



Some functions:

* glutInit: initializes GLUT, must be called before other GL/GLUT functions. It takes the same arguments as the main().

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

* glutCreateWindow: creates a window with the given title.

int **glutCreateWindow**(char \**title*)

* glutInitWindowSize: specifies the initial window width and height, in pixels.

void **glutInitWindowSize**(int *width*, int *height*)

* glutInitWindowPosition: positions the top-left corner of the initial window at (*x*, *y*). The coordinates (*x*, *y*), in term of pixels, is measured in window coordinates, i.e., origin (0, 0) is at the top-left corner of the screen; x-axis pointing right and y-axis pointing down.

void **glutInitWindowPosition**(int *x*, int *y*)

GL\_POINTS

GL\_LINES

GL\_LINE\_STRIP

GL\_LINE\_LOOP

GL\_TRIANGLES

GL\_TRIANGLE\_STRIP

GL\_TRIANGLE\_FAN

GL\_QUADS

GL\_QUAD\_STRIP

GL\_POLYGON

lab 1:code for Drawing line

#include<windows.h>

#include <GL/glut.h>

void init(void)

{

glClearColor(0.0, 0.0, 0.0, 0.0); // Set display window colour to white

glMatrixMode(GL\_PROJECTION); // Set projection parameters

gluOrtho2D(0.0, 400.0, 0.0, 400.0);

}

void drawShapes(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT); // Clear display window

//Set colour to black

glColor3f(0.0, 0.0, 0.0);

//Adjust the point size

glPointSize(10.0);

// Draw a couple of points

//Set colour to red

glColor3f(1.0, 0.0, 0.0);

// Draw a line

glBegin(GL\_LINES);

glVertex2i(20, 250);

glVertex2i(100, 80);

glEnd();

glFlush(); // Process all OpenGL routines

}

int main(int argc, char\* argv[])

{

glutInit(&argc, argv); // Initalise GLUT

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB); // Set display mode

glutInitWindowPosition(100, 100); // Set window position

glutInitWindowSize(350, 350); // Set window size

glutCreateWindow("An Example OpenGL Program"); // Create display window

init(); // Execute initialisation procedure

glutDisplayFunc(drawShapes); // Send graphics to display window

glutMainLoop(); // Display everything and wait

return 0;

}

**Drawing triangle:**

#include<windows.h>

#include <GL/glut.h>

void init(void)

{

glClearColor(1.0, 1.0, 1.0, 0.0); // Set display window colour to white

glMatrixMode(GL\_PROJECTION); // Set projection parameters

gluOrtho2D(0.0, 400.0, 0.0, 400.0);

}

void drawShapes(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT); // Clear display window

//Set colour to black

glColor3f(0.0, 0.0, 0.0);

//Adjust the point size

glPointSize(5.0);

// Draw a couple of points

//Set colour to blue

glColor3f(0.0, 0.0, 3.0);

// Draw a filled triangle

glBegin(GL\_TRIANGLES);

glVertex2i(20, 250);

glVertex2i(100, 380);

glVertex2i(180, 250);

glEnd();

glFlush(); // Process all OpenGL routines

}

**Lab 2: home**

#include<windows.h>

#include <GL/glut.h>

void init(void)

{

glClearColor(1.0, 1.0, 1.0, 0.0); // Set display window colour to white

glMatrixMode(GL\_PROJECTION); // Set projection parameters

gluOrtho2D(0.0, 400.0, 0.0, 400.0);

}

void drawShapes(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT); // Clear display window

//Set colour to black

glColor3f(0.0, 0.0, 0.0);

//Adjust the point size

glPointSize(5.0);

// Draw a couple of points

//Set colour to green

glColor3f(0.0, 0.0, 3.0);

// Draw a filled triangle

glBegin(GL\_TRIANGLES);

glVertex2i(20, 250);

glVertex2i(100, 380);

glVertex2i(180, 250);

glEnd();

//Set colour to red

glColor3f(0.0, 5.0, 0.0);

// Draw a filled quadrilateral

glBegin(GL\_QUADS);

glVertex2i(200, 250);

glVertex2i(200, 380);

glVertex2i(380, 380);

glVertex2i(380, 250);

glEnd();

/\*//Set colour to blue

glColor3f(0.0, 0.0, 1.0);

// Draw a filled octagon

glBegin(GL\_POLYGON);

glVertex2i(90, 30);

glVertex2i(30, 90);

glVertex2i(30, 174);

glVertex2i(90, 234);

glVertex2i(174, 234);

glVertex2i(234, 174);

glVertex2i(234, 90);

glVertex2i(174, 30);

glEnd();

//Set colour to black

glColor3f(0.0, 1.0, 1.0);

// Draw an outlined triangle

glBegin(GL\_LINES);

glVertex2i(100, 200);

glVertex2i(100, 50);

glVertex2i(100, 50);

glVertex2i(300, 50);

glVertex2i(300, 50);

glVertex2i(300, 200);

glVertex2i(300, 200);

glVertex2i(100, 200);

glEnd();

\*/

glFlush(); // Process all OpenGL routines

}

int main(int argc, char\* argv[])

{

glutInit(&argc, argv); // Initalise GLUT

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB); // Set display mode

glutInitWindowPosition(50, 100); // Set window position

glutInitWindowSize(400, 300); // Set window size

glutCreateWindow("An Example OpenGL Program"); // Create display window

init(); // Execute initialisation procedure

glutDisplayFunc(drawShapes); // Send graphics to display window

glutMainLoop(); // Display everything and wait

return 0;

}