CMSCAN as an NT/OpenInventor Testbed

A Possible CMSCAN on NT??

Purpose: Investigate next generation Event Display

- Stock OO Graphics API: OpenInventor from NAG/TGS
- Stock Computer: Pentium (II)
- Stock OS: NT
- Stock Programming Environment: VC++
Methodology

- Study OpenInventor
  - Take existing detector and data description (from CMSCAN Classic)
  - Re-write Frame routines from scratch
  - Implement only “interesting” shapes
    * Basic box
    * NURB-based tracks
  - Use OIV kits where possible
  - Embed dimensions in kit (debatable)
  - Test interactivity
  - Write out OIV scene in iv format
  - Translate scene to NT
- Learn NT environment
- Investigate WIN32 API as basis for GUI using transferred scenes
OIV Window Class Structure (from OIV Whitepaper)

- Designed to Integrate with Microsoft Foundation Classes (MFC)
- Includes AppWizard for VC++
- OIV X Widgets Available in Translated Form
- Supports Dynamic Link Libraries as containers for OIV objects
VC++ Programming Environment

- Editor
- Debugger
- Make Facility
- Help Library
- Class Browser
- Resource Editor
- Wizards
- Academic Pricing

28-30 January 1998
HEPVIS 98
CMSCAN NT/OIV Testbed
George Alverson
Northeastern University
Possible Useful WIN32 Widgets

- Multiple Document Window
- Tree View
- Standard Button, Slider, etc, controls

OIV SceneViewer built as a multiple document viewer
Performance for a Typical HEP Scene

Reading graph from stripped.iv
Number of frames: 60
Window size: 400 x 400 pixels

Number of nodes in scene graph: 3
Number of triangles in scene graph: 157896
Number of lines in scene graph: 0
Number of points in scene graph: 0

seconds/frame frames/second
As-Is rendering: 0.598 1.67
No Clear: 0.597 1.67
No Materials: 0.599 1.67
No Transforms: 0.589 1.70
No Fills: 0.711 1.41

Time taken by vertex transformations: 0.710 seconds/frame
Time taken to traverse scene graph: −0.004 seconds/frame

Results of running the OIV ivperf performance evaluation program on the Intergraph TD-225
### Comparison of Several Hardware/Software Configurations

Using the same test file (stripped.iv) under varying conditions:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Graphics Card</th>
<th>GL Library</th>
<th>Time Elapsed (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparc Ultra 3D Creator</td>
<td>TGS</td>
<td>6.999</td>
<td></td>
</tr>
<tr>
<td>SGI IP26/R8000</td>
<td>Sun</td>
<td>1.884</td>
<td></td>
</tr>
<tr>
<td>Intergraph TD-225</td>
<td>NA</td>
<td>2.009</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intergraph Intense</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3D Pro 2200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

\(^a\)Courtesy of Jeremy Walton, NAG
Complex Scenes

A Not Very Complex Scene

- Scene consists of one (1) sub-detector in CMS, the MSGC’s

- Time for rendering optimized scene on Integraph w/ Intense 3D Pro 2200 is 0.6 s ⇒ 1.7 fps. Smooth performance requires > 20 fps

- Rough estimate for gross model of detector: 10 × above complexity + physics representation

- CPU power doubles in 1.5 years (Moore’s Law); graphics power (extrapolated from recent results) doubles in 6 months. May just make it by LHC turnon…

- Until then, require selective visibility
  - External widget control, e.g., TreeControl
  - J. Boudreau unfolding mechanism
  - command line interface
Integration with Unix Versions

• UWIN: (Bristol Technology)
  - Exports WIN32 to unix
  - Per seat licensing costs
• NuTCRACKER - for porting X/Motif to WinNT/95
• MainSoft - for porting MS Windows API & MFC to Unix
Visibility Control Using the TreeView Control

Control Opened to Reveal Tree Structure

- Generated on-the-fly from Scene Information
- Cross selection between widget & scene

CMSCAN Classic Visibility Control for Detector Elements

- Self generating from internal tree description