Evaluation of Game-Based Visualization Tools for Military Flight Simulation

Logan A Williams, 1 Lt Matthew Lisa, Craig Eidman, Capt Clinton Kam, 1 Lt Adam Pohl
Air Force Research Laboratory, 711 HPW/RHAE
Mesa, AZ
Logan.Williams, Matthew.Lisa, Craig.Eidman, Clinton.Kam, Adam.Pohl@mesa.afmc.af.mil

ABSTRACT

PC based commercial off the shelf (COTS) flight simulators' native image generating capabilities can be harnessed to produce outstanding visual displays for a fraction of the cost of traditional image generator (IG) software. A high fidelity F-16 simulator's faceted Mobile Modular Display for Advanced Research and Training (M2DART) configuration display was driven by multiple instances of COTS game based flight simulation software. Laminar Research's X-Plane® was selected primarily for its low cost and available Software Development Kit (SDK). This paper details the advantages and limitations of using game based software as a visualization tool. The realism and fidelity of the game-based synthetic visual environment and weather simulation, the graphics scalability, effects quality and display limitations were assessed. The most significant challenge encountered in adapting COTS game based software to serve as a visualization tool was ensuring terrain and model database correlation; potential methods of solving this problem were explored. Overall, it was shown that a COTS flight simulator is a viable and potentially cost saving alternative to traditional IGs in high fidelity flight simulation for certain training applications, given proper steps are taken to adapt and integrate the program for the user's specific needs.

ABOUT THE AUTHORS

Mr. Logan A Williams is the technology development lead for immersive training environments at the Air Force Research Laboratory, Warfighter Readiness Research Division, Mesa Research Site. He has previously served as the lead systems engineer for the F-16, A-10, and KC-135 aircrew training systems, and has over a decade of experience in networked control systems, optical system design, and physics-based modeling & simulation. Mr. Williams has earned Bachelor of Science degrees in Electrical Engineering and Physics, as well as a Master of Science in Electrical Engineering from the University of Utah.

1Lt Matthew Lisa is currently assigned to the Warfighter Readiness Research Division of the 711th Human Performance Wing, Air Force Research Laboratory, in Mesa, Arizona working as the lead Warfighter Training Systems engineer for the Immersive Environments Branch. Lt Lisa holds a B.S. in Aerospace Engineering from Virginia Polytechnic Institute and State University. He has worked on several projects spanning the modeling and simulation and training spectrum including gaming technology exploitation, close air support tactics and training, and coalition training support.

Mr. Craig Eidman is an Electrical Engineer with the Air Force Research Laboratory, Warfighter Readiness Research Division, Mesa Research Site. He is the Lead Engineer for all laboratory synthetic environments, Electronic Warfare Training, and advanced targeting pod S&T efforts. Mr. Eidman attended the United States Air Force Academy where he earned a Bachelor of Science in Electrical Engineering in 1983. He served in the Air Force as a Command Pilot flying F-16, A-10, and OA-37 aircraft in multiple theaters and as an Evaluator Forward Air Controller. Additionally, he has a Masters of Aeronautical Science, Embry-Riddle Aeronautical University, With Distinction and is an outstanding Graduate of the Air War College.

Capt Clinton Kam was recently assigned to the Warfighter Readiness Research Division of the 711th Human Performance Wing, Air Force Research Laboratory, in Mesa, Arizona working as a Threat Systems Engineer for the Immersive Environments Branch. Lt Kam holds a B.S. in Aerospace Engineering from the University of Texas. He has worked on several projects spanning the modeling and simulation and training spectrum including simulation performance testing and optimization, HLA networking, and high-fidelity piloted and unpiloted threat models.

1Lt Adam Pohl is currently assigned to the Warfighter Readiness Research Division of the 711th Human Performance Wing, Air Force Research Laboratory, in Mesa, Arizona working as a Developmental Engineer for the Immersive Environments Branch. Lt Pohl holds a B.S. in Electrical Engineering from Embry-Riddle University and a M.S. in Computer Engineering from the Air Force Institute of Technology. He has worked on several projects spanning the modeling and simulation and training spectrum including visual research, game based technology, and threat system architecture.

Evaluation of Game-Based Visualization Tools for Military Flight Simulation

Logan A Williams, 2Lt Matthew Lisa, Craig Eidman, Capt Clinton Kam, 1Lt Adam Pohl
Air Force Research Laboratory, 711 HPW/RHAE
Mesa, AZ
Logan.Williams, Matthew.Lisa, Craig.Eidman, Clinton.Kam, Adam.Pohl@mesa.afmc.af.mil

INTRODUCTION

Commercial image generators offer a full suite of tools to create a highly immersive synthetic training environment for military flight simulation. This robust capability often carries a high licensing cost and requires a specialized and/or proprietary software interface with the host simulation. It is generally accepted that game-based flight simulators cannot approach the complexity and realism of the high fidelity avionics simulations employed in modern Air Force training systems. However, low cost Commercial Off the Shelf (COTS) gaming technology is rapidly approaching many of the graphics and visual display capabilities previously available only in commercial Image Generators (IGs) while also offering rapid development capabilities to the user, via fully programmable software interface and plug-in support (Smith & Denise, 2007). In 2009, the Air Force Research Laboratory (AFRL) evaluated the advantages and limitations of using a COTS flight simulator program as an out-the-window visualization tool in a high fidelity F-16 tactical simulator as part of an ongoing game-based technologies research and development project. Laminar Research's X-Plane® was selected for its low cost, powerful software development kit, ease of integration with a Distributed Interactive Simulation (DIS) interface, non-proprietary model and software plug-in generation, and extensive development support community. This paper describes the ongoing research to investigate the advantages and limitations that game-based COTS software can provide for military training and simulation, and is intended to provide a preliminary assessment of capabilities, not an in-depth comparison to any single product. The use of X-Plane® to conduct this study is not an endorsement of this product by the U.S. government, and the opinions expressed herein are solely those of the authors and not the U.S. government.

Equipment & Setup

This research evaluates the use of Laminar Research's X-Plane® in the capacity of an out-the-window visualization tool for an external simulation, using a custom DIS interface written by AFRL. AFRL's Air-to-Surface Testbed F-16 simulator was employed to provide the host avionics simulation for this research, and to validate it's applicability to high fidelity flight simulators (see Figure 1). The testbed consists of a unique high fidelity F-16 Block 30/40/50 (reconfigurable) cockpit previously built by AFRL under an Advanced Technology Demonstration (ATD) program. The testbed was operated in the Block 30 Software Concurrency Update (SCU) 6 configuration for this research.

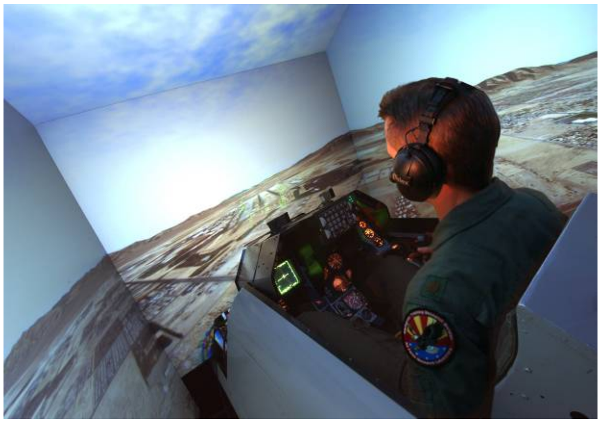

Figure 1. High Fidelity F-16 Block 30 Cockpit with M2DART Immersive Visual Display.

The Experimental Common Immersive Theatre Environment (XCITE) Version 3.0 provided the constructive entity, network, and scenario management for the simulation. XCITE is a government owned synthetic tactical environment developed by AFRL capable of modeling high-fidelity air-to-air and air-to-ground aerodynamic, RADAR, and weapons interactions (see Figure 2).

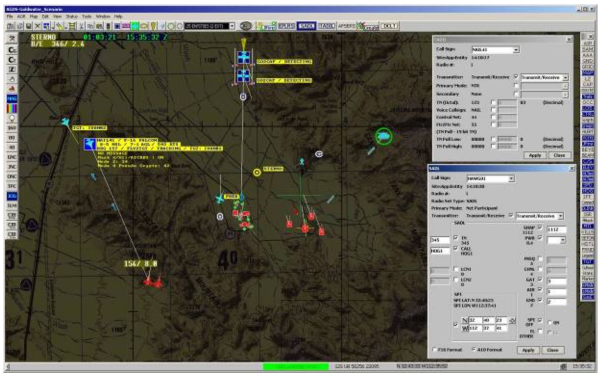

Figure 2. The XCITE Synthetic Environment

The three forward facets of the simulator's Mobile Modular Display for Advanced Research and Training (M2DART) configuration display were driven by separate instances of X-Plane®, running on independent 32 bit Windows XP workstations, utilizing Intel Core I7 920 processors, ASUS P6T6 motherboards, GeForce GTX 295 video cards, and 6 Gigabyte (3 Gigabyte accessible) ram. The forward image was rear projected by a Panasonic PT-AE3000U, while the left and right images were displayed on Panasonic PT-AE2000U projectors. The Heads Up Display (HUD) was drawn separately by the F-16 host simulation and rear projected by an Optoma EP 739. No attempt was made to drive cockpit Multi-Function Displays (MFDs) for sensor emulation with X-Plane® .

ADAPTATION OF GAME-BASED SOFTWARE

As a game-based flight simulator, X-Plane® allows the player to fly native aircraft models using traditional joystick, throttle, and keyboard controls. X-Plane® also includes a robust Software Development Kit (SDK) which proved to be the key component enabling its use as a visualization tool capable of supporting military flight simulation. The SDK allows the native controls to be modified or overridden and new functions to be quickly developed, thus granting extreme flexibility in the control and display of the simulated environment.

DIS Interface

Using the SDK, an X-Plane® plug-in was written which transmits and/or receives DIS network traffic conforming to the IEEE 1278.1a protocol, to include DIS packets generated by both the host Cockpit and XCITE software. Previous research (Eidman, Lisa, Kam, Pohl, Rogers, & Mitchell, 2009) has demonstrated the use of X-Plane® as a virtual cockpit using this technique, although this effort utilizes DIS traffic generated external to the X-Plane® application, which operates in a receive-only status (see Figure 3).

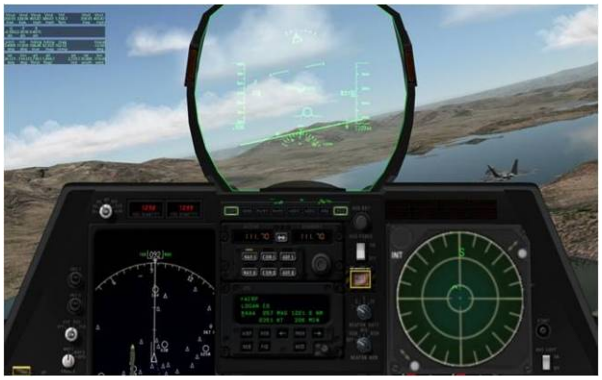

Figure 3. Use of X-Plane® as a Virtual Cockpit for Electronic Warfare Training

To facilitate this effort, DIS traffic was generated on a local network by both the host F-16 cockpit avionics simulation and the XCITE threat environment, which manages the DIS entity states for the constructive forces, including entity generation, aerodynamic modeling, weapon interactions, dead reckoning, and entity removal (Wooster, Richard, & Carr, 2006; Eidman & Kam, 2008). Entity state Protocol Data Units (PDUs) generated by these external systems are then received by the custom DIS Plugin and the corresponding entity models are rendered in the out-the-window scene. X-Plane® tracks the own-ship entity position via the site/app/entity identifier of the DIS PDU and renders the out-the-window scene from the equivalent eye point. The XCITE software calculates each constructive entity location in latitude, longitude, and altitude (including orientation - roll, pitch, and yaw) then converts this position to WGS84 geocentric coordinates to conform to DIS standards prior to broadcast. The position data of each external entity is processed by X-Plane® directly in geocentric coordinates for out-the-window display. This interface then allows X-Plane® to adopt the basic functionality of an image generator for DIS compatible systems.

X-Plane® is limited to controlling a maximum of 20 distinct entity models, though any of these 20 models may be replicated any number of times within the simulation. Notably, this limitation only applies to dynamic entities directly controlled by X-Plane®, and not to geotypical ground entities or user defined object models. Thus, when X-Plane® is used in the sole capacity of a visualization tool, DIS entities are controlled by an *external* source and merely displayed as objects by X-Plane®, allowing any number of distinct external entity models to be displayed, up to the limit of the controlling software, in this case the XCITE threat system. A configuration file was generated containing the DIS enumeration data corresponding to X-Plane's® model library. External entities received by the DIS plug-in are compared to the model library to determine which object model will be rendered to the display. The only requirement is that the requisite object models be built or imported to the X-Plane® model library (see Figures 4 & 5).

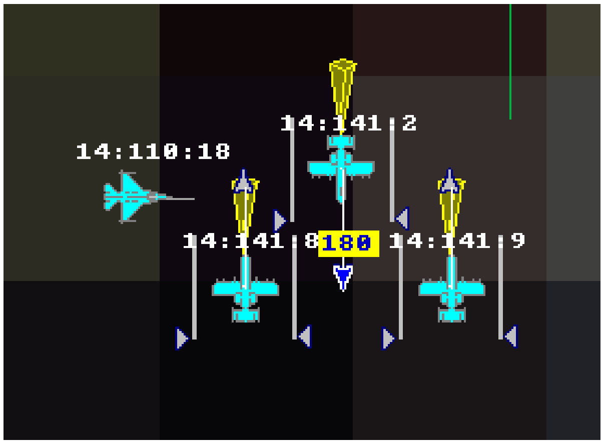

Figure 4. Constructive Entity Locations Determined by the XCITE Threat System. Note the Display of Site/App/Entity DIS Identification Tags

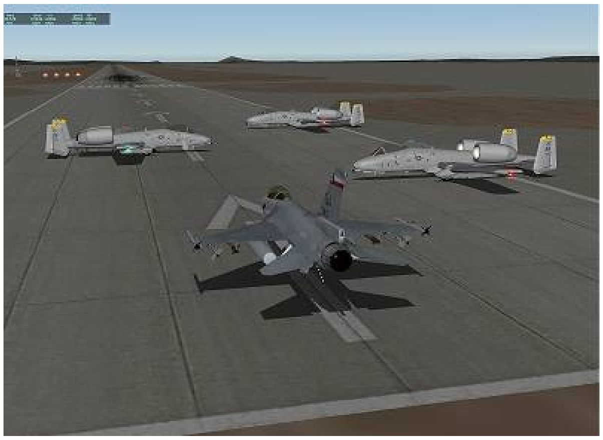

Figure 5. Corresponding Entity Locations Displayed in the X-Plane® Visualization

Database Correlation

As anticipated, the native X-Plane® terrain database is not perfectly correlated with the XCITE terrain database. Both terrain databases use Digital Terrain Elevation Data (DTED) level 1 with visual terrain overlay, although differing terrain processing algorithms and sources of visual data have been applied to each. Additionally, the XCITE software calculates entity locations in latitude, longitude, and altitude before converting to geocentric coordinates (x, y, z, φ, θ, ψ) prior to broadcast. When used as a visualization tool, X-Plane® accepts the entity state position in round earth coordinates and renders from the corresponding eye point. Both of these factors contribute to a discrepancy between the Above Ground Level (AGL) altitude derived from the entity state location, as determined by the XCITE database, and the AGL calculated from X-plane's® native terrain database (see Figure 6).

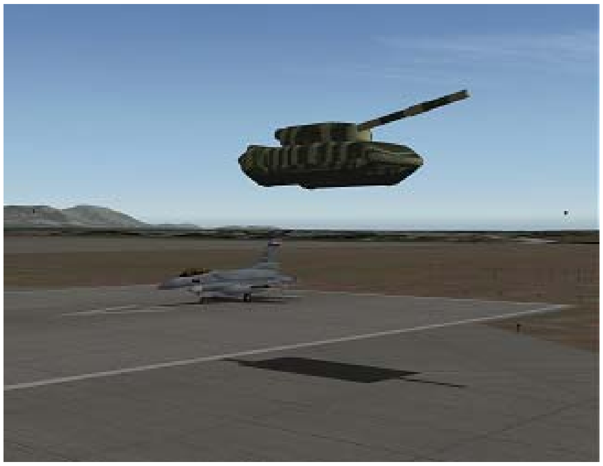

Figure 6. Correlation Errors Between Native X-Plane® Entity Position (Background F-16) and XCITE Entity Position (Foreground Tank) Received Via DIS Network.

At high altitudes this discrepancy is not immediately apparent. However, it becomes readily apparent at low altitudes, as ground entities supplied by the threat system are displayed either above or below zero AGL. Typical correlation errors were as large as ±30 meters, depending upon specific locations within the database. While the implications of uncorrelated visual databases have been previously evaluated (Stephens & Hendrix, 2003; Stokes & Stephens 2005), the most apparent problem with air-to-surface engagements was lack of target visual ID during air-to-surface target engagement (if the target was displayed below the terrain skin), and non-representative visual delivery of ordinance. However, actual weapons engagement and threat interactions are modeled solely within the XCITE threat environment and host F-16 simulation; therefore all fundamental interactions are correctly modeled and implemented regardless of the lack of visual correlation.

Two methods were considered to increase correlation between the X-Plane® and XCITE databases. The initial approach was to develop a ground-clamp algorithm using altitude data from the X-Plane® terrain locations to compel visual correlation with the XCITE terrain database. To implement this, the DIS plug-in was modified such that for each DIS entity state packet received the entity location is calculated by X-Plane®, but the model is not yet visually rendered to the display. If the entity is a ground entity, the distance between this location (which should be 0 AGL) and the terrain location (which is 0 AGL in X-Plane®) is then calculated to yield the vertical correlation error. The altitude is then corrected by this amount before the entity is visually rendered to the display (see Figure 7). Transverse error corrections (i.e. errors in the x,y plane) were not addressed in the course of this research.

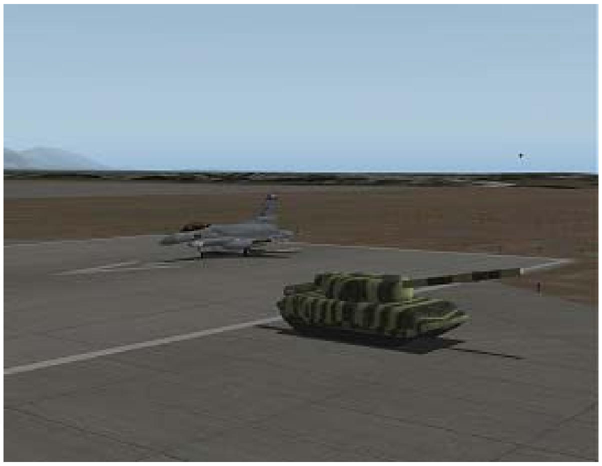

Figure 7. Implementation of Ground Clamping Algorithm to Eliminate Vertical Correlation Error for Entity Position (Foreground Tank) Received Via DIS Network.

An alternate method of increasing database correlation was considered, though not implemented due to the success of the initial ground clamping algorithm. This second approach required the XCITE terrain database to be recompiled from a known DTED level 1 data set, without post processing. This same dataset would then be substituted within X-Plane® , thus forcing the two databases to match as closely as possible. This approach, while supported by the X-Plane® SDK, was less desirable for two reasons. First, it requires the ability to recompile the XCITE source terrain data, which is a capability unique to AFRL and not widely available to many users. Second, it was anticipated that some discrepancies would remain due to the previously described conversion errors between flat- and round-earth coordinates, or perhaps any unknown or uncontrolled post processing which may occur within X-Plane®.

Synthetic Tactical Environment

The synthetic tactical environment is governed externally by both the XCITE threat system and the host simulation. Therefore, there is no need for X-Plane® to assume management of the tactical environment when used solely as an out-the-window (OTW) visualization tool. As previously described, all entity-state PDUs, aeronautical models, weapons fly-out, RADAR interactions, etc., are governed by external systems. For this application, the HUD is also externally driven by the F-16 host simulation and is not dependent upon the X-plane® visualization. No attempt was made during this effort to network multiple simulators using the X-Plane® visualization toolset, although this should be possible with minimal or no additional modification of the DIS plug-in.

However, roles have been previously identified in which the X-plane® simulation may be required to implement some tactical functionality. For example, the use of X-plane® as an aggressor/wingman station for an instructor would then require X-plane® to transmit DIS packets rather than acting in a strict receive-only capacity. To prove effective in this role, validation of the aerodynamic model for the own ship, emission of DIS PDUs, and the development of the needed control and feedback interfaces would be required. These elements would reside directly within the X-Plane® application. However, modeling of RADARs and weapons fly outs would most likely continue to be handed off to an external program, such as XCITE. Each of the required additions is possible using the included SDK and plane-maker software, and future research is planned to explore this use of game-based software.

Synthetic Visual Environment

The X-Plane® visual output is based on OpenGL. Most aspects of the display can be altered through an in-game menu. X-Plane® is capable of flat screen, cylindrical, and dome projection, including rudimentary edge blending. A facetted M2DART display was used in this research, thus cylindrical and dome projection with edge blending were not tested. Additionally, both cylindrical and dome projection require secondary licenses. The ability to incorporate plug-ins allows the user to control most aspects of the display, including drawing objects and placing entities.

Most game based flight simulators, including X-Plane®, now include geospecific terrain data for the entire world and a standard model set. Terrain features, such as buildings and trees, are geotypical (see Figure

8). However, custom terrain can be created by the user or a third party, allowing for geospecific placement of buildings and other objects. Custom models can also be created and used, allowing the user to correlate the model database with that used in other systems. This feature was used to create the tank seen in Figures 6 and 7, as well as several other models which were not contained within the default model set.

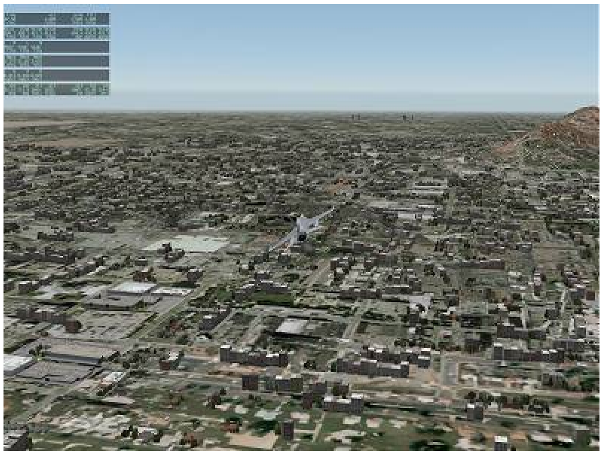

Figure 8. Daylight OTW View Typical of Game-Based Flight Simulators. Note the Geotypical Ground Models

The basic X-Plane® application simulates both red-out and black-out when the virtual pilot is stressed by negative and positive gs, respectively. This is done by blurring, and tinting the display in proportion to the number of gs and duration of the stress. The Night Vision Goggle (NVG) simulation is of poor quality. It may be useful for some limited training applications, but it does not accurately depict what one sees through actual NVGs. This is in part due to the lack of material encoding of the database and model set.

The apparent realism of the synthetic environment , when running at the high end of the software's capability, is visually outstanding, as evidenced by Figures 8 and 9. Graphics detail, resolution, anisotropic filtering and antialiasing levels can be specified by the user, providing additional versatility for aircraft simulation. (Geri & Winterbottom, (2005). The quality of water visualization can be minimized, or brought to a level where terrain and weather are accurately reflected by the water's surface. The wave properties of bodies of water can be altered through an in-game menu, which has an impact on the simulation of air-to-sea engagements. X-Plane® is capable of, and typically does display point lights. The time of day and year is easily set, realistically depicting lighting levels for both day and night. Weather can be controlled through a dedicated in-game menu, externally controlled, or networked to display current, real world weather data. The weather effects include volumetric clouds and fog, which is of significant value in training systems, as it effects the visual acquisition of other aircraft and targets (see Figure 9).

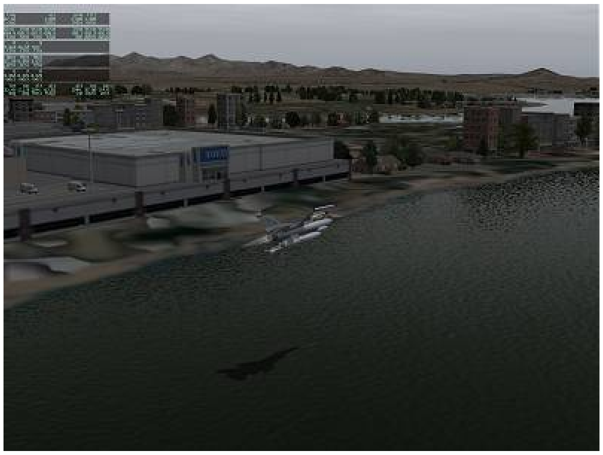

Figure 9. Graphics Capabilities Typical of Game-Based Flight Simulation.

Out-The-Window Visualization

The capabilities of this game-based software enabled the generation of OTW displays across of three screen facets. The display on each screen was driven by a dedicated instance of X-Plane®, resulting in three instances of the software running in parallel. When the front screen's instance of X-Plane® was used as a master, with the other two instances as subordinates, occasional synchronization delay (up to 120ms) occurred in both the subordinate screens. This was done using a native feature in X-Plane® that allows secondary instances of the program to display alternate fields of view for the master simulation. Alternatively, all three screens were well synchronized when each instance was run independently, each utilizing its own instance of the DIS interface plug-in to simultaneously receive network data with no measurable synchronization delay. A single instance of X-Plane® could potentially have run all three screens, given an appropriate hardware/driver configuration. X-Plane® is capable of outputting separate Fields of View (FOV) to two display channels; a third display could have been driven by extending the resolution of one display channel and splitting the image across two displays. This option was not used, however, due to significant negative impact on performance.

Overall, the quality of the display output proved sufficient for training where correlation of terrain databases is of relatively minor significance, such as training at altitude (Niall & Pierce, 2000). It also proved sufficient for basic simulation, such as Hands-on Throttle and Stick (HOTAS) familiarization and emergency procedures training. The quality could possibly be sufficient for higher fidelity training applications and in applications where terrain correlation is critical, such as close air support (CAS), only if a custom geospecific terrain database was created, sensor/pod simulations are developed, and material coding was added via a robust plug-in. This would likely require effort comparable to that of commercial database generation, but could be done in-house or by a third party without violating the COTS software license agreement.

EVALUATION OF CAPABILITIES

It is generally accepted that game-based software cannot be used to fully simulate high-fidelity aircraft avionics and/or realistic combat interactions, although it has become highly desirable to leverage the significant COTS investment into the special effects, graphics, modeling, and hardware optimization solutions which are key performance drivers within the gaming industry. However, the use of any game-based visualization technology must provide the visual fidelity, performance, and capabilities commonly supplied by commercial IGs (Meta VR Online Homepage, 2009). Predictably, game-based software both excels and falls short when compared to typical commercial IG capabilities.

Performance

The performance of X-Plane's ® visual display, like most games, is highly dependent on the hardware configuration. High-end machines, such as those used in this effort, will yield the best results, but the software can be run on lower-end machines without negatively impacting the frame rate of the simulation. This is made possible due to X-Plane's® graphics scalability, allowing extensive adjustment of the rendering options via an in-game menu. For applications such as an out-the-window view for a networked simulator, the quality of the visual output can be increased significantly without reducing the frame rate by disabling X-Plane's® physics engine. This reduces computational overhead while not affecting the fidelity of the simulation because the position of all DIS entities are governed by external software; X-Plane's® native flight models become unnecessary. Additionally, the plug-in that facilitates DIS compatibility does not measurably reduce the frame rate of the simulation, which is the rate at which the visual display buffer is updated.

Unfortunately, the use of DIS PDU's to drive the visual display does introduce an artifact which results from the use of dead reckoning algorithms. The F-16 host simulation emits an entity state PDU only when a dead reckoning threshold is breached. Under typical dead reckoning thresholds (3 degrees roll/pitch/yaw, 5 meters x/y/z) the result is scene "jitter" in the projected image, effectively reducing the perceived frame rate. This effect is not constant, but is most noticeable at low altitude and during rapid changes in position and/or orientation. Although the X-Plane® application is capable of 60+ Hz frame refresh rates, the visualization eyepoint only updates upon receipt of a new DIS entity state PDU, which is subsequently dependent upon the airspeed and maneuvers performed by the pilot. Reduction of the dead reckoning thresholds can be applied to reduce the jitter effect (i.e. increase effective frame rate) as desired, though at the cost of proportionally increased DIS network traffic. Although no attempt was made during this effort to network multiple simulators, it is anticipated that this limitation may place an upper limit on the number of simulators which can be effectively networked in a DMO scenario. The use of additional smoothing algorithms was not implemented, though this is anticipated to significantly reduce the jitter effect (Covas-Smith, Gaska, Shamp & Pierce, 2007; Slater & Covas, 2007).

Cost Savings

Game-based flight simulators have begun steadily incorporating software development capabilities via fully programmable interfaces, model generation applications, and plug-in support such as the X-Plane® SDK and (the no longer supported) Microsoft ESP®. The availability of these tools is the single enabling factor which allows COTS software to begin assuming the roles of commercial IGs while capitalizing upon the substantial investment in graphics and hardware optimization of the game industry. Thus, a substantial cost savings for military applications lies in leveraging these commercial investments for out-the-window visualization while maintaining separate high fidelity host avionics and threat environment simulations.

Many commercial IG's carry a relatively high licensing cost of several thousand, or tens of thousands of dollars per visual channel while game based software, targeted toward the consumer budget, typically falls below $100 per copy due to vast differences in application, audience, fidelity, and the available economies of scale. With the increasing requirement for ultra high resolution displays which begin to approximate eye-limiting resolution, the use of multiple projector arrays or multiple-input projectors can quickly make traditional IG solutions cost prohibitive. Thus, a natural synergy exists between the requirement for ultra high resolution visual displays and the graphics motivated, cost driven gaming industry.

The total cost for the X-Plane® software (3 licenses) used in this effort was US$120.00. Adaptation of the X-Plane® software and development of the DIS plug-in and associated utilities has required minimal software development as part of AFRL's ongoing game-based technologies research and development project. The scope of this effort is estimated at less than 200 man-hours of total development time contributed by junior level software engineers. The minimal effort to adapt X-Plane® for use as an out-the-window visualization, combined with a relatively minor software cost, make the cost savings over commercial IG's readily apparent for well suited applications.

Limitations of Game-Based Technology

Unfortunately, the use of game-based visualization tools is not without limitations. As previously identified, game based software does not have the capability of modeling host avionics or tactical environments with the degree of fidelity required for military simulation. Additionally, there is no incentive for COTS game developers to provide proper material encoding of databases or develop correspondingly accurate sensor representations of object models and terrain, as those applications remain largely military-specific. While many game-based simulations provide geospecific terrain databases and imagery overlays, without modification it is unlikely to be of high enough resolution or properly encoded to adequately support several specialized training requirements such as NVG operations, Targeting Pods, Suppression of Enemy Air Defenses (SEAD), CAS, and various air-to-ground training scenarios requiring high resolution material encoded imagery or specialized sensor operations. Modeling limitations also contribute to this deficiency, as COTS game software is not likely to posses the range of geospecific or sensor models required to meet a variety of air-to-surface training needs. The additional development required to remedy these limitations is significant, and begins to rival the effort and cost necessary to develop commercial databases for military simulation.

As with the fiscal advantages, these performance limitations are also a result of the focus toward the consumer budget. Consumer hardware, while not particularly limited in computational capability, typically suffers from the lack of storage capacity necessary to maintain several high resolution, material encoded databases. Consumers additionally have no need for bona fide material encoding or sensor representations, so these capabilities are understandably absent in game-based simulations, allowing game developers to avoid this costly aspect of simulation and entrust these roles to the commercial IG market.

These limitations of game based software make it readily apparent that there will always be a unique need for commercial IG's, and the associated high fidelity databases, for military simulation. However, it is also apparent that there are several more basic roles in which game-based visualization may act as a fully adequate, low cost training alternative.

Optimal Training Applications:

The limited capabilities of game-based software reduce the scope of what may be trained to primarily daylight, out-the-window, air-to-air centric training events, with limited air-to-surface roles. While this eliminates numerous training and rehearsal capabilities essential to the military simulation regime, there remains a substantial gamut of training events in which low cost visualization tools may provide an optimal solution. These include generalized daylight out-the-window and air-to-air centric events as well as more specific training applications in which the visual component is a helpful, though not vital component of the simulation. Such applications include introductory flight training, H Joint Close Air Support (JCAS) training for ground troops, Emergency Procedure (EP) trainers, Electronic Warfare (EW) trainers, Avionics trainers, and various other part task trainers. The game-based solution is also an ideal choice in the admittedly rare circumstance that any training systems meeting these criteria also require multiple projector arrays or multiple visual channels, in which commercial IG costs can rapidly scale out of proportion.

Current and Future Research

AFRL is currently conducting extensive research into the training applications of game-based hardware and software. This research ranges from the validation of flight models to low-cost infrared tracking systems to desktop training systems. Future research, as it pertains to the use of game-based COTS software for synthetic visual displays, will include both subjective and objective comparisons to traditional IGs and databases for resolution, scene content, special effects, and correlation to more explicitly identify and evaluate specific training capabilities. Other research applications, beyond OTW visualization for flight simulation, will include, but not be limited to, the use of this software for cockpit MFDs, desktop trainers, dome projection, and head-mounted displays (HMD), as well as networking multiple simulators using the X-Plane® visualization toolset.

CONCLUSION

This research has demonstrated that game-based COTS software can provide, in select training applications, an adequate alternative to traditional IG's for a fraction of the cost and development time. However, the investment required to match the full performance of high fidelity commercial IGs would likely become cost prohibitive. However, given the rapid advances of a continuously evolving game industry and the increasing interest in using game-based software for training applications, future developments may address some of these limitations. Currently, game-based COTS software may still be considered for a wide range of training applications, given that the software under consideration includes an SDK or equivalent method of tailoring the game to fit the needed application.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the efforts of Capt Clinton Kam and 1Lt Adam Pohl for writing much of the software that supported this effort.

REFERENCES

Covas-Smith, C.M., Gaska, J.P., Shamp, M., & Pierce, B.J. (2007). Detection threshold of model visual displacement in a networked flight simulator. *Proceedings of the International Symposium on Simulation and Training for Day and Night Missions, Defence Research and Development*, *Toronto*, *Canada*, November, 2007.

Eidman, C., & Kam, C., (2008) Computer Generated Forces for Joint Close Air Support and Live Virtual Constructive Training. *Interservice/Industry Training, Simulation and Education Conference 2008*

Eidman, C., Lisa, M., Kam, C., Pohl, A., Rogers, B., Mitchell, D. (2009). Leveraging Gaming Technology for Military Training, *Proceedings of Society for Applied Learning Technology, New Learning Technology 2009 Conference.*

Geri, G., & Winterbottom, M. (2005). Effect of display resolution and antialiasing on the discrimination of simulated-aircraft orientation. *Displays*, 26, pp. 159-169.

Meta VR Online Homepage (2009) *VRSG as a Computer Image Generator*. Retrieved June 19, 2009 from http://www.metavr.com/products/vrsg/IG.html

Niall, K.K. & Pierce, B.J. (2000). *Assessment of visual requirements, Aircrew Training: Methods, Technologies, and Assessments*. Lawrence Erlbaum Associates, Inc.

Slater, M. & Covas, C.M. (2007). Effects of Long-haul Network Connectivity on the Visual Fidelity of Real-time Flight Simulation. *Presented to the Fall Simulation Interoperability Standards Organization Simulation Interoperability Workshop*, September, 2007.

Smith, L. S. P., & Denise, N.. The utilization of low cost gaming hardware in conventional simulation, *Interservice/Industry Training, Simulation and Education Conference 2007.*

Stephens, C.W., & Hendrix, J. (2003). An Overview of Networked Simulation Technology From the Warfighter Training Research Division Perspective, *Proceedings of IMAGE 2003 Conference.*

Stokes, T. & Stephens, B. (2005). Using Open Industry Standards to Share Visual Simulation Databases. *Interservice/Industry Training, Simulation and Education Conference 2005*

Wooster, J., Richard, R., & Carr, J. (2006) Live, Virtual, and Constructive Integration, the Evolution of Distributed Mission Operations. *Military Simulation & Training Magazine, Issue 6, 2006.* pp 12-13