**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 using function to represent each object |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **void four\_side()**  **{**  **glColor3f(1,1,1);**  **glBegin(GL\_LINES);**  **glVertex2f(2,15);**  **glVertex2f(2, 2);**  **glVertex2f(2,2);**  **glVertex2f(15,2);**  **glVertex2f(-2,15);**  **glVertex2f(-2,2);**  **glVertex2f(-2,2);**  **glVertex2f(-15,2);**  **glVertex2f(-2,-2);**  **glVertex2f(-15,-2);**  **glVertex2f(-2,-2);**  **glVertex2f(-2,-15);**  **glVertex2f(2,-2);**  **glVertex2f(15,-2);**  **glVertex2f(2,-2);**  **glVertex2f(2,-15);**  **glVertex2f(0,15);**  **glVertex2f(0,9);**  **glVertex2f(0,8);**  **glVertex2f(0,7);**  **glVertex2f(0,6);**  **glVertex2f(0,5);**  **glVertex2f(0,4);**  **glVertex2f(0,3);**  **glVertex2f(0,2);**  **glVertex2f(0,1);**  **glVertex2f(0,0);**  **glVertex2f(0,-1);**  **glVertex2f(0,-2);**  **glVertex2f(0,-3);**  **glVertex2f(0,-4);**  **glVertex2f(0,-5);**  **glVertex2f(0,-6);**  **glVertex2f(0,-7);**  **glVertex2f(0,-8);**  **glVertex2f(0,-9);**  **glVertex2f(-15,0);**  **glVertex2f(-14,0);**  **glVertex2f(-13,0);**  **glVertex2f(-12,0);**  **glVertex2f(-11,0);**  **glVertex2f(-10,0);**  **glVertex2f(-9,0);**  **glVertex2f(-8,0);**  **glVertex2f(-7,0);**  **glVertex2f(-6,0);**  **glVertex2f(-5,0);**  **glVertex2f(-4,0);**  **glVertex2f(-3,0);**  **glVertex2f(-2,0);**  **glVertex2f(-1,0);**  **glVertex2f(-0.5,0);**  **glVertex2f(1,0);**  **glVertex2f(2,0);**  **glVertex2f(3,0);**  **glVertex2f(4,0);**  **glVertex2f(5,0);**  **glVertex2f(6,0);**  **glVertex2f(7,0);**  **glVertex2f(8,0);**  **glVertex2f(9,0);**  **glVertex2f(10,0);**  **glVertex2f(10,0);**  **glVertex2f(11,0);**  **glVertex2f(12,0);**  **glVertex2f(13,0);**  **glVertex2f(14,0);**  **glVertex2f(15,0);**  **glEnd();**  **}**  **void Four\_Traffic\_light()**  **{**  **//1st**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.2, 2.8);**  **glVertex2f(4.2, 2.8);**  **glVertex2f(4.2, 2.1);**  **glVertex2f(2.2, 2.1);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.6, 3.4);**  **glVertex2f(3.8, 3.4);**  **glVertex2f(3.8, 2.8);**  **glVertex2f(2.6, 2.8);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(3, 4.6);**  **glVertex2f(3.4, 4.6);**  **glVertex2f(3.4, 3.4);**  **glVertex2f(3, 3.4);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.5, 8);**  **glVertex2f(4, 8);**  **glVertex2f(4, 4.6);**  **glVertex2f(2.5, 4.6);**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=7.45-7;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.2,y+7.5 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.2,y+6.5 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(20, 90, 50);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.2,y+5.5 );**  **}**  **glEnd();**  **//2nd**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-4.2, 2.8);**  **glVertex2f(-2.2, 2.8);**  **glVertex2f(-2.2, 2.1);**  **glVertex2f(-4.2, 2.1);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-3.8, 3.4);**  **glVertex2f(-2.6, 3.4);**  **glVertex2f(-2.6, 2.8);**  **glVertex2f(-3.8, 2.8);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-3.4, 4.6);**  **glVertex2f(-3, 4.6);**  **glVertex2f(-3, 3.4);**  **glVertex2f(-3.4, 3.4);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-4, 8);**  **glVertex2f(-2.4, 8);**  **glVertex2f(-2.4, 4.6);**  **glVertex2f(-4, 4.6);**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(231, 76, 60);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.2,y+7.5 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.2,y+6.5 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.2,y+5.5 );**  **}**  **glEnd();**  **//3rd**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-4.5, -7.2);**  **glVertex2f(-2.2, -7.2);**  **glVertex2f(-2.2, -8);**  **glVertex2f(-4.5, -8);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-4.2, -6.5);**  **glVertex2f(-2.6, -6.5);**  **glVertex2f(-2.6, -7.2);**  **glVertex2f(-4.2, -7.2);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-3.6, -5.5);**  **glVertex2f(-3.2, -5.5);**  **glVertex2f(-3.2, -6.5);**  **glVertex2f(-3.6, -6.5);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-4.2, -2.2);**  **glVertex2f(-2.6, -2.2);**  **glVertex2f(-2.6, -5.5);**  **glVertex2f(-4.2, -5.5);**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(231, 76, 60);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.4,y-2.9 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.4,y-3.9 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.4,y-4.9 );**  **}**  **glEnd();**  **//4th**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.2, -7.2);**  **glVertex2f(4.8, -7.2);**  **glVertex2f(4.8, -8);**  **glVertex2f(2.2, -8);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.6, -6.5);**  **glVertex2f(4.4, -6.5);**  **glVertex2f(4.4, -7.2);**  **glVertex2f(2.6, -7.2);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(3.3, -5.2);**  **glVertex2f(3.6, -5.2);**  **glVertex2f(3.6, -6.5);**  **glVertex2f(3.3, -6.5);**  **glEnd();**  **glColor3ub(127, 140, 141);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.6, -2.2);**  **glVertex2f(4.4, -2.2);**  **glVertex2f(4.4, -5.2);**  **glVertex2f(2.6, -5.2);**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(231, 76, 60);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.5,y-2.7 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.5,y-3.7 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0, 0, 0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.45;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.5,y-4.7 );**  **}**  **glEnd();**  **}**  **void car()**  **{**  **//lower**  **glColor3ub(185, 119, 14);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.5, -1);**  **glVertex2f(6, -1);**  **glVertex2f(6, -1.8);**  **glVertex2f(2.5, -1.8);**  **glEnd();**  **//upper**  **glColor3ub(241, 196, 15);**  **glBegin(GL\_POLYGON);**  **glVertex2f(3.5, -0.2);**  **glVertex2f(5, -0.2);**  **glVertex2f(5.5, -1);**  **glVertex2f(3,-1);**  **glEnd();**  **//window 1**  **glColor3ub(240, 243, 244 );**  **glBegin(GL\_POLYGON);**  **glVertex2f(3.6, -0.4);**  **glVertex2f(3.95, -0.4);**  **glVertex2f(4.2, -0.8);**  **glVertex2f(3.6, -0.8);**  **glEnd();**  **//window 2**  **glColor3ub(240, 243, 244 );**  **glBegin(GL\_POLYGON);**  **glVertex2f(4.4, -0.4);**  **glVertex2f(4.8, -0.4);**  **glVertex2f(5, -0.8);**  **glVertex2f(4.4, -0.8);**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(40, 40, 43) ;**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=-0.40;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+4.9,y-1.60 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(40, 40, 43) ;**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=-0.40;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+3.77,y-1.60 );**  **}**  **glEnd();**  **//car2**  **//lower**  **glColor3ub(255, 127, 80);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-7, -1);**  **glVertex2f(-3, -1);**  **glVertex2f(-3, -1.8);**  **glVertex2f(-7, -1.8);**  **glEnd();**  **//upper**  **glColor3ub(250, 128, 144);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-5.5, -0.2);**  **glVertex2f(-4, -0.2);**  **glVertex2f(-3.25, -1);**  **glVertex2f(-6, -1);**  **glEnd();**  **//window1**  **glColor3ub(240, 243, 244 );**  **glBegin(GL\_POLYGON);**  **glVertex2f(-5.4, -0.4);**  **glVertex2f(-5, -0.4);**  **glVertex2f(-5, -0.8);**  **glVertex2f(-5.6, -0.8);**  **glEnd();**  **//window2**  **glColor3ub(240, 243, 244 );**  **glBegin(GL\_POLYGON);**  **glVertex2f(-4.6, -0.4);**  **glVertex2f(-4, -0.4);**  **glVertex2f(-4, -0.8);**  **glVertex2f(-4.6, -0.8);**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(40, 40, 43);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=-0.40;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-3.7,y-1.60);**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(40, 40, 43);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=-0.40;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-5.52,y-1.60);**  **}**  **glEnd();**  **//Car3**  **glColor3ub(4, 205, 246);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-1.5, -5);**  **glVertex2f(-0.5, -5);**  **glVertex2f(-0.5, -7);**  **glVertex2f(-1.5, -7);**  **glEnd();**  **glColor3ub(255, 255, 255);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-1.5, -5.5);**  **glVertex2f(-0.5, -5.5);**  **glVertex2f(-0.5, -5.8);**  **glVertex2f(-1.5, -5.8);**  **glEnd();**  **glColor3ub(4, 205, 246);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-1.5, -5.8);**  **glVertex2f(-0.5, -5.8);**  **glVertex2f(-0.5, -6.3);**  **glVertex2f(-1.5, -6.3);**  **glEnd();**  **glColor3ub(255, 255, 255);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-1.5, -6.3);**  **glVertex2f(-0.5, -6.3);**  **glVertex2f(-0.5, -6.6);**  **glVertex2f(-1.5, -6.6);**  **glEnd();**  **glColor3ub(4, 205, 246);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-1.5, -6.6);**  **glVertex2f(-0.5, -6.6);**  **glVertex2f(-0.5, -7);**  **glVertex2f(-1.5, -7);**  **glEnd();**  **}**  **/\*void circle()**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(246, 220, 4);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.70;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y );**  **}**  **glEnd();**  **}**  **\*/**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **four\_side();**  **Four\_Traffic\_light();**  **car();**  **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(-0.1,0.7,-0.1,0.3); // 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**  **gluOrtho2D(-10,10,-10,10);**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw two village scenarios for day and night using function to represent each object |
| **Graph Plot (Picture)-** |
| **Code-**  **///////day///////**  **/\*#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **/\* Initialize OpenGL Graphics**  **void initGL() {**  **// Set "clearing" or background color**  **glClearColor(0.784f, 0.98f, 0.976f, 0.0f); // Black and opaque**  **}**  **void circle(float radius, float cX, float cY)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(255,0,0);**  **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+cX,y+cY);**  **}**  **glEnd();**  **}**  **void circle1(float radius, float cX, float cY)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(0.298,0.522,0.227);**  **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+cX,y+cY);**  **}**  **glEnd();**  **}**  **/\***  **void circle2(float radius, float cX, float cY)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(0.0,0.0,0.0);**  **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+cX,y+cY);**  **}**  **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 with current clearing color**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.43, 0.53, 0.97); // Yellow**  **glVertex2f(90.0f, -15.0f);**  **glVertex2f(90.0f, 30.0f);**  **glVertex2f(-90.0f,30.0f);**  **glVertex2f(-90.0f,-15.0f);**  **glEnd();**  **////Road**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.787,0.78,0.725); // Yellow**  **glVertex2f(20.0f, 0.0f);**  **glVertex2f(12.0f, 15.0f);**  **glVertex2f(-12.0f, 15.0f);**  **glVertex2f(-20.0f,0.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.601,0.214,0.420); // Yellow**  **glVertex2f(18.0f, -15.0f);**  **glVertex2f(18.0f, 0.0f);**  **glVertex2f(-18.0f, 0.0f);**  **glVertex2f(-18.0f,-15.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.0,0.0,0.0); // Yellow**  **glVertex2f(15.0f, -10.0f);**  **glVertex2f(15.0f, -5.0f);**  **glVertex2f(10.0f, -5.0f);**  **glVertex2f(10.0f, -10.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.0,0.0,0.0); // Yellow**  **glVertex2f(-15.0f, -10.0f);**  **glVertex2f(-15.0f, -5.0f);**  **glVertex2f(-10.0f, -5.0f);**  **glVertex2f(-10.0f, -10.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.0,0.0,0.0); // Yellow**  **glVertex2f(5.0f, -15.0f);**  **glVertex2f(5.0f, -5.0f);**  **glVertex2f(-5.0f, -5.0f);**  **glVertex2f(-5.0f, -15.0f);**  **glEnd();**  **/////tree**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.98,0.64,0.39); // Yellow**  **glVertex2f(-25.0f, -15.0f);**  **glVertex2f(-25.0f, 15.0f);**  **glVertex2f(-30.0f, 15.0f);**  **glVertex2f(-30.0f, -15.0f);**  **glEnd();**  **circle1(10,-32,15);**  **circle1(10,-23,15);**  **circle1(10,-27.5,17);**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.98,0.64,0.39); // Yellow**  **glVertex2f(25.0f, -15.0f);**  **glVertex2f(25.0f, 15.0f);**  **glVertex2f(30.0f, 15.0f);**  **glVertex2f(30.0f, -15.0f);**  **glEnd();**  **circle1(10, 32,15);**  **circle1(10, 23,15);**  **circle1(10, 27.5,17);**  **/////mountain**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(60.0f, 30.0f);**  **glVertex2f(20.0f, 30.0f);**  **glVertex2f(40.0f, 60.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(40.0f, 30.0f);**  **glVertex2f(-40.0f, 30.0f);**  **glVertex2f(10.0f, 60.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(-30.0f, 30.0f);**  **glVertex2f(-90.0f, 30.0f);**  **glVertex2f(-60.0f, 60.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(90.0f, 30.0f);**  **glVertex2f(20.0f, 30.0f);**  **glVertex2f(40.0f, 60.0f);**  **glEnd();**  **////sun**  **circle(10,70,80);**  **////**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(90.0f, -90.0f);**  **glVertex2f(90.0f, -15.0f);**  **glVertex2f(-90.0f, -15.0f);**  **glVertex2f(-90.0f, -90.0f);**  **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("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();**  **gluOrtho2D(-90,90,-90,90); // Our own OpenGL initialization**  **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>**  **/\* Initialize OpenGL Graphics\*/**  **void initGL() {**  **// Set "clearing" or background color**  **glClearColor(0.0f, 0.0f, 0.0f, 0.0f); // Black and opaque**  **}**  **void circle(float radius, float cX, float cY)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(255,0,0);**  **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+cX,y+cY);**  **}**  **glEnd();**  **}**  **void circle1(float radius, float cX, float cY)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(0.298,0.522,0.227);**  **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+cX,y+cY);**  **}**  **glEnd();**  **}**  **void circle2(float radius, float cX, float cY)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(1.0,1.0,1.0);**  **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+cX,y+cY);**  **}**  **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 with current clearing color**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.43, 0.53, 0.97); // Yellow**  **glVertex2f(90.0f, -15.0f);**  **glVertex2f(90.0f, 30.0f);**  **glVertex2f(-90.0f,30.0f);**  **glVertex2f(-90.0f,-15.0f);**  **glEnd();**  **////Road**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.787,0.78,0.725); // Yellow**  **glVertex2f(20.0f, 0.0f);**  **glVertex2f(12.0f, 15.0f);**  **glVertex2f(-12.0f, 15.0f);**  **glVertex2f(-20.0f,0.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.601,0.214,0.420); // Yellow**  **glVertex2f(18.0f, -15.0f);**  **glVertex2f(18.0f, 0.0f);**  **glVertex2f(-18.0f, 0.0f);**  **glVertex2f(-18.0f,-15.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.0,0.0,0.0); // Yellow**  **glVertex2f(15.0f, -10.0f);**  **glVertex2f(15.0f, -5.0f);**  **glVertex2f(10.0f, -5.0f);**  **glVertex2f(10.0f, -10.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.0,0.0,0.0); // Yellow**  **glVertex2f(-15.0f, -10.0f);**  **glVertex2f(-15.0f, -5.0f);**  **glVertex2f(-10.0f, -5.0f);**  **glVertex2f(-10.0f, -10.0f);**  **glEnd();**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.0,0.0,0.0); // Yellow**  **glVertex2f(5.0f, -15.0f);**  **glVertex2f(5.0f, -5.0f);**  **glVertex2f(-5.0f, -5.0f);**  **glVertex2f(-5.0f, -15.0f);**  **glEnd();**  **/////tree**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.98,0.64,0.39); // Yellow**  **glVertex2f(-25.0f, -15.0f);**  **glVertex2f(-25.0f, 15.0f);**  **glVertex2f(-30.0f, 15.0f);**  **glVertex2f(-30.0f, -15.0f);**  **glEnd();**  **circle1(10,-32,15);**  **circle1(10,-23,15);**  **circle1(10,-27.5,17);**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.98,0.64,0.39); // Yellow**  **glVertex2f(25.0f, -15.0f);**  **glVertex2f(25.0f, 15.0f);**  **glVertex2f(30.0f, 15.0f);**  **glVertex2f(30.0f, -15.0f);**  **glEnd();**  **circle1(10, 32,15);**  **circle1(10, 23,15);**  **circle1(10, 27.5,17);**  **/////mountain**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(60.0f, 30.0f);**  **glVertex2f(20.0f, 30.0f);**  **glVertex2f(40.0f, 60.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(40.0f, 30.0f);**  **glVertex2f(-40.0f, 30.0f);**  **glVertex2f(10.0f, 60.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(-30.0f, 30.0f);**  **glVertex2f(-90.0f, 30.0f);**  **glVertex2f(-60.0f, 60.0f);**  **glEnd();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(90.0f, 30.0f);**  **glVertex2f(20.0f, 30.0f);**  **glVertex2f(40.0f, 60.0f);**  **glEnd();**  **////sun**  **circle(10,70,80);**  **////**  **glBegin(GL\_QUADS); // These vertices form a closed polygon**  **glColor3f(0.298,0.522,0.227); // Yellow**  **glVertex2f(90.0f, -90.0f);**  **glVertex2f(90.0f, -15.0f);**  **glVertex2f(-90.0f, -15.0f);**  **glVertex2f(-90.0f, -90.0f);**  **glEnd();**  **////**  **circle2(1,15,70);**  **circle2(2,45,70);**  **circle2(1,-15,70);**  **circle2(1,20,80);**  **circle2(1,-80,60);**  **circle2(1,-75,70);**  **circle2(1,-40,55);**  **circle2(1,-40,70);**  **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("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();**  **gluOrtho2D(-90,90,-90,90); // Our own OpenGL initialization**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |