**作業二  墨西哥帽子**

405262180

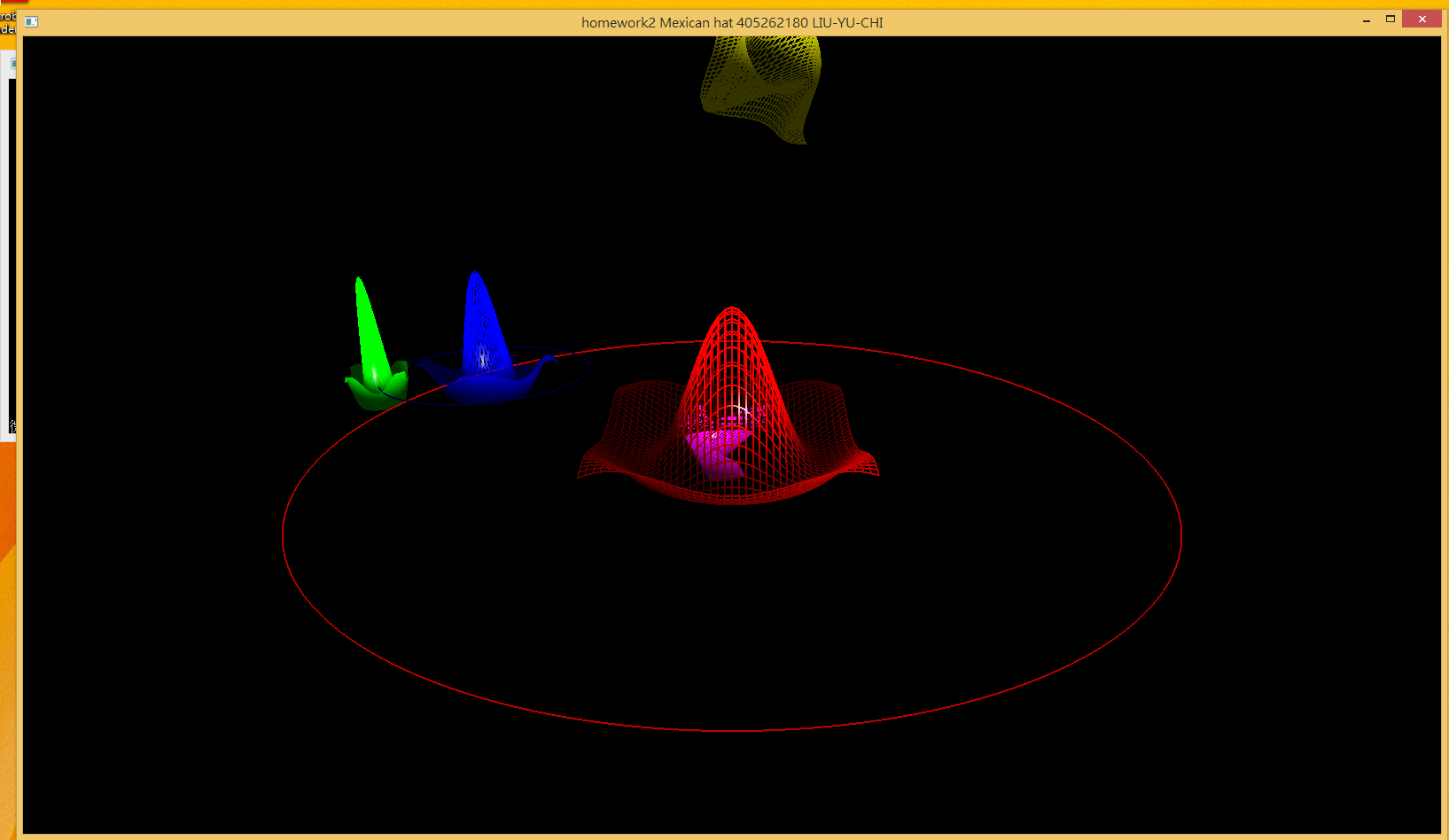
資工三乙

劉育騏

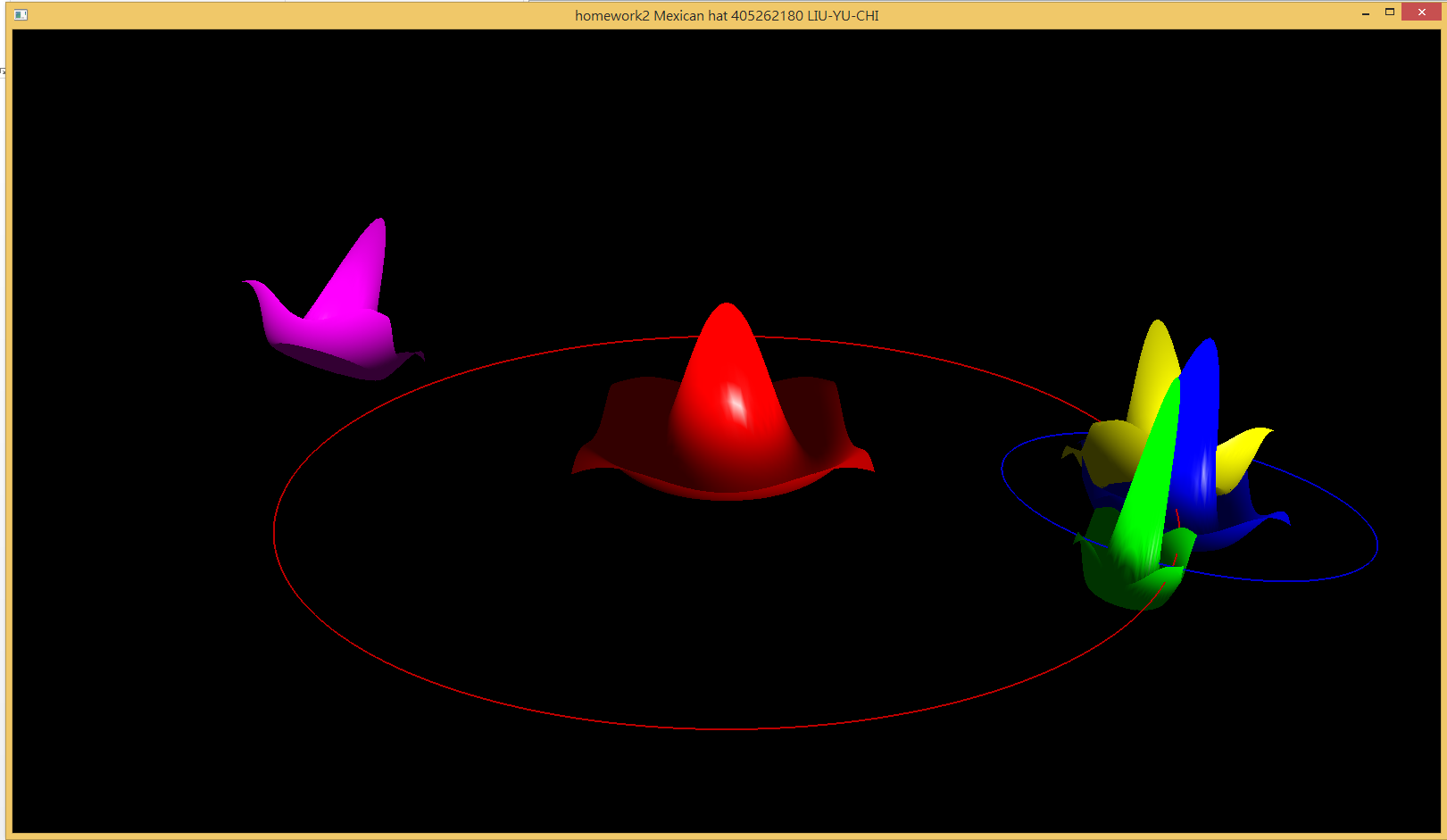
討論:本次作業，學到很多新知識，然後也知道一些新的函式，知道很多應用方法，也更熟悉一些操作。

執行畫面:

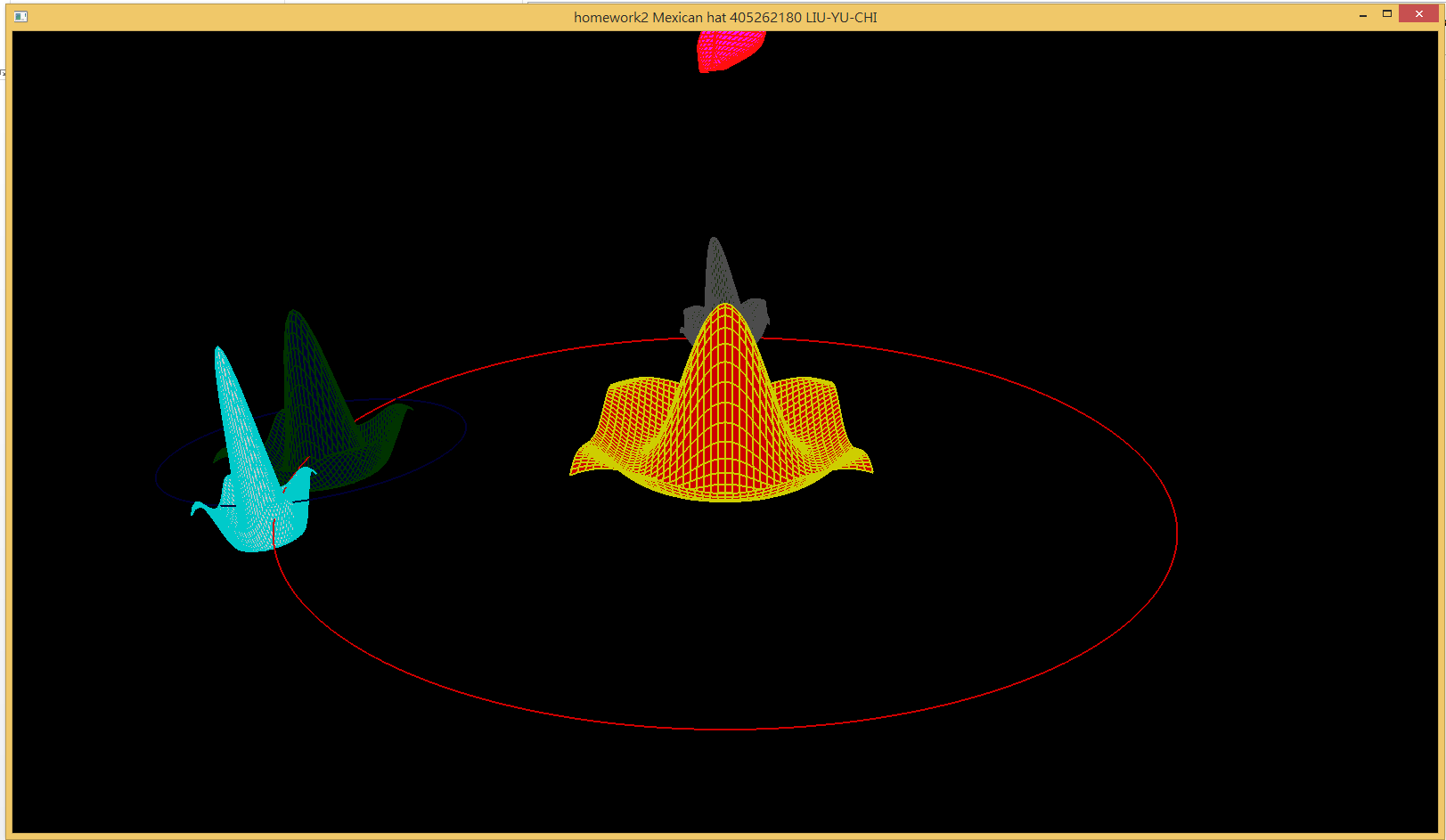
1.線框



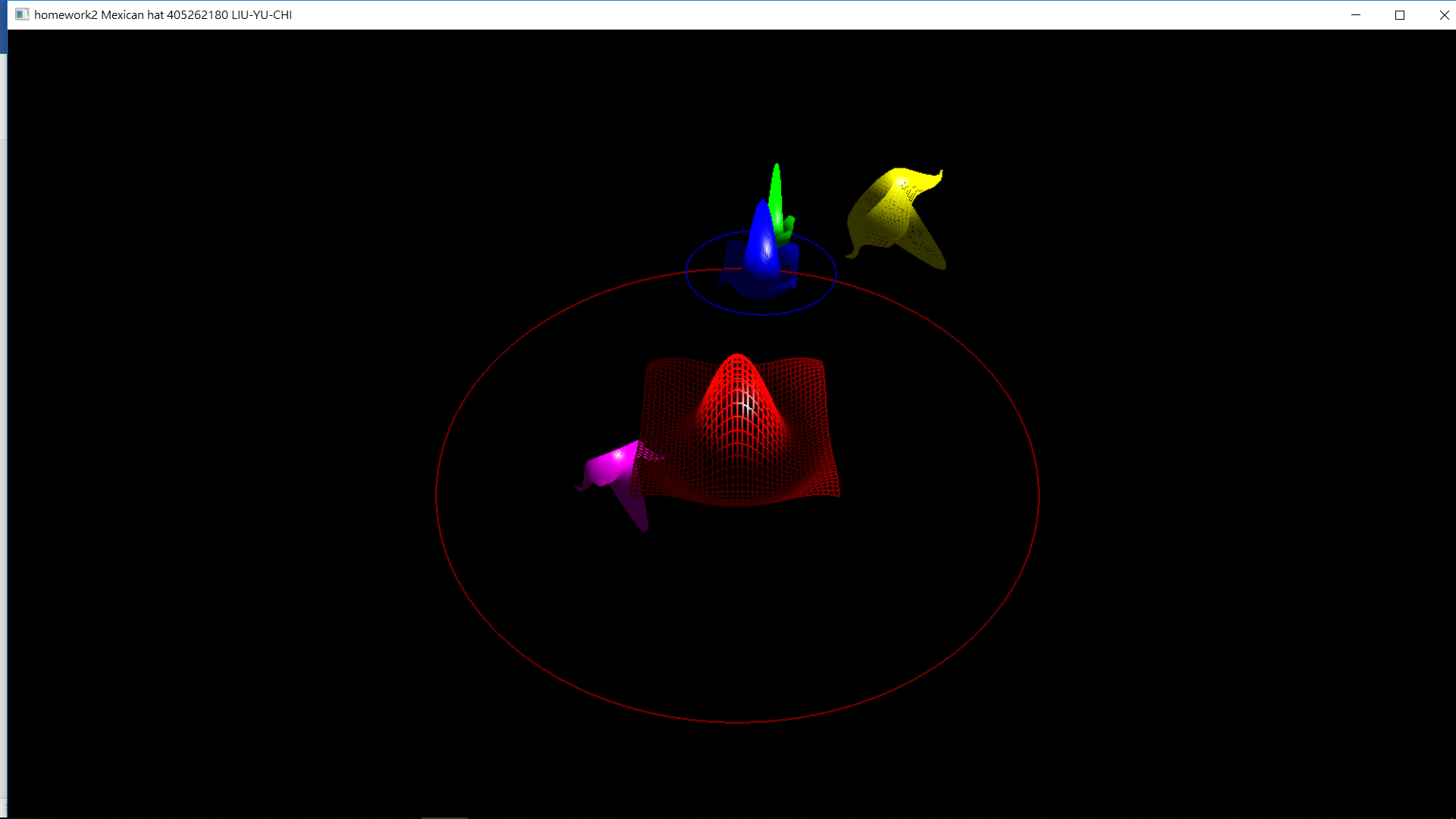
2.塗滿



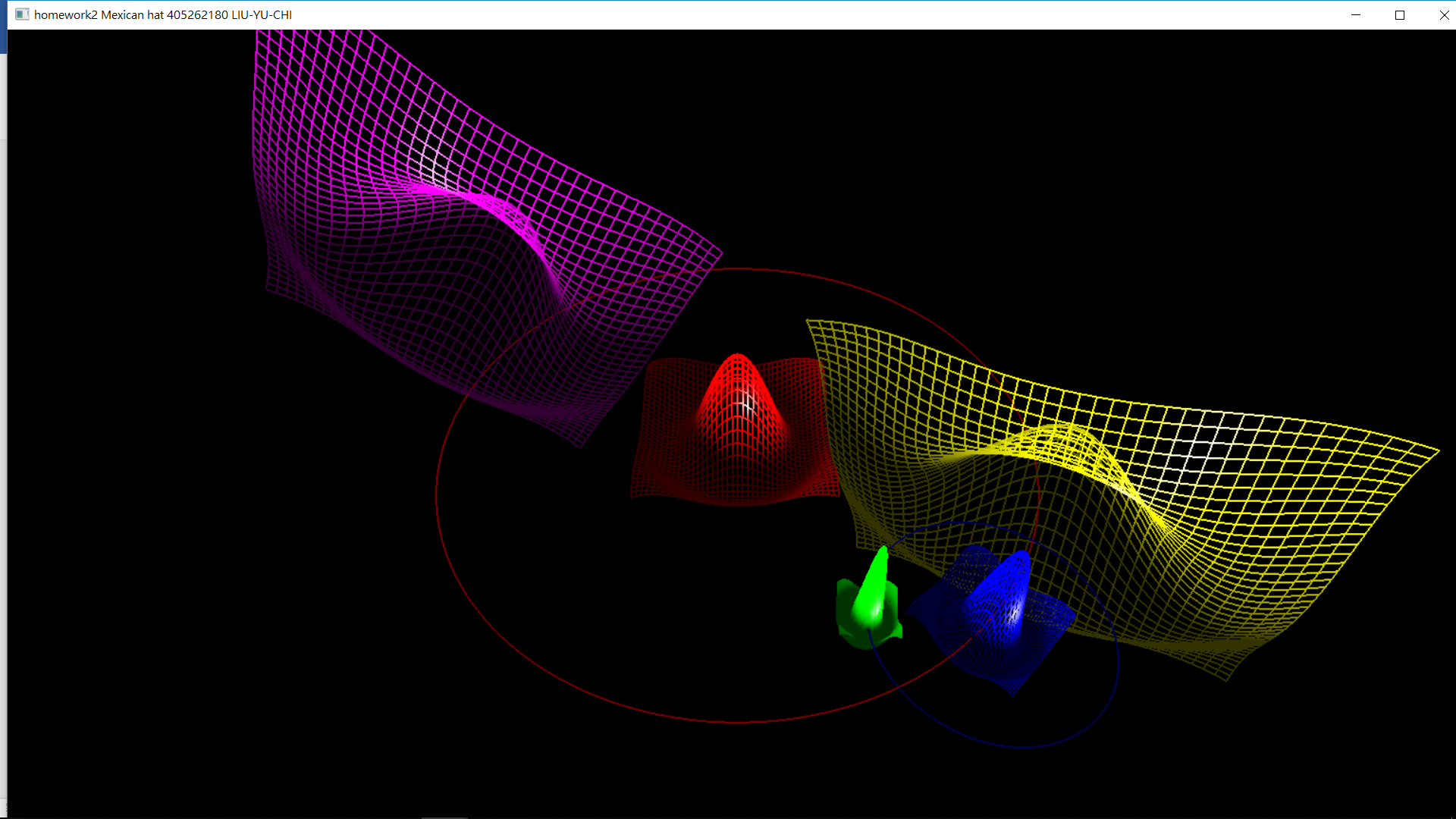
3.線框加塗滿



4.調整視角



5.star變形



**程式碼**

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

#include <stdio.h>

#include <cmath>

#define PI acos(-1)

const int nRows = 40, nCols = 40;

static int weight = 1600, height = 900;

GLenum myType = GL\_LINE\_LOOP; //線條或塗滿

GLfloat data[nRows][nCols];

int choose\_draw = 0;//選擇哪種畫法

typedef struct Color{

float r,g,b;

Color(float r1, float g1, float b1){

r = r1;

g = g1;

b = b1;

}

}C;

C red(1.0, 0.0, 0.0);

C green(0.0, 1.0, 0.0);

C blue(0.0, 0.0, 1.0);

C yellow(1.0, 1.0, 0.0);

C white(1.0, 1.0, 1.0);

C purple(1.0, 0.0, 1.0);

C lite\_blue(0.0, 1.0, 1.0);

C gray(0.299,0.587, 0.114);

struct Planet{

double speed;

double radius, loop;

GLfloat pointarr[nRows\*nCols\*4][4];

int id;//紀錄index

void pushVec4(GLfloat a, GLfloat b, GLfloat c, GLfloat d){//存array

pointarr[id][0] = a;

pointarr[id][1] = b;

pointarr[id][2] = c;

pointarr[id][3] = d;

id++;

}

void setPointarr(){//設定點

id=0;

/\*Point (two triangles/quad) from data\*/

for(int i = 0; i < nRows-1; ++i){

for(int j = 0; j < nCols-1; ++j){

pushVec4(radius\*(GLfloat)i/(GLfloat)nRows-radius/2, data[i][j], radius\*(GLfloat)j/(GLfloat)nCols-radius/2, 1.0);

pushVec4(radius\*(GLfloat)(i+1)/(GLfloat)nRows-radius/2, data[i+1][j], radius\*(GLfloat)j/(GLfloat)nCols-radius/2, 1.0);

pushVec4(radius\*(GLfloat)(i+1)/(GLfloat)nRows-radius/2, data[i+1][j+1], radius\*(GLfloat)(j+1)/(GLfloat)nCols-radius/2, 1.0);

pushVec4(radius\*(GLfloat)i/(GLfloat)nRows-radius/2, data[i][j+1], radius\*(GLfloat)(j+1)/(GLfloat)nCols-radius/2, 1.0);

}

}

}

void drawMHat(){

for(int i=0; i<id; i+=4){

glBegin(myType);

glNormal3fv(pointarr[i]);//法向量

glVertex4fv(pointarr[i]);

glNormal3fv(pointarr[i+1]);

glVertex4fv(pointarr[i+1]);

glNormal3fv(pointarr[i+2]);

glVertex4fv(pointarr[i+2]);

glNormal3fv(pointarr[i+3]);

glVertex4fv(pointarr[i+3]);

glEnd();

}

}

void drawMHat2(C c1, C c2){

glEnableClientState(GL\_VERTEX\_ARRAY);

glVertexPointer(4, GL\_FLOAT, 0, pointarr);

for(int i=0; i<id; i+=4){

glColor3f(c1.r, c1.g, c1.b);

glDrawArrays(GL\_TRIANGLE\_FAN, i, 4);

glColor3f(c2.r, c2.g, c2.b);

glDrawArrays(GL\_LINE\_LOOP, i, 4);

}

}

}sun, earth, moon, star;

struct View{

double eyeX, eyeY;

double centerX;

}myView;

const GLfloat light\_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f };

const GLfloat light\_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat light\_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat light\_position[] = { 2.0f, 5.0f, 5.0f, 0.0f };

const GLfloat mat\_ambient[] = { 0.7f, 0.7f, 0.7f, 1.0f };

const GLfloat mat\_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };

const GLfloat mat\_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat high\_shininess[] = { 100.0f };

void idle();

void myReshape(int w, int h);//重設螢幕

void display(void);//顯示用

void menu(int);//選單

void mymouse(int, int, int, int);//滑鼠

void specialKey(GLint, GLint, GLint);//keyboard

void keyboard(unsigned char, int, int);//鍵盤

void init();

void initPlanet();//初始星球

void initView();//初始眼睛位置

void paintLoopPlant(double, Color);//線條圈圈

void initPlanet(){

sun.radius = 2.0;

sun.loop = 3.0;

sun.setPointarr();

earth.speed = 1.0;

earth.radius = 1.0;

earth.loop = 1.0;

earth.setPointarr();

moon.speed = 1.0;

moon.radius = 0.5;

moon.setPointarr();

star.radius = 1.0;

star.speed = 1.0;

star.setPointarr();

}

void initView(){

myView.eyeX = 0.0;

myView.eyeY = 2.0;

myView.centerX = 0.0;

}

void init(){

int i, j;

// data for radial hat function: sin(Pi\*r)/(Pi\*r)

for(i = 0; i<nRows; ++i){

GLfloat x = PI\*(4.0\*(GLfloat)i/(GLfloat)nRows-2.0);

for(j=0; j<nCols; ++j){

GLfloat z = PI\*(4.0\*(GLfloat)j/(GLfloat)nRows-2.0);

GLfloat r = sqrt(x\*x + z\*z);

data[i][j] = r ? sin(r)/r : 1.0;

}

}

glEnable(GL\_POLYGON\_OFFSET\_FILL); //z 交叉問題處理

glPolygonOffset(1.0, 2.0);

glClearColor(0, 0, 0, 0);

glEnable(GL\_DEPTH\_TEST);

glDepthFunc(GL\_LEQUAL);// z<=當前

glShadeModel(GL\_SMOOTH);

glEnable(GL\_LIGHT0);

glEnable(GL\_NORMALIZE);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_LIGHTING);

glLightfv(GL\_LIGHT0, GL\_AMBIENT, light\_ambient);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, light\_diffuse);

glLightfv(GL\_LIGHT0, GL\_SPECULAR, light\_specular);

glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, mat\_ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, high\_shininess);

}

void myReshape(int w, int h){

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(60.0, (GLfloat)w / (GLfloat)h, 1.0, 20.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluLookAt(0.0, myView.eyeY, 5.0, myView.centerX, 0.0, 0.0, 0.0, 1.0, 0.0);

}

void paintLoopPlant(double r, Color c){

GLfloat x, z;

int i;

glLineWidth(2.0f);

glColor3f(c.r, c.g, c.b); //earth loop: green

glBegin(GL\_LINE\_LOOP);

for (i = 0; i < 1000;i++)

{

x = r \* sin(i \* 2 \* PI / 1000-PI);

z = r \* cos(i \* 2 \* PI / 1000-PI);

glVertex3f(x,0,z);

}

glEnd();

}

void display(void)

{

glLoadIdentity();

gluLookAt(0.0, myView.eyeY, 5.0, myView.centerX, 0.0, 0.0, 0.0, 1.0, 0.0);

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

const double s1 = t \* 90.0 \* earth.speed;

const double s2 = t \* 90.0 \* moon.speed;

const double s3 = t \* 90.0 \* star.speed;

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

glPushMatrix();

//SUN

if(choose\_draw==0){

glColor3f(1, 0.0, 0.0);

sun.drawMHat();

}

else{

sun.drawMHat2(red, yellow);

}

//SUN's loop

paintLoopPlant(sun.loop, red);

//EARTH

glRotatef(s1, 0.0, 1.0, 0.0);

glTranslatef(sun.loop, 0.0, 0.0);

if(choose\_draw==0){

glColor3f(0, 0, 1.0); //blue

earth.drawMHat();

}

else{

earth.drawMHat2(blue, green);

}

//EARTH's loop

paintLoopPlant(earth.loop, blue);

//MOON

glRotatef(s2, 0.0, 1.0, 0.0);

glTranslatef(0.0, 0.0, earth.loop);

if(choose\_draw==0){

glColor3f(0, 1.0, 0.0);

moon.drawMHat();

}

else{

moon.drawMHat2(white, lite\_blue);

}

glPopMatrix();

glPushMatrix();

//STAR

glRotatef(s1, 1.0, 1.0, 0.0);

glTranslatef(sun.loop, 0.0, 0.0);

glRotatef(s3, 1.0, 0.0, 0.0);

if(choose\_draw==0){

glColor3f(1.0, 1.0, 0.0);

star.drawMHat();

}

else{

star.drawMHat2(purple, red);

}

glPopMatrix();

glPushMatrix();

//STAR2

glRotatef(s1, 1.0, 0.0, 1.0);

glTranslatef(-sun.loop, 0.0, 0.0);

glRotatef(s3, 1.0, 0.0, 0.0);

if(choose\_draw==0){

glColor3f(1.0, 0.0, 1.0);

star.drawMHat();

}

else{

star.drawMHat2(gray, white);

}

glPopMatrix();

glutSwapBuffers();

}

void idle(){

glutPostRedisplay();

}

void menu(int index){

switch(index){

case 0:

choose\_draw = 0;

myType = GL\_LINE\_LOOP;

break;

case 1:

choose\_draw = 0;

myType = GL\_POLYGON;

break;

case 2:

choose\_draw = 1;

break;

default:

break;

}

}

void keyboard(unsigned char key, int x, int y){

switch (key) {

case 'u':

earth.speed += 0.1;

break;

case 'i':

earth.speed -= 0.1;

break;

case 'j':

moon.speed += 0.1;

break;

case 'k':

moon.speed -= 0.1;

break;

case '+':

star.speed += 0.1;

break;

case '-':

star.speed -= 0.1;

break;

case '8':

if(star.radius <= 4.0){

star.radius += 0.1;

star.setPointarr();

}

else

printf("too large\n");

break;

case '2':

if(star.radius >= 0.0){

star.radius -= 0.1;

star.setPointarr();

}

else

printf("too small\n");

break;

case 'q':

exit(0);

break;

default:

break;

}

glutPostRedisplay();

}

void specialKey(GLint key, GLint x, GLint y){

if(key==GLUT\_KEY\_UP) myView.eyeY +=0.1;

if(key==GLUT\_KEY\_DOWN) myView.eyeY -=0.1;

if(key==GLUT\_KEY\_LEFT) myView.centerX +=0.1;

if(key==GLUT\_KEY\_RIGHT) myView.centerX -=0.1;

glutPostRedisplay();

}

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

{

glutInit(&argc, argv);

glutInitWindowSize(weight, height);

glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE | GLUT\_DEPTH);

glutCreateWindow("homework2 Mexican hat 405262180 LIU-YU-CHI");

init();

initPlanet();

initView();

glutReshapeFunc(myReshape);

glutDisplayFunc(display);

glutIdleFunc(idle);

glutKeyboardFunc(keyboard);

//Direction Key Controller

glutSpecialFunc(specialKey);

glutCreateMenu(menu);

glutAddMenuEntry("line",0);

glutAddMenuEntry("fill",1);

glutAddMenuEntry("fill+line",2);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

glutMainLoop();

return EXIT\_SUCCESS;

}