



# Tutorial: A Tool for Teaching graphics Programming

*Michael J. Papper*

*Michael A. Gigante*

Royal Melbourne Institute of Technology  
Advanced Computer Graphics Centre <sup>1</sup>

January 8, 1992

graphics state are both inspectable and modifiable, with these changes propagated throughout the entire system.

## Keywords

computer graphics; microworlds; simulation; education

This interaction aids in: determining what are reasonable parameters for GL functions; observing how GL functions interrelate; observing how the functions modify the graphics state; and observing how the graphics state determines what is presented in the final window.

## Abstract

Tutorial is an interactive exploratory learning tool designed to aid programmers in learning graphics programming and Silicon Graphics Inc.'s Graphics Library (GL) in particular.

Tutorial is driven by four demo programs that present *simulated source code* to the student. The student uses Tutorial by executing each line of code and noticing its effects on the graphics state, representations of the internal hardware, and the final output. We intend that the exploratory nature of the system will result in deep learning of graphics programming concepts.

The user interacts with Tutorial by modifying parameters to GL function calls, choosing lines of source code to execute, and modifying the graphics state directly. Nearly all aspects of the

The poster describes Tutorial and presents four scenarios of Tutorial in use. Each scenario is a series of pictures which show the modification of function parameters and the effect of the modification on the final image. The scenarios we will show are: the use of the perspective and lookat to define the viewpoint and viewing volume; orthographic projection and how it relates to the viewport and the hardware zbuffer; how the 2-dimensional orthographic projection, viewport and charstr interact; and using the overlay and underlay framebuffers.

Each scenario demonstrates how Tutorial enables the user to come to a deeper understanding of the effects of various graphics library functions. For instance, the relationship between the orthographic projection area and the viewport (screen) area is explained. The scenarios also demonstrate the different ways the user can interact with Tutorial. These include modifying function parameters, altering the simulated source code, changing the *state* of the system directly (bypassing the simulated source code), and inspecting various parts of the graphics pipeline. As well the features of the interface and the control of the simulated source code will become apparent from the scenarios.

---

<sup>1</sup>RMIT Advanced Computer Graphics Centre, Collaborative information Technology Research Institute, 723 Swanston St., Carlton Victoria, Australia 3053, +61-3-282-2462.  
mpapp@godzilla.cgl.citri.edu.au  
mg@godzilla.cgl.citri.edu.au