\

Virtual Sphere Sample Code Release Note v1.1 Mar. 1993

Michael Chen
Human Interface Group / ATG
Apple Computer, Inc.
AppleLink: CHEN.M

Fonts Requirement

To read this document, you need Times and Helvetica fonts.

Introduction

The Virtual Sphere Sample Code package contains a C implementation of the Virtual Sphere interface for performing 3D rotation using a 2D input device. Also in the package is a small 3D graphics system to perform transformations and to display the objects used in this program. The sample code is compilable under MPW and Think C, and the application should run on all System 7 machines and most System 6 machines (see below).

The Virtual Sphere provides an intuitive interface for

\

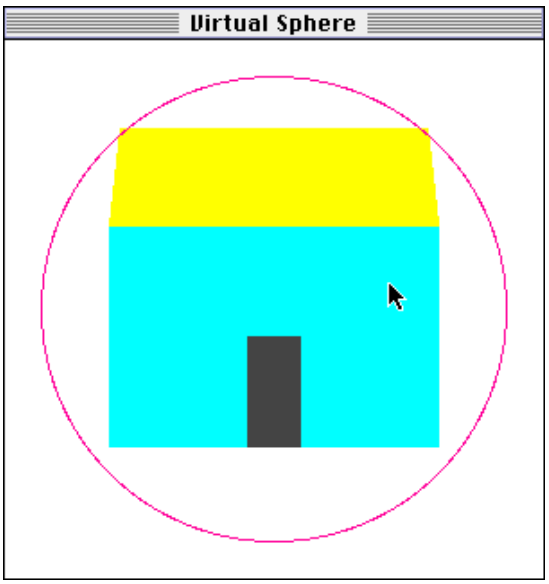
performing 3D rotation using a 2D input device such as the mouse. An implementation of the Virtual Sphere is described in *A Study in Interactive 3-D Rotation Using 2-D Control Devices* by Michael Chen, S. Joy Mountford and Abigail Sellen published in Proceedings of ACM Siggraph '88 (Volume 22, Number 4, August 1988). The paper also compared the Virtual Sphere with other more traditional rotation techniques in an experiment. The Virtual Sphere implementation enclosed is an improvement over the algorithm described in the paper. You may also want to look at two other demo applications called “Rotation Controllers” and “Rotation Experiment” that I have released earlier.

Running the Sample Application

This package contains an already compiled application of the sample code, called “VirtualSphereSample”. It should run on all machines running System 7, and machines with System 6.0.7 & Color QuickDraw. This application uses SANE for floating point math. The application named “VirtualSphereSample using FPU” requires a Mac with a math coprocessor and at least a 68020 CPU.

After the application is launched, you will see a 3D house enclosed by the Virtual Sphere controller (the circular cue):

\



The Virtual Sphere controller simulates the mechanics of a physical 3D trackball that can freely rotate about any arbitrary axis in 3-space. The user can imagine the circular cue to be a glass sphere that is encasing the object to be rotated. Rotation is a matter of rolling the sphere and therefore the object with the mouse cursor. Up-and-down and left-and-right movement at the centre of the circle is equivalent to "rolling" the imaginary sphere at its apex and produces rotation about an axis lying on the plane of the screen. Movement along (or completely outside) the edge of the circle is equivalent to rolling the sphere at the edge and produces rotation about the axis perpendicular to the screen.

The Object menu will let you replace the house object with a cube or an icosahedron (20-sided polyhedron). The Options menu offers different rendering options:

- line drawing, flat-shading or flat-shading with outline drawing modes
- drawing in black and white (using 64 shades of dithering patterns) or in color mode
- turning backfaced polygon removal on or off
- turning double buffering on or off

Normally you would want to keep backfaced polygon removal turned on when drawing in color. In line drawing mode, having backfaced polygon removal off produces wireframe rendering while having backfaced polygon removal on produces a fake version of wireframe rendering with hidden line removal.

The application will automatically go into black and white drawing mode at launch time if the machine does not have color or if the bit-depth is 1. Otherwise, color drawing mode is selected.

Compiling the Sample Program

The Sample code can be compiled under the MPW C and THINK C environments. You can compile the application using SANE or hardware floating point. If you use hardware floating point, a 68020 or better processor is assumed.

MPW C 3.2

Select “Full Build…” from the Build menu and type in “VirtualSphereSample”, or type “BuildProgram -e VirtualSphereSample” in the Worksheet.

\

The MakeFile is configured to use SANE for floating point math. To take advantage of hardware floating point, redefine COptions and RealMathLibs as instructed in the MakeFile and do a full build again.

Think C 5.0.2 using SANE

Launch THINK C by opening "VirtualSphereSample.π". Select "Run" from the Project menu, and answer "Yes" to update the project.

* If you ever need to recompile the resource file, VirtualSphereSample.π.rsrc, follow the instructions here: Launch SARez in the "…:THINK C 5.0 Utilities:Rez Utilities:" folder. Click on the "Description Files…" button. Find and highlight the file Sample.r, click the "Add" and then the "Done" button. From the pop-up menu, select "Write output to a new file…", type in "VirtualSphereSample.π.rsrc" and click "OK". Click the SARez button. At this point, the resource file "VirtualSphereSample.π.rsrc" should have been created in the directory with the sample code.

Think C 5.0.2 using Hardware Floating Point

Launch THINK C by opening "VirtualSphereSampleFPU.π". Select "Run" from the Project menu, and answer "Yes" to update the project.

* If you ever need to recompile the resource file, "VirtualSphereSampleFPU.π.rsrc", follow the instructions here: Launch SARez in the "…:THINK C 5.0 Utilities:Rez Utilities:" folder. Click on the "Description Files…" button. Find and highlight the file Sample.r, click the "Add" and then the "Done" button. From the pop-up menu, select "Write output to a new file…", type in "VirtualSphereSampleFPU.π.rsrc" and click "OK". Click the SARez button. At this point, the resource file "VirtualSphereSampleFPU.π.rsrc" should have been created in the directory with the sample code.
* If you ever need to recompile the library, "ANSI Native,020,881,4 byte int", follow the instructions here: Move the file "ANSI Native,020,881,4 byte int" into the "…:THINK C 5.0:C Libraries:" folder. Open this project. Select "Bring Up To Date" from the Project menu. This step creates an ANSI library that uses the FPU and native FPU format for floating point calculations. In addition, the library uses the 68020 instruction set and 4-byte integers.

Sample Code Documentation

Please refer to the article "3-D Rotation using a 2-D Device" in the June 93 issue of the develop magazine.

\

Comments

Comments and bug reports are welcomed. Please contact me using the AppleLink address listed at the top of this release note.

Michael.

\

Revision History

Version 1.1 (Mar 93) for June 93 issue of develop

- Extracted RGB to grayscale code in PolyColor into routine RGBToGrayscale (file Graphics3D.c)
- Renamed variables in VirtualSphere.c and SampleAdditional.c to be consistent with the develop article
- In DoRotation, sphereRadius should be declared as an Integer, not a Real (file SampleAdditional.c).
- SetRotationMatrix improved to not use any trigonometry (file Graphics.c).
- Added DotProduct3D (file Graphics.c).
- Source code for atan and asin are no longer needed but are left in for reference (File MyMath.c).

Version 1.0 (Dec 92) for Developer CD

Original release for Developer CD Series, March 93