Rectangle

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

/\* clear all pixels \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

/\* draw white polygon (rectangle) with corners at

\* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0)

\*/

glColor3f (1.0, 0.0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f (0.25, 0.25, 0.0);

glVertex3f (0.75, 0.25, 0.0);

glVertex3f (0.75, 0.75, 0.0);

glVertex3f (0.25, 0.75, 0.0);

glEnd();

/\* don't wait!

\* start processing buffered OpenGL routines

\*/

glFlush ();

}

void init (void)

{

/\* select clearing (background) color \*/

glClearColor (0.0, 0.0, 0.0, 0.0);

/\* initialize viewing values \*/

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);

}

/\*

\* Declare initial window size, position, and display mode

\* (single buffer and RGBA). Open window with "hello"

\* in its title bar. Call initialization routines.

\* Register callback function to display graphics.

\* Enter main loop and process events.

\*/

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (100, 100);

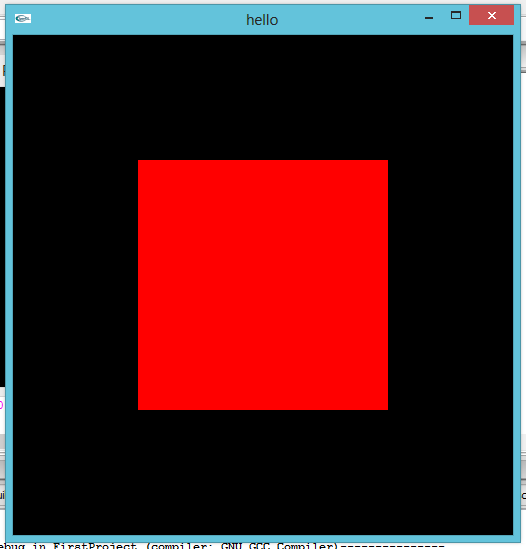
glutCreateWindow ("hello");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0; /\* ISO C requires main to return int. \*/}



Triangle 1

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

/\* clear all pixels \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

/\* draw white polygon (rectangle) with corners at

\* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0)

\*/

glColor3f (1.0, 0.0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f (0.4, 0.4, 0.0);

glVertex3f (0.6, 0.4, 0.0);

glVertex3f (0.5, 0.6, 0.0);

glEnd();

/\* don't wait!

\* start processing buffered OpenGL routines

\*/

glFlush ();

}

void init (void)

{

/\* select clearing (background) color \*/

glClearColor (0.0, 0.0, 0.0, 0.0);

/\* initialize viewing values \*/

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);

}

/\*

\* Declare initial window size, position, and display mode

\* (single buffer and RGBA). Open window with "hello"

\* in its title bar. Call initialization routines.

\* Register callback function to display graphics.

\* Enter main loop and process events.

\*/

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (500, 100);

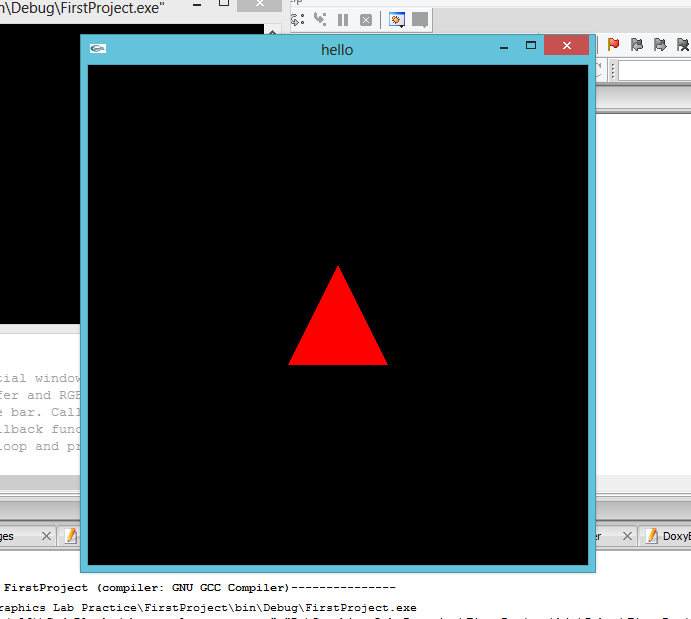
glutCreateWindow ("hello");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0; /\* ISO C requires main to return int. \*/}



Triangle 2

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

/\* clear all pixels \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

/\* draw white polygon (rectangle) with corners at

\* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0)

\*/

glColor3f (1.0, 0.0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f (0.4, 0.4, 0.0);

glVertex3f (0.6, 0.4, 0.0);

glVertex3f (0.5, 0.2, 0.0);

glEnd();

/\* don't wait!

\* start processing buffered OpenGL routines

\*/

glFlush ();

}

void init (void)

{

/\* select clearing (background) color \*/

glClearColor (0.0, 0.0, 0.0, 0.0);

/\* initialize viewing values \*/

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);

}

/\*

\* Declare initial window size, position, and display mode

\* (single buffer and RGBA). Open window with "hello"

\* in its title bar. Call initialization routines.

\* Register callback function to display graphics.

\* Enter main loop and process events.

\*/

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (500, 100);

glutCreateWindow ("hello");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0; /\* ISO C requires main to return int. \*/}

