Worldwide
Developers
Conference

QuickDraw™ 3D
Shawn Hopwood
Product Manager
David Harrington
Evangelist

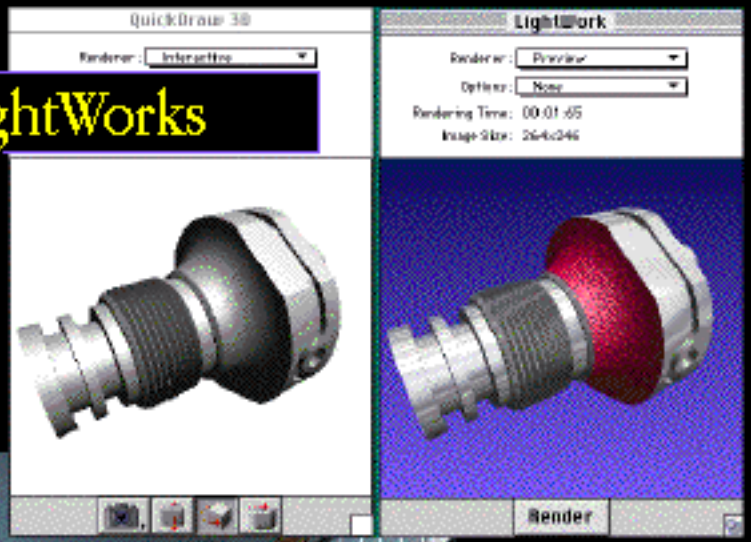
Current State of Affairs

- QD3D v1.5 Shipped 11/96
  - Won Comdex's Best of Show
  - First big step towards extensibility
    - Plug-in Rendering
    - Plug-in Groups
  - First version to showcase cross-platform (Win95/NT)
    - Simultaneous release and features
    - Uses native Windows 95/NT services

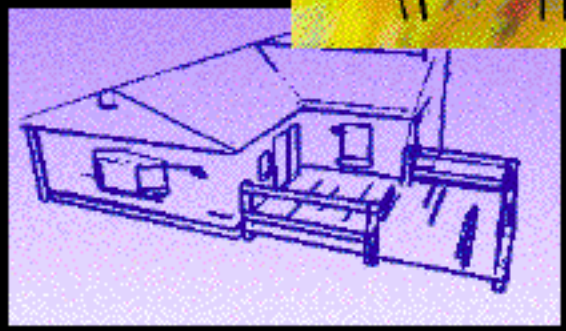
QuickDraw 3D Defined

Plug-In Rendering

LightWorks



ThinkFish



QD3D Accelerator Cards

- 3D Chips
  - 3D Labs
  - 3D F/X
  - SMOS
  - ATI
- 3D Accelerator Boards
  - Radius—Thunder 3D—(3D Labs)
  - Newer Tech—RenderPix—(3D Labs)
  - ATI—Xclaim VR/3D—(Rage II)

3Dfx Connects the Arcade and PC Markets with a Common Architecture

3Dfx connects coin-op and the PC with a scalable 3D architecture

- High value titles
- Preserves quality
- Reduces cost

Unifying Markets with a Common Platform

Consistent features, performance and software for developers

Voodoo, Voodoo Rush

- High fill rate
- Efficient triangle performance
- Texture effects
- Highly accurate rendering
- Advanced features: alpha blending, filtering, MIP mapping...

APIs: Rave3D, Glide, OpenGL, Direct3D

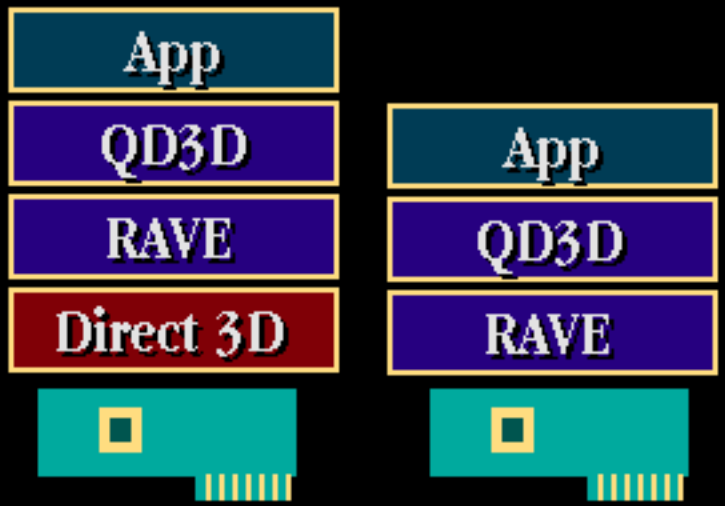
Scientific Visualization	Highly Realistic Simulation	Premium Entertainment
Arcade and Console Graphics	PC Game Graphics	Standard PC Graphics

3Dfx for the Mac Platform

- 3Dfx and Apple are working together to bring you this technology
- Commercial hardware products will be available shortly—contact Apple and the OEM suppliers
- 3Dfx is working closely with publishers and game developers—Interplay/Macsoft, GT Interactive
- We are actively seeking Internet and content creation applications developers
- Contact us!
  - Email: devprogram@3dfx.com
  - Web: http://www.3dfx.com
  - Phone: (888) FOR3DFX

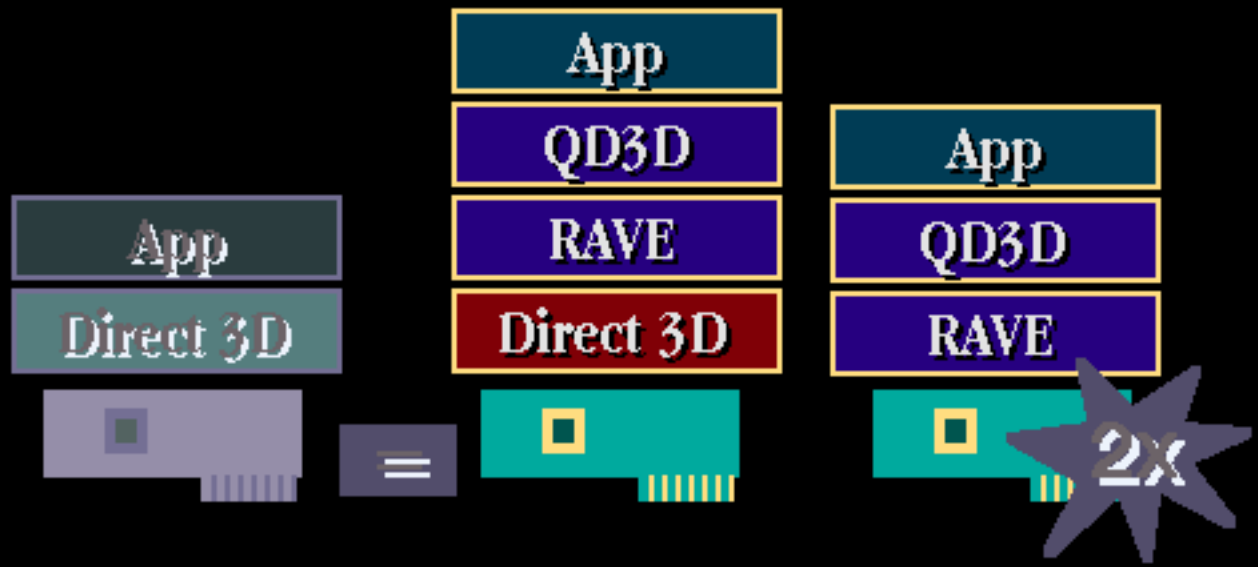
QuickDraw 3D on Windows

- Uses native file I/O
- Uses native draw context
- Will be able to use RAVE or Direct3D to access hardware acceleration



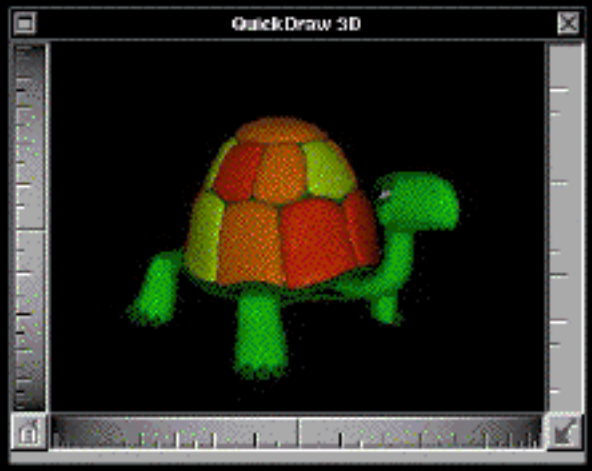
QuickDraw 3D on Windows

- Uses native file I/O
- Uses native draw context
- Will be able to use RAVE or Direct3D to access hardware acceleration



QuickDraw 3D and Rhapsody

- Blue Box (Sys.7.x)
  - QD3D will work without change
- Yellow Box (OpenStep)
  - Will be ported to support OpenStep APIs



What's Next?

- QD3D v.1.6–Q4 1997
  - Performance release
  - VRML 2.0 Rendering Compliance
  - 1st Draft of Shader Interface
- QD3D v2.0
  - Shader Interface Implementation
  - Next Generation H/W Abstraction Layer

Making QuickDraw 3D Internet Savvy

- QD3D Netscape plug-in
  - Allows for Drag & Drop from WWW to application
- 3DMF
  - File references
- Adding Walkthrough Mode in QD3D Viewer
- Enabling Plug-in Rendering
  - Cartoon Interactive Rendering
  - BSP Rendering

How Is VRML and the Macintosh Viewed Today?

- Apple Missing From VRML playing field
- No Shipping VRML Browsers
- No VRML 2.0 Authoring Tools

How Will We Change this Problem?

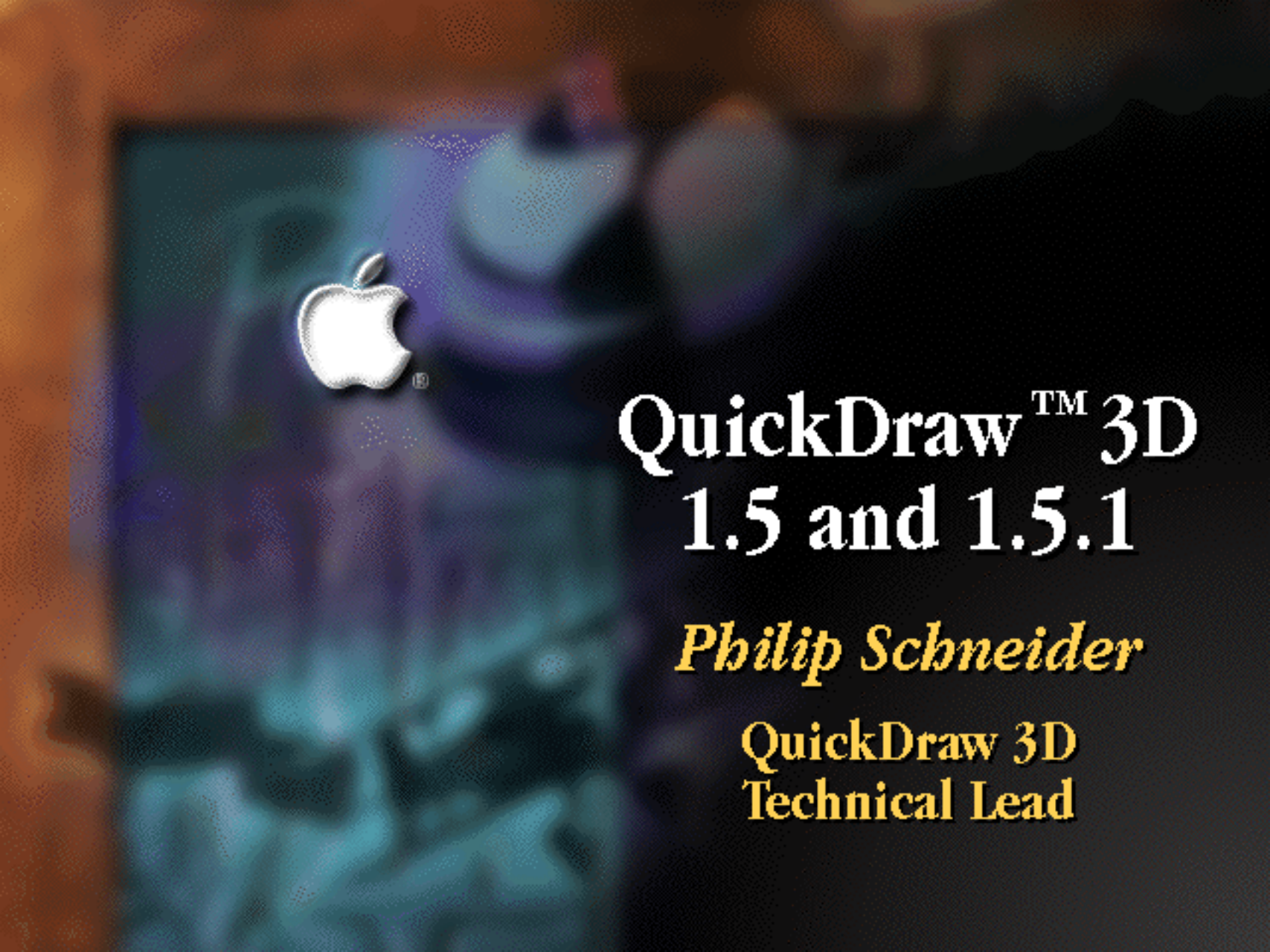
- Apple has a seat on VRML Board
- Creating alliances with Third-Party Vendors
- Fixing and adding new technologies to QuickDraw 3D

Key VRML 2.0 Partnerships Announced

- Apple, IBM and Paragraph
  - VRML 2.0 Binary Collaboration
    - Smaller
    - Faster
    - Geometry Compression
- Silicon Graphics
  - Apple to License VRML 2.0, Browser CosmoPlayer
  - Looking to partner further on VRML Technologies

Third-Party VRML Browsers and Authoring Tools

- Intervista
  - C based VRML Browser—WorldViewer
  - www.intervista.com
- Dimension-X
  - JAVA based VRML Browser—LiquidReality
  - www.dimension-X.com
- Paragraph
  - VRML Authoring Tool—VisualSpaceHomeBuilder (VSHB)
  - www.paragraph.com


QuickDraw™ 3D
1.5 and 1.5.1
Philip Schneider
QuickDraw 3D
Technical Lead

Introduction

Where to get started

- First steps
- Develop articles
  - See http://devworld.apple.com/dev/toc.shtml
- Debugging

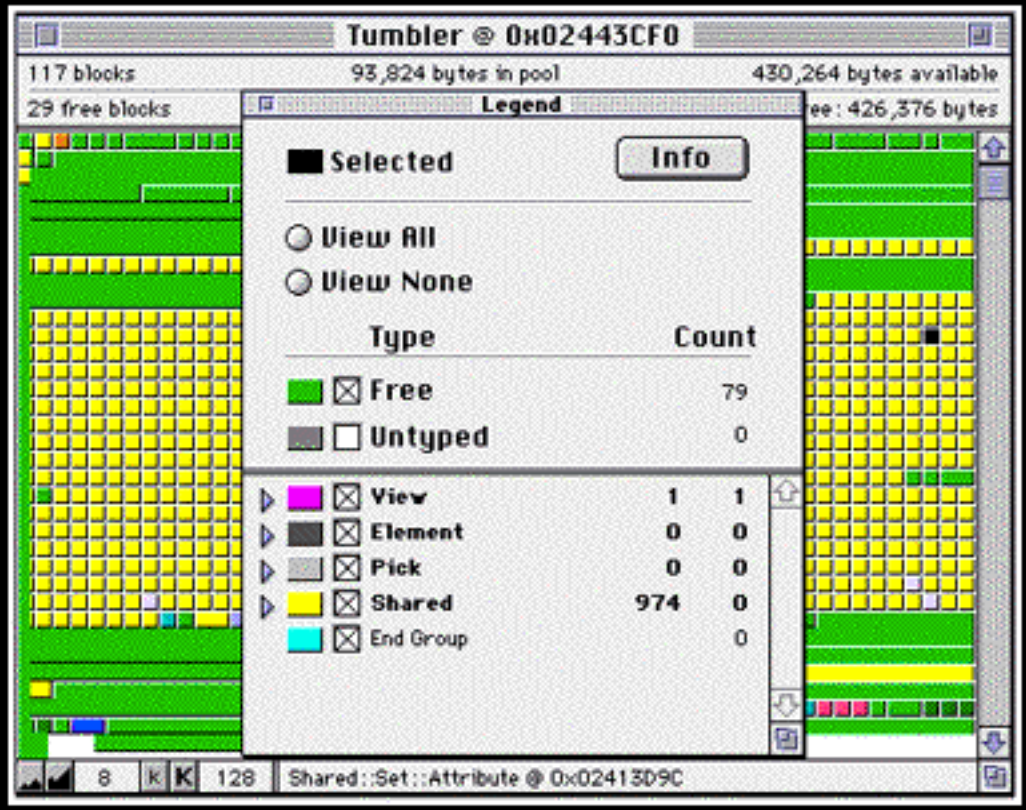
Basic Components of a QuickDraw 3D Application

First steps...

- View
  - Camera
  - Lights
  - Draw Context
  - Renderer
- Geometries, Transforms, Styles etc.
- For a good introduction to the basics see: "QuickDraw 3D: A New Dimension for Macintosh Graphics" in *develop* Issue 22

Debugging QuickDraw 3D Applications

- Use the debug version of the library
- Use 3Debug to look at memory usage



QuickDraw 3D on Windows

- Uses native file I/O
- Uses native draw contexts
- Windows-native Viewer provided
- Will be able to use RAVE to access hardware acceleration (via D3D Engine)

QuickDraw 3D on Windows 95/NT

- Cross-platform by design—only 3 platform-specific objects
  - Two Windows-specific draw contexts (interface to Windows system)
  - Windows-specific storage object

Windows 95/NT…

- Win32DC draw context
  - Compatible with existing code base using DCs
  - Must use CS_OWNDC window class style

```
typedef struct TQ3Win32DCDrawContextData {
    TQ3DrawContextData    drawContextData;
    HDC                   hdc;
} TQ3Win32DCDrawContextData;
```

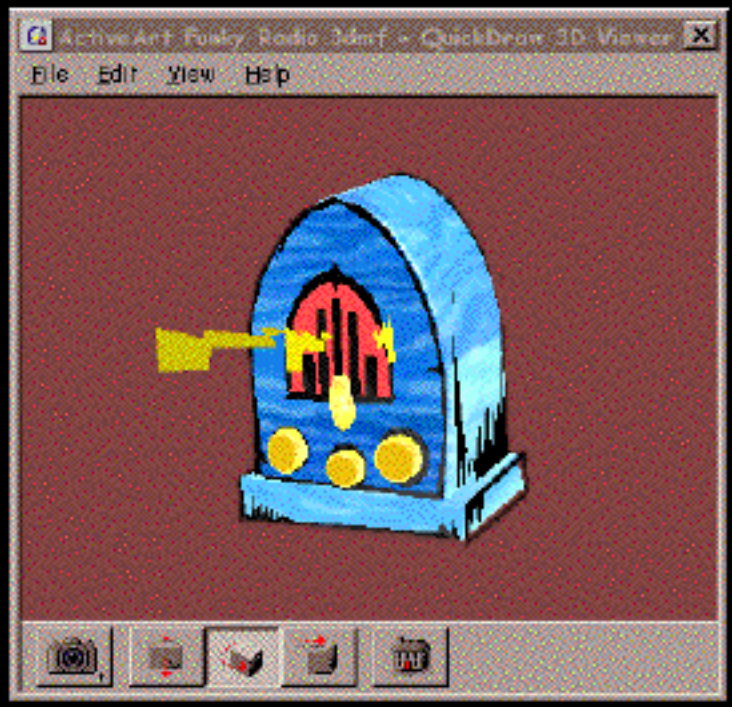
…Windows 95/NT

- DDSurface draw context
  - Can take advantage of 2D hardware acceleration for double-buffering and clearing
  - Requires more setup, but more flexible

```
typedef struct TQ3DDSurfaceDrawContextData {
    TQ3DrawContextData      drawContextData;
    TQ3DDSurfaceDescriptor ddSurfaceDescriptor;
} TQ3DDSurfaceDrawContextData;
```

Windows 95 and Windows NT

Screen Shot of QD3D Viewer



Plug-In Extensibility

- Plug-ins can be used to extend the functionality of QuickDraw 3D
- In 1.5 the following types of plug-ins are supported
  - Elements/Attributes
  - Groups
  - Renderers
- Support for plug-in shaders will be forthcoming

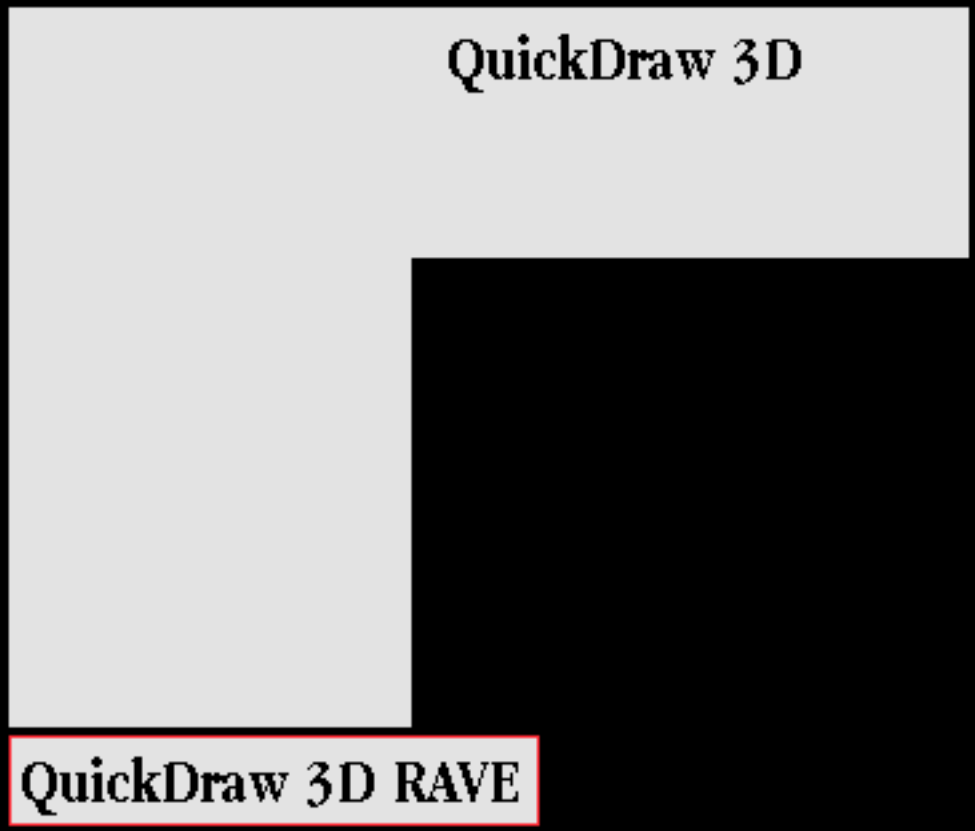
Commonality

- **Loading/Initialization**
  - On Mac OS this is handled by CFM
  - On Windows by the DLL loader
  - You need to supply a registration function
- **Metahandler**
  - Method dispatcher for the plug-in class
- **Termination**

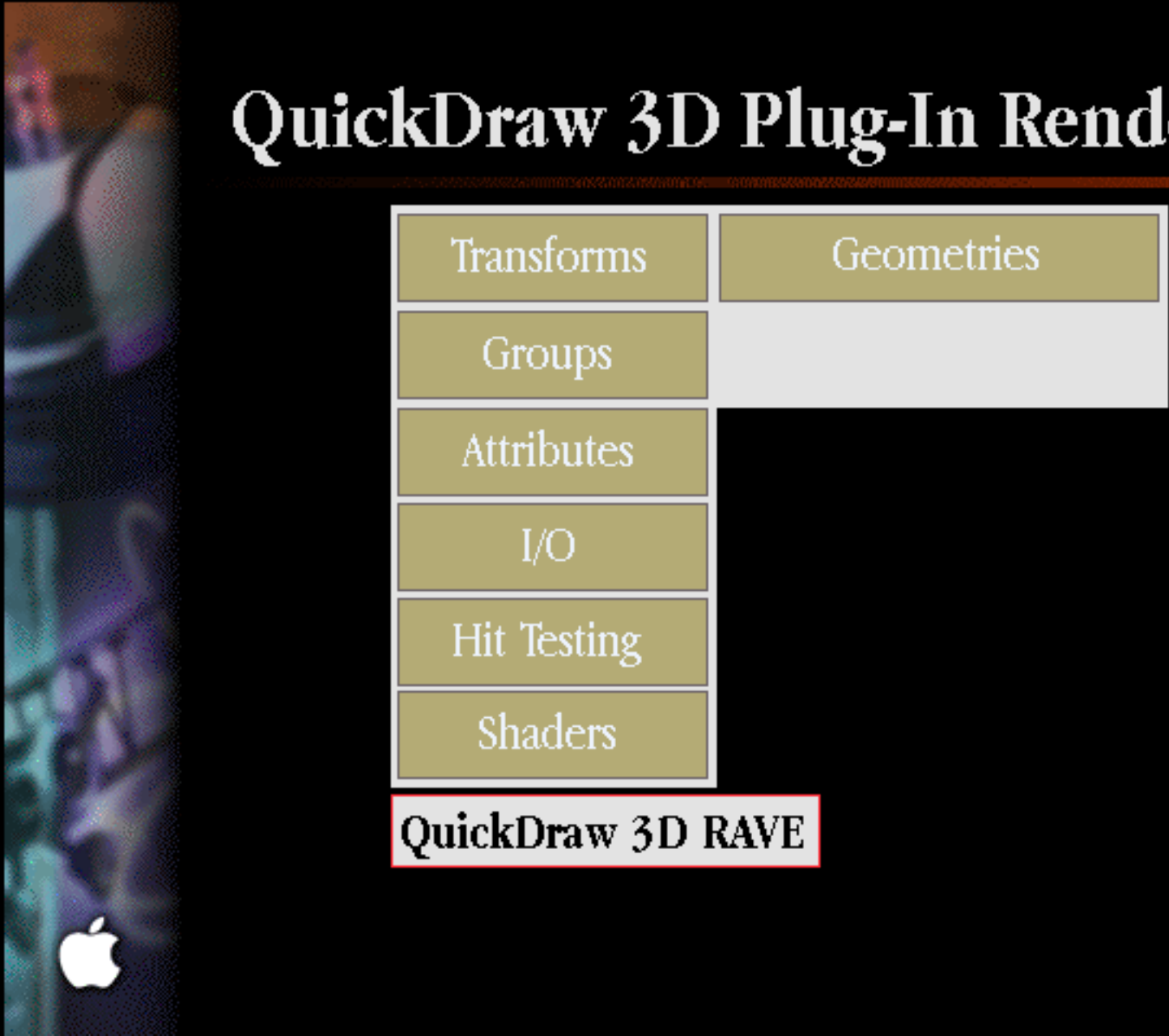
Plug-In Renderers

- 3 Steps to build a QD3D Plug-in Renderer
  - Register
  - Top Level Metahandler
  - 2nd Level MetaHandler
- Renderers are pretty involved in general, sample renderers and documentation provided on SDK
- QuickDraw 3D provides a mechanism for packaging a new or existing renderer

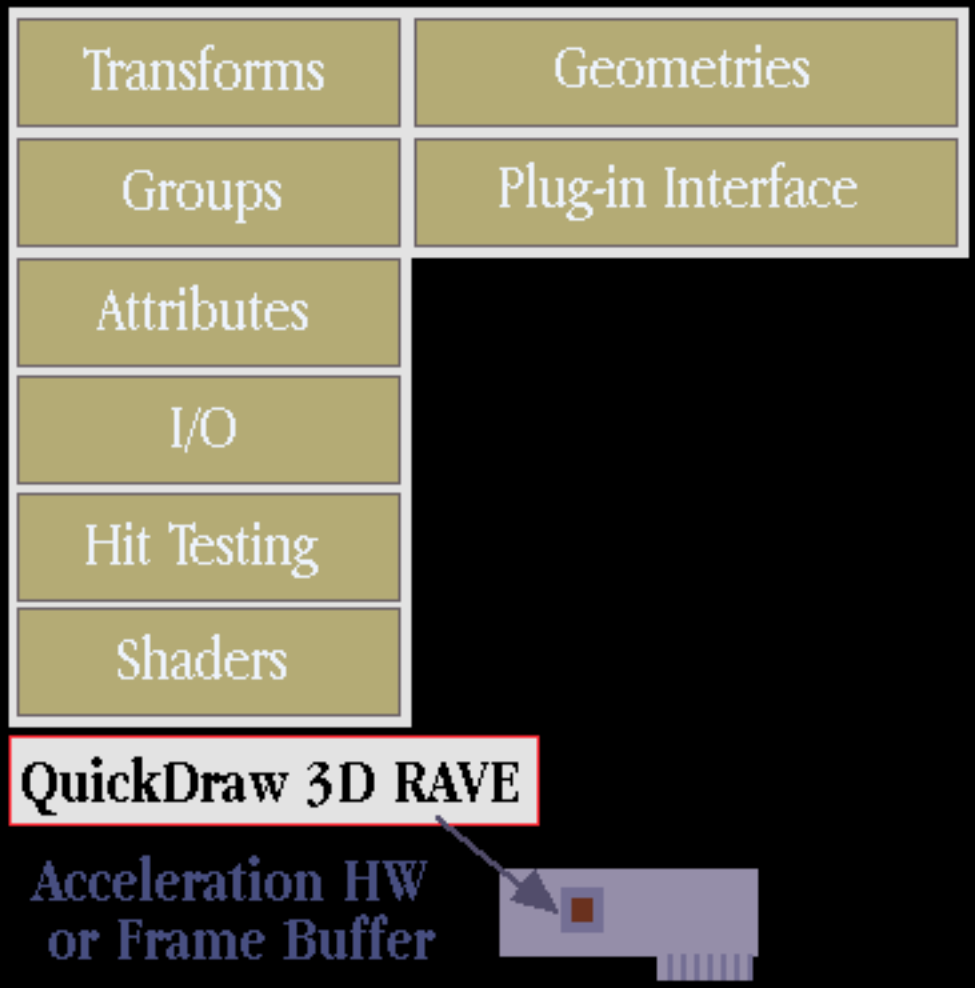
QuickDraw 3D Plug-In Renderers



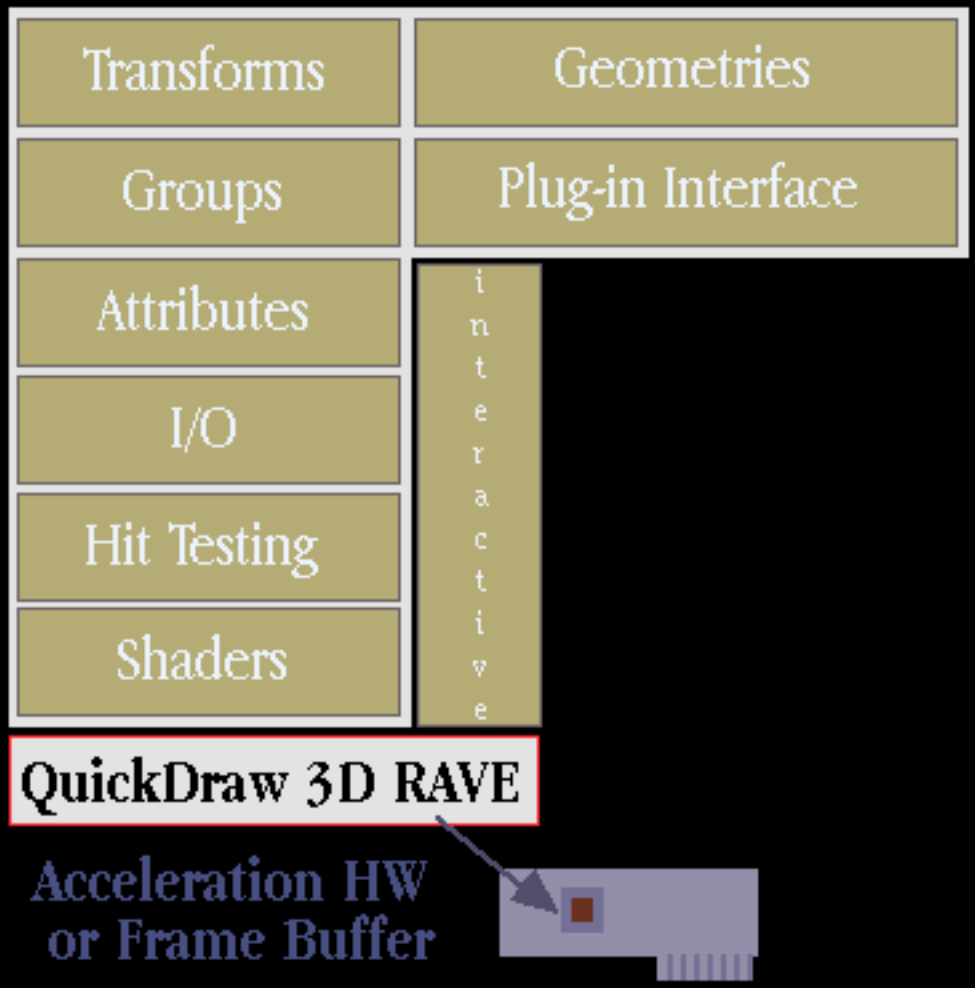
QuickDraw 3D Plug-In Renderers



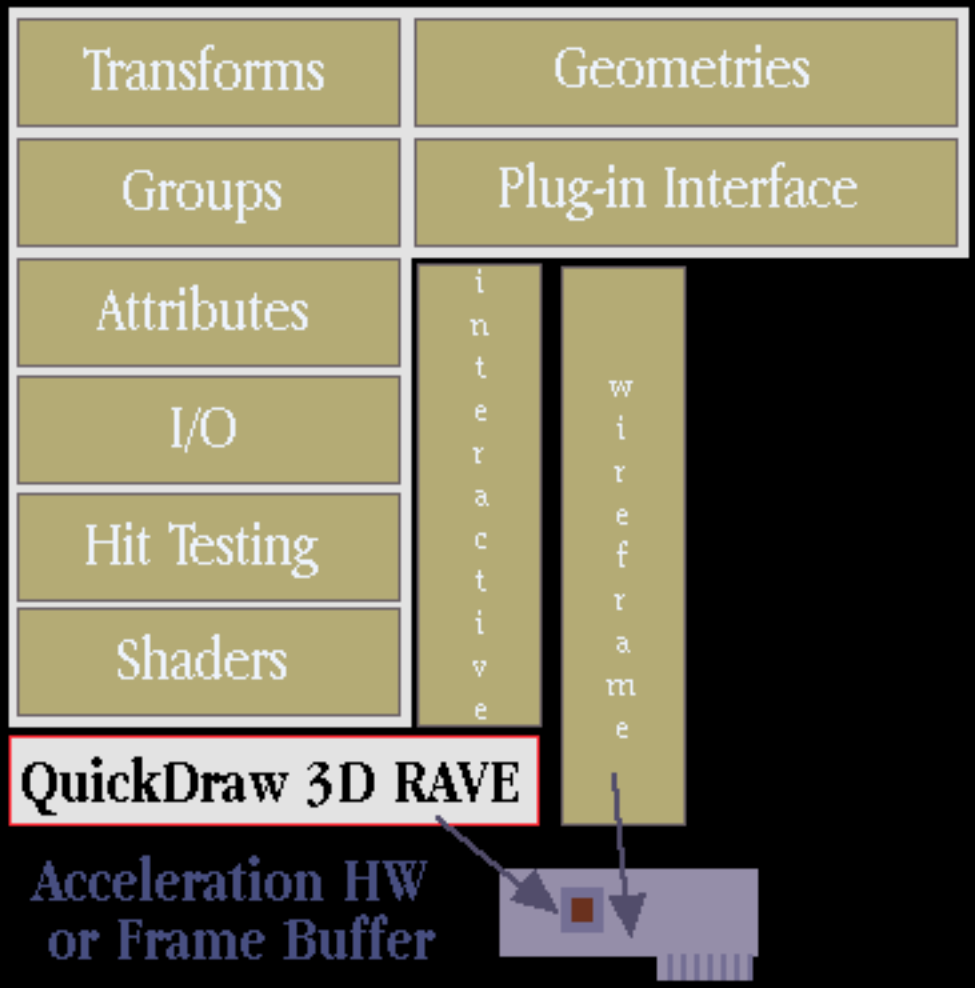
QuickDraw 3D Plug-In Renderers



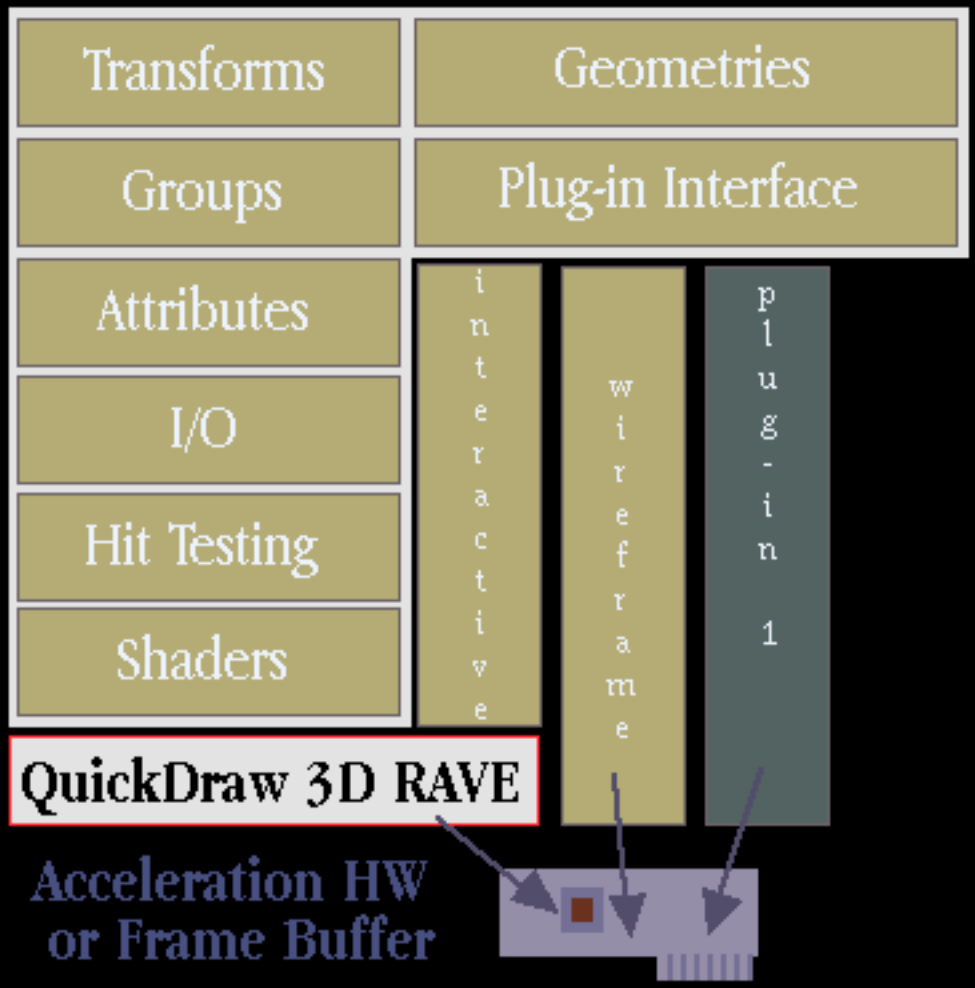
QuickDraw 3D Plug-In Renderers



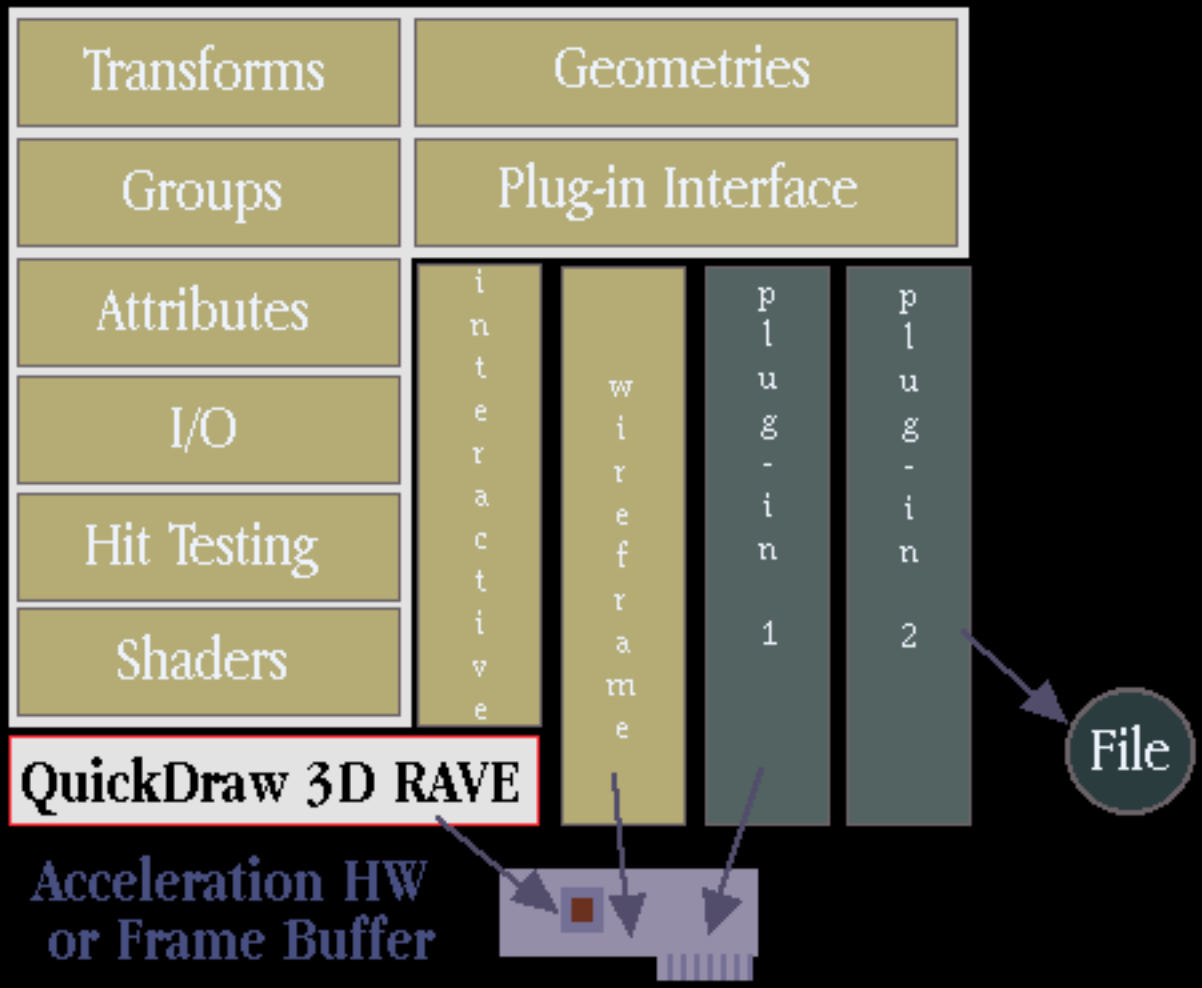
QuickDraw 3D Plug-In Renderers



QuickDraw 3D Plug-In Renderers



QuickDraw 3D Plug-In Renderers



Plug-In Renderers

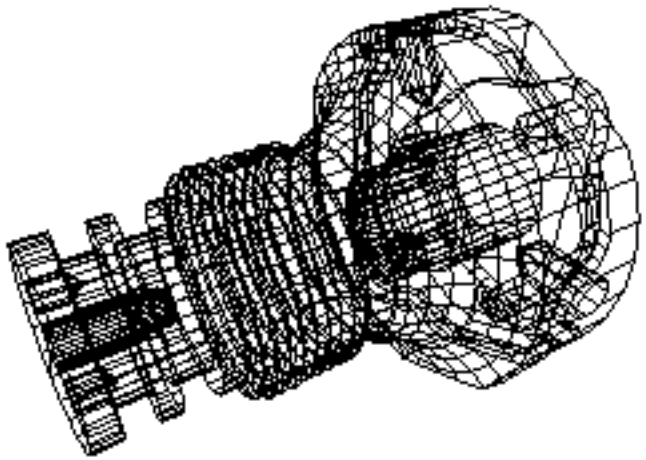
QuickDraw 3D

Renderer : Wireframe

Options : No Interpolation

Rendering Time : 00:01:21

Image Size : 264x246



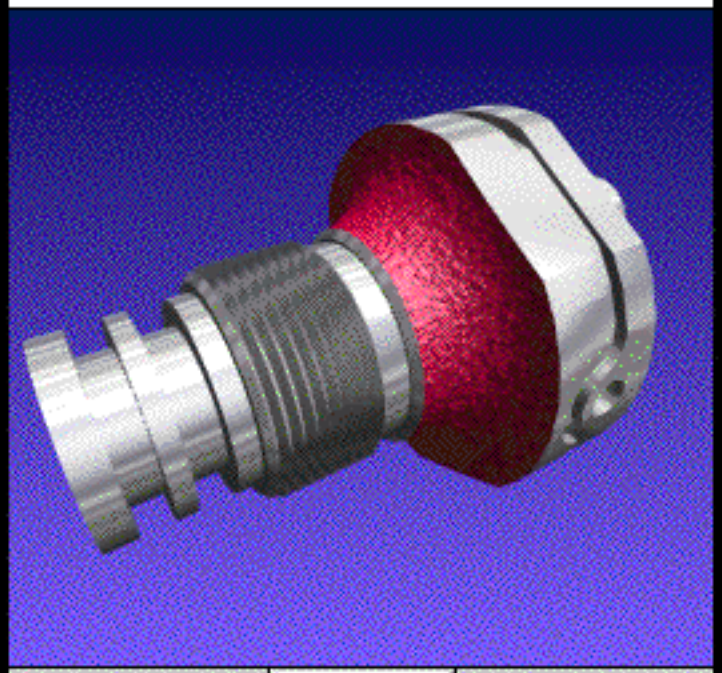
LightWork

Renderer : Raytrace Full

Options : None

Rendering Time : 00:34:55

Image Size : 264x246



Render

Metahandler

The plug-in method dispatcher

- Each class of plug-in has a set of methods that get called via the metahandler
- These methods vary depending on whether your plug-in class is
  - An Element or Attribute, a Group, or a Renderer
- The method handler is essentially a big switch statement, returning (usually) function pointers

Elements and Attributes

- See d e v e l o p Issue 26
- What's the difference between them?
  - Attributes can be inherited, elements are not
- There is a sample plug-in attribute on the SDK
- You should be migrating your attributes and elements to be plug-ins if you need to share them with other developers

Name Space Changes

Changes from d e v e l o p and the Book

- If you are reading the documentation for attributes note that the name space is altered
  - "X" is added to extensibility names
- Some examples:

```
TQ3FunctionPointer              -> TQ3XFunctionPointer
Q3View_SubmitWriteData          -> Q3XView_SubmitWriteData
kQ3MethodTypeObjectTraverse -> kQ3XMethodTypeObjectTraverse
```

Geometric Primitives

- Simple
  - Point, Line, PolyLine, Triangle, Polygon, General Polygon, Box
- Labels
  - Marker, Pixmap Marker†
- Surface modeling
  - TriGrid, Mesh, TriMesh†, Polyhedron†

†New in 1.5

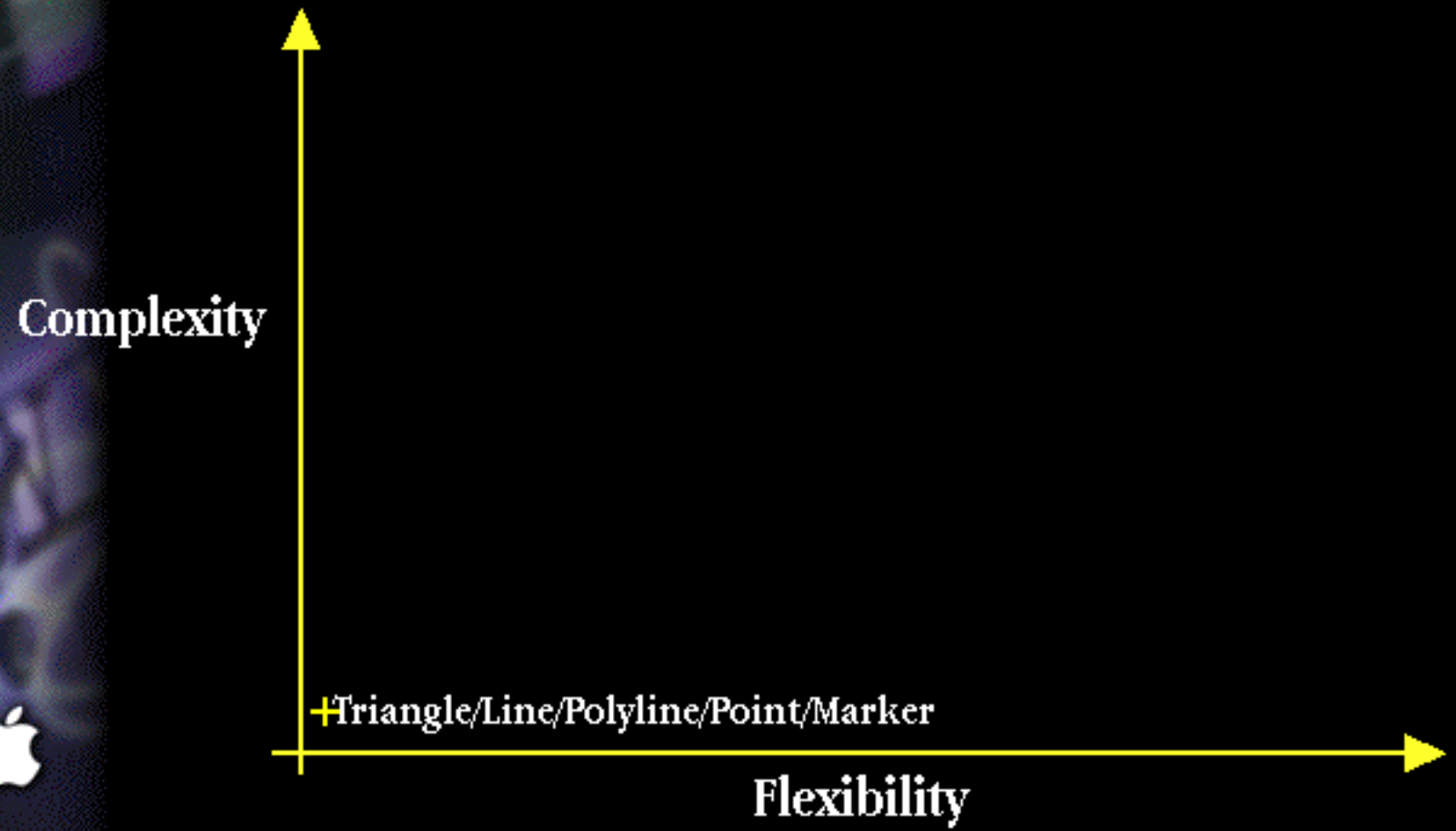
Geometric Primitives *(cont.)*

- **NURBs**
  - NURB Curve, NURB Patch
- **Conics and Quadrics**
  - Ellipse†, Disk†, Torus†, Cylinder†, Cone†, Ellipsoid†

†New in 1.5

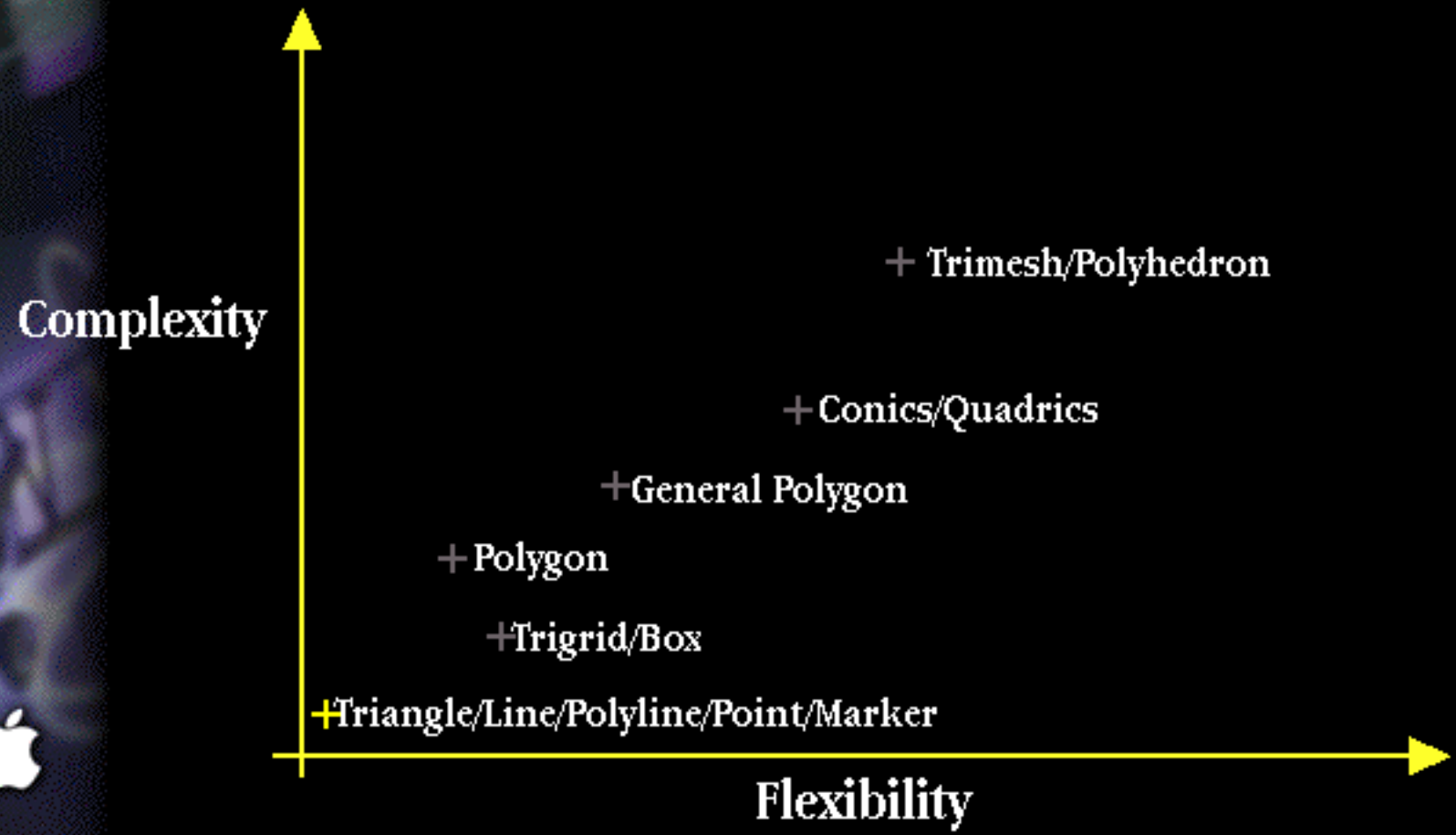
Geometries

Complexity vs. Flexibility

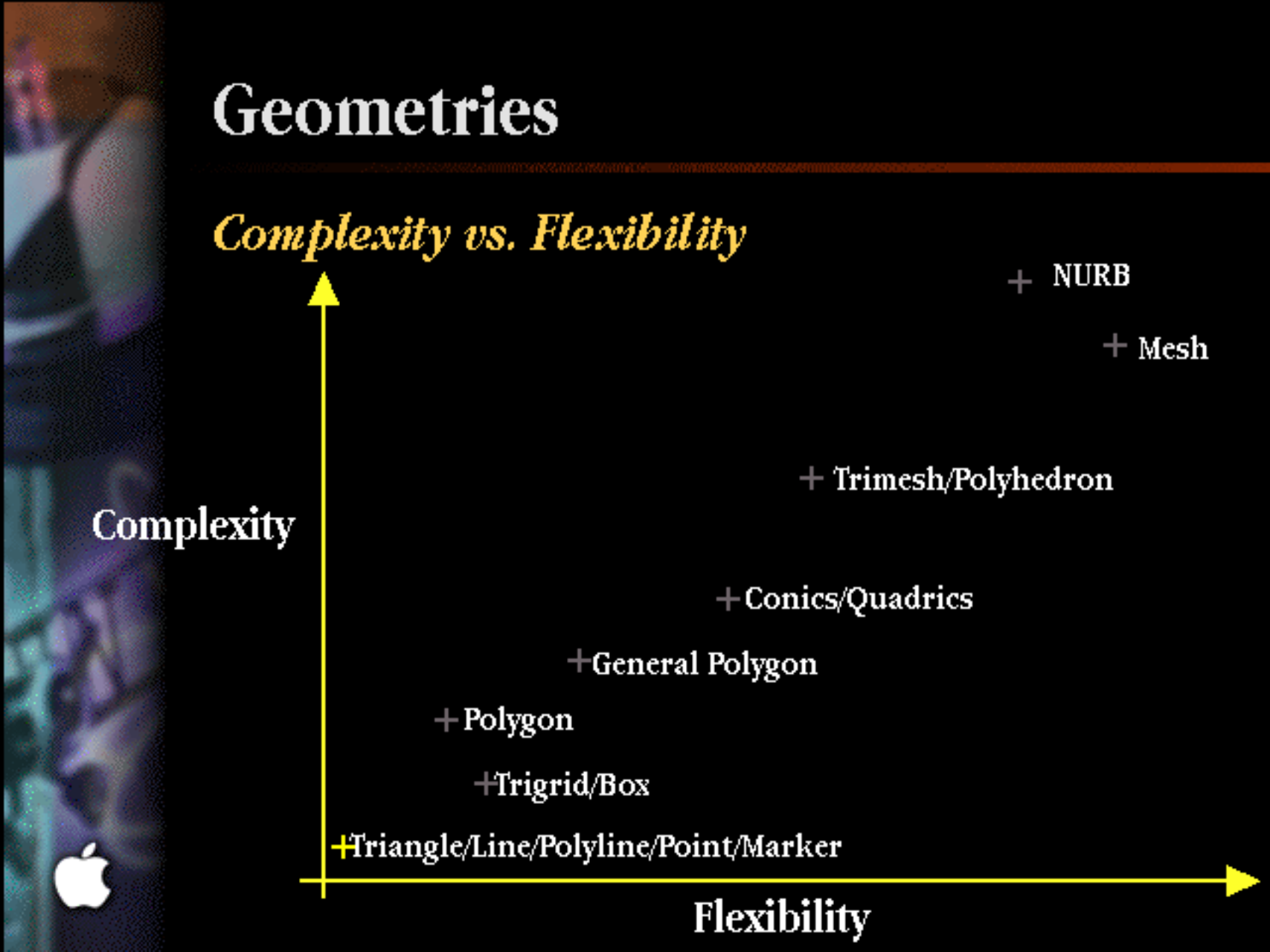


Geometries

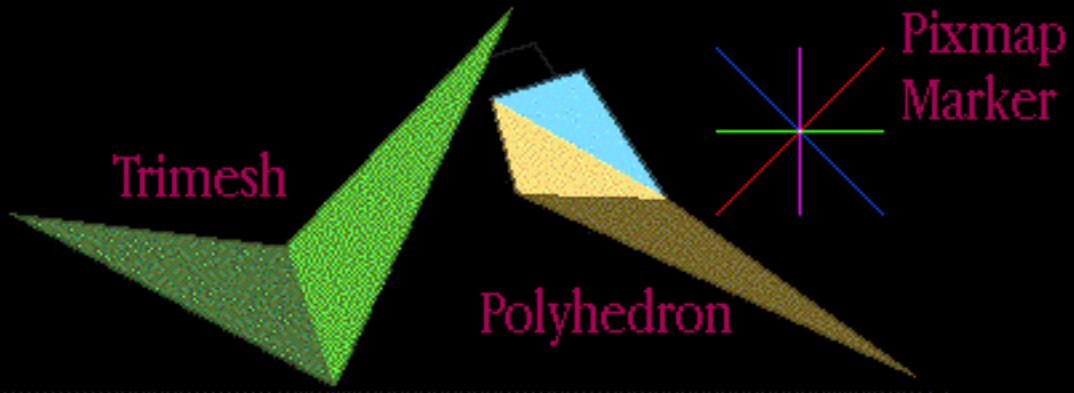
Complexity vs. Flexibility

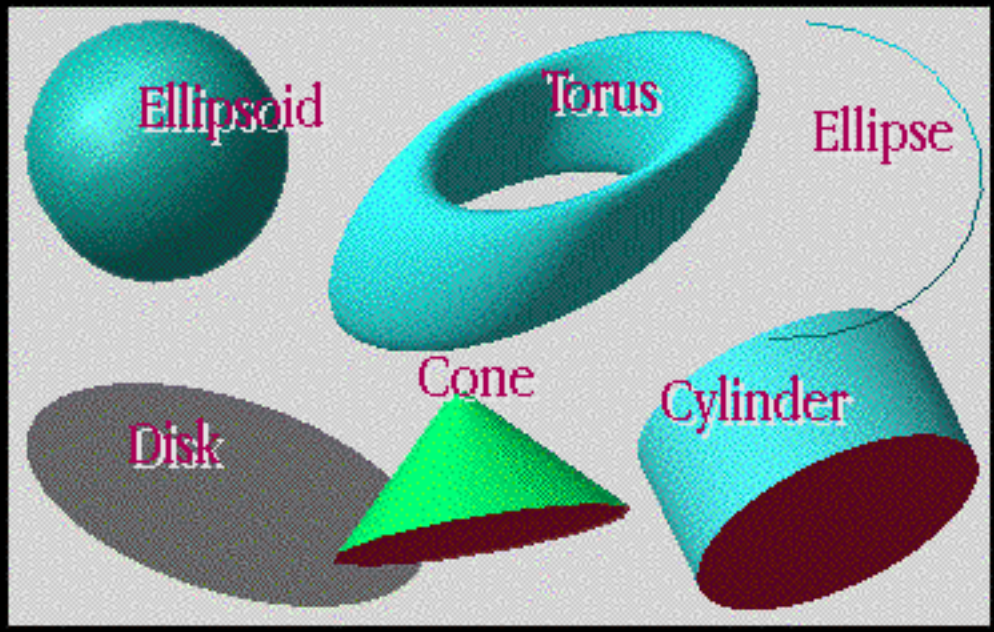


Geometries

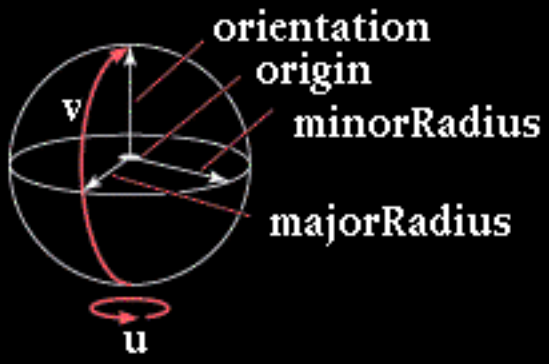

Complexity vs. Flexibility
NURB
Mesh
Trimesh/Polyhedron
Complexity
Conics/Quadrics
General Polygon
Polygon
Trigrid/Box
Triangle/Line/Polyline/Point/Marker
Flexibility

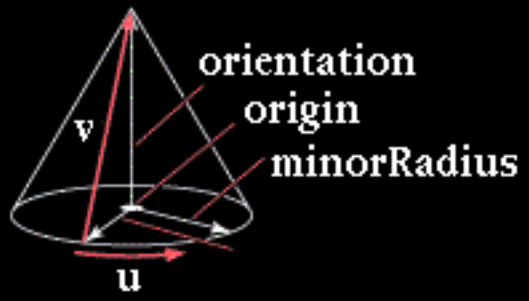
New 1.5 Geometries

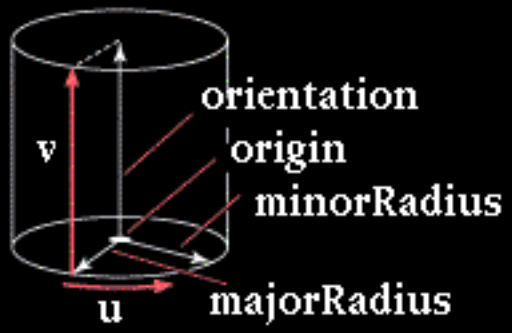


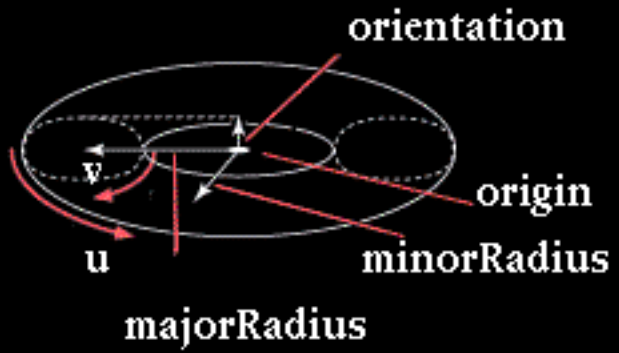


Conics and Quadrics





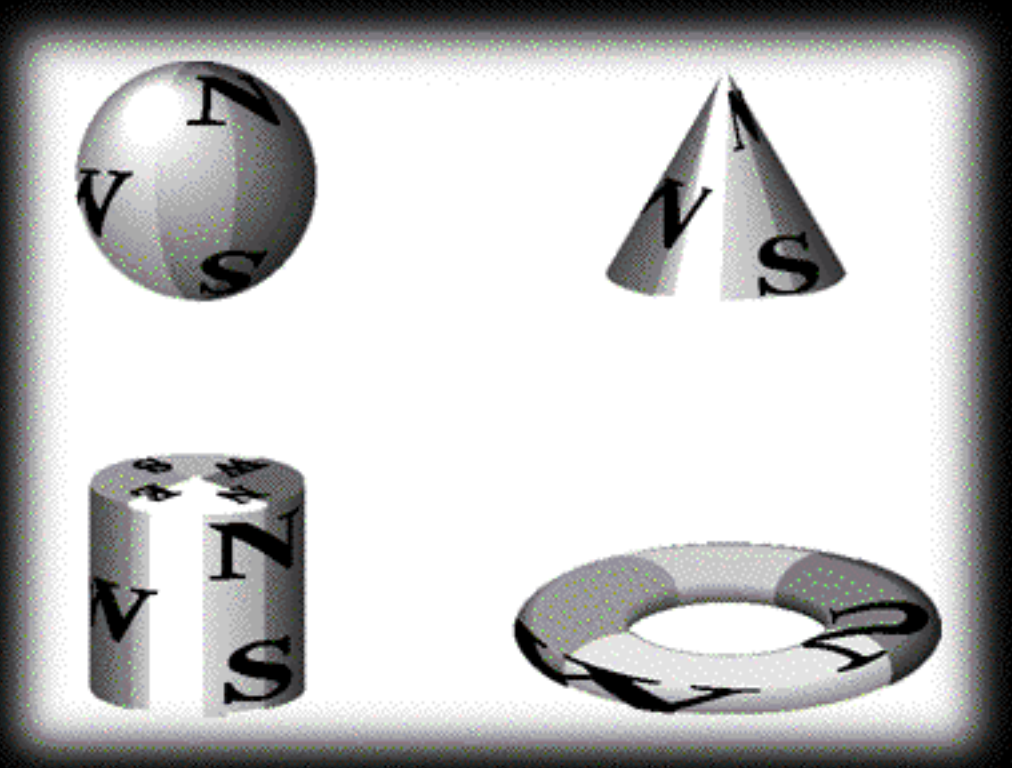




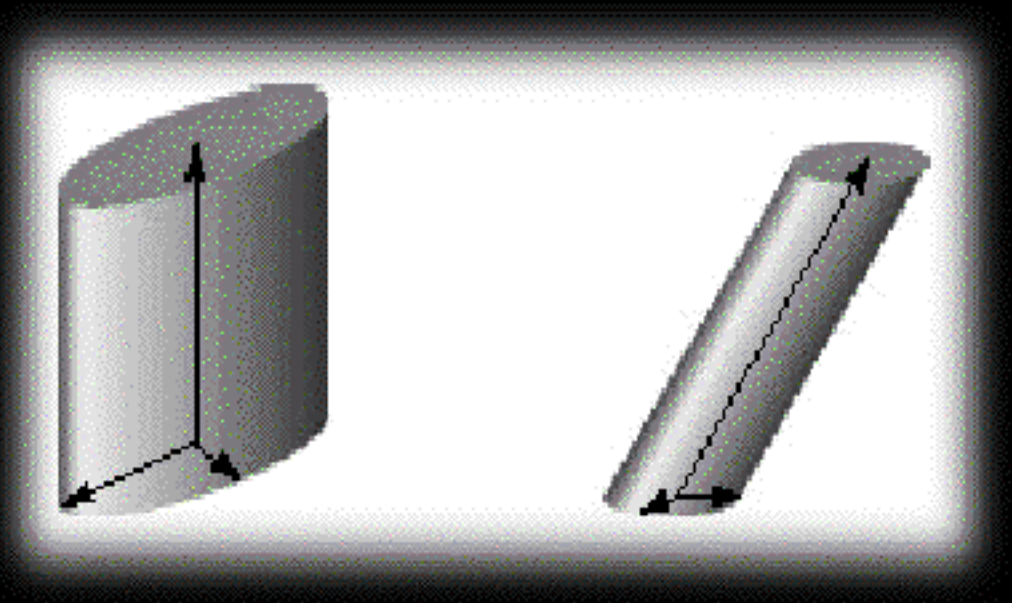
Important Things To Know™

- Generally, the axes should be orthogonal (all at 90-degree angles to one another)
- Non-orthogonal axes are permissible, and result in skewed objects
- Axes' relative lengths may be anything you want, which results in objects with elliptical cross-sections
- Skewed, elliptical cross-section objects are legal

Conics and Quadrics



Non Orthogonal Axes

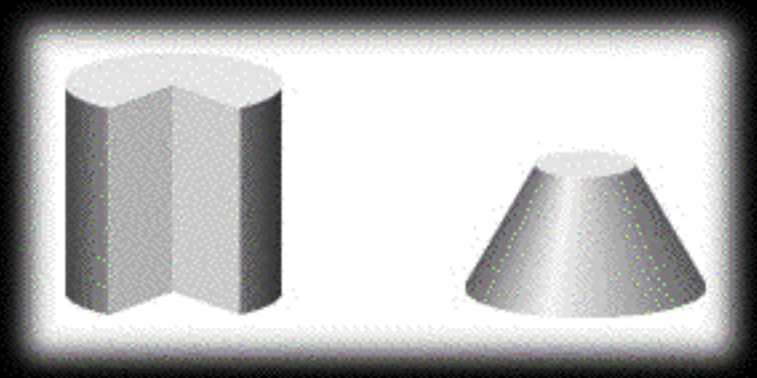


Conics—Future Directions

- The uMin and uMax, and vMin and vMax, fields allow one to make partial object
  - Hemispheres, truncated cones, etc.
- This feature is not yet released
  - In QuickDraw 3D 1.5 the min values must be set to 0 and the max values must be set to 1

Future Stuff—Post 1.5

- A partial Cylinder and a partial Cone

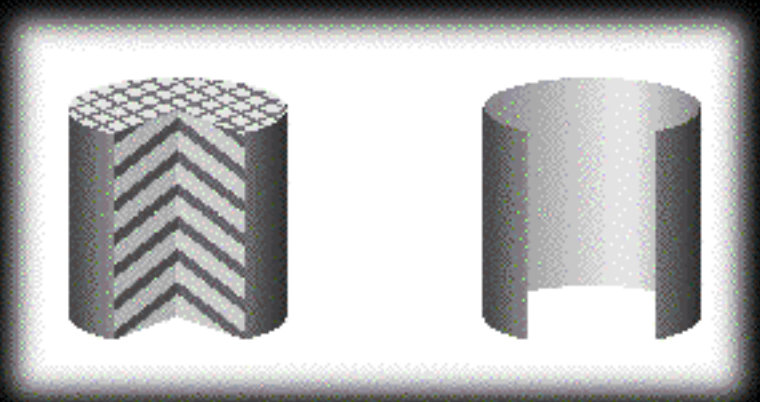


Conics—End Caps

- The “endCap” field allows one to place (or not place) end caps on the cone and cylinder
- When the partial objects are enabled, the “interior” cap, if set, will add a face over the cut-out section
  - i.e., you can make a partial ellipsoid look as if it were cut from a hollow ball if you specify no interior end cap, and from a solid ball if you do specify an interior end cap

Future Stuff—Post 1.5

- Cylinders with and without end caps or interior end caps



Conics—End Caps

1.5 Implementation

- Currently, the only legal values are:
  - kQ3EndCapNone, kQ3EndCapTop, kQ3EndCapBottom, or (kQ3EndCapTop | kQ3EndCapBottom) for the cylinder
  - kQ3EndCapNone or kQ3EndCapBottom for the cone
  - kQ3EndCapNone for the ellipsoid and torus

TriMesh and Polyhedron

- Mesh introduced in 1.0 is repositioned as
  - Free form geometry
  - For interactive authoring/manipulation
- New geometries introduced to fill in the role previously reserved for the Mesh
- Both based on connected triangular faces, which share vertices and edges

TriMesh and Polyhedron

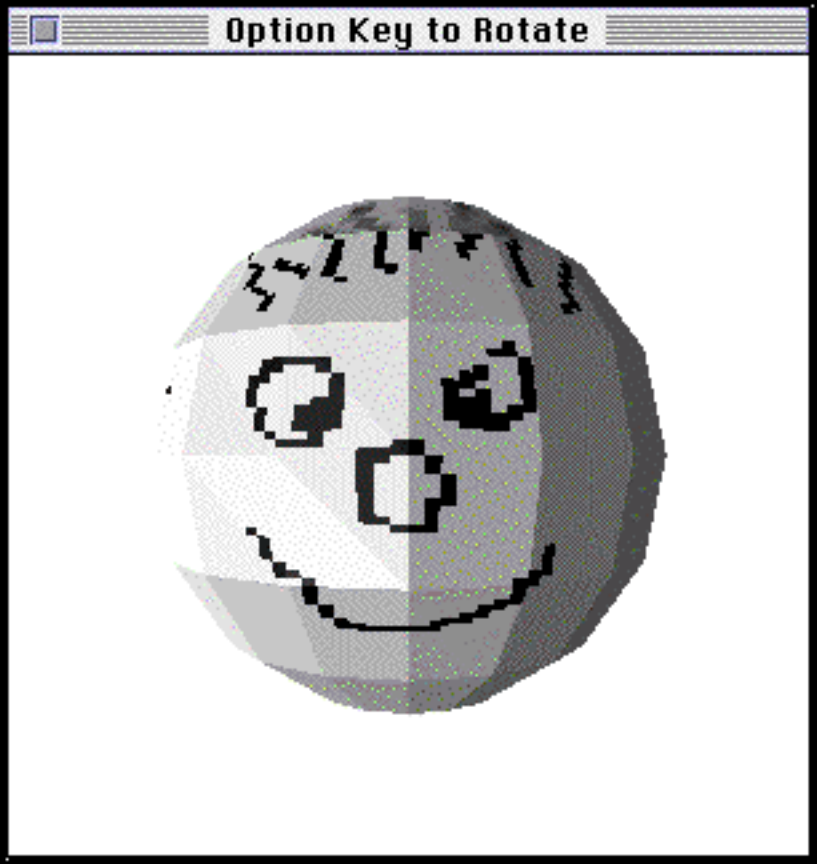
- Both geometries provide for immediate and retained modes (the Mesh continues to be only retained)
- For both geometries, you have to do your own tessellation (i.e., breaking complex faces into triangles)
- Polyhedron makes use of the same AttributeSet that are used by all other geometries, the Trimesh uses a unique flat attribute model

So Which Do I Use?

Characteristic	Polyhedron	Trimesh	Mesh	Trigrid
Memory usage	very good	good	poor	very good
File space usage	very good	good	very good	very good
Rendering speed	good	very good	good	good
Topological obj editing	poor	impossible	very good	impossible
Topological data struct. ed.	fair	fair	impossible no data str.	impossible fx topology
Geometric data structure editing	very good	very good	impossible	very good

UV Picking

- Allows cool 3D painting on Objects
- See TextureEyes demo
- Sample app on the SDK—BoxPaint



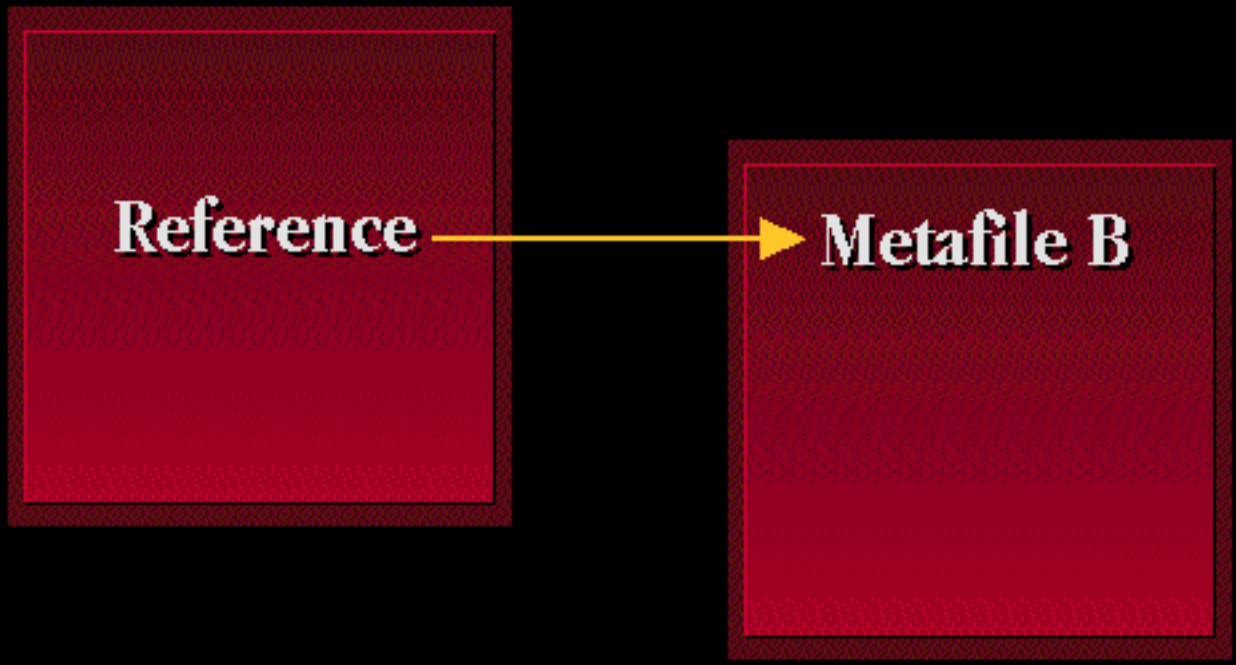
File References

Allow references to 3D data across files

- End of monolithic and gigantic files
- Allows for caching of 3D data
- Reduces bandwidth and download time
- Ships as part of QuickDraw 3D 1.5

I/O File References

- You can reference one metafile from another metafile



Levels of Detail—LOD

- Same object represented by proxies at different levels of detail
- Distance to camera controls detail selection
- Far away objects render faster
- Implemented as a DistanceProxyGroup, a subclass of DisplayGroup
- Source code for this is on the conference CD

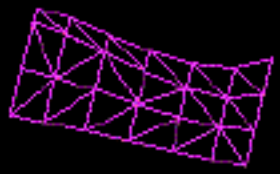
I/O: Group Traversal Control

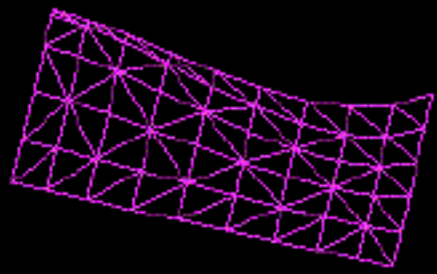
"Diving into Groups"

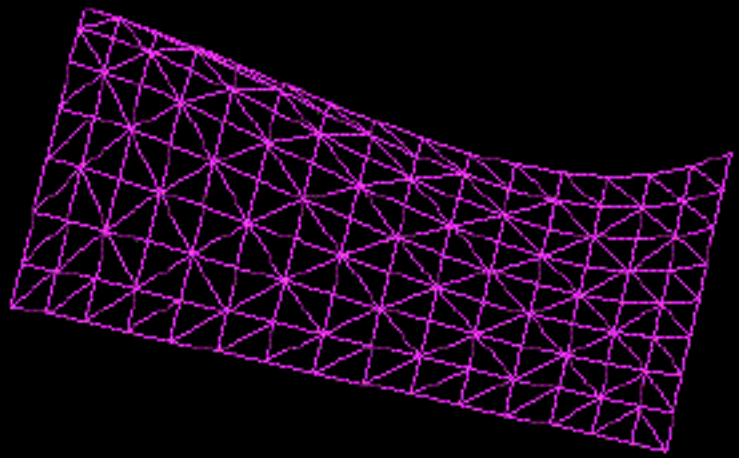
```
typedef enum TQ3FileReadGroupStateMasks {
    kQ3FileReadWholeGroup           = 0,
    kQ3FileReadObjectsInGroup       = 1 << 0,
    kQ3FileCurrentlyInsideGroup     = 1 << 1
} TQ3FileReadGroupStateMasks;

typedef unsigned long TQ3FileReadGroupState;

TQ3Status Q3File_SetReadInGroup(
    TQ3FileObject               file,
    TQ3FileReadGroupState       readGroupState);

TQ3Status Q3File_GetReadInGroup(
    TQ3FileObject               file,
    TQ3FileReadGroupState       *readGroupState);
```

Adaptive Tessellation

- Level of detail changes as object is moved “closer” to the camera location
- Will improve performance for:
  - NURB Patches
  - Conics and Quadrics



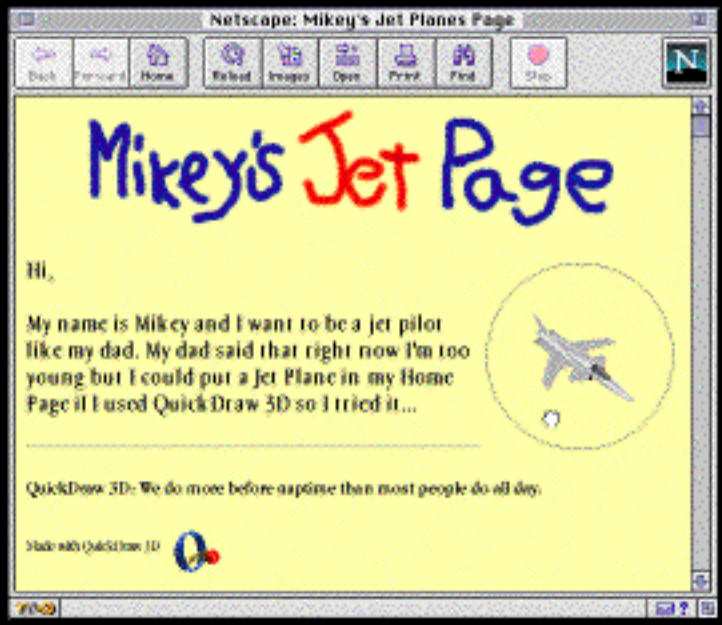




Netscape Plug-In

Now supports QuickDraw 3D 1.5

- On the QuickDraw 3D web site RSN
- Allows you to embed 3DMF in HTML



Viewer Improvements

- The UI for the Viewer has been improved
  - Better user interaction with the "virtual sphere"
  - Camera view pop-ups
  - Improved support for drag and drop
  - Improved control over drawing
    - New separate calls for control strip and content window
- See develop issue 29 for more information

QuickDraw 3D RAVE

"Renderer Acceleration Virtual Engine"

- Low-level API
  - Targeted at games, simulation, real-time applications
- Hardware Abstraction Layer
  - Hardware vendors write one RAVE driver
  - Software developers get acceleration on many cards for free
- Very thin layer
  - Hardware speeds with a software interface

RAVE Benefits

- Accelerates the rasterization and hidden surface removal (HSR) operations on 3D data including triangles, points, and lines
- Provides the fastest possible path to 3D hardware acceleration
- Provides the features that can run the most efficiently in hardware

RAVE Concepts

- **RAVE Manager**
  - Responsible for the loading of RAVE drawing engines and informing the application of all the available RAVE engines
- **Drawing Engine or RAVE Engine**
  - A shared library loaded by the RAVE Manager that implements the main RAVE drawing functions

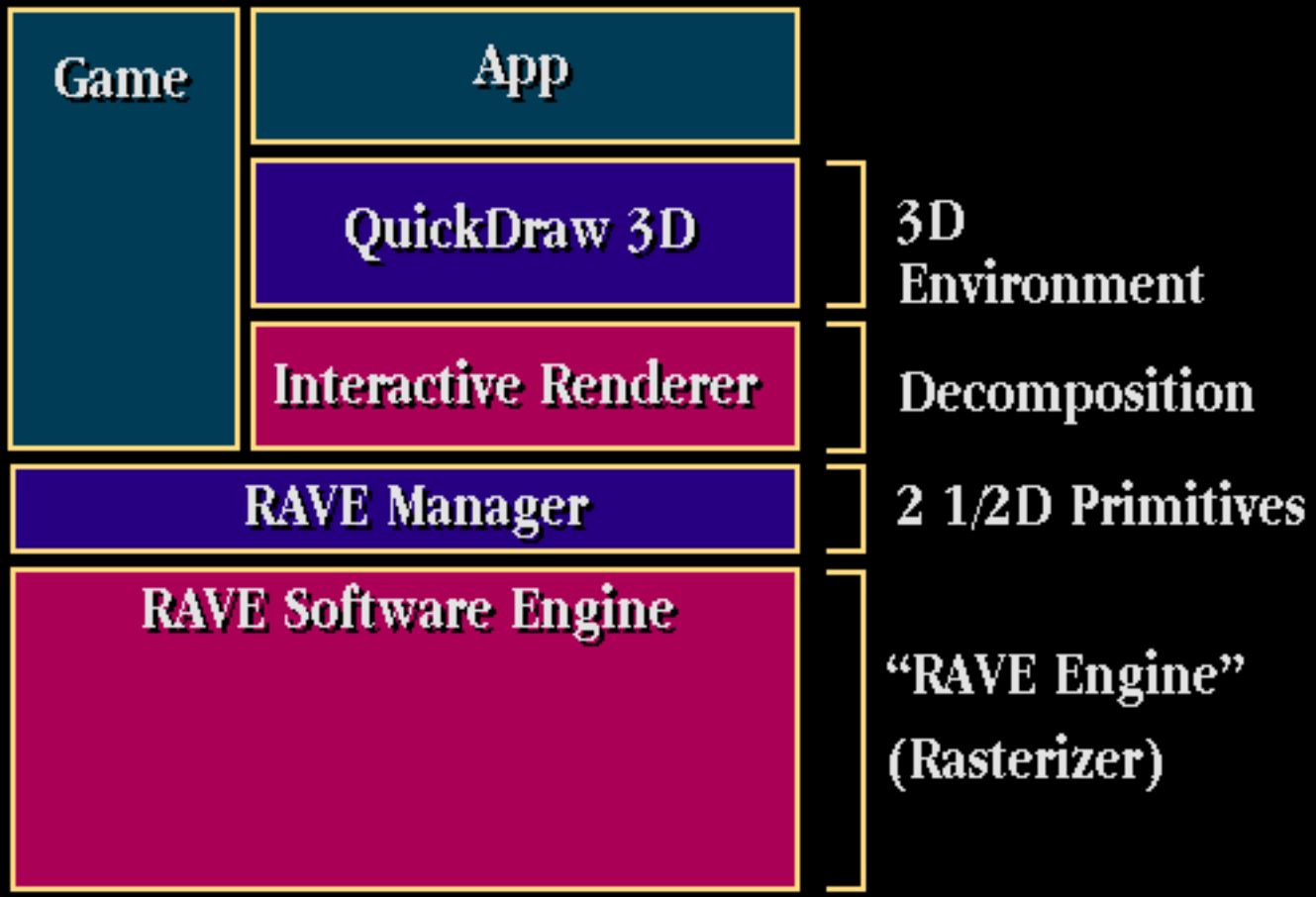
RAVE Concepts

- Draw Context
  - A rectangular region on a device specified by the application and managed by a RAVE engine into which drawing occurs
- Clipping Region
  - A platform dependent mask that prevents pixels from being written into the frame buffer—used for maintaining the window system appearance

RAVE Features

- Gouraud Shading
- Perspective correct texture mapping with full diffuse and specular modulation per channel
- Support for ZBuffer/non-ZBuffering HSR algorithms
- Fast drawing primitives with full vertex sharing—TriMesh
- Wide variety of texture formats including CL8 and CL4

RAVE Architecture



RAVE Architecture

What RAVE Does Not Support in 1.5

- Geometry processing
- Lighting model
- Geometric clipping
- Multiple monitors

Future of RAVE

These new features are under consideration

- Single pass texture compositing
- Texture mipmap selection bias
- Fog
- Shadow volumes
- ZBuffer access, utilities
- Will be backward compatible

Useful New API Calls for 1.5.1

- **Plug-in renderer support**
  - Q3Renderer_IsInteractive
  - Q3Renderer_GetConfigurationData
  - Q3Renderer_SetConfigurationData
  - Q3RendererClass_GetNickNameString

What's New in 1.5.1?

Bug Fixes!

- Panes and Masks on Macintosh Draw Context
- Pixmap Draw Context now allows kQ3PixelTypeARGB32 and kQ3PixelTypeARGB16
- ViewIdle and ViewIdleProgress methods
- Sorting Line pick hits
- Various Viewer bugs

For More Information

- develop magazine
  - QuickDraw 3D articles on the Web see: http://www.devworld.apple.com
- Mailing list
  - Send mail to:
    - quickdraw-3d-request@devtools.apple.com
    - with 'help' as the title of the E-mail
- QuickDraw 3D web site
  - http://quickdraw3d.apple.com

Awards and Kudos

DISCOVER MAGAZINE AWARDS FOR TECHNOLOGICAL INNOVATION

Discover MAGAZINE

BYTE Award of Distinction

impact

Macworld Editors Choice

MacUser Best New Technology

Other QuickDraw 3D Related Sessions

At WWDC '97

- 307 QuickDraw 3D Tips and Tricks
  - Tips on improving your application
    - Performance
    - Using plug-in renderers
    - Writing plug-ins
  - 1:50 pm, Friday, Room A1
- 392 QuickDraw 3D Feedback Forum
  - Tell us what you want us to do
  - 5:50 pm, Friday, Room J1

Worldwide
Developers
Conference