**Lab Taks-4**

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-4
* Must include resources for all the section in the table

|  |
| --- |
| **Question- 1**  Draw the scenario of a traffic signal |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **void circle(float radius, float xC, float yC, float r, float g, float b)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(r,g,b);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=radius;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+xC,y+yC);**  **}**  **glEnd();**  **}**  **/\* 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(7.5);**  **glBegin(GL\_POLYGON);//stand**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(6.0f, 0.0f);**  **glVertex2f(7.0f, 0.0f);**  **glVertex2f(7.0f, 6.0f);**  **glVertex2f(6.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//stand 2**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(4.0f, 6.0f);**  **glVertex2f(9.0f, 6.0f);**  **glVertex2f(9.0f, 18.0f);**  **glVertex2f(4.0f, 18.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//stand 3**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(6.0f, 18.0f);**  **glVertex2f(7.0f, 18.0f);**  **glVertex2f(7.0f, 22.0f);**  **glVertex2f(6.0f, 22.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//t1**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(2.0f, 16.0f);**  **glVertex2f(4.0f, 15.0f);**  **glVertex2f(4.0f, 17.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//t2**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(2.0f, 12.0f);**  **glVertex2f(4.0f, 11.0f);**  **glVertex2f(4.0f, 13.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//t3**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(2.0f, 8.0f);**  **glVertex2f(4.0f, 7.0f);**  **glVertex2f(4.0f, 9.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//t4**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(11.0f, 8.0f);**  **glVertex2f(9.0f, 9.0f);**  **glVertex2f(9.0f, 7.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//t5**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(11.0f,12.0f);**  **glVertex2f(9.0f, 13.0f);**  **glVertex2f(9.0f, 11.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);//t6**  **glColor3ub(0.0f, 0.0f, 0.0f);**  **glVertex2f(11.0f,16.0f);**  **glVertex2f(9.0f, 17.0f);**  **glVertex2f(9.0f, 15.0f);**  **glEnd();**  **circle(1.5,6.5,16,200,0,0);**  **circle(1.5,6.5,12,232,220,36);**  **circle(1.5,6.5,8,0,200,0);**  **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("OpenGL Setup Test");**  **gluOrtho2D(0,20,0,20); // Create a window with the given title**  **glutInitWindowSize(320, 320);// Set the window's initial width & height**  **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 two village scenarios for day and night |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code-**  **//DAY**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **void circle(float radius, float xC, float yC, float r, float g, float b)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(r,g,b);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=radius;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+xC,y+yC);**  **}**  **glEnd();**  **}**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(7.5);**  **glClearColor(20.0f,167.0f,242.0f,1.0f); // Set background color to black and opaque**  **glBegin(GL\_POLYGON); //HOME 1**  **glColor3ub(215, 67, 7 );**  **glVertex2f(6.0f, 0.0f);**  **glVertex2f(22.0f, 0.0f);**  **glVertex2f(22.0f, 22.0f);**  **glVertex2f(6.0f, 22.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //HOME 2**  **glColor3ub(16, 167, 191 );**  **glVertex2f(6.0f, 22.0f);**  **glVertex2f(22.0f, 22.0f);**  **glVertex2f(14.0f, 33.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //HOME 3**  **glColor3ub(0.0f, 0.0f, 0.0f );**  **glVertex2f(11.0f, 0.0f);**  **glVertex2f(17.0f, 0.0f);**  **glVertex2f(17.0f, 6.0f);**  **glVertex2f(11.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //TREE BELOW**  **glColor3ub(190, 111, 17 );**  **glVertex2f(30.0f, 0.0f);**  **glVertex2f(38.0f, 0.0f);**  **glVertex2f(38.0f, 22.0f);**  **glVertex2f(30.0f, 22.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //TREE 1**  **glColor3ub(38.0f, 137.0f, 5.0f );**  **glVertex2f(26.0f, 22.0f);**  **glVertex2f(42.0f, 22.0f);**  **glVertex2f(34.0f, 30.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //TREE 2**  **glColor3ub(38.0f, 137.0f, 5.0f );**  **glVertex2f(26.0f, 26.0f);**  **glVertex2f(42.0f, 26.0f);**  **glVertex2f(34.0f, 36.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 1**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(2.0f, 0.0f);**  **glVertex2f(4.0f, 0.0f);**  **glVertex2f(4.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 2**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(4.0f, 0.0f);**  **glVertex2f(6.0f, 0.0f);**  **glVertex2f(2.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 3**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(2.0f, 0.0f);**  **glVertex2f(1.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 4**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(23.0f, 0.0f);**  **glVertex2f(25.0f, 0.0f);**  **glVertex2f(26.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 5**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(25.0f, 0.0f);**  **glVertex2f(27.0f, 0.0f);**  **glVertex2f(24.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 5**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(39.0f, 0.0f);**  **glVertex2f(41.0f, 0.0f);**  **glVertex2f(42.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 6**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(43.0f, 0.0f);**  **glVertex2f(45.0f, 0.0f);**  **glVertex2f(44.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //GRASS 7**  **glColor3ub(0.0f, 137.0f, 0.0f );**  **glVertex2f(44.0f, 0.0f);**  **glVertex2f(40.0f, 0.0f);**  **glVertex2f(40.0f, 6.0f);**  **glEnd();**  **circle(6,42,50,230, 227, 10 );**  **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("OpenGL Setup Test");**  **gluOrtho2D(0,50,0,60); // Create a window with the given title**  **glutInitWindowSize(320, 320);// Set the window's initial width & height**  **glutDisplayFunc(display);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}**  **//NIGHT**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **void circle(float radius, float xC, float yC, float r, float g, float b)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(r,g,b);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=radius;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+xC,y+yC);**  **}**  **glEnd();**  **}**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(7.5);**  **glClearColor(0.0f,0.0f,0.0f,1.0f); // Set background color to black and opaque**  **glBegin(GL\_POLYGON); //HOME 1**  **glColor3ub(215, 67, 7 );**  **glVertex2f(6.0f, 0.0f);**  **glVertex2f(22.0f, 0.0f);**  **glVertex2f(22.0f, 22.0f);**  **glVertex2f(6.0f, 22.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //HOME 2**  **glColor3ub(16, 167, 191 );**  **glVertex2f(6.0f, 22.0f);**  **glVertex2f(22.0f, 22.0f);**  **glVertex2f(14.0f, 33.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //HOME 3**  **glColor3ub(0.0f, 0.0f, 0.0f );**  **glVertex2f(11.0f, 0.0f);**  **glVertex2f(17.0f, 0.0f);**  **glVertex2f(17.0f, 6.0f);**  **glVertex2f(11.0f, 6.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //TREE BELOW**  **glColor3ub(190, 111, 17 );**  **glVertex2f(30.0f, 0.0f);**  **glVertex2f(38.0f, 0.0f);**  **glVertex2f(38.0f, 22.0f);**  **glVertex2f(30.0f, 22.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //TREE 1**  **glColor3ub(38.0f, 137.0f, 5.0f );**  **glVertex2f(26.0f, 22.0f);**  **glVertex2f(42.0f, 22.0f);**  **glVertex2f(34.0f, 30.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); //TREE 2**  **glColor3ub(38.0f, 137.0f, 5.0f );**  **glVertex2f(26.0f, 26.0f);**  **glVertex2f(42.0f, 26.0f);**  **glVertex2f(34.0f, 36.0f);**  **glEnd();**  **circle(6,42,50,255,255, 255 );**  **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("OpenGL Setup Test");**  **gluOrtho2D(0,50,0,60); // Create a window with the given title**  **glutInitWindowSize(320, 320);// Set the window's initial width & height**  **glutDisplayFunc(display);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

A computer screen with a black screen

Description automatically generated