliable and unreliable transport modes. By restricting use of reliability to only times when it is really needed, control channel traffic required for reliability is greatly reduced, furthering both scaleability and performance (Ref. 4).

Any HLA/RTI infrastructure will need to provide for the simulation host to switch between unreliable and reliable data receipt in a manner that does not exceed the real-time requirements of the exercise (i.e., 100 milliseconds). An elegant option is to provide for simulations to send and receive reliable and unreliable traffic on different ports. This implementation proved cumbersome however in practice when conducting simulation exercises. The time required to switch between ports in a single threaded application pushed the simulation past the 100 millisecond latency threshold. Multithreaded application would suffer from coordinating rendezvous amongst the processes' threads. The solution in reference 4 was to piggyback the unreliable and reliable traffic through a single port. Any HLA implementation will need to address how simulation hosts can differentiate between unreliable and reliable transmissions without losing real-time performance.

8.0 REFERENCES

1. Small Business Innovative Research (SBIR) RFP A96-164 “Virtual Reality Scene Generation By Means of Open Standards”.

2. Gustavson, P., “Developer's Forecast for the Simulation World”, Proceedings from the fifteenth Workshop on Standards for the Interoperability of Distributed Simulations, September 1996.

3. http://www.microsoft.com/msdownload/directx3.htm

4. Smith, W.G. and A. Koifman, "A Distributed Interactive Simulation Intranet Using RAMP, a Reliable Adaptive Multicast Protocol", Proceedings from the Fourteenth Workshop on Standards for the Interoperability of Distributed Simulations, Orlando, Florida, March, 1996.

9.0 ACKNOWLEDGMENTS

Figures 1 and 2 were reprinted by permission from 3DFx Interactive, Inc.

The author drew heavily upon the stated requirements for a Virtual Reality Scene Generation By Means of Open Standards in SBIR A96-164 for the design and implementation of the VRSG and in writing this paper.

10.0 ABOUT THE AUTHOR

W. Garth Smith is a member of the MetaVR, Inc. virtual company. Garth was formerly a senior member of the technical staff at TASC. He also was one of a team of simulation engineers responsible for the Bolt, Beranek and Newman German version (AGPT) of SimNet.