Computer Graphics Lab

H M A Mohit Chowdhury

13.01.04.136

Section C Group C-1

Department of Computer Science and Engineering

Date of Submission: November 28, 2016



Ahsanullah University of Science and Technology

void glutInit(int \*argc, char \*\*argv);

Initialize the glut function

Parameters:

* argc – A pointer to the *unmodified* argc variable from the main function.
* argv – A pointer to the *unmodified* argv variable from the main function.

void glutInitWindowPosition(int x, int y);

Set the position of the window

Parameters:

x – the number of pixels from the left of the screen -1 is the default value, meaning it is up to the window manager to decide where the window will appear. If not using the default values then you should pick a positive value, preferably one that will fit in your screen.

y – the number of pixels from the top of the screen. The comments mentioned for the *x* parameter also apply in here.

void glutInitWindowSize(int width, int height);

Set the size of the window

Parameters:

* *width* – The width of the window
* *height* – the height of the window

void glutInitDisplayMode(unsigned int mode)

Parameters:

* *mode* – specifies the display mode

GLUT\_RGBA or GLUT\_RGB – selects a RGBA window. This is the default color mode.

GLUT\_INDEX – selects a color index mode.

GLUT\_SINGLE – single buffer window

GLUT\_DOUBLE – double buffer window, required to have smooth animation.

GLUT\_ACCUM – The accumulation buffer

GLUT\_STENCIL – The stencil buffer

GLUT\_DEPTH – The depth buffer

int glutCreateWindow(char \*title);

Create the window with title

Parameters:

title – sets the window title

void glutDisplayFunc(void (\*funcName)(void));

this tells GLUT that the function renderScene should be used whenever the window requires to be painted. GLUT has a function that takes as a parameter the name of the function to use when redrawing is needed.

void glutMainLoop(void)

GLUT provides a function that gets the application in a never ending loop, always waiting for the next event to process.

void glutReshapeFunc(void (\*func)(int width, int height));

when window change it calculate the height and width ration otherwise the object will be distorted

Parameters:

*func* – The name of the function that will be responsible for setting the correct perspective when the window changes size.

void glutIdleFunc(void (\*func)(void));

tell GLUT that when the application is idle, the render function should be called.

Parameters:

* + *func* – The name of the function that will be called whenever the application is idle.

glRotate — multiply the current matrix by a rotation matrix

|  |  |
| --- | --- |
| void **glRotated**( | GLdouble *angle*, |
|  | GLdouble *x*, |
|  | GLdouble *y*, |
|  | GLdouble *z*); |

|  |  |
| --- | --- |
| void **glRotatef**( | GLfloat *angle*, |
|  | GLfloat *x*, |
|  | GLfloat *y*, |
|  | GLfloat *z*); |

**Parameters**

*angle*

Specifies the angle of rotation, in degrees.

*x*, *y*, *z*

Specify the *x*, *y*, and *z* coordinates of a vector, respectively.

glutSolidSphere and   glutWireSphere render a solid or wireframe sphere respectively.

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 modeling coordinates origin of the specified radius. The sphere is subdivided around the Z axis into slices and along the Z axis into stacks.

glutSolidCone and   glutWireCone render a solid or wireframe cone respectively.

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.

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

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 modeling coordinates origin whose axis is aligned with the Z axis.

References

[1] <https://www.opengl.org/resources/libraries/glut/spec3/node1.html>

[2] http://www.lighthouse3d.com/tutorials/glut-tutorial/