作業二　墨西哥帽子

405262180

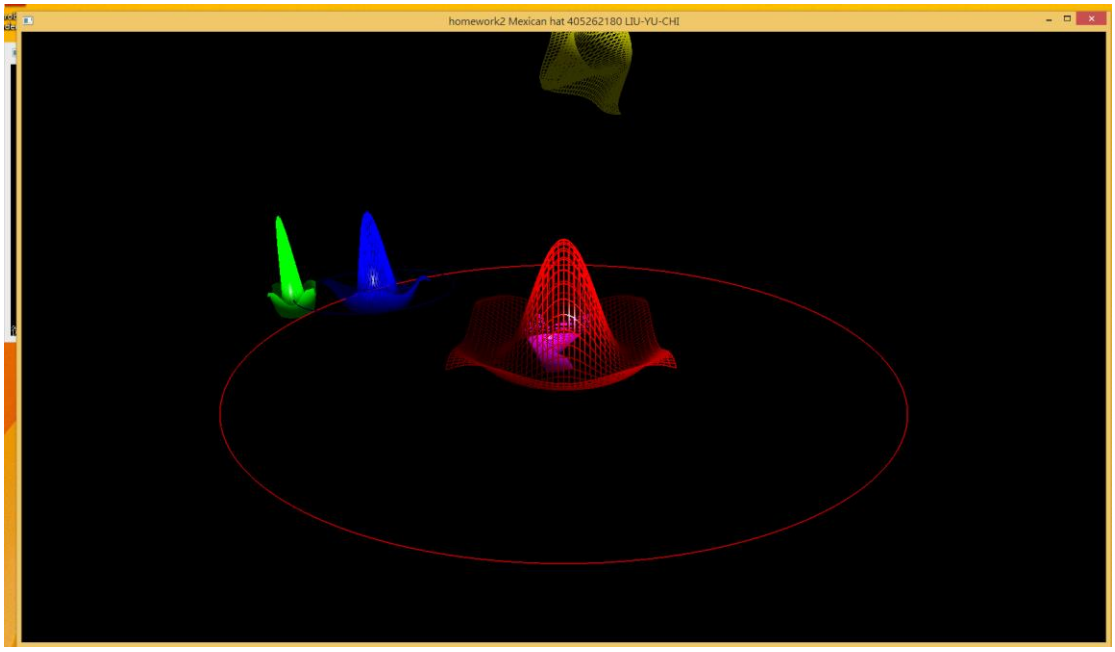
資工三乙

劉育騏

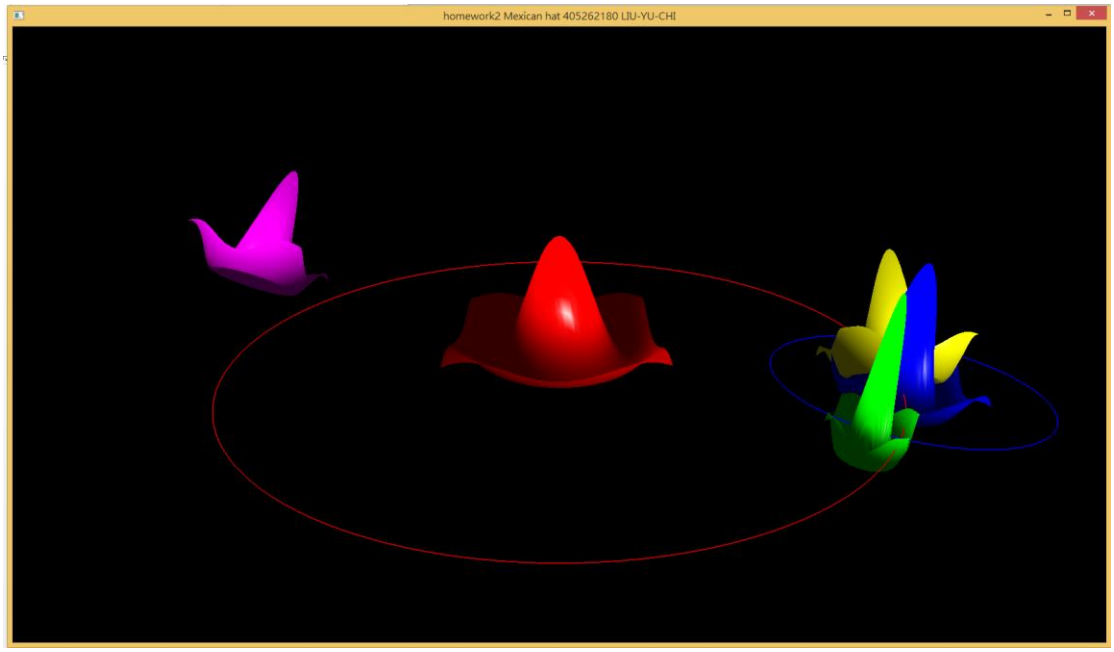
討論:本次作業，學到很多新知識，然後也知道一些新的函式，知道很多應用方法，也更熟悉一些操作。

執行畫面:

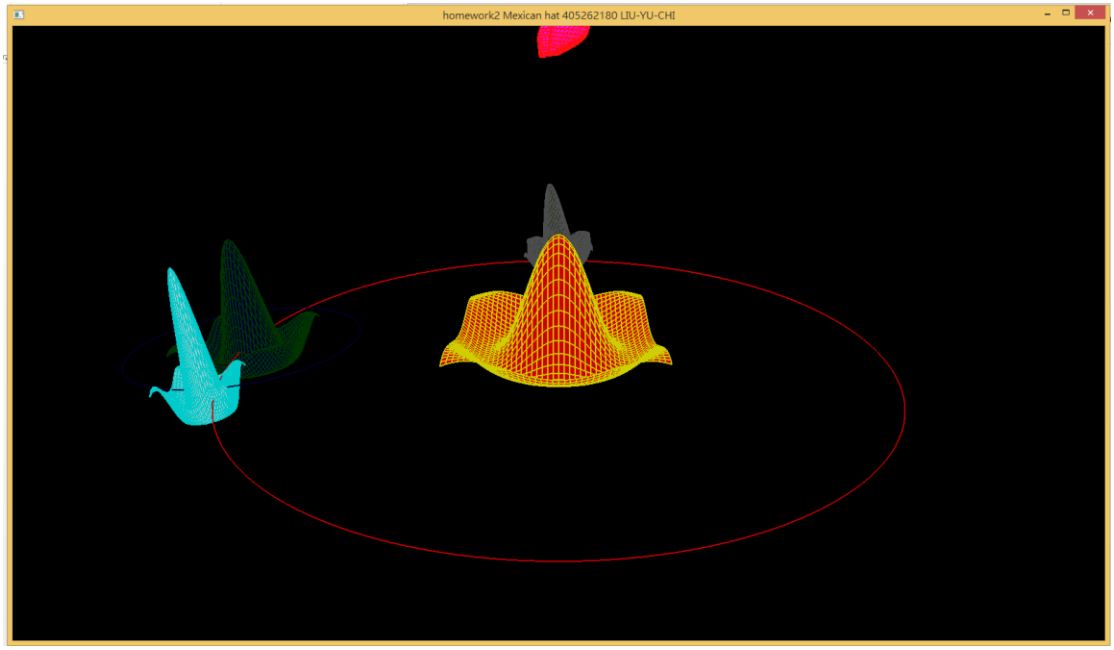
1.線框



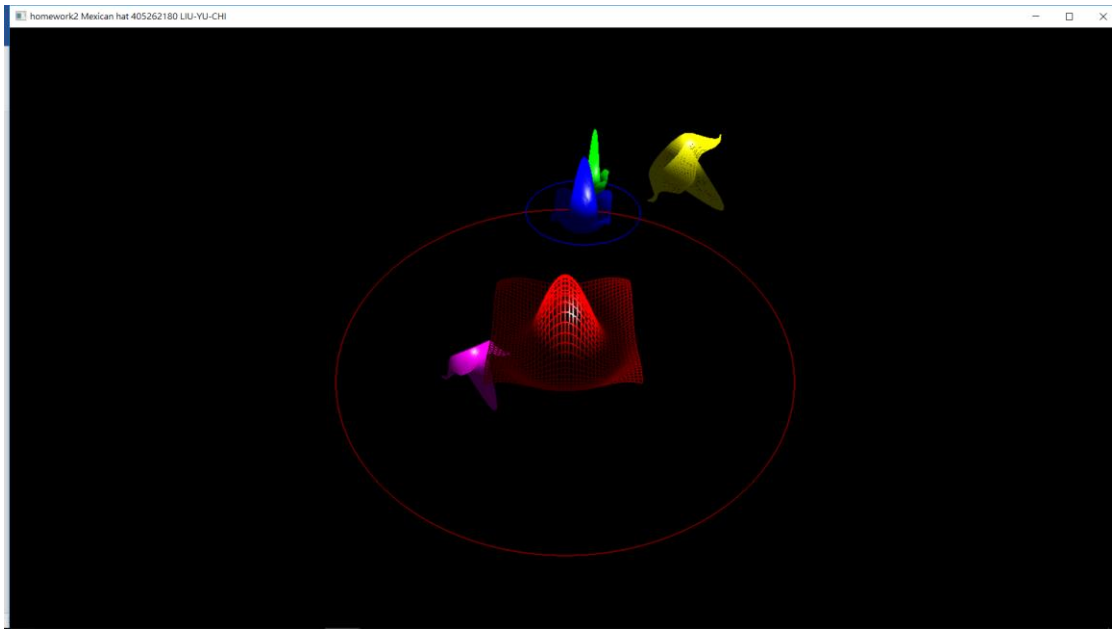
2.塗滿



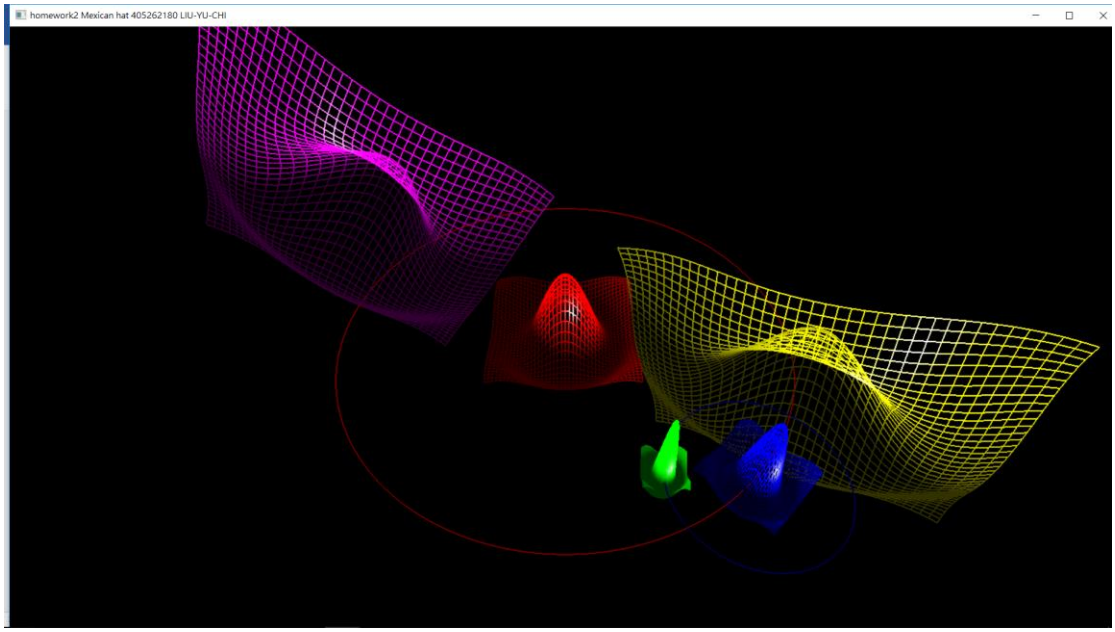
3.線框加塗滿



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;
}
```