Mars Exploration Rover (MER) Project

Viz 2.2.1
User Guide (UG)

Version 1.0

Authors:

Judd Bowman, Viz Engineer
Laurence Edwards, Viz Project Lead

Custodian:

Laurence Edwards, Viz Project Lead

March 29, 2002

NASA Ames Research Center

1.0 INTRODUCTION

1.1 Identification

The Viz User Guide describes the purpose, user interaction, and operating environment of the Viz software application as a Science Analysis Software Element for the MER Project. It is intended to help new users of Viz become acquainted with the application.

This guide responds to the requirements of the MER Software Development and Management Plan (SDMP), and insures compliance with NASA and JPL requirements for software development and management.

1.2 Overview

Viz is a cross-platform software application that handles 3D visualization and user interaction with a simulated environment. It is the latest application in a series of software tools produced by the Intelligent Robotics Group at Ames Research Center. As both a desktop application and networked graphics server, Viz delivers a high degree of flexibility to support the needs of NASA's evolving robotic planetary exploration program.

Viz was developed primarily for the Mars Polar Lander (MPL) mission, and was used in pre-landing MPL operational readiness tests and several FIDO and K9 rover tests.

For use in the MER Project, Viz has been adapted to operate with the data products produced by the Athena Science Payload and the MER platform. New features have also been added to Viz to increase its usefulness for science analysis and operations. A description of the role of Viz in the MER Project is contained in the Athena Science Implementation Plan.

1.3 Related Documents

Title	Designation
Science Development and Management Plan	420-1-202, JPL D-20719
Athena Science Implementation Plan	420-1-201, JPL D-20458

2.0 EXECUTION PROCEDURES

2.1 System Overview

Viz is a desktop application that allows a user to manipulate and study a simulated 3D environment. The simulated environment is loaded in to the application from Virtual Reality Markup Language models or XYZ maps generated from stereo image pairs. For the MER Project, Viz will be able to data derived from Pancam, Navcam, and Hazcam stereo image pairs. Both terrain and robot (rover) models are allowed in the simulated environment and both can be manipulated in useful ways.

To simplify the task of the user, Viz divides functionality in to two categories: 1) loading and maintaining the models in the simulated environment, and 2) interacting with and studying the environment. In both cases, all actions are done through a graphical user interface once the application has been launched.

2.2 Launching and Terminating Viz

After Viz has been properly installed on a workstation, it can be launched at any command prompt by typing:

>> viz

Alternatively, a Viz icon ( ) may appear in the application menu. Selecting the icon will also launch the application. The user is not required to enter any command line arguments when launching Viz.

A successful launch of Viz will create a new window on the desktop titled "Main – Viz". While an operating session in Viz is active, this window will always be visible on the desktop or in the taskbar. To terminate a session, close the *Main* window.

2.2 Loading and Maintaining the Simulated Environment

The functionality for loading and maintaining the simulated environment is located in the *Main* window. The window contains an object tree displaying the relationship between all available objects in the environment. This tree is synonymous with the simulated environment.

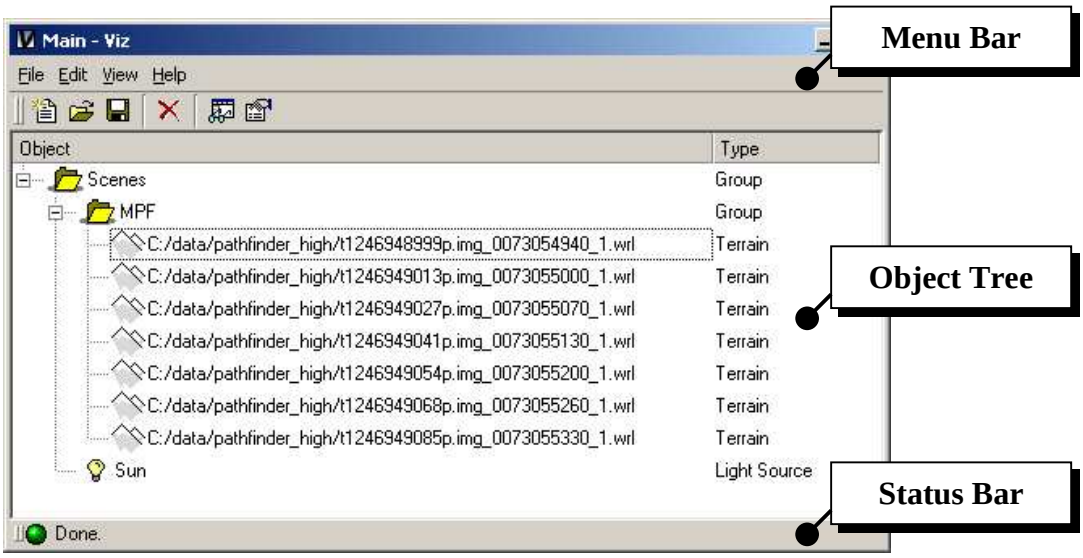

Figure 1. *Main* window (containing objects for the Mars Pathfinder environment)

Models or XYZ maps that have been loaded in to Viz are visible in the object tree. Each object in the tree is assigned a type at load-time so that Viz can handle it appropriately. The object types include terrains and robots (rovers) as mentioned previously, but also cameras, lights, groups, and abstract objects. See Appendix A for more information about object types and how they are used by Viz.

A menu bar containing buttons for the operations available to load and manipulate objects in the tree is also contained in the *Main* window. The operations to manipulate the object tree are, for the most part, familiar file operations that one would expect to encounter in other applications. A list and description of these operations is contained in Appendix B.

The final element of the *Main* Window is a status bar displaying information related to the current activity of the application.

2.3 Interacting with the Simulated Environment

The user interacts with the environment in a *Viewer* window. Multiple *Viewer*s can be open at one time, each displaying all or part of the object tree. Much of the primary functionality of Viz is accessible from within this window.

A *Viewer* consists primarily of a rendering area where the user can view and interact with the simulated environment. Several tool panels contain operations and displays for directing the interaction. The primary operations available for interacting with and studying the environment are divided into view tools, point tools, and grid tools. Appendix C discusses each of these tools.

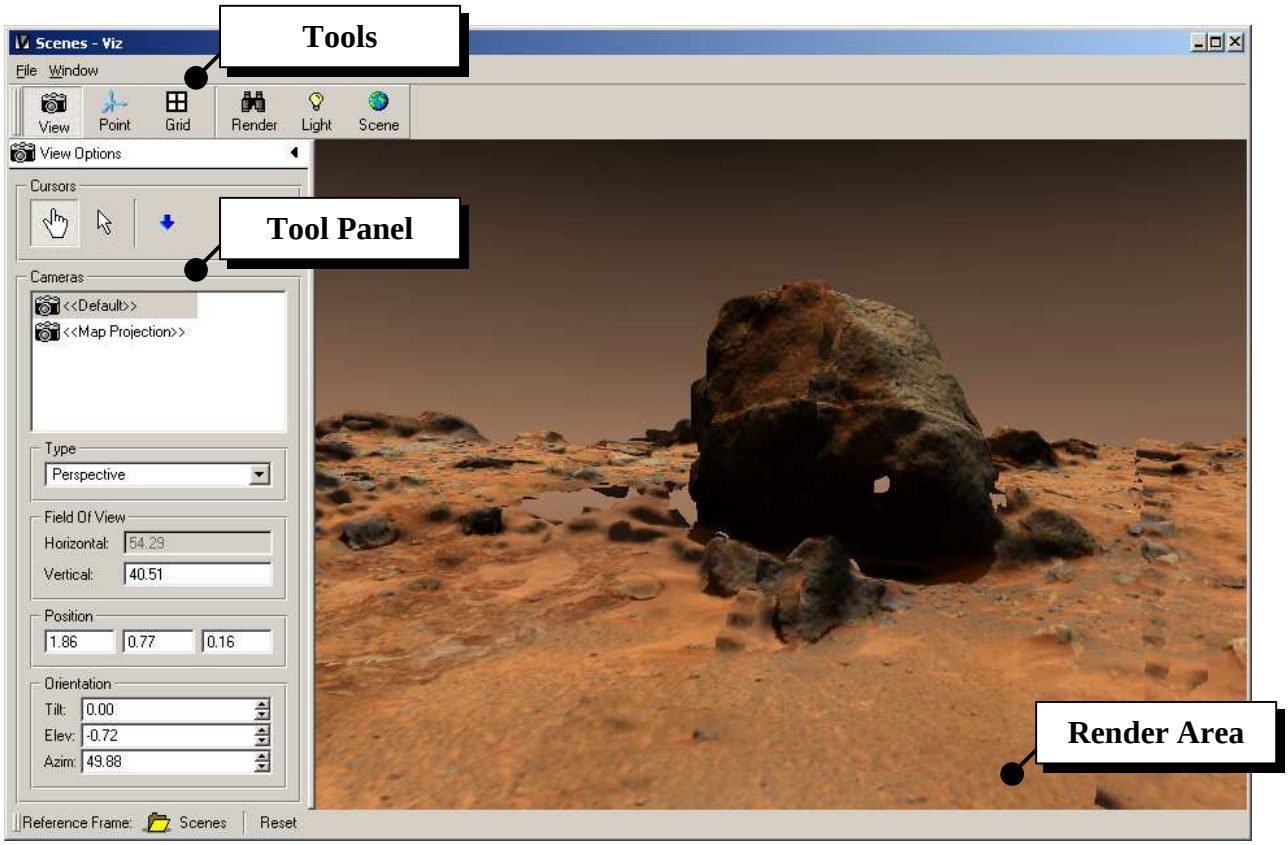

Figure 2. *Viewer* window (displaying Mars Pathfinder environment)

The user can also configure various parameters of each *Viewer* using its *Options* dialog. Stereo rendering, wire frame drawing, and lighting conditions can be specified with this dialog; which is described in Appendix D.

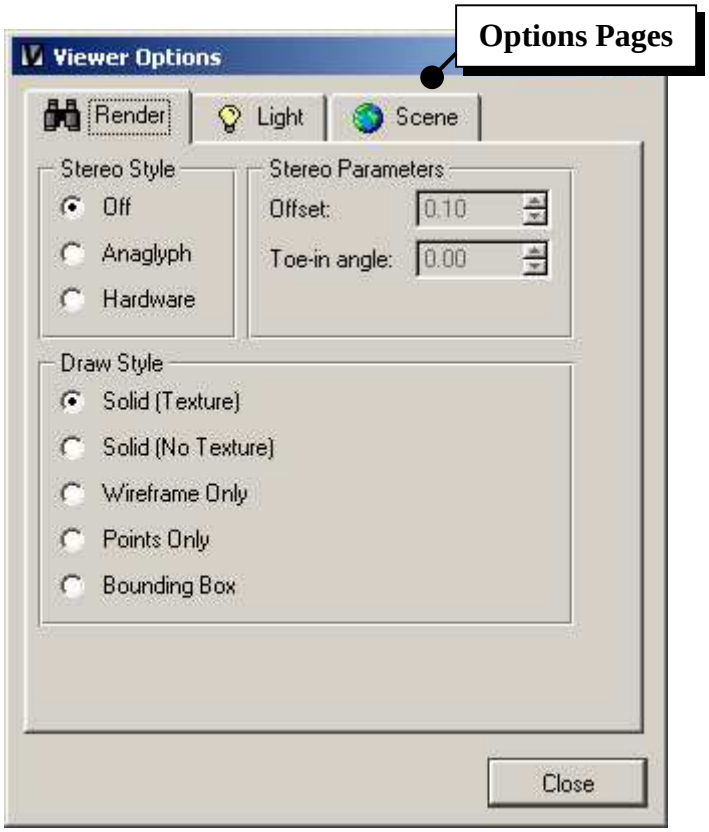

Figure 3. *Viewer Options* dialog

APPENDIX A: Object Types

Viz distinguishes between 7 primary object types. All object types share certain basic properties. These common attributes include a title, a parent, a transformation relative to the parent, and material hints. Below is a table describing the object types and their unique properties individually.

Type	Description
 Abstract Object	In some cases, it is useful to add objects to the object tree that do not represent specific items. For example, graphs that display data, or shapes that change size or color in response to an external input, would fall in to this category. This is the least used type of object in Viz and has no additional properties associated with it.
 Camera	Although each viewer has its own camera(s), the object tree can hold cameras as well. Two classes of cameras are availabe: 1) perspective, and 2) orthographic. Each camera class has a field of view. For a perspective camera, the field of view is given in degrees, whereas for an orthographic camera, the field of view indicates the dimensions of the camera.
 Group	This type of object is similar to a folder or directory. Its purpose is to contain other objects in logical groups.
 Light	Scenes must be illuminated by light to make them visible. A viewer has its own light, called ambient light, but the object tree can hold lights as well. Three classes of lights are available: 1) directional, 2) point-source, and 3) spotlights. All three classes of lights share the basic property of color. A transform affects each class of lights in a different way. For directional lights, only the rotation component of the transform is used, for point-source lights, only the translation component is used, and for spotlights, the entire transform is used.
 Robotic Component	Objects that are robots or parts of robots are assigned this type.
 Terrain	Objects that represent the surface of the environment are terrains. This is typically the primary object type, since the ground is the part of the environment of interest.
 Target	Important locations can be marked as targets to help the user keep track of information. In some cases, targets can be shared with WITS to facilitate working with both applications.

APPENDIX B: Object Loading and Manipulation Actions

Action	Description
About	Displays a small dialog containing the version number of the application and other ancillary information.
About Qt	Displays a small dialog containing the version number and brief description of the shared library of Qt used by the application.
Apply Default Colors	Instructs all terrain objects in the environment to use the default coloring or texturing.
Apply Elevation-Map Colors	Instructs all terrain objects in the environment to use a color map based on elevation instead of the default coloring or texturing.
Apply Slope-Map Colors	Instructs all terrain objects in the environment to use a color map based on surface tangents instead of the default coloring or texturing.
Delete Object	Removes the currently highlighted object (and all of its children) from the object tree.
Location Window	Displays the location dialog to specify the planet, latitude, longitude, and UTC time needed to calculate the sun and shadow positions.
New...	Displays a dialog similar to the properties dialog that allows the user to create an empty group, a new camera, or a new light.
Object Properties	Displays a properties dialog containing the type, parent, and other information associated with the object. In some circumstances, the user can edit an object's properties as well.
Open...	Displays a standard open file dialog to allow the user to select a *.viz file to open. Other file types, such as *.wrl and *.iv can be imported here, too. After selecting a file, the open file dialog may be followed by an additional dialog similar to the properties dialog. The user will be prompted to enter information describing the object being opened or imported.
Save As...	Writes the object tree structure of the currently highlighted object and its children to disk. A standard save file dialog is displayed to allow the user to enter a file name. Note: This file may not be sufficient to transfer data to another computer or user. Some parts of data objects, such as the images used to texture terrains, may not be included directly in this file, but rather linked indirectly.
Status Window	Displays a simple dialog showing the last 100 status message.
View Object	Creates a new Viewer Window to render the currently highlighted object and its children. Read more about Viewer windows in Section 2.2

APPENDIX C: Tools for Interacting with the Simulated Environment

Tool	Description
Grid	**Overview** Several types of grids are available to superimpose on the scene. These include a cartesian grid and a radial grid. The grid can be translated and rotated. In addition, its color and transparency can be specified. **Cursor** When a *Viewer* is in grid tool mode, the cursor can be used to rotate and position the grid: • *Left button:* Click and drag to rotate the grid • *Right button:* Click to re-center the grid
Point	**Overview** The user can perform measurements on the scene by using the point tool. This panel provides three measurements: individual point, distance line, and poly line. • *Individual point:* Displays the XYZ position, surface normal, and sun incidence angle for the picked point. The distance and azimuth from the origin of the current reference frame is also calculated. • *Distance line:* Displays the distance and azimuth between two points, as well as the individual point data. • *Poly line:* Allows the user to outline a region for calculating surface area and volume. **Cursor** When a *Viewer* is in point tool mode, the cursor can be used to pick points in the environment: • *Left button: Pick point*

 View	**Overview** Each viewer provides two cameras. 1) The <<Default>> camera is a fully configurable camera for interacting with the scene. The user can modify its position either in the tool panel, or by using the mouse in the rendering area. 2) The <<Map Projection>> camera is a partially configurable camera that simulates an orthographic map of the scene (viewed down the gravity-axis). The dimensions of the map are equivalent to the camera's field of view. Additional cameras can be used to render the scene if they are in the object tree. These cameras are not configurable from the tool panel and cannot be positioned using the mouse in the rendering area. **Cursors** When a *Viewer* is in point tool mode, two cursors are available to reposition the selected camera in the environment: the standard navigation cursor and the focal point cursor. Standard Navigation Cursor: • *Left button:* Click and drag to swing the camera • *Middle button:* Click and drag up/down to zoom in/out • *Middle wheel:* Rotate to zoom in/out • *Right button:* Click and drag to slide the camera Focal Point Cursor: • *Left button:* Set focal point and zoom in • *Middle button:* Click and drag up/down to zoom in/out • *Middle wheel:* Rotate to zoom in/out • *Right button:* Set focal point and pan camera

APPENDIX D: Viewer Options

The *Options* dialog of a *Viewer* window provides the ability to customize the rendering and appearance of the simulated environment for each window. Settings are divided in to three pages as listed below:

Page	Description
Light Options	Each viewer allows the selection of which lights in the object tree are used to illuminate the environment it renders from this page.
Render Options	*Viewers* typically render a scene from a single camera. But they can also render a scene using 2 cameras to simulate stereo vision. Viz supports 2 types of stereo rendering: 1) hardware stereo rendering such as for CrystalEyes® by StereoGraphics, and 2) anaglyph stereo rendering for red/blue glasses.
Scene Options	The user can set a background color or image (which is mapped around a sphere) here. The ambient light is also configurable in this panel, in addition to simple fog options.