//#include <windows.h>

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

//Εδώ ορίζουμε το μέγεθος του παραθύρου στην οθόνη του υπολογιστή μας

const int Width=1024;

const int Height = 768;

int DrawAxis = 0;

GLfloat lightPos[] = { 0.0f, 0.0f, 0.0f, 1.0f };

GLfloat whiteLight[] = { 0.2f, 0.2f, 0.2f, 1.0f };

GLfloat sourceLight[] = { 0.8f, 0.8f, 0.8f, 1.0f };

struct STAR

{

GLfloat x;

GLfloat y;

GLfloat size;

};

#define NUM\_STARS 600

STAR Stars[NUM\_STARS];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Συνάρτηση που σχεδιάζει τους άξονες του συστήματος

κόσμου

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void drawAxis()

{

if( DrawAxis )

{

//ο άξονας Χ σε κόκκινο

glColor3f(1,0,0);

glBegin(GL\_LINES);

glVertex3f(-10,0,0);

glVertex3f(10,0,0);

glEnd();

//ο άξονας Υ σε πράσινο

glColor3f(0,1,0);

glBegin(GL\_LINES);

glVertex3f(0,-8,0);

glVertex3f(0,8,0);

glEnd();

//ο άξονας Ζ σε μπλέ

glColor3f(0,0,1);

glBegin(GL\_LINES);

glVertex3f(0,0, -10);

glVertex3f(0,0,10);

glEnd();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Συνάρτηση που σχεδιάζει τα αντικείμενα

στην οθόνη

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void renderScene(void)

{

//ζήτα τον χρόνο που έχει περάσει από την αρχή του προγράμματος

const double t = glutGet(GLUT\_ELAPSED\_TIME) / 1000.0;

//a θα είναι η γωνία περιστροφής αυτή τη χρονική στιγμή

const double a = t\*60.0;

//Καθόρισε το πίνακα ModelView σαν τον ενεργό πίνακα

glMatrixMode(GL\_MODELVIEW);

//κάνε τον ModelView ίσο με τον μοναδιαίο πίνακα

glLoadIdentity();

//καθόρισε που θα κοιτά η κάμερα

gluLookAt(5,6,13, 0,0,0, 0,1,0);

//καθάρισε την οθόνη

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glDisable(GL\_LIGHTING);

drawAxis();

//Τμήμα που ζωγραφίζει τα αστέρια σαν φόντο \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//θέσε το χρώμα των αστεριών σε ανοικτό γκρί

glColor3f(0.6,0.6,0.6);

//ζωγράφισε NUM\_STARS αστέρια ως τριδιάστατα σημεία διαβαζοντας

//την θέση και το μέγεθος του από τον πίνακα Stars

for (int i = 0 ; i < NUM\_STARS ; i++ )

{

//καθόρισε το μέγεθος κάθε σημείου

glPointSize(Stars[i].size);

//ζωγράφισε ένα σημείο.

glBegin(GL\_POINTS);

glVertex3f(Stars[i].x,Stars[i].y, -20.0f);

glEnd();

}

//Τμήμα που ζωγραφίζει τον ήλιο στο κέντρο του συστήματος \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//θέσε το τρέχον χρώμα σε κίτρινο

glColor3f(1,1,0);

//σώσε τον πίνακα ModelView στην στοίβα

glPushMatrix();

//κλιμάκωσε τον ModelView (άλλαξε μέγεθος του ήλιου)

glScalef(1,1,1);

//ζωγράφισε μια σφαίρα (ήλιος)

glutSolidSphere(1,64,64);

//φόρτωσε τον πίνακα ModelView από την στοίβα

glPopMatrix();

glEnable(GL\_LIGHTING);

glLightfv(GL\_LIGHT0,GL\_POSITION,lightPos);

//Τμήμα που ζωγραφίζει τη γη \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//θέσε το τρέχον χρώμα σε μπλε

glColor3f(0,0,1);

//περίστρεψε τη γη κατά γωνία a γύρω από τον άξονα Y

glRotatef(a,0,1,0);

//μετακίνησε την γή κατά -6 κατά μήκος του άξονα Ζ

glTranslatef(0,0,-6);

//σώσε τον πίνακα ModelView στην στοίβα

glPushMatrix();

//κλιμάκωσε την γη

glScalef(0.5,0.5,0.5);

//ζωγράφισε την γη σαν σφαίρα

glutSolidSphere(1, 16, 16);

//φόρτωσε τον πίνακα ModelView από την στοίβα

glPopMatrix();

//Τμήμα που ζωγραφίζει τη σελήνη \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//θέσε το τρέχον χρώμα σε γκρί

glColor3f(0.5f,0.5f,0.5f);

//περίστρεψε τη σελήνη κατά γωνία -4a γύρω από τον άξονα Y

glRotatef(-4\*a,0,1,0);

//μετακίνησε την σελήνη κατά -1.5 κατά μήκος του άξονα Ζ

glTranslatef(0,0,-1.5);

//κλιμάκωσε την σελήνη

glScalef(0.25,0.25,0.25);

//ζωγράφισε την σελήνη σαν σφαίρα

glutSolidSphere(1, 16, 16);

glutSwapBuffers();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Συνάρτηση που θα καλεστεί όταν ο χρήστης

πατήσει ένα πλήκτρο στο πληκτρολογιο

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void keyPress(unsigned char key, int x, int y)

{

switch (key)

{

//έλεγξε αν ο χρήστης έχει πατήσει τα πλήκτρα 'ΕSC' ή 'q'

case 27 :

case 'q':

//έχει πατήσει κάποιο από τα δυο πλήκτρα, έξοδος από το πρόγραμμα

exit(0);

break;

case 'a':

case 'A':

DrawAxis = 1 - DrawAxis;

}

glutPostRedisplay();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Αρχικοποίηση της εφαρμογής. Εδώ εκτελούμε

εντολές που πρέπει να τρέξουν μία μόνο φορά

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void init(void)

{

float fAspect = ((float)Width)/Height;

//Καθορισμός χρώματος στο οποίο θα καθαρίζεται το παράθυρο

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

//καθορισμός ενεργού πίνακα. Οτι μετασχηματισμούς εφαρμόσουμε τώρα

//θα επηρεάσουν τον πίνακα προβολής

glMatrixMode (GL\_PROJECTION);

//κάνε τον πίνακα προβολής ίσο με τον μοναδιαίο πίνακα

glLoadIdentity ();

//όρισε μια προβολή με προοπτική.

gluPerspective(45.0f, fAspect, 1.0, 400.0);

//"φτιάξε" NUM\_STARS αστέρια σε τυχαίες θέσεις και σώσε τις θέσεις

//τους στο πινακα Stars

for (int i = 0 ; i < NUM\_STARS ; i++ )

{

float x = ((float)((rand() % 800) - 400) \* 0.1f);

float y = ((float)((rand() % 800) - 400) \* 0.1f);

float size = 1 + rand() % 3;

Stars[i].x = x;

Stars[i].y = y;

Stars[i].size = size;

}

glEnable(GL\_CULL\_FACE);

glCullFace(GL\_BACK);

glEnable(GL\_DEPTH\_TEST);

glDepthFunc(GL\_LESS);

glEnable(GL\_LIGHT0);

glEnable(GL\_NORMALIZE);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_LIGHTING);

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT,whiteLight);

glLightfv(GL\_LIGHT0,GL\_DIFFUSE,sourceLight);

}

/\* Αρχική συνάρτηση. Με αυτή τη συνάρτηση θα αρχίσει η εκτέλεση του προγράμματος \*/

int main(int argc, char \*\*argv)

{

//αρχικοποίηση GLUT

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

//δημιουργία παραθύρου στο οποίο θα δείξουμε την εικόνα

glutInitWindowSize(Width, Height);

glutInitWindowPosition(150,150);

glutCreateWindow("Computer Graphics - Lab 2");

//καθορισμός συνάρτησης που θα καλείται σε περίπτωση που το παράθυρο χρειάζεται

//επανασχεδίαση

glutDisplayFunc(renderScene);

//καθορισμός συνάρτησης που παρακολουθεί την είσοδο από το πληκτρολόγιο

glutKeyboardFunc(keyPress);

//καθορισμός συνάρτησης που θα καλείται όταν to GLUT δεν εξυπηρετεί κάποιο συμβάν.

glutIdleFunc(renderScene);

init();

glutMainLoop();

return EXIT\_SUCCESS;

}