**Lab Taks-2**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* Must submit within time that will be discussed in class VUES to the section named Lab Tak-2
* Must include resources for all the section in the table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Question- 1**  Draw a Rainbow Flag   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h>**  **#include <GL/glut.h>**  **void initGL() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **}**  **void display()**  **{**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **//purple**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 01.0f);**  **glVertex2f(-0.8, 0.4);**  **glVertex2f(0.8, 0.4);**  **glVertex2f(0.8, 0.6);**  **glVertex2f(-0.8, 0.6);**  **glEnd();**  **//blue**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 01.0f);**  **glVertex2f(-0.8, 0.2);**  **glVertex2f(0.8, 0.2);**  **glVertex2f(0.8, 0.4);**  **glVertex2f(-0.8, 0.4);**  **glEnd();**  **//blue**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 1.0f, 01.0f);**  **glVertex2f(-0.8, 0.0);**  **glVertex2f(0.8, 0.0);**  **glVertex2f(0.8, 0.2);**  **glVertex2f(-0.8, 0.2);**  **glEnd();**  **//green**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 1.0f, 0.0f);**  **glVertex2f(-0.8, -0.2);**  **glVertex2f(0.8, -0.2);**  **glVertex2f(0.8, 0.0);**  **glVertex2f(-0.8, 0.0);**  **glEnd();**  **//green**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.5f, 0.0f);**  **glVertex2f(-0.8, -0.4);**  **glVertex2f(0.8, -0.4);**  **glVertex2f(0.8, -0.2);**  **glVertex2f(-0.8, -0.2);**  **glEnd();**  **//yellow**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 01.0f, 0.0f);**  **glVertex2f(-0.8, -0.6);**  **glVertex2f(0.8, -0.6);**  **glVertex2f(0.8, -0.4);**  **glVertex2f(-0.8, -0.4);**  **glEnd();**  **//red**  **glBegin(GL\_POLYGON);**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(-0.8, -0.8);**  **glVertex2f(0.8, -0.8);**  **glVertex2f(0.8, -0.6);**  **glVertex2f(-0.8, -0.6);**  **glEnd();**  **glFlush();**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title**  **glutInitWindowSize(320, 320); // Set the window's initial width & height**  **glutInitWindowPosition(50, 50); // Position the window's initial top-left corner**  **glutDisplayFunc(display); // Register callback handler for window re-paint event**  **initGL(); // Our own OpenGL initialization**  **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>**  **#include <GL/glut.h>**  **void initGL() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **}**  **void display()**  **{**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **//outline**  **glBegin(GL\_LINES);**  **glColor3f(0,0,0);**  **glVertex2f(-4,4);**  **glVertex2f(4,4);**  **glVertex2f(-4,4);**  **glVertex2f(-4,-4);**  **glVertex2f(4,-4);**  **glVertex2f(-4,-4);**  **glVertex2f(4,-4);**  **glVertex2f(4,4);**  **glEnd();**  **//B1**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-4, 2);**  **glVertex2f(-2, 2);**  **glVertex2f(-2, 4);**  **glVertex2f(-4, 4);**  **glEnd();**  **//B2**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(0, 2);**  **glVertex2f(2, 2);**  **glVertex2f(2, 4);**  **glVertex2f(0, 4);**  **glEnd();**  **//B3**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-2, 0);**  **glVertex2f(0,0);**  **glVertex2f(0,2);**  **glVertex2f(-2,2);**  **glEnd();**  **//B4**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(2, 0);**  **glVertex2f(4,0);**  **glVertex2f(4,2);**  **glVertex2f(2,2);**  **glEnd();**  **//B5**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-4,-2);**  **glVertex2f(-2,-2);**  **glVertex2f(-2,0);**  **glVertex2f(-4,0);**  **glEnd();**  **//B6**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(0,-2);**  **glVertex2f(2,-2);**  **glVertex2f(2,0);**  **glVertex2f(0,0);**  **glEnd();**  **//B7**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(-2,-4);**  **glVertex2f(0,-4);**  **glVertex2f(0,-2);**  **glVertex2f(-2,-2);**  **glEnd();**  **//B8**  **glBegin(GL\_POLYGON);**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(2,-4);**  **glVertex2f(4,-4);**  **glVertex2f(4,-2);**  **glVertex2f(2,-2);**  **glEnd();**  **glFlush();**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title**  **glutInitWindowSize(720, 720); // Set the window's initial width & height**  **glutInitWindowPosition(20, 20); // Position the window's initial top-left corner**  **glutDisplayFunc(display); // Register callback handler for window re-paint event**  **initGL(); // Our own OpenGL initialization**  **gluOrtho2D(-5,+5,-5,+5);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 3**  Create the batman logo given below- |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(5.2, 4.4);  glVertex2f(5.2, 5.8);  glVertex2f(5.8, 4.8);  glVertex2f(5.8, 4.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(5.8, 4.4);  glVertex2f(5.8, 4.8);  glVertex2f(6.2, 4.8);  glVertex2f(6.2, 4.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(6.2, 4.4);  glVertex2f(6.2, 4.8);  glVertex2f(6.8, 5.8);  glVertex2f(6.8, 4.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(4.2, 4.8);  glVertex2f(3.8, 5.8);  glVertex2f(3.7, 6.3);  glVertex2f(1.3, 6.3);  glVertex2f(1.7, 6.1);  glVertex2f(2.5, 5.6);  glVertex2f(2.7, 5.3);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(2.7, 5.3);  glVertex2f(2.6, 4.9);  glVertex2f(2.2, 4.5);  glVertex2f(2.2, 4.5);  glVertex2f(2.2, 4.5);  glVertex2f(1.8, 4.2);  glVertex2f(1.2, 4);  glVertex2f(3, 4);  glVertex2f(3.9, 5.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(2.2, 4);  glVertex2f(5, 2.6);  glVertex2f(7, 2.6);  glVertex2f(6.8, 4.4);  glVertex2f(5.2, 4.4);  glVertex2f(4.4, 4.4);  glVertex2f(3.9, 5.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(4.6, 3.2);  glVertex2f(5, 2.6);  glVertex2f(5.4, 1.8);  glVertex2f(5.6, 1.4);  glVertex2f(5.8, 1);  glVertex2f(6, 0.4);  glVertex2f(6.2, 1);  glVertex2f(6.4, 1.4);  glVertex2f(6.6, 1.8);  glVertex2f(7, 2.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(7.6, 4.4);  glVertex2f(7.8, 4.8);  glVertex2f(8, 5.2);  glVertex2f(8.3, 6.3);  glVertex2f(10.8, 6.3);  glVertex2f(10.1803694012847, 5.9873936853086);  glVertex2f(9.8, 5.8);  glVertex2f(9.5, 5.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(8.8, 4);  glVertex2f(8, 4);  glVertex2f(6.8, 4.4);  glVertex2f(7.26, 4.31);  glVertex2f(7.6, 4.4);  glVertex2f(8, 5.2);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(8.2, 5.8);  glVertex2f(9.2, 5.2);  glVertex2f(9.2, 4.8);  glVertex2f(9.4, 4.5);  glVertex2f(9.8, 4.3);  glVertex2f(10.07, 4.20);  glVertex2f(10.8, 4);  glVertex2f(8, 4);  glVertex2f(6.0, 0.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f);  glVertex2f(6.8, 4.4);  glVertex2f(7, 2.6);  glVertex2f(7.4, 3.2);  glVertex2f(7.8, 3.8);  glVertex2f(8, 4);  glEnd();  glFlush();  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(720, 720); // Set the window's initial width & height  glutInitWindowPosition(20, 20); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL();  gluOrtho2D(-15,15,-15,15) ; // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |