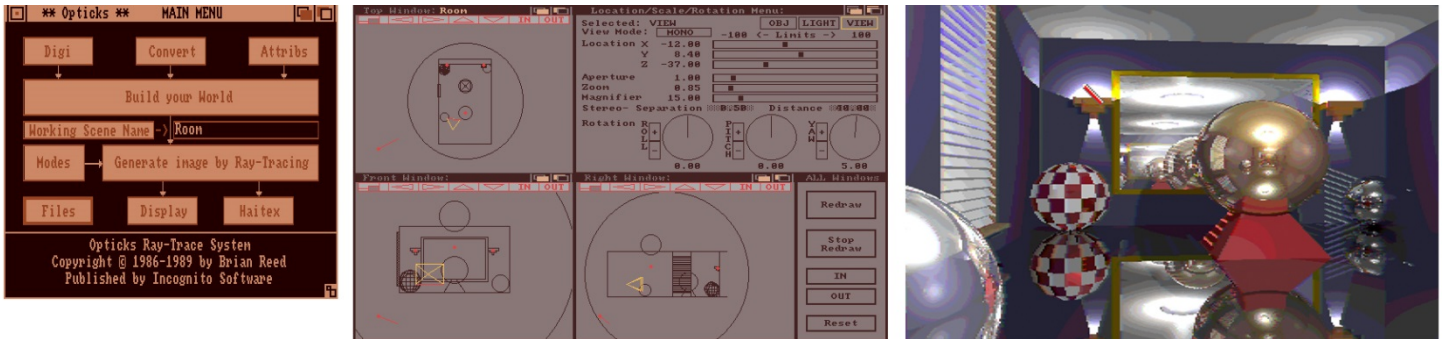


# OPTICKS 1.2

Version 1.2.1 – December 2, 2020

Opticks was released in 1989 as a two disk set. The first disk has all the primary programs for building 3D objects, editing attributes, building a 3D scene, rendering, and viewing. The second disk has a demo slideshow, maps, utility programs with some source code, and alternate compilations of Build and Render with the 68020 & 68881 options.



The programs were written by Brian Reed and published by Incognito Software. Many thanks to the original Incognito crew: Doug Brooks, Mike Rudas, Abe Hamade and the testers! In 2020, with assistance and motivation I have it running again.

Some tweaks have been made to the boot files, particularly `s/Startup-Sequence`. It has been simplified, and optimized for >512K memory. The original was squeezed to run in 512K memory but emulators make available memory a non-issue. See the comments inside `Startup-Sequence` to minimize startup memory for use on a 512K machine.

I have seen two versions of Opticks online, and in some cases they have been tagged as v1.0 and v1.1. It may be that the early one was a pre-release for testers and reviewers. The definitive ways to tell the difference is that the final release has the Haitex app, and the Render program has a wireframe preview option in HAM mode which helps with understanding the camera view quickly. To minimize confusion I'm calling this v1.2 which can be identified by the Startup message, the deprecated files in the `/s` folder with ".1989" in the name, and three new cameras up to 4k resolution (named Simple...). Note that all versions say v1.0 in the Render status window.

**Version 1.2.1: The ONLY differences from v1.2.0 are these notes and other minor Opticks-Intro edits, and while the contents of the ADF files are unchanged their name has been updated to v1.2.1.**

\* The 2020 renders used the WinUAE emulator with Kickstart ROM "KS ROM v1.3 (A500,A1000,A2000) rev 34.5 (256k) [315093-02]" and Workbench 1.3 (34.20). Opticks may or may not work with other Kickstart or Workbench versions.

\* One user reported that the Scene file may be getting corrupted by the Build program. Make ".sce" file backups often so that you don't lose too much work, in case you experience the same.

\* Home page: <https://ReedOnline.com/Opticks>

# Getting started with Opticks

## Your first test of Opticks:

- 1) Boot off the OPTICKS1 disk.
- 2) When the “First!” application opens, click on ASSIGN.
- 3) In MAIN MENU, click on “Working Scene Name” and select the Room scene.
- 4) Click on “Generate image by Ray-tracing” and you should see a status window as objects load, then a wireframe preview, and then pixels start drawing (rendering) in HAM display mode. It will be a bit blocky because the Resolution Multiplier is greater than 1 for quicker preview.
- 5) After closing the Render window, click on Build to see the Room in the scene designer. Use the OBJ, LIGHT and VIEW buttons to select an item to work on. The selected item is displayed in yellow.

## The MAIN MENU is a flowchart:

Digi – A basic grid to draw an outline then either extrude it as a flat object, or lathe (spin) it to make a round version of the drawn cross section

Convert – Import 3-Demon or Sculpt3D objects (somewhat more useful than Digi?)

**Attribs – An important app to create items like Global Illumination, Lamp, Gel (to color a Lamp), Camera, and Surface Material to be used in a scene**

**Build – This is where you build your scene!**

**Render – Generate the scene’s image by ray-tracing!**

**Modes – The render settings including resolution and intensity scaling, display, and file save options**

Display – A basic app for displaying images

Haitex – For use with the Haitex 3D glasses when plugged into a real Amiga (cannot work in emulators)

When you boot from the Opticks1 disk the Startup-Sequence will run the “First!” app to make you assign three environment variables. Click on the ASSIGN button to set the variables, clicking on RETAIN means cancel (no change). The assignments are required.

If you do not boot off Opticks1 you can run “First!” manually by clicking on its icon (“Assign Dirs”), where you can assign the directories to suit your system.

The path assignments can be automated in your Startup-Sequence:

Assign OptPIC: OPTICKS1:Opticks/Pictures ; The Render output directory  
Assign OptDAT: OPTICKS1:Opticks ; The Scene data directories  
Assign OptBIN: OPTICKS1:Opticks ; The Opticks program directory

# Tips for using Opticks

**For best render output use the “Save 24 bit” option in Modes to save to a Targa (TGA) file.** Rendering to on-screen HAM display is primarily for previewing the scene. Rendering to a 24-bit file does not require the HAM or IFF options to be enabled. Recent 24-bit renders have been as high as 8192x6144. An interesting feature of rendering to TGA is that each line is saved to disk, and that a stopped TGA render can be resumed.

When setting render modes, be sure to select a render output: *HAM screen*, *IFF file*, or *24-bit file*. Even if none are selected the render will still run, but there will be no output. There is no warning for this condition.

**A note about a possible error in the file dialogs.** When the list of files is longer than the window allows, some may not initially appear. It seems we must click on the scrollbar to see / scroll through all, every time it opens.

**The Intensity Multiplier** is for overall brightness adjustment of a scene. Recent testing makes me wonder if it scales differently for HAM display than TGA save. Also, HAM display may have a minimum setting of 16.

**The Resolution Multiplier** is simply a setting for previews. Final renders should use 1, but by using the multiplier we get larger but "blockier" results. It's a way to preview while developing a scene, sometimes you just want a quick idea if you like a change you made.

**To get out of a HAM display window**, left-click the mouse.

**The .bru files in MAPS seem to be renamed .iff files.** I think I did this renaming for maps that I expected to be used as repeating patterns.

## Disk Contents

### OPTICKS1

In the Opticks directory are the GUI programs and data directories:

- Main, Digi, Convert, Att, Modes, Build, R, Hai, First!
- SCENES, OBJECTS, SRFS, LAMPS, CAMERAS, MAPS, PICTURES
  - Except for image files in Maps and Pictures, Opticks data files can be opened in a text editor.

### OPTICKS2

MAPS – Maps that didn't fit on Disk1

PICTURES – The Demo pictures

UTILS – Basic image manipulation tools, use a modern image app instead

DOCS – Miscellaneous docs and some TGA file manipulation source code samples

68881 – This directory contains versions of BUILD and R compiled with the 68020 & 68881 options. I'm not sure we had a chance to test them back then. Initial attempts in the WinUAE emulator were not successful, but that may be due to me not understanding emulator settings enough. If I recall correctly, BUILD and R on OPTICKS1 were compiled using a software floating point library, so these might prove to be speedy. To try the 68020 & 68881 versions put them in OptBIN:

# Polygonal Object File Format

The .obj file format is a text file and is fairly straightforward. All polygons MUST be planar and convex! Polygons are described in a clock-wise direction to determine the front face. See the Opticks manual for related diagrams.

Here is a cube object with data descriptions. The “;” (semicolon) and everything after it is used here as comments but they must not be in the file.

```
OBJECTv1.0      ; ID and object version number
1              ; Number of objects in file (ONLY 1 supported in v1.0)
11            ; This is a polygonal object
8 6 24        ; Point count, polygon count, polygon vertex count
-0.5 -0.5 -0.5 ; Eight points - XYZ coordinates for points 1 to 8
-0.5 0.5 -0.5
0.5 0.5 -0.5
0.5 -0.5 -0.5
-0.5 -0.5 0.5
-0.5 0.5 0.5
0.5 0.5 0.5
0.5 -0.5 0.5
2 3 4 -1 1    ; Six polygons (squares in this object), defined by point numbers
1 5 6 -2 1    ; See how surface numbers 1, 2, 3 & 4 are applied to 6 polygons
3 7 8 -4 2
7 6 5 -8 3
1 4 8 -5 4
2 6 7 -3 4
```

The negative sign in each polygon indicates the last point of the polygon, and the last number on the line is the surface number for that polygon. This cube object demonstrates 4 surfaces that need to be assigned in Build. In the OBJECTS directory, Cube.obj has 6 surfaces, Cube1.obj has 1 surface.

There are 24 vertices here, 6 polygons with 4 vertices each. The surface number is not included in the vertex count.

# Amiga Emulation

Thanks to Kevin Braun for contributing this section:

Here I'll explain emulator configuration options I use to make the most of Opticks.

My favorite way to use Opticks via emulation (FS-UAE in my case) is to create a virtual hard drive folder using UAE and copy the contents of Opticks disks there. This virtual hard drive folder is viewed by the virtual Amiga as a hard drive, with all the speed and capacity advantages that affords. Equally powerful is that the virtual hard drive folder is treated by the Mac or PC (running the Amiga emulator) as a regular folder. This allows you to drag and drop files to and from it. You can access any config files with a Windows or Mac text editor and edit them in your modern editor of choice.

Once you have Opticks running properly, the Holy Grail is speed. Back when the Amiga was contemporary, you set up a scene and went to bed and perhaps awoke to a completed render. That would be an accurate recreation of the experience from back then, but to me, it is more fun to explore Opticks by trying ideas and the faster you can render, the more ideas you can try. So assuming you have Opticks set up correctly and can render a scene, change the processor to the 68060. I am currently running an emulated Amiga 1000 in UAE but with the 68060. This renders far faster than the standard machine. No doubt you can experiment with the emulator settings and eke out more speed with an emulated accelerator board, but I am satisfied with the stability of the 68060.

If you do experiment with the emulator virtual hardware, I recommend you change only one hardware variable at a time. Troubleshooting multivariate changes will drive you nuts. This lesson also relates to your scene building and rendering. I would change one variable at a time in regards to lighting changes, material changes, and camera changes.

# Useful References

AmigaOS command reference:

[https://wiki.amigaos.net/wiki/AmigaOS\\_Manual:\\_AmigaDOS\\_Command\\_Reference](https://wiki.amigaos.net/wiki/AmigaOS_Manual:_AmigaDOS_Command_Reference)

<https://en.wikipedia.org/wiki/AmigaDOS>

The official 3.9 manuals are online, and while there's a lot of stuff that doesn't work in 1.3, many basic commands and principles are the same:

[http://amiga.nvg.org/amiga/reference/AmigaOS3.9\\_Manual/](http://amiga.nvg.org/amiga/reference/AmigaOS3.9_Manual/)

ED (text editor) commands:

<https://amigalove.com/viewtopic.php?t=190>

Startup-Sequence command info:

<https://www.stone-oakvalley-studios.com/post.php?id=000915072014001907>

A very custom Startup Sequence sample:

<http://www.amigalog.com/amigados-part-two-startup-sequences/>

Share a folder to UAE for easy file access between UAE and the host system:

See the “Setting Up A Hard Disk Folder” section here:

<http://mirrors.arcadecontrols.com/easyemu/winuaeguide/uae-hdd.htm>