Analysis of Rogers

Jed Margolin

{Company's} Mr. {Person's} Prior Art References

> Mathematical Elements for Computer Graphics, David F. Rogers (McGraw-Hill 1976 ISBN 0-07-053527-2). General text discussing computer graphics elements, including rotations, translations and perspective transformations. Professor Rogers was one of my instructors at the U.S. Naval Academy.

My copy of Mathematical Elements for Computer Graphics, (McGraw-Hill 1976 ISBN 0-07-053527-2) lists two authors: David F. Rogers and J. Alan Adams. I believe I bought my copy in 1979, but since the ISBN numbers are the same, it's the same book that Mr. {Person} lists. Mr. {Person} left off one of the authors. *{Ref 6a}*

Mr. {Person} is correct insofar as it is a general text discussing computer graphics elements. Again, if {Person} wants to argue that any invention that uses computer graphics is obvious (and therefore, invalid) it would be a tough sell in any Federal Court.

However, the book is more than computer graphics elements. It is a snapshot of computer graphics in the late 1970s.

Although I am tempted to quote extensively from it (I lived through that era), I will restrict myself to this section:

> 1-1 OVERVIEW OF COMPUTER GRAPHICS
>
> Computer graphics as defined above can be a very complex and diverse subject. It encompasses fields of study as diverse as electronic and mechanical design of the components used in computer graphics systems and the concepts of display lists and tree structures for preparing and presenting pictures to an observer using a computer graphics system. **A discussion of these aspects of interactive computer graphics is given in the book by Newman and Sproul** (Ref. 1-1) Here we will attempt to present only those aspects of the subject of interest from a user's point of view. From this point of view, computer graphics can be divided into the following areas:
>
> Representing pictures to be presented
> Preparing pictures for presentation
> Presenting previously prepared pictures
> Interacting with the picture
>
> Here the word "picture" is used in its broadest sense to mean any collection of lines, points, text, etc., to be displayed on a graphics device. A picture may be as simple as a single line or curve, or it may be a fully scaled and annotated graph or a complex representation of an aircraft, ship, or automobile.

The standard computer graphics text at the time was Principles of Interactive Computer Graphics by Newman & Sproul. It contained all the material contained in Rogers & Adams, and more. Much, much more. I still have my copy of the Second Edition (McGraw-Hill, ISBN 0-07-046338-7) that I bought in the early 1970s. *{Ref. 6b}*

Newman & Sproul was supplanted by Fundamentals of Interactive Computer Graphics by Foley & van Dam, which came out in 1982. *{Ref. 6c}*

None of these references taught synthetic vision. The computer graphics community was mostly interested in pretty pictures. Realtime 3D was too computationally expensive.

The Second Edition of Foley & Van Dam Computer Graphics Principles and Practice was by Foley, van Dam, Feiner, and Hughes in 1990 *{Ref. 6d}.* This edition has something interesting to say about the state of realtime 3D in the early 1990s. On Page 919:

> **18.11.6 Real-Time Flight Simulators**
>
> The systems that "put it all together" to generate the most realistic simulation of 3D scenes for interactive tasks are the multimillion-dollar flight simulators. Flight simulation is not unique in being able to benefit from truly real-time systems. It is, however, the one application for which customers have consistently been willing to spend millions of dollars for a single system, largely because of the cost and danger of training pilots solely in actual airplanes. Because the community of flight-simulator users is fairly small, and because the systems tend to use proprietary, hardware-specific software provided by the manufacturer, detailed information about flight-simulator architectures has not appeared in the literature.
>
> Early flight simulators include General Electric's NASA II (see Section 18.9) and Evans & Sutherland designs based on the scan-line systems developed at the University of Utah in the late 1960s. Some of these early flight-simulator systems could display 1000 or more polygons in real time, but all used simple shading methods and provided few image enhancements. Later systems have not substantially increased the number of primitives that can be displayed. For example, Evans & Sutherland's current high-end system, the ESIG-1000, displays only 2300 polygons at 60 Hz [EUAN89]. Rather, system developers have increased scene realism and reduced distracting artifacts by incorporating features such as antialiasing, haze and fog, point light sources, clouds, and filtered textures [SCHA83]. The effectiveness of these techniques can be seen in Color Plates 1.5(a) and I.5(b).
>
> Flight simulators from major manufacturers such as Evans & Sutherland, General Electric, McDonnell-Douglas, and Singer/Link all share several architectural themes: Since flight simulation involves predictable interactions with very large datasets, these systems tend to use more specialized processing than do other graphics systems. For example, custom processors are frequently built to manage the image database, to transform primitives, to rasterize the image, and to perform image-enhancement operations afterward. A typical simulator system is composed of a long pipeline of proprietary processors [SCHA83].
>
> Certain simplifications can sometimes be made in a flight simulator that are not possible in more general graphics systems. For example, since a typical simulator dataset involves a small number of moving objects and a wide, unchanging backdrop, the generality of the z-buffer visibility algorithm may not be needed and a simpler depth-sort algorithm may suffice.

> Flight simulators also must manage complex databases without hesitation. A typical database may represent a region 100 miles square. Detail needed for low-level flight cannot be displayed when the airplane is at 40,000 feet. This requires the system to maintain object descriptions with different levels of detail that can be swapped in and out in real time. The architecture also must handle overloading gracefully, since image complexity may increase drastically at the most crucial times, such as during takeoffs, during landings, and in emergency situations [SCHU80]. Frames must be generated at least 30 times per second - even in these situations.

Given the multimillion dollar cost of these realtime 3D systems it should be easy to understand why no one in the Aerospace or Computer Graphics industries thought of putting one in an aircraft and doing synthetic vision.

There was one other company doing realtime 3D graphics. The hardware was not as capable as the multimillion flight simulators just mentioned. It also didn't cost multimillion dollars. It cost about $7,500 to the arcade operator.

Color Plate 1.7 in Foley & van Dam is a picture of Atari Games' Hard Drivin' game. *{Ref. 6c}*

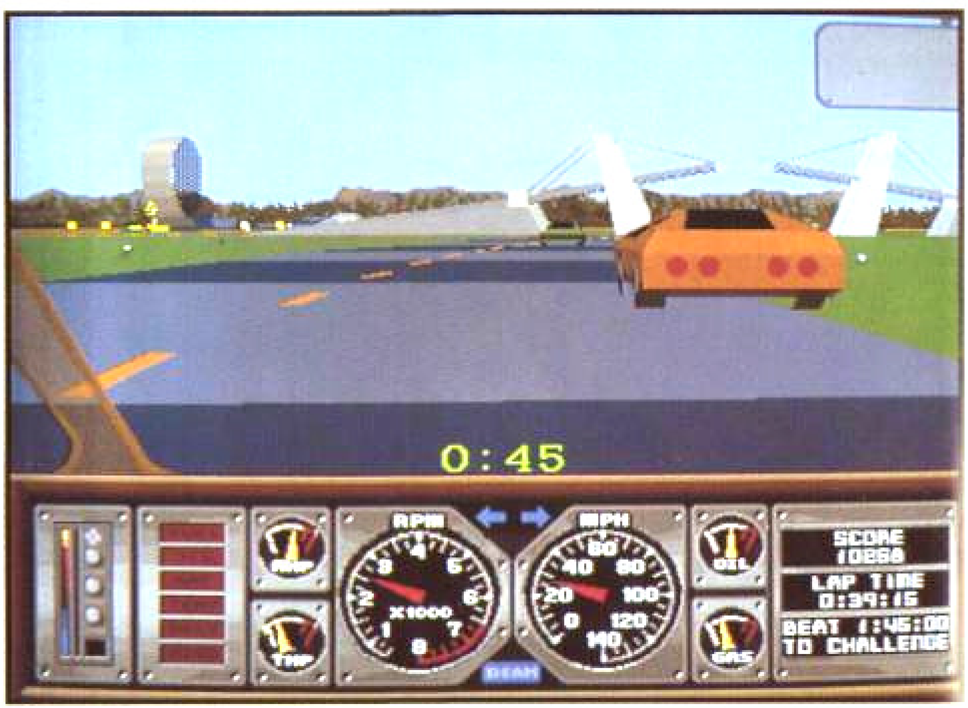

Plate I.7 *Hard Drivin'* arcade video game. (Courtesy of Atari Games Corporation, copyright © 1988 Atari Games Corporation.)

I was the Project Engineer for Hard Drivin'. I was also the Hardware Engineer, provided the core 3D algorithms, and was responsible for getting the game into production.

My professors at The University of Michigan might not have been famous authors (except for my Metallurgy Professor, Dr. Lawrence van Vlack, Elements of Materials Science and Engineering {Ref. 6e}) but my game was in Foley & van Dam.

References

Ref. 6a - **Mathematical Elements for Computer Graphics;** David F. Rogers and J. Alan Adams, McGraw-Hill 1976 ISBN 0-07-053527-2

Ref. 6b - **Principles of Interactive Computer Graphics**, Second Edition; Newman & Sproul. McGraw-Hill, ISBN 0-07-046338-7

Ref. 6c - **Fundamentals of Interactive Computer Graphics;** Foley & van Dam, Addison-Wesley, ISBN 0-201-14468-9, 1982

Ref. 6d - **Computer Graphics Principles and Practice,** Second Edition; Foley, van Dam, Feiner, and Hughes, 1990, Addison-Wesley ISBN 0-201-12110-7

Ref. 6e - **Dr. Lawrence Van Vlack.** http://www.mse.engin.umich.edu/undergraduate/vvul/about