

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Ciência da Computação

Atividade 6 - Sistemas de Coordenadas e Movimentos de Objetos

BCC327 - Computação Gráfica

Matheus Peixoto Ribeiro Vieira - 22.1.4104
Professor: Rafael Alves Bonfim de Queiroz

Ouro Preto
31 de março de 2025

Sumário

1	Introdução	1
1.1	Especificações da máquina	1
2	Implementação	1

Lista de Figuras

1	Sistema Solar	2
2	Sistema Solar após movimentações	2

Lista de Códigos Fonte

1	Raios e órbitas	1
2	Planetas e orbitas	1

1 Introdução

Para este trabalho foi escolhida a implementação do Sistema Solar contendo o Sol, a Terra e a Lua. Sendo que o Sol está fixo, a Terra gira entorno dele e a Lua orbita a Terra.

1.1 Especificações da máquina

A máquina onde o desenvolvimento foi realizado possui a seguinte configuração:

- Processador: Intel Core i5-9300H
- Memória RAM: 16Gb.
- Sistema Operacional: WSL 2.0 com Ubuntu 22.04.5 LTS

2 Implementação

Para a implementação do código, junto ao OpenGL, foi utilizada a biblioteca glut.

Inicialmente, é definido os raios para as esferas que irão compor o Sol, Terra e Lua. Em seguida, é definido para a Terra e para a Lua o raio de suas órbitas, como pode ser visto no código 1.

```
1 #define SUN_RADIUS      1.0
2 #define EARTH_ORBIT_RADIUS 4.0
3 #define EARTH_RADIUS    0.3
4 #define MOON_ORBIT_RADIUS 0.7
5 #define MOON_RADIUS     0.1
```

Código 1: Raios e órbitas

Na função `main` é definido um timer pela função `glutTimerFunc` [3], onde nela é definido o incremento da angulação da Terra e o da Lua, onde a última possui uma velocidade de rotação maior, simulando o universo real de forma não fidedigna.

A função `display` é a responsável por exibir na tela os objetos e realizar suas rotações. Após as configurações de câmera, o Sol é desenhado com a `glutSolidSphere` [2].

Em seguida é desenhada a órbita da Terra ao redor do Sol. E depois é feita uma pilha de matrizes com a função `glPushMatrix` [1], dessa forma, é feita a hierarquia dos objetos, onde o pai é o elemento mais externo e o filho o mais interno. Assim, a Terra é desenhada.

O desenho da órbita da Lua permanece sempre ao lado da Terra, dessa forma, ele pode ser mantido dentro da hierarquia da mesma, uma vez que mudanças aplicadas ao planeta deverão influenciar o seu movimento, que deve ser igual.

Como a Lua deve seguir o movimento da Terra, uma nova hierarquia é criada com o `glPushMatrix` e, dentro dela, a Lua é desenhada. Dessa forma, o objeto segue o movimento do seu Pai.

No código 2 podem ser vistos os códigos que fazem a hierarquia dos objetos

```
1 // Desenha a Terra
2 glPushMatrix();
3     // Desenha a Terra
4     glRotatef(angleEarth * 180.0 / M_PI, 0.0, 1.0, 0.0);
5     glTranslatef(EARTH_ORBIT_RADIUS, 0.0, 0.0);
6     glColor3f(0.0, 0.0, 1.0);
7     glutSolidSphere(EARTH_RADIUS, 20, 20);
8
9     // Desenha a órbita da Lua em torno da Terra
10    glColor3f(0.5, 0.5, 0.5);
11    glBegin(GL_LINE_LOOP);
12        for (int i = 0; i < 360; i++) {
13            float theta = i * M_PI / 180.0;
14            glVertex3f(MOON_ORBIT_RADIUS * cos(theta), 0, MOON_ORBIT_RADIUS *
15                      sin(theta));
16        }
17    glEnd();
```

```

17
18 // Desenha a Lua
19 glPushMatrix();
20     glRotatef(angleMoon * 180.0 / M_PI, 0.0, 1.0, 0.0);
21     glTranslatef(MOON_ORBIT_RADIUS, 0.0, 0.0);
22     glColor3f(0.8, 0.8, 0.8);
23     glutSolidSphere(MOON_RADIUS, 20, 20);
24     glPopMatrix();
25
26 glPopMatrix();

```

Código 2: Planetas e orbitas

A figura 1 exibe o resultado final do código, exibindo o sistema solar. Já a figura 2 exibe o estado do sistema solar após alguns segundos de movimentação.

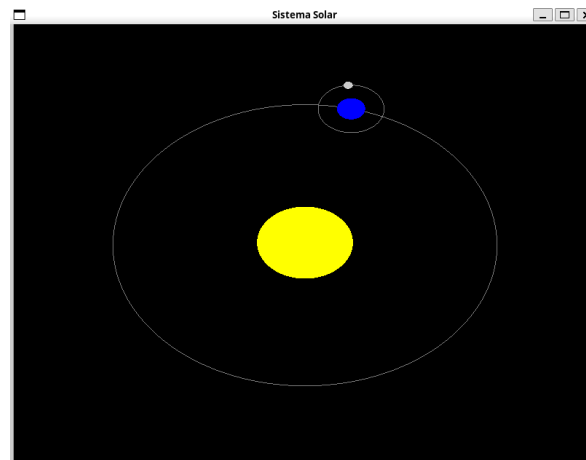


Figura 1: Sistema Solar

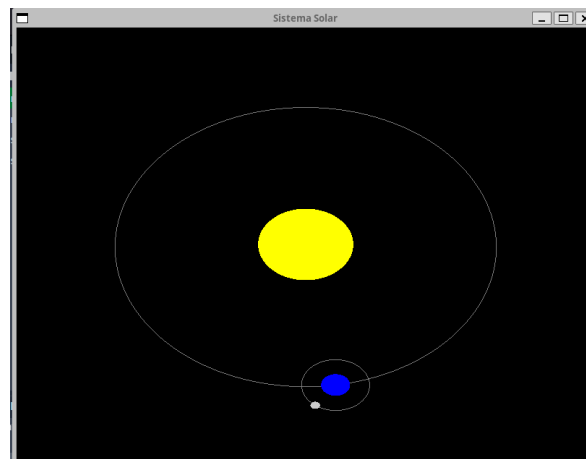


Figura 2: Sistema Solar após movimentações

Para a compilação e execução do código o seguinte comando deve ser executado:

```

gcc atividade.c -o atividade -lGL -lGLU -lglut -lm
./atividade

```

Referências

- [1] Adam. `glpushmatrix`. <https://stackoverflow.com/questions/23971752/understanding-glpushmatrix-and-glpopmatrix>. [Online; acessado em 14-Fevereiro-2025].
- [2] OpenGL. `glutsolidSphere`. <https://www.opengl.org/resources/libraries/glut/spec3/node81.html>. [Online; acessado em 14-Fevereiro-2025].
- [3] OpenGL. `glutTimerFunc`. <https://www.opengl.org/resources/libraries/glut/spec3/node64.html>. [Online; acessado em 14-Fevereiro-2025].