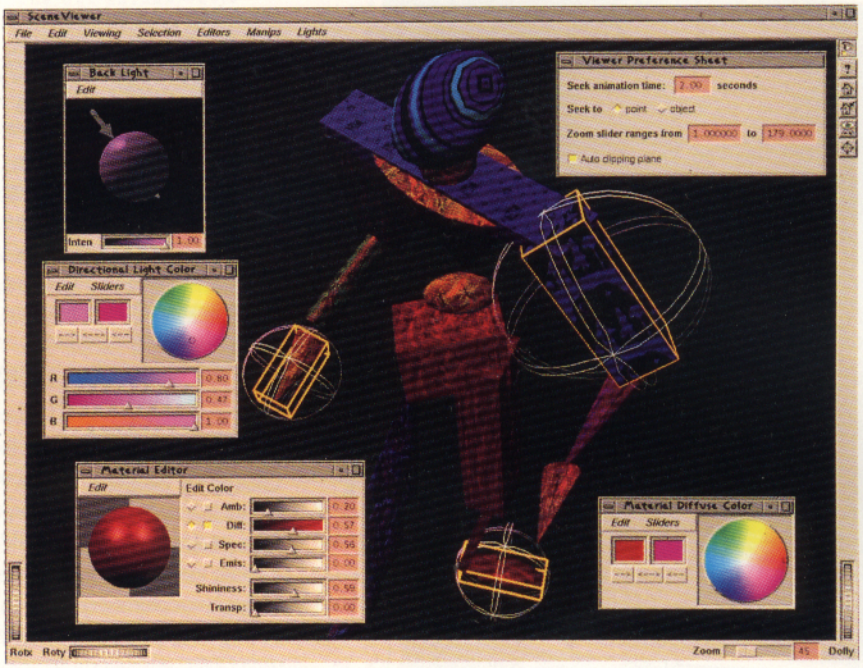
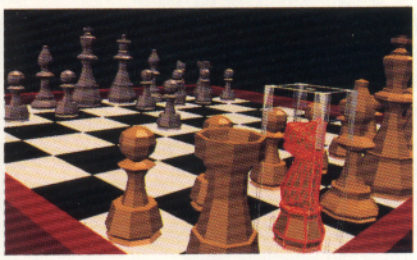
Introducing IRIS Inventor

Visual Highlights from the IRIS Inventor 3D Object Toolkit

By Rikk Carey



Most of the code for this interactive 3D program lives in the toolkit; writing this program was a simple matter of choosing the appropriate toolkit objects.



This image shows a handle box manipulator, attached to a chess piece, and automatic selection highlighting. This is a prototype for a 3D chess game that works only in 3D — no 2D widgets were used.

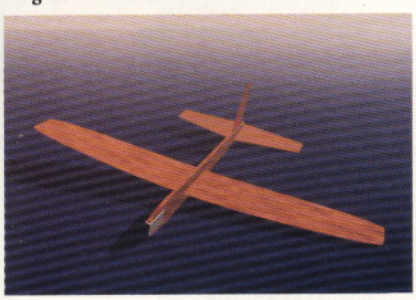


Image rendered and modeled by IRIS Inventor.

Writing interactive 3D graphics applications traditionally has been a tedious, time-consuming, and difficult task requiring a high level of programming expertise. Because of this, developers typically have taken one of two approaches. They have either invented their own software abstraction above the low level graphics commands, or produced portable, lowest-common-denominator applications with little or no direct 3D interaction. The first choice yields short term solutions that are rarely given sufficient design and implementation effort. The second approach results in disjoint user interfaces in which users can view 3D objects but interact only through remote user interface widgets or keyboard command languages.

IRIS Inventor, an object oriented 3D toolkit, offers a comprehensive solution to these traditional programming problems — without the compromises. It presents a new programming model based on a 3D scene database. Created using a 3D scene library, this database describes entire scenes comprised of 3D data and objects. IRIS Inventor's database makes it easy to develop 3D applications because the database forms an intuitive 3D model based on the physical world and not on a display list of drawing commands. This allows the programmer to focus on "what it is" and not "how to draw it," hence the application solution and not the graphics hardware implementation. In addition to rendering, the 3D database provides a number of other operations such as picking, printing, event handling and I/O. If a specific object or operation needed for a program does not exist, IRIS Inventor's object oriented architecture allows for its easy creation.

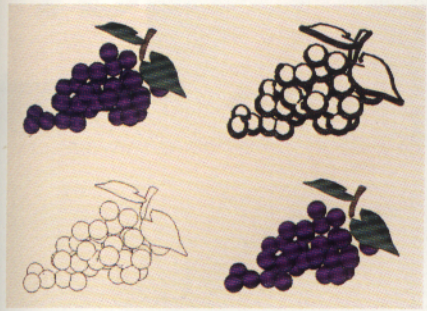
IRIS Inventor also provides an innovative event model that paves the way for 3D interaction. Just as a 2D window system sends events to windows and widgets, IRIS Inventor distributes events to 3D objects in the scene database. A new class of 3D objects, called manipulators, "listen" and react to user events by editing other objects in the scene database. A rich set of manipulators is provided with IRIS Inventor, as well as the means for creating new customized manipulator objects.

Utility libraries for both the X Window System and GL windows programming environments are included with IRIS Inventor. Both libraries contain initialization, main loop, and simple window objects. Additionally, the X Window System utility library contains a set of components, pre-built Xt widgets, that provide commonly needed 3D user interfaces (i.e., the examiner viewer and the material editor).

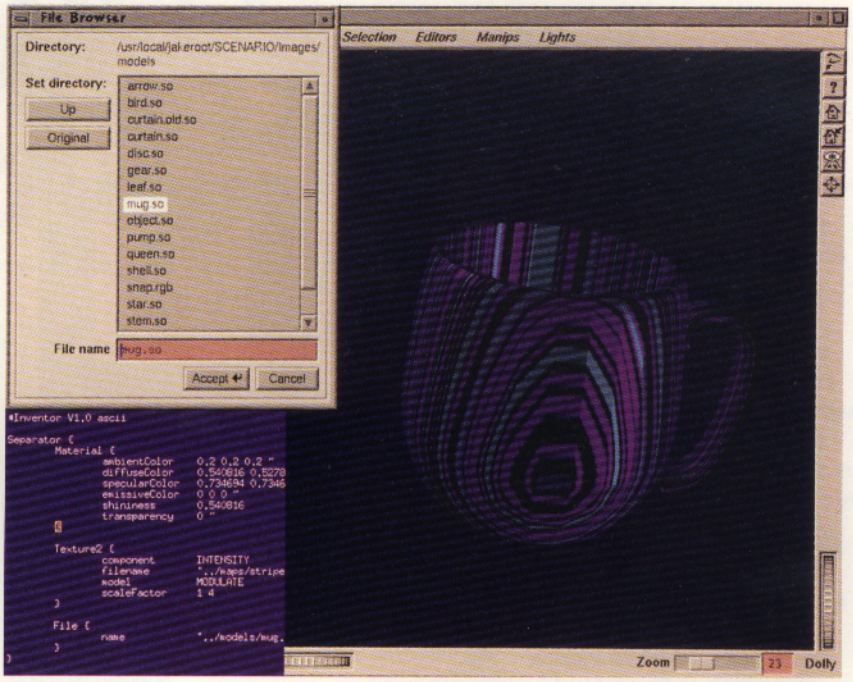
IRIS Inventor is written in C++ and is finely tuned to run fast and efficiently across the entire Silicon Graphics product line. It has C and C++ bindings, and can be mixed with raw GL code if needed. IRIS Inventor defines an ASCII and binary 3D Interchange File Format for data exchange and 3D cut and paste. A programming guide, manual pages, help cards, sample programs and data are included with the software.

Rikk Carey works with Silicon Graphics Advanced Systems Division.

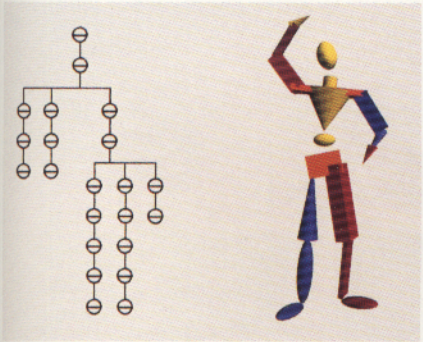
A beveled 3D text object defined by a font and a profile curve, not a set of polygons.



The upper left image is GL rendered, the lower left and upper right images are hidden line output to PostScript, and the lower right image is full color output to PostScript.



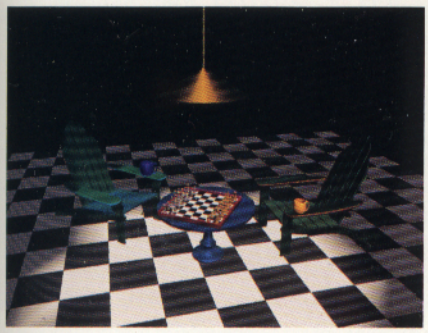
The File Gizmo is courtesy of IRIS Showcase and the ASCII file from which the cup was read. IRIS Inventor's 3D Interchange File Format is intended to be used as a common format for exchanging 3D data across applications. IRIS Inventor supports both ASCII and binary (XDR) formats.



The rendered robot is accompanied by a diagram of the 3D scene hierarchy used to represent it. Each circle in the diagram represents a group object in the hierarchy; all other objects (geometry and properties) are not shown.

This teapot, also rendered by IRIS Inventor, has texture mapped NURBS surfaces. Note that IRIS Inventor can apply textures to geometry objects. If texture coordinates are specified, then a default projection is used. Similarly, IRIS Inventor generates default normals for programs that do not specify them.

This complex 3D scene was composed using 3D manipulator objects. Manipulators are 3D objects that "listen" for user events and respond by editing other objects and are a 3D analog to traditional 2D widgets.

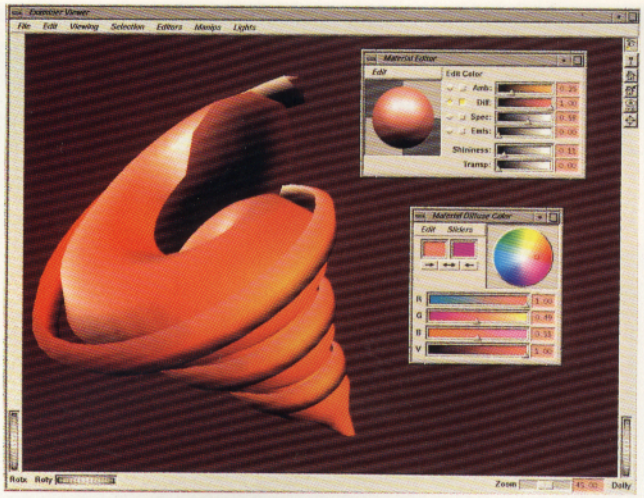


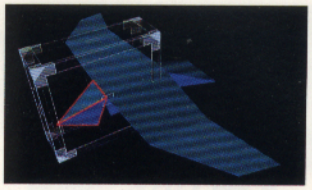
This virtual room was specified as a 3D hierarchical scene of objects, lights, properties and camera.

This image depicts a collection of objects with a manipulator attached.

This image of a 3D hierarchical robot has a trackball manipulator attached to its shoulder joint. The arm can be rotated easily by clicking and dragging the mouse on the trackball.

This 3D object is in the process of being edited via the Material Editor and Color Editor widgets. IRIS Inventor includes several editor widgets in a library of commonly used 3D editors.

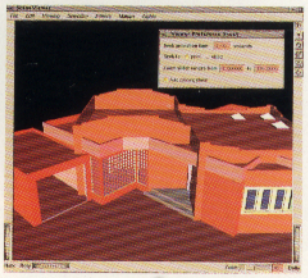




The model of a bird also has a handle box manipulator attached to it which provides an easy to use interface for positioning and scaling objects in 3D space.

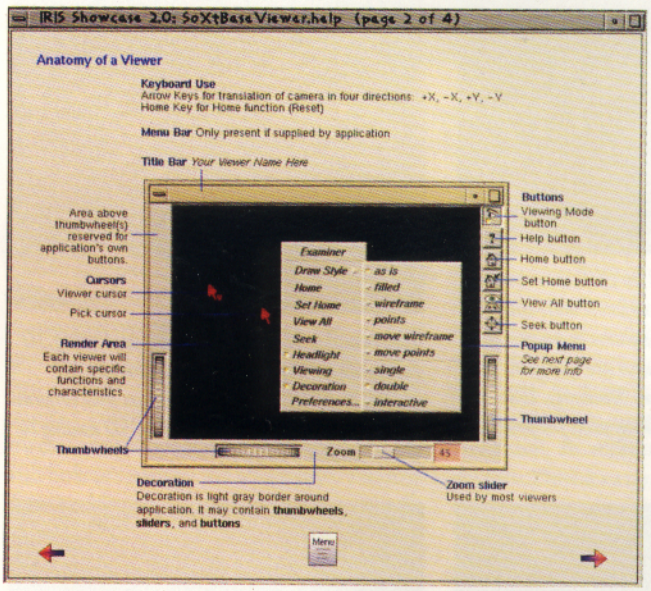
This image resulted from the Hello Cone program.

The Examiner Viewer widget renders a complex architectural model (output from a radiosity simulation). IRIS Inventor includes several viewer widgets, each with a unique user interface paradigm designed for a specific viewing task.

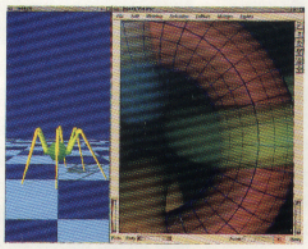


```
console
main( int, char ** )
{
    SoGL::init();

    // Make a scene containing a red cone
    SoSeparator *root = new SoSeparator;
    root->ref();
    SoPerspectiveCamera *camera = new SoPerspectiveCamera;
    root->addChild( camera );
    root->addChild( new SoDirectionalLight );
    SoMaterial *material = new SoMaterial;
    material->diffuseColor.setValue( 1.0, 0.0, 0.0 );
    root->addChild( material );
    root->addChild( new SoCone );
    camera->viewAll( root );

    SoGLRenderArea *ra = new SoGLRenderArea( "2.1.Hello Cone" );
    ra->setSceneGraph( root );
    ra->show();

    SoGL::mainLoop();
}
```

C++ source code for Hello Cone program.



Depicted is the top level help card for 3D viewer widgets. IRIS Inventor viewers are Xt widgets that integrate 3D programming smoothly into the X development environment.



GL and IRIS Inventor are able to be rendered within the same GL window. IRIS Inventor is window system independent. Windows can be created using either GLX mixed-mode windows or traditional GL windows (winopen). IRIS Inventor provides several convenience functions to make this easy. Programmers who choose to use the GL window and event model have several choices within IRIS Inventor for managing input.

A chess board scene with the Directional Light Editor widget attached to a light source. Notice the 3D arrow and sphere object in the light editor window, which was built out of IRIS Inventor manipulators. This is an example of how 3D user interface objects can greatly improve program usability.

For more information regarding IRIS Inventor, call your local Silicon Graphics sales office or contact the IRIS Inventor Product Manager, Scott Bovenizer, by calling 415/335-7090, or e-mailing to bove@sgi.com.

SILICON GRAPHICS, INC. EDUCATION CENTER COURSE CALENDAR

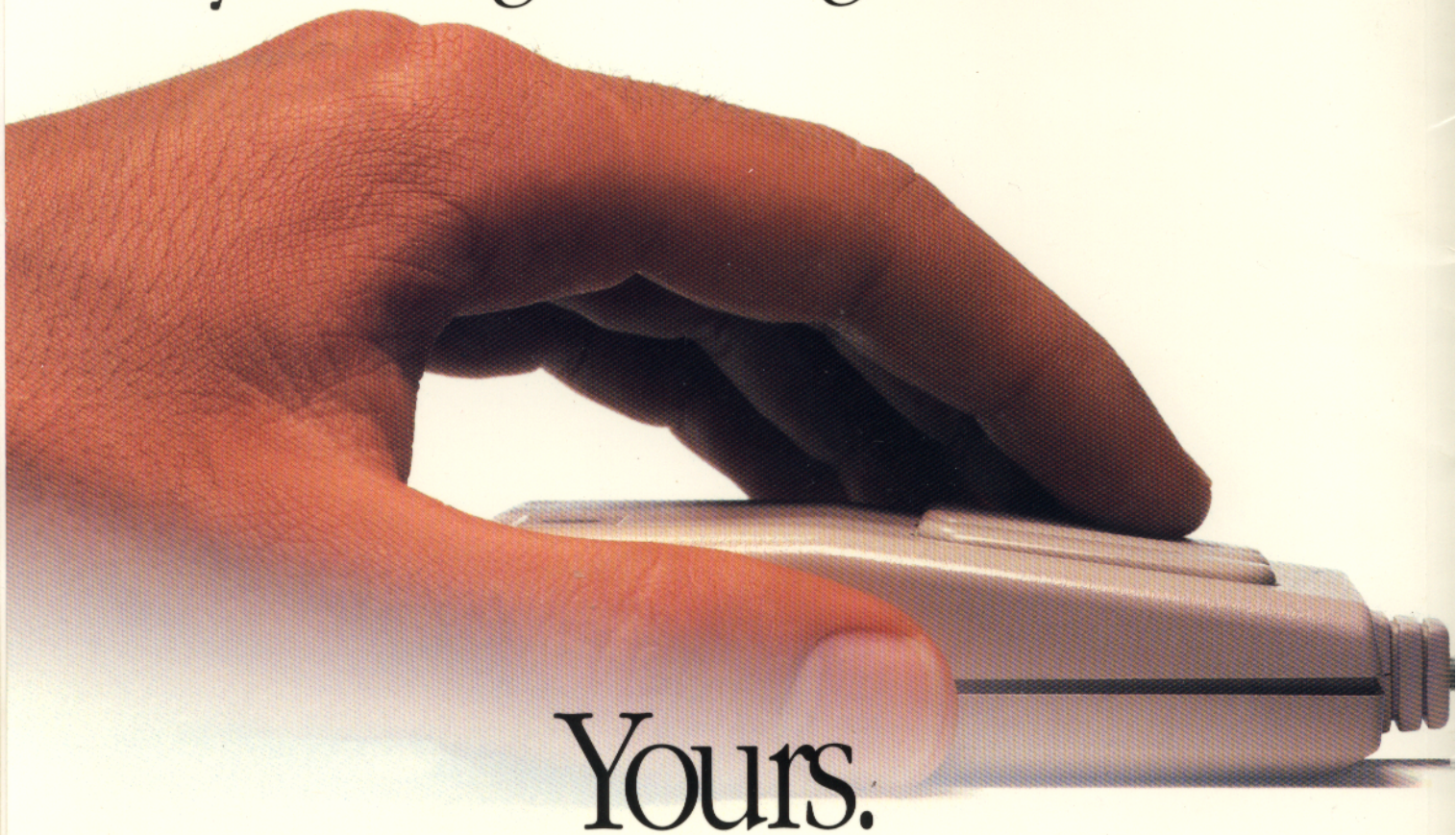
Through December 1992

4D SERIES COURSES	LOCATION* WEC	EEC	SEC
Graphics Library Programming 1 4.5 days	August 3, 1992 October 5, 1992 November 30, 1992	July 6, 1992 October 19, 1992	September 14, 1992 November 9, 1992
Graphics Library Programming 2 & POWERVision 4.5 days	August 10, 1992 October 12, 1992 December 7, 1992	July 13, 1992 October 26, 1992	September 21, 1992
Parallel Programming 4.5 days	November 9, 1992	September 28, 1992	N/A
Real-Time Programming 4.5 days	September 14, 1992 November 30, 1992	October 12, 1992	N/A
Motif Programming 4.0 days	July 13, 1992 September 21, 1992 November 16, 1992	August 31, 1992 December 14, 1992	N/A
Mastering IRIX 4.5 days	July 6, 1992 August 17, 1992 September 14, 1992 October 19, 1992 November 9, 1992 November 30, 1992	July 20, 1992 November 2, 1992	August 3, 1992 October 12, 1992 November 30, 1992
System Administration 4.5 days	July 13, 1992 August 24, 1992 September 21, 1992 October 26, 1992 November 16, 1992 December 7, 1992	July 27, 1992 September 14, 1992 Noember 30, 1992	August 10, 1992 October 19, 1992 December 7, 1992
Network Administration 4.5 days	July 20, 1992 December 14, 1992	October 5, 1992	August 17, 1992
System Maintenance Power Series 10.0 days	July 13, 1992 September 21, 1992 October 26, 1992 December 7, 1992	August 3, 1992 November 9, 1992	N/A
Advanced System Administration 4.5 days	August 31, 1992 November 2, 1992	September 21, 1992 December 7, 1992	N/A
IRIS Inventor 4.5 days	August 10, 1992 October 5, 1992	N/A	November 16, 1992
IRIS Explorer 4.5 days	July 20, 1992 August 24, 1992 October 26, 1992 December 14, 1992	N/A	N/A
End User Fundamentals 2.0 days	July 27, 1992	N/A	N/A

KEY: WEC—Western Education Center, Mountain View, CA.
EEC—Eastern Education Center, SGI Federal, Bethesda, MD.
SEC—Southern Education Center, Dallas, TX.

*The SGI Education Center reserves the right to cancel classes due to insufficient enrollment.
To register or obtain more information, call 800-800-4SGI.

With Wavefront's Video Composer, your images are in good hands.

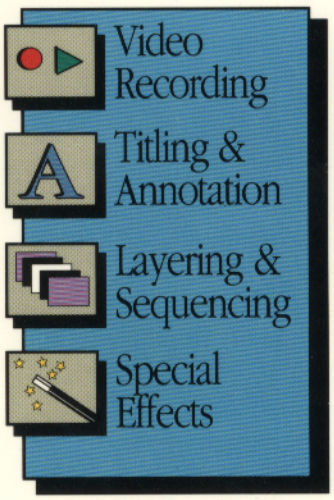


Yours.

Now, without leaving your workstation, you can manipulate video and create the kind of eye-grabbing special effects you've had to rely on outside sources for! Discover a whole new approach to creating compelling videos using Wavefront's new Video Composer™ — the only choice for desktop video production on your Silicon Graphics® workstation.

With its simple menu-driven interface you can bring images together from multiple sources. Combine computer images and live video. And blend traditional 2D effects into the 3D domain. All at a fraction of typical post-production costs, and with unlimited creative freedom.

From text for titles and annotation to broadcast quality effects, Wavefront's Video Composer lets you orchestrate it all, easily and economically.



Animate effects using time-based curves • Control VTRs on an existing network • Browse image files on multiple filesystems • Automatically regenerate images • Enhance rendered images • Color-correct images with 16-bit fidelity • Combine images in unlimited combinations of layers • Create mattes automatically from framegrabbed images • Build image sequences and the transitions between them • Supports a wide variety of video peripherals including Silicon Graphics' VideoCreator™, VideoFramer™, VideoLab™ and GenLock™.

Take control — of your images and audience. Call **1-800-545-WAVE** for more information and a free video! Wavefront Technologies, 530 E. Montecito St., Santa Barbara, CA 93103.

Desktop Video Production for your Silicon Graphics Workstation.

WAVEFRONT TECHNOLOGIES