****

* **UNITED STATES INTERNATIONAL UNIVERSITY -AFRICA**
* **NJERU OLIVER NJIRU**
* **ID NO: 663565**
* **COURSE: APT2090: COMPUTER GRAPHICS**
* **ASSIGNMENT COMPILATION**
* **LECTURER: PROFESSOR TOBIAS MWALILIS**

**Draw Constillation, The Big Dipper, Assignment**

/\*  
\* GL02Primitive.cpp: Vertex, Primitive and Color  
\* Draw Simple 2D colored Shapes: quad, triangle and polygon.  
\*/  
#include <windows.h> // for MS Windows  
#include "glut.h" // GLUT, include glu.h and gl.h  
void myInit(void)  
{  
glClearColor(0.0,0.0,0.0,0.0); // set white background color  
glPointSize(4.0); // a ‘dot’ is 4 by 4 pixels  
glMatrixMode(GL\_PROJECTION);  
glLoadIdentity();  
gluOrtho2D(0.0, 640.0, 0.0, 480.0);  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< myDisplay >>>>>>>>>>>>>>>>>  
void myDisplay(void)  
{  
glColor3f(1.0f, 1.0f, 1.0f); // set the drawing color  
glClear(GL\_COLOR\_BUFFER\_BIT); // clear the screen  
glBegin(GL\_POINTS);  
glVertex2i(450, 200);  
glVertex2i(550, 180);  
glVertex2i(600, 250);  
glVertex2i(400, 300);  
glVertex2i(300, 350);  
glVertex2i(200, 400);  
glVertex2i(100, 470); // draw three points

glEnd();  
glFlush(); // send all output to display  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< main >>>>>>>>>>>>>>>>>>>>>>  
int main(int argc, char\*\* argv)  
{  
glutInit(&argc, argv); // initialize the toolkit  
glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); // set display mode  
glutInitWindowSize(640,480); // set window size  
glutInitWindowPosition(100, 150); // set window position on screen  
glutCreateWindow("Draw DOTS"); // open the screen window  
glutDisplayFunc(myDisplay); // register redraw function  
myInit();  
glutMainLoop(); // go into a perpetual loop  
return 0;  
}

**Draw Kenyan Flag Assignment**

/\*  
\* GL02Primitive.cpp: Vertex, Primitive and Color  
\* Draw Simple 2D colored Shapes: quad, triangle and polygon.  
\*/  
#include <windows.h> // for MS Windows  
#include "glut.h" // GLUT, include glu.h and gl.h  
void myInit(void)  
{  
glClearColor(1.0,1.0,1.0,0.0); // set white background color  
glPointSize(4.0); // a ‘dot’ is 4 by 4 pixels  
glMatrixMode(GL\_PROJECTION);  
glLoadIdentity();  
gluOrtho2D(0.0, 640.0, 0.0, 640.0);  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< myDisplay >>>>>>>>>>>>>>>>>  
void myDisplay(void)  
{  
glClear(GL\_COLOR\_BUFFER\_BIT); // clear the screen

glColor3f(0.0f, 1.0f, 0.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(50, 50); // GREEN  
glVertex2i(600, 50);  
glVertex2i(600, 190);  
glVertex2i(50, 190);  
glEnd();

glColor3f(1.0f, 0.0f, 0.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(50, 200); // RED  
glVertex2i(600, 200);  
glVertex2i(600, 340);  
glVertex2i(50, 340);  
glEnd();

glColor3f(0.0f, 0.0f, 0.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(50, 350); //  BLACK  
glVertex2i(600, 350);  
glVertex2i(600, 490);  
glVertex2i(50, 490);  
glEnd();

glFlush(); // send all output to display  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< main >>>>>>>>>>>>>>>>>>>>>>  
int main(int argc, char\*\* argv)  
{  
glutInit(&argc, argv); // initialize the toolkit  
glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); // set display mode  
glutInitWindowSize(640,480); // set window size  
glutInitWindowPosition(100, 150); // set window position on screen  
glutCreateWindow("Draw DOTS"); // open the screen window  
glutDisplayFunc(myDisplay); // register redraw function  
myInit();  
glutMainLoop(); // go into a perpetual loop  
return 0;  
}

**Draw House Assignment**

/\*  
\* GL02Primitive.cpp: Vertex, Primitive and Color  
\* Draw Simple 2D colored Shapes: quad, triangle and polygon.  
\*/  
#include <windows.h> // for MS Windows  
#include "glut.h" // GLUT, include glu.h and gl.h  
void myInit(void)  
{  
glClearColor(1.0,1.0,1.0,0.0); // set white background color  
glPointSize(4.0); // a ‘dot’ is 4 by 4 pixels  
glMatrixMode(GL\_PROJECTION);  
glLoadIdentity();  
gluOrtho2D(0.0, 640.0, 0.0, 480.0);  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< myDisplay >>>>>>>>>>>>>>>>>  
void myDisplay(void)  
{  
glClear(GL\_COLOR\_BUFFER\_BIT); // clear the screen

glColor3f(1.0f, 0.5f, 0.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(170, 100); // draw three points  
glVertex2i(330, 100);  
glVertex2i(330, 250);  
glVertex2i(170, 250);  
glEnd();

glColor3f(0.0f, 0.0f, 0.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(150, 250); // draw three points  
glVertex2i(350, 250);  
glVertex2i(250, 400);  
glEnd();

glColor3f(1.0f, 1.0f, 1.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(275, 110); // draw three points  
glVertex2i(310, 110);  
glVertex2i(310, 230);  
glVertex2i(275, 230);  
glEnd();

glColor3f(1.0f, 1.0f, 1.0f); // set the drawing color  
glBegin(GL\_POLYGON);  
glVertex2i(210, 190); // draw three points  
glVertex2i(250, 190);  
glVertex2i(250, 230);  
glVertex2i(210, 230);  
glEnd();

glFlush(); // send all output to display  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< main >>>>>>>>>>>>>>>>>>>>>>  
int main(int argc, char\*\* argv)  
{  
glutInit(&argc, argv); // initialize the toolkit  
glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); // set display mode  
glutInitWindowSize(640,480); // set window size  
glutInitWindowPosition(100, 150); // set window position on screen  
glutCreateWindow("Draw DOTS"); // open the screen window  
glutDisplayFunc(myDisplay); // register redraw function  
myInit();  
glutMainLoop(); // go into a perpetual loop  
return 0;  
}

**Draw Cloud Assignment**

/\*  
\* GL02Primitive.cpp: Vertex, Primitive and Color  
\* Draw Simple 2D colored Shapes: quad, triangle and polygon.  
\*/  
#include <windows.h> // for MS Windows  
#include "glut.h" // GLUT, include glu.h and gl.h  
#include <math.h>  
void myInit(void)  
{  
glClearColor(1.0,1.0,1.0,0.0); // set white background color  
glColor3f(0.0f, 0.0f, 0.0f); // set the drawing color  
glPointSize(4.0); // a ‘dot’ is 4 by 4 pixels  
glMatrixMode(GL\_PROJECTION);  
glLoadIdentity();  
gluOrtho2D(0.0, 640.0, 0.0, 480.0);  
}  
//<<<<<<<<<<<<<<<<<<<<<<<< myDisplay >>>>>>>>>>>>>>>>>  
void outwardCircle(float cx, float cy, float r, int num\_segments){  
    glBegin(GL\_LINE\_LOOP);  
    glColor3f(0,0.5,0);  
    for(int j = 0; j < num\_segments; j++) {  
        float theta = 2.0f \* 3.1415926f \* float(j) / float(num\_segments);  
        float x = r \* cosf(theta);  
        float y = r \* sinf(theta);  
        glVertex2f(x + cx, y + cy);  
    }  
    glEnd();  
}

void innerCircle(float cx, float cy, float r, int num\_segments){  
    glBegin(GL\_POLYGON);  
    glColor3f(1,1,1);  
      
    for(int j=0; j < num\_segments; j++) {  
        float theta = 2.0f \* 3.1415926f \* float(j) / float(num\_segments);  
        float x = r \* cosf(theta);  
        float y = r \* sinf(theta);  
        glVertex2f(x + cx, y + cy);  
    }  
    glEnd();  
}

void myDisplay(void)  
{  
glClear(GL\_COLOR\_BUFFER\_BIT); // clear the screen

outwardCircle(160,260,70,100);  
outwardCircle(200,300,70,100);  
outwardCircle(300,320,70,100);  
outwardCircle(400,290,73,100);  
outwardCircle(500,300,70,100);  
outwardCircle(260,200,70,100);  
outwardCircle(350,170,73,100);  
outwardCircle(440,190,70,100);  
outwardCircle(530,240,70,100);  
outwardCircle(400,250,70,100);

innerCircle(200,268,70,100);  
innerCircle(195,288,70,100);  
innerCircle(260,288,70,100);  
innerCircle(370,280,70,100);  
innerCircle(470,280,70,100);  
innerCircle(502,280,70,100);  
innerCircle(484,240,70,100);  
innerCircle(400,198,70,100);  
innerCircle(300,207,70,100);

glFlush(); // send all output to display  
}

//<<<<<<<<<<<<<<<<<<<<<<<< main >>>>>>>>>>>>>>>>>>>>>>  
int main(int argc, char\*\* argv)  
{  
glutInit(&argc, argv); // initialize the toolkit  
glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); // set display mode  
glutInitWindowSize(640,480); // set window size  
glutInitWindowPosition(100, 150); // set window position on screen  
glutCreateWindow("Draw DOTS"); // open the screen window  
glutDisplayFunc(myDisplay); // register redraw function  
myInit();  
glutMainLoop(); // go into a perpetual loop  
return 0;  
}