Experiment name:2D transformation ( Home) :

Introduction: In this project, we use how to fundamental forms of triangle and rectangle to build a front-facing perspective of a home in OpenGL. Here, I'm using a 2D transformation and a straightforward shape to draw a house.

Code:

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

glClear (GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

//house structure

glColor3f(0.1f, 0.1f, 0.1f);//Dark grey

glVertex3f(0.0f,0.0f, 0.0f);

glVertex3f(40.0f, 0.0f, 0.0f);

glVertex3f(40.0f, 30.0f, 0.0f);

glVertex3f(0.0f, 30.0f, 0.0f);

//door

glColor3f (0.0, 0.0, 1.0);

glVertex3f(15.0f,0.0f, 0.0f);

glVertex3f(25.0f, 0.0f, 0.0f);

glVertex3f(25.0f, 10.0f, 0.0f);

glVertex3f(15.0f, 10.0f, 0.0f);

//window1

glColor3f(1.0f, 0.0f, 0.0f);//RED

glVertex3f(5.0f,15.0f, 0.0f);

glVertex3f(15.0f, 15.0f, 0.0f);

glVertex3f(15.0f, 25.0f, 0.0f);

glVertex3f(5.0f, 25.0f, 0.0f);

//window2

glColor3f(1.0f, 0.0f, 0.0f);//red

glVertex3f(25.0f,15.0f, 0.0f);

glVertex3f(35.0f, 15.0f, 0.0f);

glVertex3f(35.0f, 25.0f, 0.0f);

glVertex3f(25.0f, 25.0f, 0.0f);

//Translation for window 1 . tx=2,ty=2;

glColor3f(1.0f, 0.0f, 0.0f);//RED

glVertex3f(7.0f,17.0f, 0.0f);

glVertex3f(17.0f, 17.0f, 0.0f);

glVertex3f(17.0f, 27.0f, 0.0f);

glVertex3f(7.0f, 27.0f, 0.0f);

// Scaling for door

// sx=2,sy=2. and it is uniformglColor3f (0.0, 0.0, 1.0);

glVertex3f(30.0f,0.0f, 0.0f);

glVertex3f(50.0f, 0.0f, 0.0f);

glVertex3f(50.0f, 20.0f, 0.0f);

glVertex3f(30.0f, 20.0f, 0.0f);

// sharing with window along x axis

//where shx=2;

glColor3f(1.0f, 0.0f, 0.0f);//red

glVertex3f(75.0f,15.0f, 0.0f);

glVertex3f(105.0f, 15.0f, 0.0f);

glVertex3f(105.0f, 25.0f, 0.0f);

glVertex3f(75.0f, 25.0f, 0.0f);

//refection for window 2 along with x axis

glColor3f(1.0f, 0.0f, 0.0f);//red

glVertex3f(25.0f,-15.0f, 0.0f);

glVertex3f(35.0f, -15.0f, 0.0f);

glVertex3f(35.0f, -25.0f, 0.0f);

glVertex3f(25.0f, -25.0f, 0.0f);

glEnd();

//home 1

glBegin(GL\_TRIANGLES);

glColor3f(0.0, 1.0, 0.0);

glVertex3f(0.0f, 30.0f, 0.0f);

glVertex3f(40.0f, 30.0f, 0.0f);

glVertex3f(20.0f, 45.0f, 0.0f);

glEnd();

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(-120.0, 120.0, -80.0, 80.0, -10.0, 10.0);

}

/\*

\* 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 (600, 600);

glutInitWindowPosition (100, 100);

glutCreateWindow ("Farzana 2579");

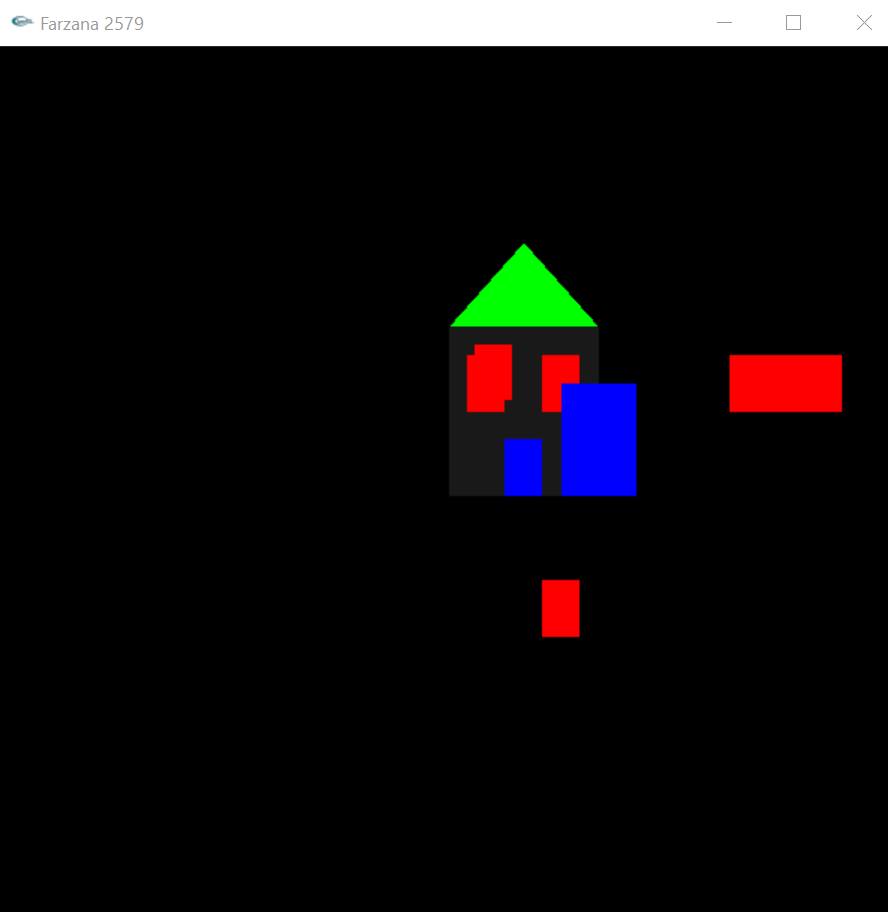
init ();

glutDisplayFunc(display);

glutMainLoop();

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

}

Output:

Conculation: Initialize the toolkit using the function glutInit (&argc, argv).Set the display mode and specify the color scheme using the function glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB).Specify the window size using the function glutInitWindowSize(600, 600).Set the starting position for the window using the function glutInitWindowPosition (100, 100).Initialize the window and set the title using the function glutCreateWindow(“color”).Initialize the myInit() function and perform the following steps:Set the background color to orange using the function glClearColor(1.0, 1.0, 1.0). Initialize the myDisplay() function and perform the following steps:Clear the screen using the function glClear(GL\_COLOR\_BUFFER\_BIT).The rectangular part of the hut can be drawn using the function .Set the drawing color to glColor3f(1.0f, 0.0f, 0.0f).Create the 2 windows, the door, the top triangle, and the main rectangle of the house using the polygon command and setting their vertices by using the functions:glBegin(GL\_POLYGON);glVertex2i ( x, y );. Using angle: function to set the position which position rotate this home. Color the polygons in the above steps using the function glColor3f(R, G, B).