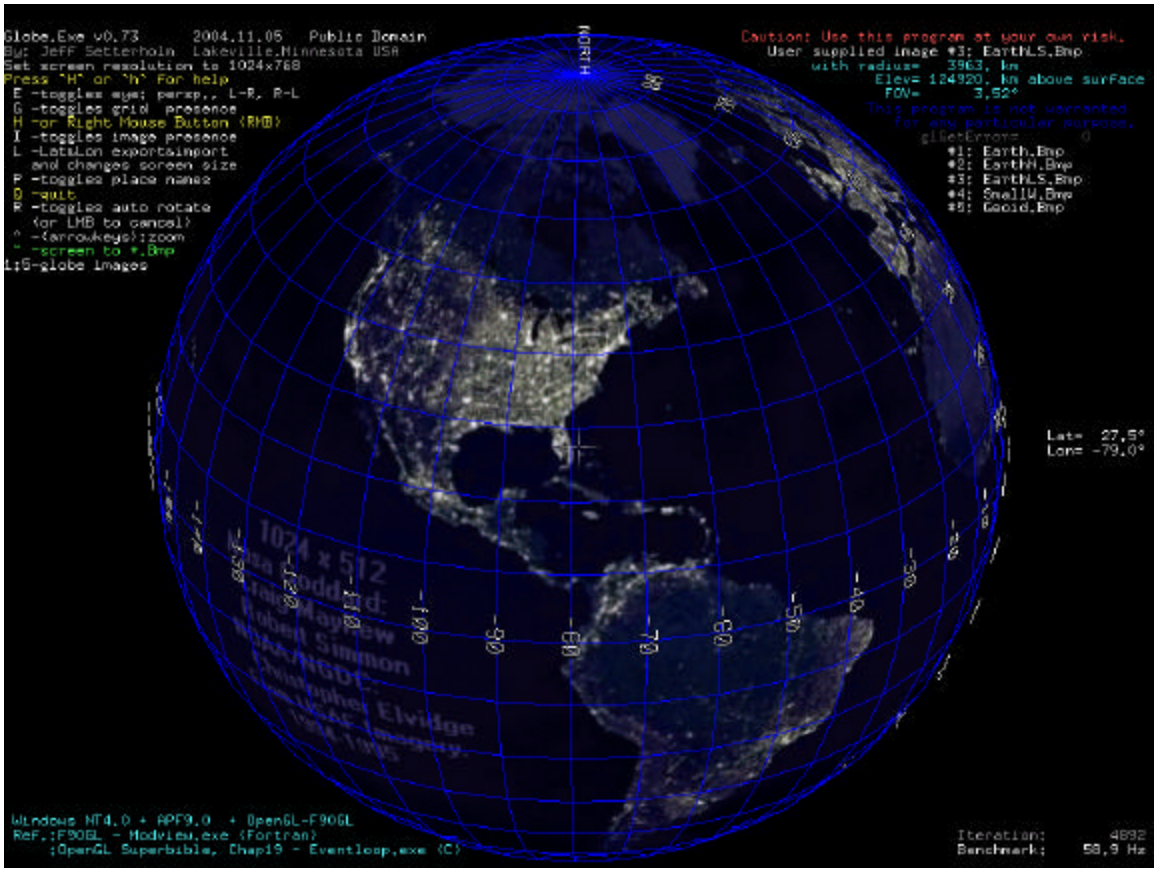
Globe.Exe Version 0.73

A download from www.setterholm.com



Want to wrap five of your own full-color *.Bmp globe images onto a quantitative rotatable sphere? **Globe.Exe** allows you to do that.

Globe073.Exe - is a *self-extracting* **WinZip** *file* that includes several images of the whole Earth. When executed, it defaults to creating a "Globe" subdirectory at the present location and extracting the zipped files to that subdirectory. (The user has the ability to intervene & redefine the target directory.) The unzipping itself happens in the blink of an eye.

Globe.Exe version 0.73 - is a public domain rotatable sphere viewer.
It self-extracts when **Globe073.exe** is run.
*****Read The Disclaimers*****
herein before attempting to run Globe.Exe.

When the globe displays, "1","2", "3","4", & "5" immediately access five different images. ("Globe.Txt" contains the image list.)

Version 0.73 Caution: "Use this program at your own risk."

DETAILS:

This program: displays 1-to-5 user supplied .Bmp images on a textured sphere which the user can switch among & rotate using the mouse. (Globe.Txt contains the user's image list.)

Globe.Exe was programmed on an **Athlon 700** processor with a **GeForce3** video card under **Windows NT 4.0** service pack 6. **Absoft Pro Fortran 9.0** and NIST's **F90GL** Fortran bindings to **OpenGL** were used. The intended screen resolution is 1024 x 768 with "32bit" color.

GLUT32.DLL should be in the same directory with Globe.Exe, if it is not already in your computer's path elsewhere. (At runtime, **Windows 2000** automatically looked for the **Glut32**.DLL in the same directory, and found it.) **GLUT** is an alternative to **GLAUX**, & bypasses almost all the complexities of **Microsoft "Windows"** programming.

Globe.Txt: Line 1: total number of bitmap images (1 minimum, 5 maximum)
(8 or 24 bit/pixel .bmp's only)
Line 2: 1st .bmp image - path (20 characters max.)
Line 3: - sphere radius in kilometers
Line 4: 2nd .bmp image - path (20 characters max.)
Line 5: - sphere radius in kilometers
...etc.

Controls include:

>**H/h** - Press **"H"** or **"h"** for help This brings controls information onto the screen.
>**E/e** - toggles eye through: perspective
: left-right stereo
: right-left stereo (for crossed-eye viewing)
>**G/g** - toggles latitude/longitude grid and grid labels presence.
>**I/i** - toggles the image presence.
>**L/l** - performs cursor latitude/longitude export and import (and changes the screen size)
"Globe.Out: Line 1: latitude (degrees)
Line 2: longitude (degrees)
When L/l is pressed to reduce screen size , export occurs.
When L/l is pressed to maximize screen size, import occurs.
>**Q/q** - quit
>**P/p** - toggles place names
>**R/r** - toggles auto-rotate
>Zoom - via **up and down arrows**.
Full zoom-out depicts the geometry of the Earth at Lunar range.
>**Mouse motion** - actively & passively tied to sphere rotation.
>" ~" - screen dump: to "ScrnDmp*.Bmp" *:[1,5]
directly from the OpenGL frame buffer.

Beeps on completion of each screen dump.
The program halts after the 5th dump.
>**glGetError output** - upper right of screen.
Checked once per iteration. (Not robust.)
Non-zero values are display in red.

Features include:

>"**Windows**" screen frame (for familiar appearance)
>Horizon-to-horizon view at all zoom levels, mono and stereo.
>Grayscale cursor & black backing for Lat/Lon numerics (for improved visibility).
>Display of user-supplied globe radius, field of view, and distance above the surface.
>Lat/Lon Grid & readout.
>Benchmark: In seconds below 5 Hz. In Hz above 5 Hz. My system:~ 60Hz monocular
(Approximate sample interval: 1 second.)

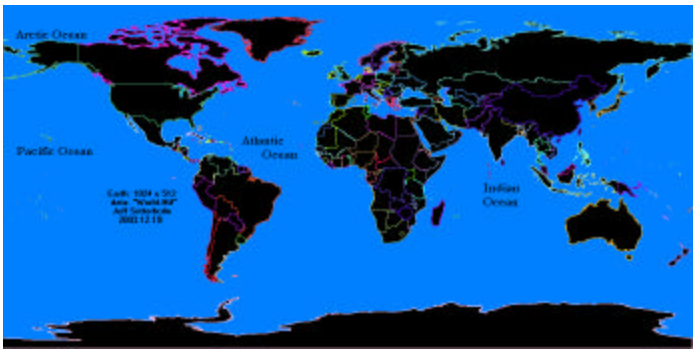
The globe images provided herein are:

Earth.Bmp
512 x 256 pixels
#1
NIMA
For Earth: use radius= 3963. kilometers
This appears to be the Sunlit Earth.

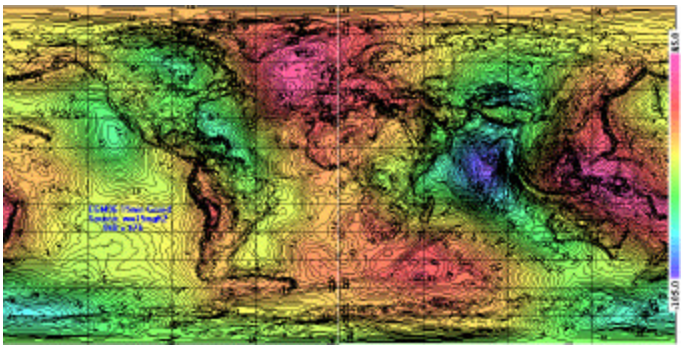
EarthN.Bmp
504 x 252 pixels
#2
NIMA
This appears to be the Moonlit Earth.

EarthLS.Bmp
1024 x 512 pixels
#3
NGDC – from USAF Photography.
This appears to be the night Earth
without moonlight.

SmallW.Bmp
1024 x 512 pixels
#4
Setterholm - using "World.Mif"
You might find this useful as a starting point for creating your own map content to wrap onto the Globe. (Note: Globe only reads 8 or 24 bit-per-pixel .bmp's.) Each pixel is .3515625 degrees square (~21 nautical miles high, varying width).



Geoid.Bmp
948 x 474 pixels
#5
NGDC – from file **"ww15mgh2.gif"**
The WGS-84 ellipsoid is close to Earth's true shape(~ +/- 100 meters). The geoid height-adjusts the ellipsoid to sea level. The elevation color scale is at the right edge.

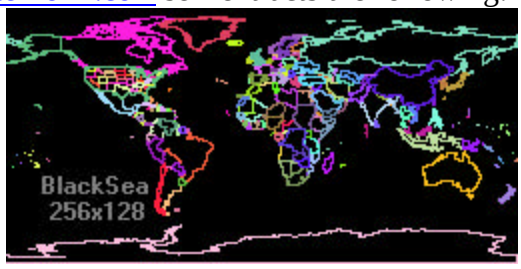


Download File:

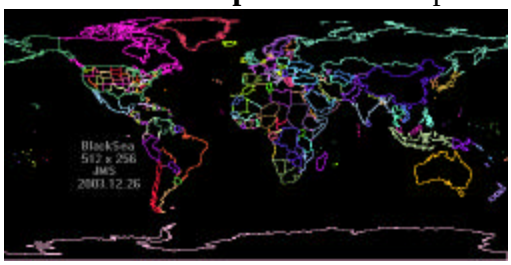
Date	Time	Bytes	Name	Further Information
11/05/04	03:00p	1,457,643	Globe073.exe	:self-extracting archive
11/05/04	03:00p	1,427,343	Globe073.ZIP	:non-self extracting archive
		----------	:contents:	----------
11/05/04	02:38p	448,031	Globe073.pdf	:Documentation
09/25/99	12:37p	393,272	EARTH.BMP	:Image#1
12/19/03	06:42a	525,366	EarthLS.bmp	:Image#3
01/04/02	03:52p	381,078	EarthN.Bmp	:Image#2
12/19/03	07:59a	1,348,110	Geoid.Bmp	:Image#5
11/05/04	01:51p	745,472	Globe.exe	:The application
11/05/04	12356p	545	Globe.txt	:File list used by Globe.Exe
11/08/01	01:27a	237,568	glut32.dll	:GLUT
12/19/03	08:23a	1,572,918	SmallW.Bmp	:image#4

Supplemental Images:

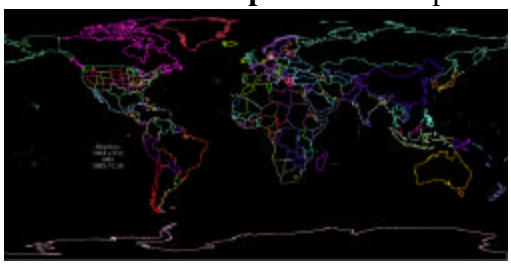
BlackSea.Exe in the root directory of www.setterholm.com self-extracts the following:
BlackSea256.Bmp 256x128 pixels



BlackSea512.Bmp 512x256 pixels



BlackSea1024.Bmp 1024x512 pixels



Compressed file size: ~ 305 kilobytes.

BlackSea2048.Bmp 2048x1024 pixels

BlueSea4096.Bmp 4096x2048 pixels

and this **Globe.Txt**:

```
5
BlackSea256.Bmp
3963.d0
BlackSea512.Bmp
3963.d0
BlackSea1024.Bmp
3963.d0
BlackSea2048.Bmp
3963.d0
BlueSea4096.Bmp
3963.d0
```

EarthL2400x1200.Exe self-extracts:

EarthL2400.Bmp 2400x1200 pixels

Usage notes:

1. **If Globe.Exe exits** before displaying the globe for the first time, you can run Globe from the DOS command line **to see the errors** that Globe reports. If Globe reports an **OpenGL** hex error, your graphics card probably isn't powerful enough to display the larger of the images listed in **Globe.Txt**.

2. **OpenGL** allocates texture memory in powers of two. The images shown here are imported as textures for wrapping onto Globe's sphere. Images that are exactly twice as wide as they are high should be used. Hence: 512 x 256, 1024 x 512, 2048 x 1024, and 4096 x 2048 pixel images make the most productive use of OpenGL video memory.

3. Pressing the Right Mouse Button, "H", or "h" adds 'help' guidance to the screen.

4. My labeling on the .Bmp Globe texture images was done using MS "**Paint**".

5. Globe version 0.73 was programmed using Absoft Pro Fortran version 9.0, which has thus far produced stable Fortran/F90GL executables - an excellent product which also natively supports C++ programming.

6. The upper left corner of the .Bmp's is:
-180 degrees (W) longitude, +90 degrees (N) latitude
The lower right corner of the .Bmp's is:
+180 degrees (E) longitude, -90 degrees (S) latitude
The entire top row of the image maps at the North pole, and the entire bottom row of the image maps at the South Pole. The leftmost and rightmost columns meet in the vicinity of the International Date Line.

7. If your .Bmp Globe image has only a few colors, **WinZip**'s compression is excellent.

8. Native **OpenGL** is programmed in C.
I learned how to texture spheres from: **OpenGL Superbible** , chapter 19:"Eventloop"
(ISBN 1-57169-164-2) Earth.Bmp is there.
F90GL – written by William Mitchell of NIST, provides the Fortran bindings. An excellent piece of work that brought commercial Fortran out of its "dark ages".

The following Intellectual Property Notice is part of the GLUT distribution:

The **OpenGL** Utility Toolkit distribution for Win32 (**Windows NT & Windows 95**) contains source code modified from the original source code for **GLUT** version 3.3 which was developed by Mark J. Kilgard. The original source code for **GLUT** is Copyright 1997 by Mark J. Kilgard. **GLUT for Win32** is Copyright 1997 by Nate Robins and is not in the public domain, but it is freely distributable without licensing fees. It is provided without guarantee or warrantee expressed or implied. It was ported with the permission of Mark J. Kilgard by Nate Robins.

THIS SOURCE CODE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.|

OpenGL (R) is a registered trademark of Silicon Graphics, Inc.

I added the blue coloring. **Words in blue** **are** (**or may be**) trademarks of other entities.

If this constitutes insufficient IP notice for this application, call me at (952) 461-3445 with the wording and/or coloring that you want added, deleted, or changed, and leave a callback number.

Disclaimer:

THE SOFTWARE, DATA, AND IMAGES HERE ARE PROVIDED AS IS WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, FITNESS FOR ANY PARTICULAR PURPOSE.

Jeffrey M. Setterholm
Lakeville, Minnesota, USA
November 5, 2004

email: info@setterholm.com