

به نام خدا



دانشگاه شهید بهشتی

دانشکده علوم ریاضی

گزارش تمرین دوم گرافیک (کار با کتابخانه OpenGL)

زهره دهقانی تفتی (۹۶۲۲۲۰۳۷)

تاریخ تحویل : ۱۰ اردیبهشت ۱۴۰۰

فهرست مطالب

۲..... تمرین دوم- یک سیستم دوار با ۴ جز چرخشی. اجزای سیستم (خورشید، زمین، ماه، سیاره ۲)

۳..... پاسخ:

۶..... خروجی:

تمرین دوم- یک سیستم دوار با ۴ جز چرخشی. اجزای سیستم (خورشید، زمین، ماه، سیاره ۲)

خورشید: شعاع ۵، موقعیت اولیه $(0,0,0)$ ،

چرخش ۱- سرعت چرخش: 10° درجه در هر 30 میلی ثانیه، محور چرخش y ، pivot point: local $(0,0,0)$

زمین: شعاع ۳، موقعیت اولیه $(10,0,0)$

چرخش ۱- سرعت چرخش: 30° درجه در هر 30 میلی ثانیه، محور چرخش y ، pivot point: local $(0,0,0)$

چرخش ۲- سرعت چرخش: 10° درجه در هر 30 میلی ثانیه، محور چرخش y ، pivot point: global $(0,0,0)$

ماه: شعاع ۰.۵، موقعیت اولیه $(10,5,0)$ ،

چرخش ۱، سرعت چرخش: 10° درجه در هر 30 میلی ثانیه، محور چرخش y ، pivot point: local $(0,0,0)$

چرخش ۲، سرعت چرخش 30° درجه در هر 30 میلی ثانیه، محور چرخش x ، pivot point: earth local $(0,0,0)$

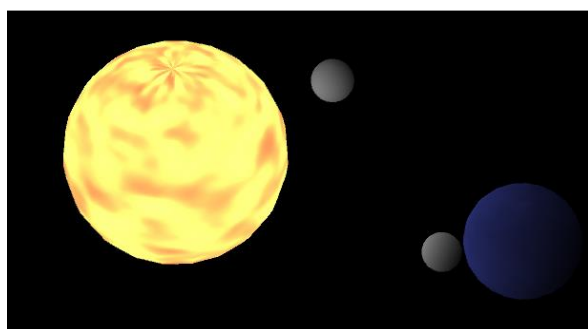
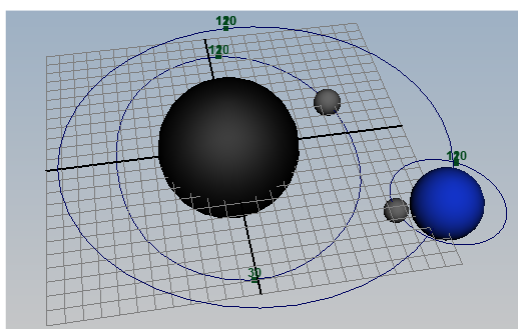
چرخش ۳، سرعت چرخش: 10° درجه در هر 30 میلی ثانیه، محور چرخش y ، pivot point: global $(0,0,0)$

سیاره ۲: شعاع ۲، موقعیت اولیه $(5,0,0)$

چرخش ۱- سرعت چرخش: 30° درجه در هر 30 میلی ثانیه، محور چرخش y ، pivot point: local $(0,0,0)$

چرخش ۲- سرعت چرخش: 10° درجه در هر 30 میلی ثانیه، مسیر حرکت یک بیضی می باشد Pivot point: global

$(0,0,0)$ ، قطر بزرگ ۵، قطر کوچک ۴



پاسخ:

در این تمرین ۴ جز دوار داریم. این ۴ جز را با استفاده از توابع آماده Glut و تغییرات موردنیاز برای چرخش، جابجایی و تغییر اندازه رسم می کنیم. در این تمرین مسیر حرکت سیاره ما بیضی است که ما برای اینکار معادله پارامتری بیضی را در تابع translate که روی سیاره اعمال می شود گذاشتیم که مختصات این سیاره با توجه به تغییر زاویه چرخش آن تغییر می کند.

در این تمرین باید از glpushmatrix و glpopmatrix در مواقع مورد نیاز استفاده شود تا تغییرات به صورت مناسب اعمال شوند.

ابتدا کتابخانه های مورد نیاز فراخوانی می شوند:

```
#include <windows.h> // for MS Windows
#include <GL/glut.h> // GLUT, include glu.h and gl.h
#include <math.h>
```

سپس متغیرهای مورد نیاز را تعریف می کنیم که شامل زاویه های چرخش اجزا حول محورهای مختلف و تایم رفرش انیمیشن است.

```
/* Global variables */
char title[] = "Solar System";
GLfloat angleSun = 0.0f; // Rotational angle for SUN
GLfloat angleEarth_g = 0.0f; // global axis
GLfloat angleEarth_l = 0.0f; // local axis
GLfloat angleMoon_y1 = 0.0; // local y axis
GLfloat angleMoon_yg = 0.0; // global y axis
GLfloat angleMoon_xe = 0.0;
GLfloat anglePlanet_y1 = 0.0; // local y axis
GLfloat anglePlanet_g = 0.0; // global axis
int refreshMills = 30; // refresh interval in milliseconds
```

سپس در تابع initGl مقادیری که می خواهیم یکبار در برنامه تنظیم شوند مثل نورپردازی، متریال، پاکسازی پنجره، فعال کردن بافر عمق و سایه روشن را مشخص می کنیم.

```
void initGL() {

    glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque
    glClearDepth(1.0f); // Set background depth to farthest
    glEnable(GL_DEPTH_TEST); // Enable depth testing for z-culling
    glDepthFunc(GL_LEQUAL); // Set the type of depth-test
    glShadeModel(GL_SMOOTH); // Enable smooth shading

    GLfloat light_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f };
    GLfloat light_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat light_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat light_position[] = { 2.0f, 5.0f, 5.0f, 0.0f };

    GLfloat mat_ambient[] = { 0.0f, 1.0f, 0.3f, 1.0f };
    GLfloat mat_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };
    GLfloat mat_specular[] = { 0.5f, 1.0f, 1.0f, 1.0f };
    GLfloat high shininess[] = { 80.0f };

    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient); //light0
    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);

glEnable(GL_LIGHTING);           // so the renderer considers light
glEnable(GL_LIGHT0);             // فعال کردن منبع نور اول (source: gl-light0)
glEnable(GL_COLOR_MATERIAL);     // برای کار با رنگ فعلی اجسام
}

```

سپس در تابع `display` اجزای صحنه را مشخص می کنیم. این صحنه متشکل از خورشید، زمین، ماه و سیاره است که کد مربوط به هر قسمت را در زیر می بینیم. چون موقعیت اولیه و قطر بزرگ و کوچک سیاره در سوال به گونه ای بود که از داخل خورشید عبور می کرد آن را تغییر دادم.

```

void display() {

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Clear color and depth
    buffers

    //////////// SUN ////////////

    glPushMatrix();
    glColor3f(1, 1, 0);
    glRotatef(angleSun, 0.0f, 1.0f, 0.0f); // Rotate about the (0,1,0)-axis
    glutSolidSphere(5, 16, 16); //sun
    glPopMatrix();

    //////////// EARTH ////////////

    glPushMatrix();
    glRotatef(angleEarth_g, 0, 1, 0);
    glTranslatef(10, 0.0, 0.0);
    glRotatef(angleEarth_l, 0, 1, 0); // Rotate about (0,1,0)-axis
    glColor3f(0, 0, 1);
    glutSolidSphere(3, 16, 16); //earth
    glPopMatrix();

    //////////// MOON ////////////

    glPushMatrix();
    glRotatef(angelMoon_yg, 0, 1, 0);
    glTranslatef(10, 0, 0);
    glRotatef(angelMoon_xe, 1, 0, 0);
    glTranslatef(0, 5, 0);
    glRotatef(angelMoon_yl, 0, 1, 0); // Rotate about (0,1,0)-axis
    glColor3f(0.5, 0.5, 0.5);
    glutSolidSphere(0.5, 16, 16); //moon
    glPopMatrix();

    //////////// PLANET ////////////

    glPushMatrix();
    glTranslatef(8 * cos(angelPlanet_g), 0, 7 * sin(angelPlanet_g)); //elliptical
    orbit(معادله پارامتری بیضی);
    glRotatef(angelPlanet_yl, 0, 1, 0); // Rotate about (0,1,0)-axis
    glColor3f(0.73, 0.16, 0.22);
}

```

```

glutSolidSphere(2, 16, 16); //planet
glPopMatrix();

glutSwapBuffers(); // Swap the front and back frame buffers (double buffering)

// Update the rotational angle after each refresh
angleSun += 10;
angleEarth_g += 10;
angleEarth_l += 30;
angelMoon_y1 += 10;
angelMoon_yg += 10;
angelMoon_xe += 30;
angelPlanet_y1 += 30;
angelPlanet_g += 10;
}

```

از تابع `timer` برای ایجاد انیمیشن و کنترل افزایش زاویه های چرخش اجزا حول هر محور استفاده میشود. همانطور که در تمرین قبل توضیح داده شد مهم ترین جز این تابع `glutpostredisplay` است که تابع `timer` را فراخوانی می کند و صحنه را رسم می کند و در واقع باعث ایجاد لوپ می شود.

```

void timer(int value) {
    glutPostRedisplay(); // Post re-paint request to activate display()
    glutTimerFunc(refreshMills, timer, 0); // next timer call milliseconds later
}

```

تابع `reshape` اطلاعات مربوط به دوربین را دارد. تابع `lookat` مشخص می کند که دوربین در کجا قرار دارد به کجا نگاه می کند و جهت بالای دوربین چیست. تابع `perspective` نیز به ما دیدی مانند دید چشم انسان می دهد.

```

void reshape(GLint w, GLint h) {

    glMatrixMode(GL_PROJECTION);
    gluPerspective(50.0, (GLfloat)w / (GLfloat)h, 3.0, 90.0);
    gluLookAt(8.0, 16.0, 80.0, 0, 0, 0, 0, 1, 0);
}

```

تابع `main` به صورت زیر می باشد. در تابع `main` اطلاعات مربوط به پنجره خروجی را داریم. در این تابع باید از تابع هایی که در بدنه برنامه معرفی کردیم استفاده کنیم. برای هموارتر کردن انیمیشن از دو بافر استفاده می شود.

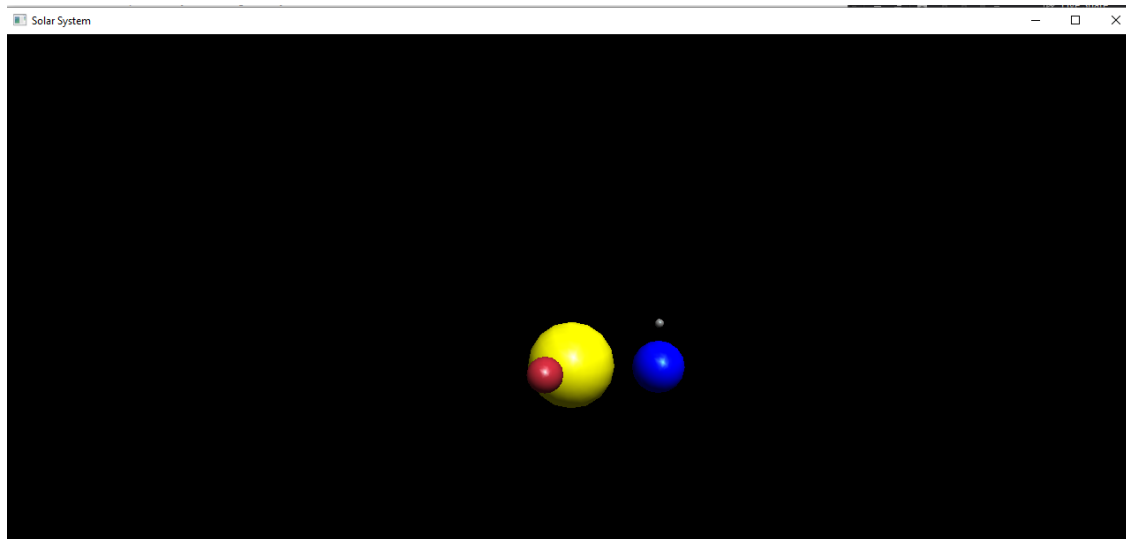
```

int main(int argc, char** argv) {
    glutInit(&argc, argv); // Initialize GLUT
    glutInitDisplayMode(GLUT_DOUBLE); // دو بافر داریم
    glutInitWindowSize(1280, 800); // سایز پنجره
    glutInitWindowPosition(50, 50); // Position the window's initial top-left corner
    glutCreateWindow(title); // Create window with the given title
    glutDisplayFunc(display); // تمام رویدادهای مورد نیاز برای طراحی مجدد صحنه را در اینجا میگذاریم
    glutReshapeFunc(reshape); // Register callback handler for window re-size
    event
    initGL(); // Our own OpenGL initialization
    glutTimerFunc(0, timer, 0); // مربوط به کنترل انیمیشن است و تابع تایمر را فراخوانی می کند
    glutMainLoop(); // Enter the infinite event-processing loop
    تابعی که همیشه برنامه را آپدیت نگه میدارد و لوپ ایجاد می کند
    return 0;
}

```

خروجی:

همانطور که انتظار داشتیم خورشید در مرکز تصویر رسم شده و به دور خود می چرخد. زمین نیز با فاصله ۱۰ متری از مرکز تصویر رسم شده و علاوه بر چرخش به دور محور خودش به دور محور کلی γ نیز میچرخد. ماه نیز همانطور که مشخص است علاوه بر چرخش حول محور خودش، حول محور γ و دور زمین می چرخد. سیاره قرمز رنگ علاوه بر چرخش به دور محور خودش در مسیر بیضی وار به دور خورشید می چرخد. همانطور که در تصویر می بینیم اثرات نورپردازی و متریال برای هر جسم مشخص است.



در این تمرین با اعداد گفته شده برای سیاره، حرکت خیلی سریعی دارد که برای ملایم تر شدن تغییرات انیمیشن آن، می توان تغییر زاویه آن را کمتر کرد.