

Sprawozdanie z ćwiczenia 7

OpenGL – Mini Projekt

Prowadzący:
Dr inż. Jan Nikodem
PN 7:30 TP

Autor: Aleksander Kopciński
Indeks: 252827

1. Wstęp teoretyczny

1.1 Prawa Keplera

W astronomii prawa Keplera dotyczące ruchu planet opisują orbity planet wokół Słońca. Prawa te opisują orbity trajektoriami eliptycznymi oraz wyjaśniają, jak zmienia się prędkości planet.

Trzy prawa to:

1. Orbita planety to elipsa ze Słońcem w jednym z dwóch ognisk.
2. Odcinek linii łączący planetę i Słońce omiata równe obszary w równych odstępach czasu.
3. Kwadrat okresu orbitalnego planety jest proporcjonalny do sześcianu długości wielkiej półosi jej orbity.

1.2 Układ słoneczny

Prawa Keplera dotyczące ruchu planet opisują orbity obiektów wokół Słońca. Zgodnie z prawami Keplera, każdy obiekt porusza się po elipsie ze Słońcem w jednym ognisku. Obiekty bliżej Słońca (z mniejszymi półosiami wielkimi) podróżują szybciej, ponieważ grawitacja słoneczna ma na nie większy wpływ. Na orbicie eliptycznej odległość ciała od Słońca zmienia się w ciągu roku. Ciało znajdujące się najbliżej Słońca nazywa się jego peryhelium, podczas gdy jego najbardziej oddalony punkt od Słońca nazywa się jego aphelium.

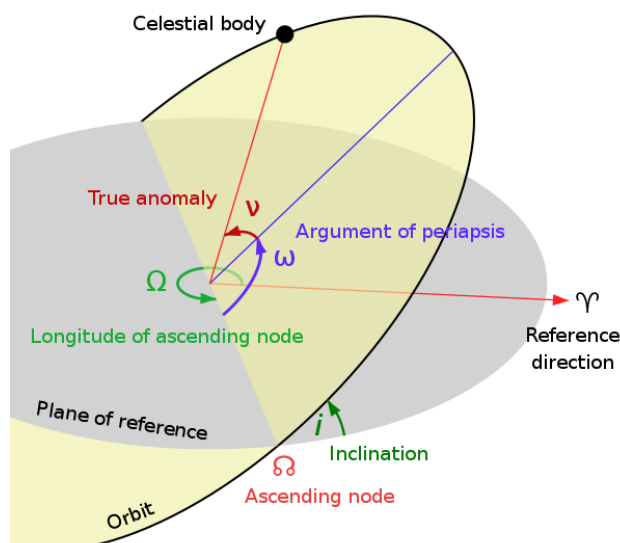
1.3 Płaszczyzna orbity

Płaszczyzna orbity obracającego się ciała jest płaszczyzną geometryczną, na której leży jego orbita. Do wyznaczenia płaszczyzny orbity wystarczą trzy nie współliniowe punkty w przestrzeni.

Płaszczyzna orbity jest definiowana w stosunku do płaszczyzny odniesienia przez dwa parametry: nachylenie* (i) i długość geograficzną węzła wstępującego (Ω)**. Z definicji płaszczyzną odniesienia dla Układu Słonecznego jest zwykle płaszczyzna orbity Ziemi, która określa ekliptykę, kołową ścieżkę na sferze niebieskiej, po której wydaje się podążać Słońce w ciągu roku.

* Nachylenie orbity mierzy nachylenie orbity obiektu wokół ciała niebieskiego. Jest wyrażony jako kąt pomiędzy płaszczyzną odniesienia a płaszczyzną orbity lub osią kierunku orbitującego obiektu.

** Długość geograficzna węzła wstępującego jest jednym z elementów orbitalnych używanych do określenia orbity obiektu w przestrzeni. Jest to kąt od określonego kierunku odniesienia do kierunku węzła wstępującego. Węzeł wstępujący to punkt, w którym orbita obiektu przechodzi przez płaszczyznę odniesienia.



Argument perycentrum(ω) jest jednym z elementów orbitalnych ciała orbitującego. Parametrycznie ω jest kątem od węzła wstępującego ciała do jego perycentrum, mierzonym w kierunku ruchu.

1.4 Elipsa

1. Elipsa

Elipsa jest krzywą płaską otaczającą dwa ogniska, tak że dla wszystkich punktów na krzywej suma dwóch odległości do punktów ogniskowych jest stała. Wydłużenie elipsy mierzy się jej mimośrodem.

2. Parametry elipsy

Wielka oś elipsy to najdłuższa średnica. Jest to odcinek, który przebiega przez środek i oba ogniska, z końcami w dwóch najbardziej oddalonych punktach obwodu. Wielka półoś jest połową wielkiej osi, a zatem biegnie od środka przez ognisko do obwodu. Półoś mała elipsy to odcinek linii, który jest prostopadły do wielkiej półosi. W szczególnym przypadku okręgu długości półosi są równe promieniowi okręgu.

3. Mimośród

Mimośród to parametrem, który określa wielkość, o jaką jego orbita wokół innego ciała odchyła się od idealnego koła. Wartość 0 to orbita kołowa, wartości od 0 do 1 tworzą orbitę eliptyczną, 1 to paraboliczna orbita, a większa niż 1 to hiperbola.

$$e = \frac{c}{a} = \sqrt{1 - \left(\frac{b}{a}\right)^2}$$

4. Semi-latus rectum

$$\ell = \frac{b^2}{a} = a(1 - e^2)$$

1.5 Orbita

1. Orbita

Orbita jest zakrzywioną trajektorią obiektu, taką jak trajektoria planety wokół gwiazdy. W większości sytuacji ruch orbitalny jest odpowiednio aproksymowany przez mechanikę Newtona. Jednak ogólna teoria względności Alberta Einsteina, która wyjaśnia grawitację spowodowaną krzywizną czasoprzestrzeni, z orbitami podążającymi za geodezją, zapewnia dokładniejsze obliczenia i zrozumienie dokładnej mechaniki ruchu orbitalnego.

2. Orbita eliptyczna

Orbita eliptyczna to orbita Keplera z mimośrodem* mniejszym niż 1; obejmuje to szczególny przypadek orbity kołowej, z mimośrodem równym 0. W ścisłym sensie jest to orbita Keplera z mimośrodem większym niż 0 i mniejszym niż 1.

3. Okres orbity

Okres orbity to czas, w którym dany obiekt astronomiczny wykonuje jedną orbitę wokół innego obiektu. Przy standardowych założeniach okres orbitalny T ciała poruszającego się po orbicie eliptycznej można obliczyć jako:

$$T = 2\pi\sqrt{\frac{a^3}{\mu}}$$

gdzie:

Mianownik ułamka to standardowy parametr grawitacyjny,
Licznik ułamka to długość wielkiej półosi.

2. Cel ćwiczenia

Celem ćwiczenia było napisanie programu symulującego układ słoneczny. Specyfikowana była odpowiednia złożoność symulacji, to znaczy uwzględnić płaszczyzny orbit, szybkość obrotu planet, traktowanie orbit jako elips, a także wykorzystywanie informacji z poprzednich laboratoriów do tekstuowania, ustawiania światła i interakcji z użytkownikiem.

3. Realizacja zadania

3.1 Parametry ciał niebieskich

Ustawianie parametrów dla ciała niebieskiego

```
Planet::Planet(float firstAxis, float secondAxis, float radius, float SAV, angle3 spinSpeed)
{
    float startAngle = 0.0;
    if (firstAxis > secondAxis)
    {
        maA = firstAxis;
        miA = secondAxis;
    }
    else
    {
        maA = secondAxis;
        miA = firstAxis;
    }

    this->radius = radius;
    ells = SAV;
    p = pow(miA, 2.0) / maA;
    e = sqrt(1.0 - (pow(miA, 2.0) / pow(maA, 2.0)));
    angle = startAngle;
    this->spinSpeed[0] = spinSpeed[0];
    this->spinSpeed[1] = spinSpeed[1];
    this->spinSpeed[2] = spinSpeed[2];

    angles[0] = spinSpeed[0];
    angles[1] = spinSpeed[1];
    angles[2] = 0.0;
}
```

Obliczanie mimośrodów i semi-latus rectum

```
p = pow(miA, 2.0) / maA;
e = sqrt(1.0 - (pow(miA, 2.0) / pow(maA, 2.0)));
```

3.2 Rysowanie kuli

Polecenia OpenGL

- `gluNewQuadric` tworzy i zwraca wskaźnik do nowego obiektu kwadryki. Do tego obiektu należy się odwoływać podczas wywoływania funkcji renderujących i sterujących kwadrykami
- `gluQuadricTexture` określa, czy współrzędne tekstury powinny być generowane dla kwadryków renderowanych za pomocą `quad`. Jeśli wartość tekstury to `GLU_TRUE`, to generowane są współrzędne tekstury, a jeśli tekstura to `GLU_FALSE`, nie. Wartość początkowa to `GLU_FALSE`.
- `gluSphere` rysuje kulę o podanym promieniu wyśrodkowaną wokół początku. Kula jest podzielona wokół osi *z* na plastry i wzdłuż osi *z* na stopy

```
quad = gluNewQuadric();

gluQuadricTexture(quad, 1);
gluSphere(quad, radius, radius * 5, radius * 5);
```

3.3 Ustawianie parametrów dla płaszczyzny orbity

```
void Planet::setPlane(GLfloat angle1, GLfloat angle2, GLfloat angle3)
{
    planeAngles[0] = angle1;
    planeAngles[1] = angle2;
    planeAngles[2] = angle3;
}
```

3.4 Obsługa klawiatury

Klawisz spacji powoduje zatrzymanie symulacji

'r' powoduje zrestartowanie symulacji

'p' powoduje wyłączenie/włączenie orbit

Esc – zakończenie programu

```
void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case ' ':
            stopPro = !stopPro;
            break;
        case 'r':
            theta[0] = theta[1] = 0.0;
            R = 10.0f;
            status = -1;
            break;
        case 'p':
            drawPath = !drawPath;
            break;
        case 27:
            exit(0);
        default:
            break;
    }

    glutPostRedisplay(); //odświeżenie zawartości aktualnego okna
}
```

3.5 Tekstury

Wykonywanie pętli ładującej tekstury dla ciał niebieskich

```
for (size_t i = 0; i < 10; i++)
{
    stringstream name;
    name << "textures/" << textureNames[i];
    pBytes = LoadTGAImage(name.str().c_str(), &ImWidth, &ImHeight, &ImComponents, &ImFormat);
    glBindTexture(GL_TEXTURE_2D, textureId[i]); //Tell OpenGL which texture to edit

    /******
    // Zdefiniowanie tekstury 2-D
    glTexImage2D(GL_TEXTURE_2D, 0, ImComponents, ImWidth, ImHeight, 0, ImFormat, GL_UNSIGNED_BYTE, pBytes);

    /******
    // Zwolnienie pamięci
    free(pBytes);
    pBytes = nullptr;
}
```

Ustawienie tekstur dla odpowiednich obiektów

```
sun.setTextureId(textureId[0]);
mercury.setTextureId(textureId[1]);
venus.setTextureId(textureId[2]);
earth.setTextureId(textureId[3]);
moon.setTextureId(textureId[4]);
mars.setTextureId(textureId[5]);
jupiter.setTextureId(textureId[6]);
saturn.setTextureId(textureId[7]);
uranus.setTextureId(textureId[8]);
neptune.setTextureId(textureId[9]);
```

3.6 Liczenie nowej pozycji obiektu

```
if (maA != 0.0 && miA != 0.0)
{
    float r = fr1(p, e, angle);
    angle += ellS;
    while (angle >= 360.0)
        angle -= 360.0;

    ellS = (area * 2.0f) / pow(r, 2.0f);

    position[0] = r * cos(angle * M_PI / 180.0);
    position[1] = r * sin(angle * M_PI / 180.0);
}

angles[2] += spinSpeed[2];
while (angles[2] >= 360.0)
    angles[2] -= 360.0;
```

3.7 Ustawienie nowej pozycji obiektu

```
glPushMatrix();
glLoadMatrixf(modelMatrix);
glRotatef(planeAngles[0], 1.0, 0.0, 0.0);
glRotatef(planeAngles[1], 0.0, 1.0, 0.0);
glRotatef(planeAngles[2], 0.0, 0.0, 1.0);
glGetFloatv(GL_MODELVIEW_MATRIX, transformedMatrix);
glPopMatrix();

glPushMatrix();
glLoadMatrixf(transformedMatrix);
glTranslatef(position[0], 0.0, position[1]);
glGetFloatv(GL_MODELVIEW_MATRIX, transformedMatrix);
glPopMatrix();
```

4. Wnioski

Nie zdołałem dokończyć programu w trakcie zajęć. Występują błędy kompilacji, dlatego nie ma wersji wykonywalnej.

5. Literatura

1. [https://en.wikipedia.org/wiki/Orbital_plane_\(astronomy\)](https://en.wikipedia.org/wiki/Orbital_plane_(astronomy))
2. https://en.wikipedia.org/wiki/Orbital_period
3. <https://en.wikipedia.org/wiki/Ellipse>
4. https://en.wikipedia.org/wiki/Solar_System#Earth
5. https://en.wikipedia.org/wiki/Elliptic_orbit
6. https://en.wikipedia.org/wiki/Kepler%27s_laws_of_planetary_motion
7. https://en.wikipedia.org/wiki/Orbit#Specifying_orbits
8. https://en.wikipedia.org/wiki/Orbital_inclination
9. https://en.wikipedia.org/wiki/Longitude_of_the_ascending_node

10. https://en.wikipedia.org/wiki/Argument_of_periapsis
11. https://en.wikipedia.org/wiki/Semi-major_and_semi-minor_axes