Geometric Editing with BRL-CAD™ 6.0 for Microsoft® Windows®

Christopher M. Pitts
CG2, Inc
Chris.Pitts@rdec.redstone.army.mil

Kimberly C. Williams
U. S. Army Aviation and Missile
Research Development and Engineering
Center (AMRDEC)
Kim.Williams@rdec.redstone.army.mil

Geometric Editing with BRL-CAD 6.0 for Microsoft Windows

- Role Of AMRDEC
- Conversion to Microsoft Windows platform
- BRL-CAD to Microsoft Windows History
- BRL-CAD 6.0 Conversion
- General Observations on the Conversion Process
- Changes to Library Functions Changes to MGED
- Conclusions
- Demonstration

Role Of AMRDEC

- The AMRDEC is responsible for providing data and analyses in the development of missile systems & aviation platforms

- BRL-CAD (raytracing and MGED only) is utilized :

    (1) in the design level studies to determine ordinance system requirements (depth of penetration, etc.)

    (2) missile terminal engagements (impact location and terminal angle optimizations)

    (3) Map impact locations from high fidelity flight simulations to Pk/h cell by cell maps

Conversion to Windows platform

- **Silicon Graphics computers are**
    - **(1) Expensive to purchase**
    - **(2) Difficult to keep on maintenance**
    - **(3) Slower than personal computers**

- **Personal computers are faster, cheaper and easier to justify the expense**

- **LINUX is not the OS of choice for our computer programmers**

So -

Porting to Microsoft Windows became the

Challenge

BRL-CAD to Windows History

Version 4.4   **1998**

       LIBRT and LIBWDB only**

Version 5.3   **2001**

       MGED and Corresponding libraries

Version 6.0   **2002**

       MGED and Corresponding libraries

** There was not an easy way to port MGED to Windows at that time

BRL-CAD 6.0 Conversion

- **The following tools and libraries were converted:**

MGED	**LIBBN**
RT	**LIBBU**
NIRT	**LIBDM**
LIBBN	**LIBFB**
LIBBU	**LIBOPTICAL**
LIBDM	**LIBRT**
	LIBSYSV
	LIBTERMIO
	LIBWDB

General Observations on the Conversion Process

• **TCL is not 100% cross platform, there are UNIX/Microsoft Windows only routines and MGED uses some of these UNIX only routines**

• **Windows is not as forgiving on memory as UNIX (i.e. memory leaks appeared)**

General Observations on the Conversion Process

Upgraded from: **to:**

    TCL 8.3   **TCL 8.4**

    TK 8.3   **TK 8.4**

    ITCL 3.2   **ITCL 3.2.1**

    IWIDGETS 3.0.0 IWIDGETS 4.0.0

- **These upgrades solved problems with path names, mainly / (UNIX) vs \ (Windows)**

- **In upgrading the above libraries, some TCL scripts had to be updated with the new ways to call the functions**

Changes to Library Functions

LIBBN, LIBOPTICAL, LIBSYSV and LIBWDB:
Required little or no changes (updates to reflect current C programming practices)

LIBTERMIO:
Made all functions stubs; the function calls are there, but have no functionality

LIBDM:
Added an OpenGL display manager for Windows

Changes to Library Functions

LIBBU:

- **Hard coded path to BRL-CAD location**

- **Changed \ to // for all paths to be compatible with Windows**

- **No parallel support was added**

- **Commented out this line in the routine that frees malloc memory**

$$*((int *)ptr) = -1; /* zappo! */$$

*   **It causes MGED to crash under Windows when a material name is undefined**

Changes to Library Functions

LIBRT:

- **Changed the pipes calls to Windows pipe calls**

- **Renamed variables *near*, *far* and *small*, to avoid Windows keywords**

- **Minor changed to various system function calls to convert to the Windows format**

Changes to Library Functions

LIBFB:

- Frame buffer server functions are stubs

- Added an OPENGL frame buffer interface for Windows

- Only the transient windows function at this time

Changes to MGED

- **Forced the default file open to be binary; otherwise the database file could not be opened**

- **Changed the pipes calls to Windows pipe calls**

- **Changed the routines that catch the data from *stdout* to place in the MGED command window**

- **Lots of minor changes to the calling syntax of functions for Windows compatibility**

- **Added editing functions (*edcodes*, *edmater*, etc.) to open *notepad***

DEMONSTRATION