Flight Simulator as Geospatial Visualisation Platform

John Hildebrandt (POC)
DSTO C3 Research Centre
Fernhill Park
Department of Defence
Canberra ACT 2600
Australia

Phone: (02) 625 66102
Fax: (02) 625 66110
Email: john.Hildebrandt@dsto.defence.gov.au

Daryl Bossert
Royal Australian Air Force
Department of Defence
Australia

Abstract

The use of 3D visualizations is increasing in Defence applications. However the cost of high-end visualization tools still limit the extent of their deployment. In this paper a common commercial flight simulator package is adapted for use in terrain visualization, situation displays and as a low cost simulation engine. A JAVA application has been developed to ingest military geospatial formats and convert these into a format usable by the flight simulator package. Multiplayer interfaces within the flight simulator package have been leveraged to allow extraction of track information into simulation systems and ingest of track information to support the generation of 3D situation displays.

Introduction

The planning and execution of military operations involves the visualization of geospatial data. For example, the terrain appreciation process will benefit from terrain visualization, and situation displays typically use some form of geospatial display. The use of 3D visualizations and displays are increasing as they give the user a fuller representation of the battle space. However high-end 3D visualization packages can often be expensive assets that restrict their use to specialized areas.

The game industry is now one of the key drivers in the development of 3D visualizations and these products are provided at mass market cost levels. Further the performance of commodity PC graphics hardware now offers performance levels previously only available in high-end 3D workstations and these performance levels are improving at an exponential rate. In this paper we report on our experiences in adapting a commercial off the shelf (COTS) 3D visualization tool, namely Microsoft Flight Simulator 2000 to visualize terrain and geospatial information. This has the potential to deliver a very low cost 3D geospatial visualization capability, which fully

exploits commodity PC graphics engines, for situations where high-end capabilities are not required or unaffordable.

We have developed software to ingest military geospatial formats [CIB], [DTED] and convert this data into a form usable by Flight Simulator. This JAVA software allows the user to select CIB and DTED level 1 data of a region and then converts this into the terrain format used by Flight Simulator 2000 (FS2000). Other commercial software can perform similar conversions for a single image and terrain datasets [TERRABUILDER], however our software takes a full CIB dataset that consists of multiple images and generates a dataset covering the full region covered by these images.

The use of the FS2000 platform to visualize simulated platform locations and to output simulated platform locations was investigated. This allows the tool to provide a low-end 3D situation display and inject data into other simulation systems. An XML output format was employed to facilitate connection to multiple applications. In the future connection to standard simulation protocols such as DIS and HLA is planned.

In enterprise settings it is likely that the imagery and geospatial information will be obtained via libraries or repositories. In the future we intend to investigate ways to link the data transformation software to back end data repositories to simplify the steps the user must take to generate 3D terrain visualization.

Flight Simulator 2000 and DirectX

Microsoft Flight Simulator 2000 is a commercial off the shelf flight simulator for Windows PC's developed for the entertainment market. It provides a worldwide terrain database and a collection of aircraft types for use in simulation. Although it is designed and marketed primarily for the entertainment area the software has been used as an adjunct to other commercial and military training programs. An instance of flight simulator can be designated as a game server and then other instances of flight simulator can connect to it. This allows multiple players to fly within the same simulated world.

A variety of software developer kits (SDK) are available for Microsoft Flight Simulator [MSSDK] to support adding terrain information, developing custom aircraft, and to support multi player interfaces. These interfaces were essential for interfacing to Flight Simulator in this work and were the reason it was chosen over other commercial simulator packages. Currently the SDK's available include the following:

- Adventure Programming Language SDK
  Used for building lessons and scenarios.
- Multiplayer SDK
  Used to connect into the products Multiplayer Infrastructure.
- Aircraft Container SDK
  Used to design aircraft exteriors.
- Panels SDK
  Used to design internal aircraft control panel layouts.

- Terrain SDK
  Used to ingest Digital Elevation data.
- Scenery SDK
  Contains details on adding data to scenery files.

Microsoft Flight Simulator is built on the DirectX libraries. DirectX provides a 3D graphics library supported by most graphics card manufacturers and in addition provides libraries for sound, multimedia, and multiplayer collaboration. The multiplayer features of Flight Simulator are provided via the DirectX multiplayer interface that is referred to as DirectPlay. The DirectX libraries will also be supported in the future X-Box gaming console that would offer an even lower cost option for deploying low cost visualations and simulations.

Other systems

Several other systems exist for providing terrain displays for example we have examined higher end tools such as Autometrics BattleScape [BAT] and ERDAS VirtualGIS [ERDAS] to provide 3D terrain visualization. While these tools provide greater functionality we thought it was useful to examine how far a common low cost tool could be taken.

The Virtual Terrain Project (VTP) [VTP] are providing free tools for terrain visualization so would offer good support for the visualization component of this work but does not address the simulation aspects. Open source flight simulator tools are also becoming available and would provide full access to source code for customization purposes.

Data Ingest Client

The FS2000 terrain and scenery software developer kits (SDK) provided information on the formats used in FS2000 to describe terrain information. Terrain in Flight simulator 2000 is described using a scenery description assembly language that is then compiled into a binary format for use by the software. The scenery description includes height information for tiles on the Earth's surface and descriptions of any objects within the world. To support draping of image textures over terrain the scenery file includes file name references to bit map files. An extract of a scenery file showing a reference to a texture file and embedded terrain height information is shown in figure 1.

```
Header(1 -8.976982 -9.046036 126.049861
125.975069)
LatRange( -9.046036 -8.976982)
set( areamx 64 )
GRP(  -9.046036 125.975069 )
Area(B -8.988491 125.987535 60)
     PerspectiveCall(:L000001)
     Jump(:L000000)
:L000001
     Perspective
     RefPoint( 2 :L000002 1.000000 -9.000000
125.975069 v1= 0 v2= 0 E= 0.000000)
     SurfaceColor( 1 69 )
     Smoothing( 1 )
     Bitmap( db000000.bmp 0 0 0 0 )
     BitmapMode( 0 )
     TexRelief( 8 8 342 319 0 0
          0    0    120; -9.000000 125.975069
          31   0    146; -9.000000 125.978186
          63   0    178; -9.000000 125.981302
          95   0    129; -9.000000 125.984418
          127  0    112; -9.000000 125.987535
          159  0    142; -9.000000 125.990651
          191  0    132; -9.000000 125.993767
          223  0    156; -9.000000 125.996884
          255  0    92;  -9.000000 126.000000
          0    31   137; -8.997123 125.975069
```

Figure 1: Scenery file (.sca) extract.

Our source of terrain information was Digital Terrain Elevation Data (DTED) that provides terrain heights over a spatial grid. It was important to be able to use this format to enable the use of data from military geospatial information (MGI) producing agencies.

To provide a realistic 3D view we overlaid overhead imagery of the region onto the terrain surface. One source of imagery that we addressed was the Controlled Image Base (CIB) imagery that is produced by combining several satellite scenes of a region into a multi file standard product.

So the task required was to transform the DTED and CIB data into the formats required by the FS2000 application. To achieve this we developed a JAVA application that allowed input of any image format supported by the JAVA Advanced Imaging (JAI) libraries and would output bitmap (BMP) formatted image tiles as required by FS2000. The DTED data and the coverage region of the supplied imagery were combined to generate the scenery description file that included references to the bitmap texture files. This file was then compiled to the binary format required by FS2000. Compilers for Microsoft scenery files are available from a variety of sources on the web. Importantly this application could convert a complete CIB dataset that consisted of multiple images in one session without user intervention. This was the main driver leading us to develop our own application rather than using commercial transformation tools.

This process is an example of a data transformation process where to use a collection of data one must transform the data between various formats. Such transformations are common where a variety of imagery and geospatial products are in use. These

transformations can become a burden for the end user and in the longer term we would seek to automate the production of the data in the required formats.

Outputting track information

To access positional information from a linked set of flight simulator packages a C++ application that links to the flight simulator server using the DirectPlay interfaces was developed. This application connected to the FS2000 server and extracted the navigational information for all aircraft registered with the server. The information was then packaged in an XML format and broadcast for use by other applications. Each XML message included an aircraft identifier, position and other information as available from the DirectPlay infrastructure. In the future outputting to some standard simulation protocol such as IEEE DIS could be implemented.

Simple web based map display clients were developed which connected to the XML stream and displayed aircraft positions in 2D and 3D. In the future it is intended to use standard situation displays via the support for the required interfaces.

Inputting track information

Once one is connected to the Flight Simulator server via DirectPlay one can also inject navigational information for other entities. This allows us to use the package to visualize tracks of other objects in 3D space. Again an XML format for the track information is employed to facilitate simple connection to other systems, but in the future a standard DIS connection could also be implemented.

By identifying the entity as an aircraft type known to the FS2000 application it will then render a representation of the aircraft at the indicated position. To display abstract track symbols a special purpose 'aircraft type' could be defined using the aircraft construction SDK. The operation of inputting and outputting track information is illustrated in figure 2.

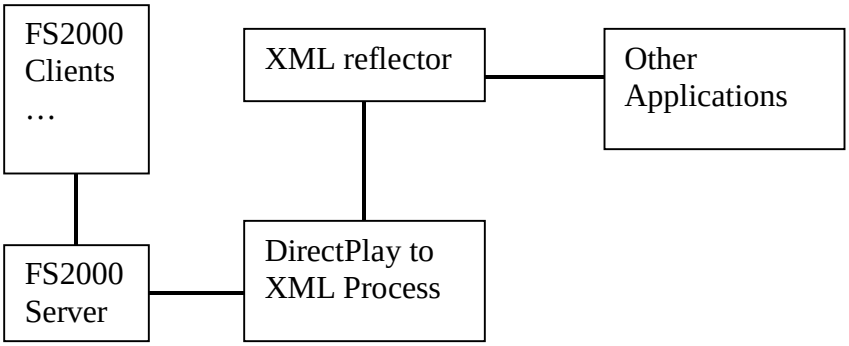

Figure 2: Exchanging DirectPlay position data with other applications.

Future Work

The experiments with Microsoft flight simulator 2000 (FS2000) have proved that a low cost package can provide useful terrain visualizations on commodity hardware. In this work a JAVA application was developed that took CIB imagery and DTED level 1 data as input and output data files compatible with FS2000. This is an example of the need for a service that will take datasets of some formats and transform those into data formats required by some over application or service.

In a large enterprise system the imagery and geospatial information will be stored in managed repositories rather than requiring knowledge of the file systems location of the data. The next logical step in developing the FS2000 capability is to demonstrate it as an example Data Transformation service that provides a direct link to Imagery and Geospatial services to source data for conversion into the files required by FS2000. One could envisage a web based application that allows the user to select the region of the world they are interested in. Then the service would connect to the image and geospatial service to obtain imagery and DTED data of the area and convert it into the FS2000 files. The files would then be placed at some location for use in FS2000.

A further enhancement to a transformation service to a file would be to provide on the fly access to the data services from the client application. In the case of Flight Simulator this is difficult due to the lack of an on the fly API to obtain terrain data. However a partially dynamic solution could be provided in the following manner.

One of the FS2000 files describes the terrain domain for a session and is loaded up front. As this is loaded at application startup it must be precompiled. Since it contains terrain height information this means all the DTED data is required up front. This file also lists the file names for the texture files that are generated from the imagery, so the file names must be determined up front. However the actual image texture files could be generated on an as required basis. This would require the system to keep track of the position one is at in the world and then use this to direct the population of the image texture files in a cache file system that FS2000 is directed to. To obtain the position information the multi-player features and API of FS2000 would be leveraged.

If a large number of image tiles are involved a cache management system would be required to delete least recently used tiles as the cache file system is filled.

The approach used here could be used to provide data transformation services to support other client applications. Particularly those that don't provide API access for data. This could lead to a family of data transformation services targeted at multiple applications and scenarios.

Conclusions

In this paper we have shown how a low cost simulation package can be leveraged to obtain significant visualization and simulation capability at very low cost. While the capability will not match fidelity or flexibility of high-end tools it would be useful to provide limited capabilities to a much wider audience. For example the high cost of

many visualization tools in the past has limited their use to specialist cells within an organization. The availability of a low cost option would allow deployment of some capability to all users that could take advantage of it.

Bibliography

BAT Autometrics Battlescape web site. http://www.autometric.com/AUTO/PRODUCTS/EDGE/BScape.html

CIB Controlled Image Base: Military Standard MIL-PRF-89041, available at NIMA web site, http://164.214.2.59/publications/specs/index.html.

DTED Digital Terrain Elevation Data: Military Standard MIL-PRF-89020A, available at NIMA web site, http://164.214.2.59/publications/specs/index.html.

ERDAS ERDAS Web Site http://www.erdas.com/

MSSDK Flight Simulator SDK's http://www.microsoft.com/games/fs2000/devdesk_sdk_fs2000.asp

TERRABUILDER http://www.terrabuilder.com/

VTP Virtual Terrain Project http://vterrain.org/