**Lab Taks-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Question- 1**  Draw a Rainbow Flag   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(10.0);**  **// Draw a Red 1x1 Square centered at origin**  **glColor3f(1.0f,0.0f,0.0f);//For RED**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(4.0f,0.0f);**  **glVertex2f(4.0f,0.5f);**  **glVertex2f(0.0f,0.5f);**  **glEnd();**  **glColor3f(1.0f,1.0f,0.0f);//For Yellow**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.5f);**  **glVertex2f(4.0f,0.5f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(0.0f,1.0f);**  **glEnd();**  **glColor3ub(255, 165, 0);//For Orange**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,1.0f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(4.0f,1.5f);**  **glVertex2f(0.0f,1.5f);**  **glEnd();**  **glColor3f(0.0f,1.0f,0.0f);//For Green**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,1.5f);**  **glVertex2f(4.0f,1.5f);**  **glVertex2f(4.0f,2.0f);**  **glVertex2f(0.0f,2.0f);**  **glEnd();**  **glColor3ub(0,191,255);//For Sky**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,2.0f);**  **glVertex2f(4.0f,2.0f);**  **glVertex2f(4.0f,2.5f);**  **glVertex2f(0.0f,2.5f);**  **glEnd();**  **glColor3f(0.0f,0.0f,1.0f);//For Blue**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,2.5f);**  **glVertex2f(4.0f,2.5f);**  **glVertex2f(4.0f,3.0f);**  **glVertex2f(0.0f,3.0f);**  **glEnd();**  **glColor3ub(135,0,130);//For Purple**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,3.0f);**  **glVertex2f(4.0f,3.0f);**  **glVertex2f(4.0f,3.5f);**  **glVertex2f(0.0f,3.5f);**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("Rainbow Flag"); // Create a window with the given title**  **glutInitWindowSize(320, 320);**  **gluOrtho2D(-6.0,6.0,-6.0,6.0); //resize the axis size**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw 4X4 Chess Board |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClearColor(0.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(10.0);**  **// Draw a Red 1x1 Square centered at origin**  **//first column**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(1.0f,0.0f);**  **glVertex2f(1.0f,1.0f);**  **glVertex2f(0.0f,1.0f);**  **glEnd();**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,1.0f);**  **glVertex2f(1.0f,1.0f);**  **glVertex2f(1.0f,2.0f);**  **glVertex2f(0.0f,2.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,2.0f);**  **glVertex2f(1.0f,2.0f);**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(0.0f,3.0f);**  **glEnd();**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,3.0f);**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(1.0f,4.0f);**  **glVertex2f(0.0f,4.0f);**  **glEnd();**  **//second column**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(1.0f,0.0f);**  **glVertex2f(2.0f,0.0f);**  **glVertex2f(2.0f,1.0f);**  **glVertex2f(1.0f,1.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(1.0f,1.0f);**  **glVertex2f(2.0f,1.0f);**  **glVertex2f(2.0f,2.0f);**  **glVertex2f(1.0f,2.0f);**  **glEnd();**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(1.0f,2.0f);**  **glVertex2f(2.0f,2.0f);**  **glVertex2f(2.0f,3.0f);**  **glVertex2f(1.0f,3.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(2.0f,3.0f);**  **glVertex2f(2.0f,4.0f);**  **glVertex2f(1.0f,4.0f);**  **glEnd();**  **//Third column**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(2.0f,0.0f);**  **glVertex2f(3.0f,0.0f);**  **glVertex2f(3.0f,1.0f);**  **glVertex2f(2.0f,1.0f);**  **glEnd();**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(2.0f,1.0f);**  **glVertex2f(3.0f,1.0f);**  **glVertex2f(3.0f,2.0f);**  **glVertex2f(2.0f,2.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(2.0f,2.0f);**  **glVertex2f(3.0f,2.0f);**  **glVertex2f(3.0f,3.0f);**  **glVertex2f(2.0f,3.0f);**  **glEnd();**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(2.0f,3.0f);**  **glVertex2f(3.0f,3.0f);**  **glVertex2f(3.0f,4.0f);**  **glVertex2f(2.0f,4.0f);**  **glEnd();**  **//Fourth Column**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(3.0f,0.0f);**  **glVertex2f(4.0f,0.0f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(3.0f,1.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(3.0f,1.0f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(4.0f,2.0f);**  **glVertex2f(3.0f,2.0f);**  **glEnd();**  **glColor3f(1.0f,1.0f,1.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(3.0f,2.0f);**  **glVertex2f(4.0f,2.0f);**  **glVertex2f(4.0f,3.0f);**  **glVertex2f(3.0f,3.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(3.0f,3.0f);**  **glVertex2f(4.0f,3.0f);**  **glVertex2f(4.0f,4.0f);**  **glVertex2f(3.0f,4.0f);**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("Chess Board"); // Create a window with the given title**  **glutInitWindowSize(320, 320);**  **gluOrtho2D(-5.0,5.0,-5.0,5.0); //resize the axis size**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 3**  Create the batman logo given below- |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(20.0);**  **// Draw a Red 1x1 Square centered at origin**  **//For First background layer under the logo(Black)**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-11.0f,-18.0f);**  **glVertex2f(11.0f,-18.0f);**  **glVertex2f(15.0f,-10.0f);**  **glVertex2f(15.0f,10.0f);**  **glVertex2f(11.0f,20.0f);**  **glVertex2f(-11.0f,20.0f);**  **glVertex2f(-15.0f,10.0f);**  **glVertex2f(-15.0f,-10.0f);**  **glEnd();**  **//For Second background layer under the logo(Yellow)**  **glColor3f(1.0f,1.0f,0.0f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-10.0f,-17.0f);**  **glVertex2f(10.0f,-17.0f);**  **glVertex2f(14.0f,-9.0f);**  **glVertex2f(14.0f,9.0f);**  **glVertex2f(10.0f,19.0f);**  **glVertex2f(-10.0f,19.0f);**  **glVertex2f(-14.0f,9.0f);**  **glVertex2f(-14.0f,-9.0f);**  **glEnd();**  **//For first part of logo positive x, positive y axis**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(3.0f,0.0f);**  **glVertex2f(3.0f,10.0f);**  **glVertex2f(2.0f,7.0f);**  **glVertex2f(0.0f,7.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(3.0f,0.0f);**  **glVertex2f(5.0f,0.0f);**  **glVertex2f(5.0f,3.5f);**  **glVertex2f(3.0f,4.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(5.0f,0.0f);**  **glVertex2f(10.0f,0.0f);**  **glVertex2f(10.0f,3.5f);**  **glVertex2f(5.0f,3.5f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(5.0f,3.5f);**  **glVertex2f(10.0f,3.5f);**  **glVertex2f(10.0f,5.0f);**  **glVertex2f(6.0f,5.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(6.0f,5.0f);**  **glVertex2f(10.0f,5.0f);**  **glVertex2f(9.0f,10.0f);**  **glVertex2f(7.0f,10.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(7.0f,10.0f);**  **glVertex2f(9.0f,10.0f);**  **glVertex2f(5.0f,15.0f);**  **glEnd();**  **//For second part of logo negative x, positive y axis(same as first part just change the x axis value in negative)**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(-3.0f,0.0f);**  **glVertex2f(-3.0f,10.0f);**  **glVertex2f(-2.0f,7.0f);**  **glVertex2f(0.0f,7.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-3.0f,0.0f);**  **glVertex2f(-5.0f,0.0f);**  **glVertex2f(-5.0f,3.5f);**  **glVertex2f(-3.0f,4.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-5.0f,0.0f);**  **glVertex2f(-10.0f,0.0f);**  **glVertex2f(-10.0f,3.5f);**  **glVertex2f(-5.0f,3.5f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-5.0f,3.5f);**  **glVertex2f(-10.0f,3.5f);**  **glVertex2f(-10.0f,5.0f);**  **glVertex2f(-6.0f,5.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-6.0f,5.0f);**  **glVertex2f(-10.0f,5.0f);**  **glVertex2f(-9.0f,10.0f);**  **glVertex2f(-7.0f,10.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_TRIANGLES);**  **glVertex2f(-7.0f,10.0f);**  **glVertex2f(-9.0f,10.0f);**  **glVertex2f(-5.0f,15.0f);**  **glEnd();**  **//For fourth part of logo positive x, negative y axis**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(0.0f,-12.0f);**  **glVertex2f(3.0f,-7.0f);**  **glVertex2f(3.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(3.0f,-7.0f);**  **glVertex2f(5.0f,-5.0f);**  **glVertex2f(5.0f,0.0f);**  **glVertex2f(3.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(5.0f,-5.0f);**  **glVertex2f(6.5f,-8.0f);**  **glVertex2f(6.5f,0.0f);**  **glVertex2f(5.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(6.5f,0.0f);**  **glVertex2f(6.5f,-12.0f);**  **glVertex2f(8.0f,-10.0f);**  **glVertex2f(8.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(8.0f,0.0f);**  **glVertex2f(8.0f,-10.0f);**  **glVertex2f(10.0f,-3.0f);**  **glVertex2f(10.0f,0.0f);**  **glEnd();**  **//For third part of logo negative x, negative y axis(same as fourth part just change the x value in negative)**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f,0.0f);**  **glVertex2f(0.0f,-12.0f);**  **glVertex2f(-3.0f,-7.0f);**  **glVertex2f(-3.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-3.0f,-7.0f);**  **glVertex2f(-5.0f,-5.0f);**  **glVertex2f(-5.0f,0.0f);**  **glVertex2f(-3.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-5.0f,-5.0f);**  **glVertex2f(-6.5f,-8.0f);**  **glVertex2f(-6.5f,0.0f);**  **glVertex2f(-5.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-6.5f,0.0f);**  **glVertex2f(-6.5f,-12.0f);**  **glVertex2f(-8.0f,-10.0f);**  **glVertex2f(-8.0f,0.0f);**  **glEnd();**  **glColor3f(0.0f,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-8.0f,0.0f);**  **glVertex2f(-8.0f,-10.0f);**  **glVertex2f(-10.0f,-3.0f);**  **glVertex2f(-10.0f,0.0f);**  **glEnd();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("Batman Logo"); // Create a window with the given title**  **glutInitWindowSize(320, 320);**  **gluOrtho2D(-30.0,30.0,-30.0,30.0); //resize the axis size**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |