GLUT objects

Extract from The OpenGL Utility Toolkit (GLUT) Programming Interface

http://www.opengl.org/resources/libraries/glut/glut-3.spec.pdf

11 Geometric Object Rendering

GLUT includes a number of routines for generating easily recognizable 3D geometric objects. These routines reflect functionality available in the `aux` toolkit described in the *OpenGL Programmer's Guide* and are included in GLUT to allow the construction of simple GLUT programs that render recognizable objects. These routines can be implemented as pure OpenGL rendering routines. The routines do *not* generate display lists for the objects they create.
The routines generate normals appropriate for lighting but do not generate texture coordinates (except for the teapot).

11.1 glutSolidSphere, glutWireSphere

`glutSolidSphere` and `glutWireSphere` render a solid or wireframe sphere respectively.

Usage

```
void glutSolidSphere(GLdouble radius,GLint slices, GLint
stacks);
void glutWireSphere(GLdouble radius,GLint slices, GLint
stacks);
```

`radius` The radius of the sphere.
`slices` The number of subdivisions around the Z axis (similar to lines of longitude).
`stacks` The number of subdivisions along the Z axis (similar to lines of latitude).

Description
Renders a sphere centered at the modelling coordinates origin of the specified `radius`. The sphere is subdivided around the Z axis into slices and along the Z axis into stacks.

11.2 glutSolidCube, glutWireCube

`glutSolidCube` and `glutWireCube` render a solid or wireframe cube respectively.

Usage

```
void glutSolidCube(GLdouble size);
void glutWireCube(GLdouble size);
```

`size` Length of each edge.

Description

`glutSolidCube` and `glutWireCube` render a solid or wireframe cube respectively. The cube is centered at the modelling coordinates origin with sides of length `size`.

11.3 glutSolidCone, glutWireCone

`glutSolidCone` and `glutWireCone` render a solid or wireframe cone respectively.

Usage

```
void glutSolidCone(GLdouble base, GLdouble height, GLint
slices, GLint stacks);
void glutWireCone(GLdouble base, GLdouble height, GLint slices,
GLint stacks);
```

`base` The radius of the base of the cone.
`height` The height of the cone.
`slices` The number of subdivisions around the Z axis.
`stacks` The number of subdivisions along the Z axis.

Description

`glutSolidCone` and `glutWireCone` render a solid or wireframe cone respectively oriented along the Z axis. The base of the cone is placed at Z = 0, and the top at Z = `height`. The cone is subdivided around the Z axis into slices, and along the Z axis into stacks.

11.4 glutSolidTorus, glutWireTorus

`glutSolidTorus` and `glutWireTorus` render a solid or wireframe torus (doughnut) respectively.

Usage

```
void glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius,
GLint nsides, GLint rings);
void glutWireTorus(GLdouble innerRadius, GLdouble outerRadius,
GLint nsides, GLint rings);
```

`innerRadius` Inner radius of the torus.
`outerRadius` Outer radius of the torus.
`nsides` Number of sides for each radial section.
`rings` Number of radial divisions for the torus.

Description

`glutSolidTorus` and `glutWireTorus` render a solid or wireframe torus (doughnut) respectively centered at the modelling coordinates origin whose axis is aligned with the Z axis.

11.5 glutSolidDodecahedron, glutWireDodecahedron

`glutSolidDodecahedron` and `glutWireDodecahedron` render a solid or wireframe dodecahedron (12-sided regular solid) respectively.

Usage

```
void glutSolidDodecahedron(void);
void glutWireDodecahedron(void);
```

Description

`glutSolidDodecahedron` and `glutWireDodecahedron` render a solid or wireframe dodecahedron respectively centered at the modelling coordinates origin with a radius of square root of 3.

11.6 glutSolidOctahedron, glutWireOctahedron

`glutSolidOctahedron` and `glutWireOctahedron` render a solid or wireframe octahedron (8-sided regular solid) respectively.

Usage

```
void glutSolidOctahedron(void);
void glutWireOctahedron(void);
```

Description

`glutSolidOctahedron` and `glutWireOctahedron` render a solid or wireframe octahedron respectively centered at the modelling coordinates origin with a radius of 1.0.

11.7 glutSolidTetrahedron, glutWireTetrahedron

`glutSolidTetrahedron` and `glutWireTetrahedron` render a solid or wireframe tetrahedron (4-sided regular solid) respectively.

Usage

```
void glutSolidTetrahedron(void);
void glutWireTetrahedron(void);
```

Description

`glutSolidTetrahedron` and `glutWireTetrahedron` render a solid or wireframe tetrahedron respectively centered at the modelling coordinates origin with a radius of square root of 3.

11.8 glutSolidIcosahedron, glutWireIcosahedron

`glutSolidIcosahedron` and `glutWireIcosahedron` render a solid or wireframe icosahedron (20- sided regular solid) respectively.

Usage

```
void glutSolidIcosahedron(void);
void glutWireIcosahedron(void);
```

Description

`glutSolidIcosahedron` and `glutWireIcosahedron` render a solid or wireframe icosahedron respectively.
The icosahedron is centered at the modelling coordinates origin and has a radius of 1.0.

11.9 glutSolidTeapot, glutWireTeapot

`glutSolidTeapot` and `glutWireTeapot` render a solid or wireframe teapot respectively.

Usage

```
void glutSolidTeapot(GLdouble size);
void glutWireTeapot(GLdouble size);
```

`size` Relative size of the teapot.

Description

`glutSolidTeapot` and `glutWireTeapot` render a solid or wireframe teapot respectively. Both surface normals and texture coordinates for the teapot are generated. The teapot is generated with OpenGL evaluators.